AppMill

TASK AND ONBOARDING PROCESS MANAGEMENT SOLUTION

Example of Usage and Instruction

Table of Contents

Table of Contents 2
1 Introduction 12
1.1 Overview of the Application and Features 12
1.1.1 Task Management Application Overview 12
1.1.2 Onboarding Process Application Overview 13
1.1.3 Benefits of Using AppMill Solutions 14
1.1.4 Implementation and Development from Scratch 14
1.1.5 Implementation and Development Using Provided AppMill Solution 15
1.1.6 Summary 15
2 Service and Clients Installation 17
2.1 Service Installation 17
2.2 Clients Installation 17
3 Initial Actions in Render 18
3.1 First Login to Render 18
4 Initial Settings in Architect 18
4.1 First Login to Architect 18
4.1.1 Predefined Data 18
4.1.2 Reload Aliases 19
4.1.3 Set up Document Devices 19
4.1.4 Architect User Documentation 20
5 Set up Account Settings 21
5.1 Edit User Details 21
5.2 Add Logo Images 21
5.3 Edit Tenant 22
6 Set up Login Functionality 23
6.1 Create Login Group 23
6.2 Create Login User 24
6.3 Create Security Filter for Login User 24
6.4 Create Login Menu 25
6.5 Create Login Screen 25
6.6 Create Group Screen for Login Screen 27
6.7 Set up Start Screen 27
6.8 Configure Guest User mode 28
6.9 Log in to Render via Login Screen 28

6.10 Update Login Screen 28
6.11 Update Styles for Login User 29
7 Example of Usage: Task Manager Tool 31
7.1 User Roles 31
7.2 Screens 31
7.2.1 Start Screen for HR User 32
7.2.2 Start Screen for Manager Users 32
7.2.3 Start Screen for Employee Users 33
7.3 Use Cases 33
7.3.1 Admin Use Cases List 33
7.3.2 HR Use Cases 34
7.3.3 Detailed HR Use Cases 34
7.3.4 Manager Use Cases 34
7.3.5 Detailed Manager Use Cases 35
7.3.6 Employee Use Cases 36
7.3.7 Detailed Employees Use Cases 37
8 Set up Environment for HR, Managers and Employees 39
8.1 Create Groups for HR, Managers and Employees 39
8.2 Set up Security Filters 39
8.2.1 Set up Security Filter for HR and Managers 39
8.2.2 Create Security Filter for Employee 40
8.3 Create all Users 41
8.4 Edit UI Styles for Users 45
8.5 Create Tables to link Managers and Employees 46
8.6 Set up Auto Sequence 47
9 Set up Environment for HR Use Cases 48
9.1 Create Menu for HR 48
9.2 Create Screens required for the HR user in the Render 48
9.2.1 Create Custom Query 48
9.2.2 Create Screen: Add Roles (Add-Roles) 50
9.2.3 Create Group Screen for Add Roles (Add-Roles) Screen 52
9.2.4 Check Add-Roles Screen in Render 52
9.2.5 Create Screen: Edit Roles (Edit-Roles) 53
9.2.5.1 Copy Add Roles Screen 53
9.2.5.2 Adjust Edit Roles Screen 53

9.2.6 Create Group Screen for Edit Roles (Edit-Roles) screen 54
9.2.7 Create Custom Multi Fields 54
9.2.8 Create Screen: All Roles (All-Roles) 56
9.2.9 Create Group Screen for All Roles (All-Roles) screen 59
9.3 Set up Start Screen for HR Group 59
9.4 Update Screens and Group Screens Weight 60
9.5 Check Screens for HR User in Render 60
10 Set up Environment for Manager Use Cases 62
10.1 Create Menu for Managers 62
10.2 Create todo_tasks Table 62
10.3 Set up Auto Sequence 63
10.4 Create Screens required in the Render for the Manager user 64
10.4.1 Create Custom Query 64
10.4.2 Copy Key Files and Templates for Writing Plugins 65
10.4.3 Create Plugin File 65
10.4.4 Create Extension Function 66
10.4.5 Restart Service 67
10.4.6 Create Screen: Create Task (M-Create-Task) 67
10.4.7 Create Group Screen for Create Tasks (M-Create-Task) Screen 71
10.4.8 Check Create Task (M-Create-Task) Screen in Render 71
10.4.9 Create Screen: Edit Task (M-Edit-Task) 72
10.4.10 Create Group Screen for Edit Task (M-Edit-Task) Screen 76
10.4.11 Create Manager_Task_Creation Workflow 76
10.4.12 Check Manager_Task_Creation Workflow in Render 78
10.4.13 Create Screen: Update Task (M-Update-Task) 78
10.4.13.1 Copy Edit Task Screen 78
10.4.13.2 Adjust Update Task (M-Update-Task) Screen 79
10.4.14 Create Group Screen for Update Tasks (M-Update-Task) Screen 79
10.4.15 Create Screen: All Tasks (M-All-Tasks) 80
10.4.16 Create Group Screen for All Tasks (M-All-Tasks) Screen 84
10.4.17 Create Alias Relationship for All Tasks Group Screen 85
10.5 Set up Start Screen for Managers Group 85
10.6 Create Task Priority (TaskPriority) List 86
10.7 Update Screens, Group Screens, and todo_tasks Table Parameters 87
10.8 Check Screens for Manager Users in Render 88

11 Set up Environment for Employee Use Cases 89
11.1 Create Menu for Employee 89
11.2 Create Screens required in the Render for the Employee user 89
11.2.1 Create Custom Query 89
11.2.2 Create Screen: Create Task (E-Create-Task) 90
11.2.2.1 Create E-Create-Task Screen from M-Create-Task Screen Copy 90
11.2.3 Create Group Screen for Create Tasks (E-Create-Task) Screen 91
11.2.4 Create Screen: Edit Task (E-Edit-Task) 91
11.2.5 Create E-Edit-Task Screen from M-Edit-Task Screen Copy 92
11.2.6 Create Group Screens for Edit Task (E-Edit-Task) Screen 92
11.2.7 Create Employee_Task_Creation Workflow 93
11.2.8 Create Screen: Update Task (E-Update-Task) 94
11.2.8.1 Create E-Update-Task Screen from M-Update-Task Screen Copy 94
11.2.9 Create Group Screen for Update Task (E-Update-Task) Screen 95
11.2.10 Create Screen: All Tasks (E-All-Tasks) 96
11.2.10.1 Create E-All-Tasks Screen from M-All-Tasks Screen Copy 96
11.2.11 Create Group Screen for All Tasks (E-All-Tasks) Screen 97
11.2.12 Create Alias Relationship for All Tasks Group Screen 97
11.3 Set up Start Screen for Employees Group 98
11.4 Check Screens for Employee Users in Render 99
12 User Onboarding Process 100
12.1 Presetting 100
12.1.1 Set up Architect Screen Access 101
12.1.2 Add Admin Group Rights to Users 101
12.1.2.1 Add Admin Group Rights to HR 102
12.1.2.2 Add Admin Group Rights to Managers 102
12.1.3 Check New Admin Users Access to Architect 103
12.1.4 Create Database Table: user_info 103
12.1.5 Create Document Family and Document Group 104
12.1.5.1 Create Document Family 104
12.1.5.2 Create Document Group 105
12.1.6 Add Document Group to Users 105
12.1.7 Create List of Employment Types (EmploymentType) 106
12.1.8 Create Days List 107
12.1.9 Create Custom Query for Job Title (JobTitle) 108

12.1.10 Create Custom Query for User Names (Usernames) 109
12.1.11 Create Custom Query for user_info_id 110
13 Set up User Information Tab 111
13.1 Create Screen: User Info 111
13.1.1 User Info Screen Header Configuration 113
13.1.2 User Info Screen Section Configuration 115
13.2 Create User Info Menu 122
13.3 Create Group Screens for User Info Screen 122
13.4 Check User Info Screen in Render 123
14 Set up Certificates Tab 124
14.1 Create certificates Table 124
14.2 Create Screen: Certificates 125
14.3 Set up Certificate Assignment to Users 128
14.4 Check Certificates Screen in Render 128
15 Set up Equipment Tab 129
15.1 Create equipment Table 129
15.2 Create Screen: Equipment 130
15.3 Set up Equipment Assignment to Users 134
15.4 Check Equipment Screen in Render 135
16 Set up Instructions Tab 136
16.1 Add Fields to user_info Table 136
16.2 Create Screen: Instructions 137
16.3 Create Custom Query for Instructions Screen 143
16.4 Add Custom Queries to Instructions Screen 145
16.5 Check Instructions Screen in Render 145
17 Set up Onboarding Process Workflow 147
17.1 Create Workflow Diagram for Onboarding Process 147
17.2 Check User Info Workflow in Render 149
18 Set up Notifications 151
18.1 Set up Email Server for Tenant 151
18.2 Update Email for Recipient (Manager User) 152
18.3 Create Notification Templates 152
18.3.1 Notifications for Group 152
18.3.2 Notifications for Custom Recipient 153
18.3.3 Notifications for Users 154

18.4 Configure Notifications for User Info Screen 155
18.5 Configure Notifications for Workflow Process 156
18.6 Check Notification in Render 157
18.6.1 Check Notification for Group 157
18.6.2 Check Notification for Custom Recipient 157
18.6.3 Check Notification for Users 158
18.7 Set up Extension Functions for Notification 158
18.7.1 Create Extension Function for Notification 158
18.7.2 Add Extension Function to Workflow Link 160
18.7.3 Check Extension Function for Workflow Link in Render 160
18.7.4 Create Extension Functions for Execute Screen 161
18.7.5 Create List with Extension Functions 163
18.7.6 Update user_info Table 164
18.8 Create Screen: Execute 165
18.8.1 Create Execute Menu 168
18.8.2 Create Group Screens for Execute Screen 168
18.8.3 Check Execute Screen in Render 169
19 Configure Reports 170
19.1 Prerequisites 170
19.1.1 Set up Reports Folder 170
19.1.2 Set up Custom Aliases 171
19.2 Create System Report 173
19.2.1 Create Report Template via QtRPT application 173
19.2.1.1 Create Report Template via QtRPT web application 173
19.2.1.2 Update Report Template via QtRPT web application 175
19.2.1.3 Create Report Template via QtRPT desktop application 176
19.2.1.4 Update Report Template via QtRPT desktop application 178
19.2.1.5 Create Environment for Reports Generation 179
19.2.1.6 Set up Document Settings 179
19.2.1.7 Set up System Report 180
19.2.1.8 Create Reports Menu for HR and Managers 181
19.2.1.9 Create Screen: Report 182
19.2.1.10 Create Group Screen for Reports 184
19.2.2 Check System Report Generation in Render 184
19.3 Create PDF Form Report Type 185

19.3.1 Create Report Template with PDF Form via Online PDF Editor 185
19.3.2 Create Report Template with PDF Form via Desktop PDF Editor 186
19.3.3 Create PDF Form Report 187
19.3.4 Update Report Screen 189
19.3.5 Check PDF Form in Render 190
19.4 Create Editable Report 190
19.4.1 Create Form Letter Template 190
19.4.2 Create Reports Template 192
19.4.3 Create Report Template with Editable Report (DS1) Type 194
19.4.4 Update Report Screen 197
19.4.5 Check Editable Report Generation in Render 198
20 Create Screen with all Field Types 200
20.1 Presetting 200
20.1.1 Create Demo Menu 201
20.1.2 Create Screen: Field Types 201
20.1.3 Create Group Screen for Field Types Screen 202
20.1.4 Update User Info Table 203
20.1.5 Create Custom Query 204
20.1.6 Check in Render 205
20.2 Configure Header Level of Field Types Screen 205
20.2.1 Check Field Types Screen Header in Render 213
20.3 Configure Sections Level of the Field Types Screen 213
20.3.1 Set up Sections' Switch 213
20.3.1.1 Configure tab for Full-time Employment Type 213
20.3.1.2 Configure Switched Section - Part-time tab 214
20.3.1.3 Configure Switched Section - Contract tab 214
20.3.1.4 Check Sections' Switch in Render 215
20.3.2 Configure the Switched Sections 215
20.3.2.1 Check the Switched Sections in Render 223
20.3.3 Configure the second section 223
20.3.3.1 Check the second section in Render 229
20.3.4 Configure the third section: Linked List and With Dependent Field 230
20.3.4.1 Check the Linked List and With Dependent Field Section in Render 234
20.3.5 Configure the fourth section: Label Iframe 234
20.3.5.1 Check Label Iframe Section in Render 234

20.3.6 Configure the fifth section (Edit Type): Document Types 235
20.3.6.1 Check Document Types Section in Render 236
20.3.7 Configure the sixth section (Document Type): Document Types Section 237
20.3.7.1 Check Document Types Section in Render 237
20.4 Create Screen: Table Type 237
20.4.1 Check Table Type Screen in Render 238
20.5 Configure the first Table section 239
20.5.1 Check the Table section in Render 242
20.6 Configure the second Table section 242
20.6.1 Check the second Table Section in Render 245
21 Create Environment for Orders Screen 246
21.1 Create user_oder Table 246
21.2 Set up Auto Sequence 247
21.3 Create Screen: Orders 247
21.4 Create Group Screen for Orders Screen 251
21.5 Check the Alias Dependencies and Relationship 252
21.6 Check Orders Screen in Render 252
22 Create Chart Screens 253
22.1 Create Screen: Chart_1 253
22.1.1 Chart-Pie Type 254
22.1.2 Chart-Doughnut Type 255
22.1.3 Chart-Line Type 255
22.1.4 Chart-Time-Series Type 256
22.2 Check Chart_1 Screen in Render 257
22.3 Create Screen: Chart_2 257
22.3.1 Chart-Bar-Horizontal Type 258
22.3.2 Chart-Bar-Vertical Type 259
22.3.3 Chart-Scatter-With-Linear-Regression Type (sort by x-axis) 260
22.3.4 Chart-Scatter-With-Linear-Regression Type (without sort by x-axis) 260
22.4 Check Chart_2 Screen in Render 261
23 Part of Form Functionality 262
23.1 Create Screen: Part of Form 262
23.1.1 General Screen Data 262
23.1.2 Header level 263
23.1.3 Section level 264

23.2 Create sections for modal window 264
23.3 Set up sections to display in a modal window 266
23.4 Create Group Screen 267
23.5 Check Part of Form Feature in Render 267
24 API Access from Screen 268
24.1 Create Guest Users 268
24.1.1 Create User 1 (GuestUser) 268
24.1.2 Create User 2 (GuestSystemUser) 269
24.2 Create Security Filters 269
24.3 Create Screen: GuestApi (without user authorization) 270
24.3.1 Copy Screen 270
24.3.2 Edit GuestApi Screen 270
24.4 Create Group Screen 270
24.5 Check GuestApi screen (without user authorization) 271
24.6 Create Screen: UserApi (with user authorization) 271
24.6.1 Create UserApi Screen from GuestApi Screen Copy 271
24.6.2 Create UserApi Group Screen 272
24.7 Check UserApi screen (with user authorization) 272
25 Screen (Edit type) with Fields Filled in via API 272
25.1 Create Basic Screen for API usage demonstration 273
25.1.1 General Screen Data 273
25.1.2 Header level 273
25.1.3 Section level 275
25.2 Create Group Screen 275
25.3 Check ApiLists Screen in Render 275
25.4 Configure Edit Type Section 276
25.5 Add Demo Data to the user_info Table 276
25.6 Add fields to be filled in via API 277
25.6.1 Add fields to be filled in via REST API 277
25.6.2 Add fields to be filled in via AppMill API 280
25.7 Check Screen in Render 283
26 Screen (Table type) with Fields Filled in via API 283
26.1 Add Demo Data to the equipment Table 283
26.2 Add fields to be filled in via API 284
26.2.1 Configure Table Type Section 284

26.2.2 Configure Columns 285
26.2.3 Configure Fields to be Filled in via REST API 285
26.2.4 Configure Fields to be Filled in via AppMill API 288
26.3 Check Screen in Render 289
27 External (third-party) plugins and libraries 290
27.1 Prerequisites 290
27.1.1 Update Document Family 290
27.1.2 Update Document Group 290
27.1.3 Update Report Screen 290
28 Third-Party Plugins Usage 291
28.1 Use Third-Party PHP Library in Plugin 291
28.1.1 Add Composer for Local Use 292
28.1.2 Install Spreadsheet Library via Composer 292
28.1.3 Create Plugin File and Auxiliary Folder 292
28.1.4 Create Extension Function 293
28.1.5 Add Button with Execute (possible document return) Action 293
28.2 Check PHP Library Usage in Render 294
29 Files Used in this Instruction 294

1 Introduction

This document will provide you a comprehensive overview and instruction of how to work with the AppMill solution, including what settings and activities to initially take in the Architect and Render applications. We will explore the simple, easy-to-follow instructions which guide you through setting up the Architect and Render applications using real-life usage examples that can be easily applied to any company processes. We will create and set up the functionality of the Task Manager and Onboarding Process Management system.

The Task Manager will contain the features to show how Task Management can be organized via AppMill solutions. In the provided instructions, we will demonstrate the entire setup procedure.

The Onboarding Management functionality helps any company to automate and streamline the onboarding process for new employees. You can build and customize any workflow that ensures new hires feel welcomed, informed, and prepared.

This guide will assist you in understanding how to configure the initial setup, go on with further settings, and fully rely on the Architect and Render tools. Additionally, we will add video instructions to the main and most complicated parts.

1.1 Overview of the Application and Features

The application being developed is a comprehensive Task and Onboarding Process Management system. The Task Manager is designed to streamline and automate the management of tasks. And the Onboarding Process Management system can improve the efficiency, effectiveness, and satisfaction of both new hires and HR teams, leading to better business outcomes overall.

The system leverages AppMill Architect and Render applications to demonstrate how these processes can be efficiently managed and customized to suit specific business needs.

1.1.1 Task Management Application Overview

The Task Management application being developed as a demonstration will provide automating and streamlining the workflows by providing clear task assignments, deadlines, and priorities. Employees can focus on high-impact tasks and reduce time wasted on redundant or unclear assignments.

The entire Task Management use cases will be covered by the application that is being developed and delivered with this instruction:

- **Task Assignment and Prioritization**: Easily assign tasks to team members, set priority levels, and establish clear deadlines. team productivity by ensuring everyone knows what they need to do and when, leading to faster project completion.
- **Progress Tracking**: Track the status of each task, from “to-do” to “in-progress” to “completed,” for clear project visibility. Managers can easily monitor task progress and deadlines, reducing bottlenecks and improving project delivery timelines.
- **Custom Notifications and Reminders**: Set up automated reminders and notifications to keep team members updated on approaching deadlines or changes.

By implementing a task management system, businesses can increase efficiency, improve collaboration, and ensure project success—all while maintaining accountability and clear communication.

1.1.2 Onboarding Process Application Overview

The HR application being developed as a demonstration will provide automating and streamlining the Onboarding Process for new employees.

By leveraging technology and a well-structured workflow, you can ensure HR users and Managers facilitate a smooth transition for new hires without placing too much administrative burden.

The entire Onboarding process use cases will be covered by the HR application that is being developed and delivered with this instruction:

- **Automated workflow** configured in the application will offer a standardized process that may grow effectively without losing a seamless and user-friendly experience for new employees.
- The **configured screens** will assist with compliance and ensure you do not lose or misfile important new hire information:
    - The **User Information screen** will allow HR users to fill in required personal information as well as job details of new hires.
    - The **Certification screen** will provide the possibility to upload and store the certificates and documents confirming the employee’s qualifications and compliance with the position.
    - The **Equipment screen** with the integrated approval process will ensure new employees are equipped with everything they need on the workstation.
        - The **automated approval process** (workflow hold step) that will be integrated into the Equipment step will automatically send out a notification to the manager and speed up the approval process. Managers of new hires often need to approve or re-assign certain requested equipment or configurations. Integrating the automated steps will save time and effort.
    - The **Instructions screen** will provide the possibility of an organized and transparent procedure for familiarizing new employees with the corporate standards and organizational guidelines, rules, and instructions, including defining practical employment norms and outlining written guidelines for internal cooperation.
- The **administrative part** will provide the environment of the application to cover the Onboarding process and the business logic we will demonstrate:
    - Building clear **permissions for multiple types of users** and giving them the corresponding **access rights** to the parts of application and screens.
    - Preparing all **system and demo data** (including database tables).
    - Configuration of **different field types**, their **layout** and **display** on UI according to the use cases (list of options, data lists, select box, check boxes, etc.).

- Organizing the **documentation management**. The documents that will be easily categorized according to purposes of usage and divided to the Document Groups (e.g. Common, Reports etc.). It will be possible to manage and control the access permissions and rights allowing to perform actions on documents (files) for different groups of users.
- Creating ready-to-use UI with **menu items**, **screens** with all required information, tabs to cover the Onboarding process use cases, etc.
- Building **automated workflow** with the **integrated notification** feature to ensure no important steps are missed and the responsible users are notified.
- Creating the **reports** to track each new employee's progress through the onboarding stage, and monitor the completion status.

Thanks to the configured and well-managed onboarding application, HR and managers can track in real time that new employees feel welcomed, informed, and prepared, leading to higher satisfaction and better integration.

1.1.3 Benefits of Using AppMill Solutions

By using AppMill Architect and Render applications, the time and effort required to set up and manage tasks and onboarding processes are significantly reduced compared to traditional programming methods. This allows you to quickly adapt to changing business needs, improve efficiency, and ensure a smoother experience for both administrators and employees.

Let us compare the time estimations for implementing and developing such an application from scratch versus using the provided AppMill solution.

1.1.4 Implementation and Development from Scratch

1. **Task Manager Tool:**
   - **Admin Use Cases:** 40 hours
   - **HR Use Cases:** 30 hours
   - **Manager Use Cases:** 40 hours
   - **Employee Use Cases:** 30 hours
   - **Customizable Task Screens:** 50 hours
   - **Automated Task Workflow:** 60 hours
2. **Onboarding Process:**
   - **Initial Setup:** 20 hours
   - **User Creation and Management:** 30 hours
   - **Onboarding Screens:** 40 hours

- **Setting up Workflow Automation and Notifications:** 50 hours
- **Configuring Reports:** 50 hours
- **Configuring Field Types Screen:** 40 hours

3. **API Access from Screen:** 40 hours
    - **Setting up Total Time for Implementation and Development from Scratch: 520 hours**

1.1.5 Implementation and Development Using Provided AppMill Solution

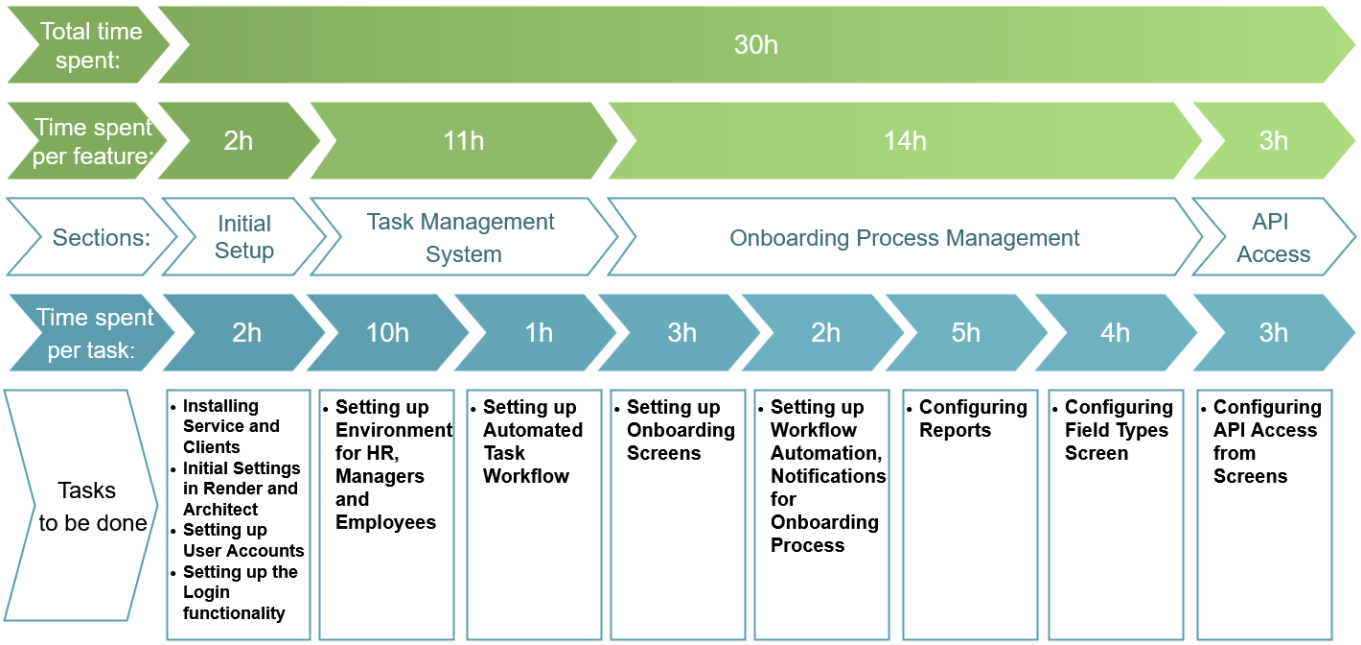
1. **Initial Setup:** 2 hours
2. **Task Manager Tool:**
    - **Environment for HR, Managers and Employees**: 2 hours
    - **HR Use Cases:** 2 hours
    - **Manager Use Cases:** 3 hours
    - **Employee Use Cases:** 3 hours
    - **Automated Task Workflow:** 1 hour
3. **Onboarding Process:**
    - **Initial Setup and Creating Onboarding Screens:** 3 hours
    - **Setting up Workflow Automation and Notifications:** 2 hours
    - **Configuring Reports:** 5 hours
    - **Configuring Field Types Screen:** 5 hours
4. **API Access from Screen:** 3 hours

Total Time for Implementation and Development Using Provided Solution: 30 hours

1.1.6 Summary

- **Total Time from Scratch:** 520 hours
- **Total Time Using Provided Solution:** 30 hours

By using the provided AppMill Architect and Render solutions, the time required to set up and develop the Task and Onboarding Process Management application is significantly reduced from 520 hours to 30 hours. This represents a substantial saving in time and effort, allowing companies to achieve faster deployment and improved efficiency.


Total time spent:
30h
Time spent per feature:
2h
11h
14h
3h
Sections:
Initial Setup
Task Management System
Onboarding Process Management
API Access
Time spent per task:
2h
10h
1h
3h
2h
5h
4h
3h
Tasks to be done
• Installing Service and Clients
• Initial Settings in Render and Architect
• Setting up User Accounts
• Setting up the Login functionality
• Setting up Environment for HR, Managers and Employees
• Setting up Automated Task Workflow
• Setting up Onboarding Screens
• Setting up Workflow Automation, Notifications for Onboarding Process
• Configuring Reports
• Configuring Field Types Screen
• Configuring API Access from Screens

2 Service and Clients Installation

Installing the Service and Clients applications is the first step.

2.1 Service Installation

Let us start with the initial necessary steps: setting up the client and the service.

Installing the Service should be completed first, while installing the Clients should come afterwards.

Open the Service install file and run it to start the installation of the service. Check the Service Installation Guide for further information.

Here, we configure the fundamental settings and parameters. The Installation Guide contains the comprehensive description and instructions.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Service Application Installation***.

2.2 Clients Installation

You can install the Client components after installing the service.

Since the AppMill Clients depend on the AppMill Service installation, the AppMill PHP Clients must be installed after the Service.

Here we set up the basic parameters and settings. The Installation Guide document provides the entire description and instructions.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Clients Application Installation***.

3 Initial Actions in Render

The following describes the first Render settings that need to be made in order to set up the system.

3.1 First Login to Render

Since the Login page has not been configured yet, you will need to sign in to Render with the Administrator name and password you created during the Service installation's Service Configuration phase in order to access the client applications.

Change Password is the first screen you encounter when you log into the Render for the first time. It is possible to reset the password and create a new one. Alternatively, you may go to Account Settings > Users in the Architect and turn off the Force change function.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***First Login to Render***.

4 Initial Settings in Architect

You have to complete the initial settings in the Architect in order to configure the system for your business needs. These configurations contain the precise order of actions that, irrespective of your company's type and daily schedule, must be followed prior to establishing the primary use cases.

Before defining the business logic, the following configurations are set up:

1. Log in to Architect - there are some specifics that should be considered.
2. Set up Account Settings:
    a. Add Account details (remove Force changes, add email, phone, etc.).
    b. Add logo images.
    c. Define Styles, Time zone, etc. (Edit Tenant).
3. Set up the Login functionality and specify the Login data required for the login to the Render.

We will investigate each step in further detail.

4.1 First Login to Architect

The initial login to the Architect application is the next step.

4.1.1 Predefined Data

The following information will be automatically predefined for you (as an Admin user) in the Account Settings when you first log into Architect with an empty database:

- **Users** - there is now only one user in the system: the Admin User.

- **Groups** - the Admin_Group will be automatically created by the system, and you, as the Admin User, will be assigned to this user group. Pay attention that the Start screen for the Admin group is not defined.
  The Login_Screen_Group (as well the Login screen as a Start screen) is created and assigned to the LoginScreenUser.

- **Tenant 1** - the Tenant will be automatically created by the system and be further used to create new users in the system. It provides the Styles and Regional settings (language, currency, date format, etc.) that are further inherited by all assigned users.

- **Security Filter** - for the combination of Tenant, Account Type and User Type values.

When you log in to the Architect application for the first time, use the password that was set in the First Login to Render step. In case the password was not changed in the First Login to Render step, use the password that was created in the Service Configuration step during Service installation.

The password that was set in the First Login to Render phase should be used the first time you log into the Architect application.

If the password was not modified during the First Login to Render step, apply the password that was generated during the Service Configuration step of the Service Installation process.

4.1.2 Reload Aliases

After logging in to the Architect and before the first screen creation, we will update the aliases.

This should be done to display the tables' list in the Primary Table drop-down list further during the first screen creation.

For that, in the Architect application, we open Features > Management and click the **Reload aliases** button. As a result, the aliases will be updated, and the tables' list will be shown in the Screen Builder.

4.1.3 Set up Document Devices

In this section, we set up the Document device (folder) in the Architect application where the files uploaded in the Architect and Render applications will be stored.

In this application we will use the Document device to store logo images (added in the Add Logo Images section) and the reports (generated for the Onboarding process). For more details about report generation, refer to the Configure Report section.

But first, make sure that the **system** document family contains the right document device name. In our case **C_SystemDevice**.

For that, in the Architect application, open Document Access > Document family. Here, open the **system** document family by clicking the Edit (pencil) icon. On the Update system page, the Device ID field should have the **C_SystemDevice** value.

Then, go to Document Access > Document devices and open the **C_SystemDevice** device update page. Pay attention that the device path should contain the right path to the directory or folder where the files

(images, reports, documents, etc.) uploaded by users in the Render and Architect applications will be stored. In our case **C:\\Codiac\\FileStorage**.

If you do not have the folder created automatically by the system:

- You can create a new folder for the existing path.

  Or

- You can create a new Document device and change the Device ID field value on the **system** document family (Document Access > Document family).

On the Document Family page (Document Access > Document Family), we can specify the retention time (in days) to store files in the Document device (folder). This can be useful to save space on your device. If the retention time is not defined, all files will be stored there until the folder is full. Note that if the folder is full of files, new files will replace the old ones.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Document Devices***.

4.1.4 Architect User Documentation

In case you need more details or information about any functionality in the Architect application, refer to the Architect User Documentation.

You can access Architect User Documentation from every page of the Architect application. On the Application menu, click the **Help** option to be redirected to the corresponding Architect User Documentation page. Use the documentation sidebar to browse for any feature description you need.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***First Login to Architect***.

Let us take a closer look at these pages and settings that should be performed.

5 Set up Account Settings

In the next step, we will set up the Account Settings in the Architect application.

Account Settings are designed to manage the data related to users such as users' details, user groups, tenants, logos used in the application, security filters, etc. The Account Settings features provide the possibility to set up custom branding, user access restrictions, etc.

Set up the following Account Settings:

1. Edit User details (remove Force changes, add email, phone, etc.).
2. Add logo images.
3. Define Styles, Time zone, etc. (Edit Tenant).

5.1 Edit User Details

Currently, the only user existing in the system is the Admin user. In the Architect application, open Account Settings > Users, then open the Admin User detail page.

If you did not change the password in the Render, you can set a new Password and disable the Force Change option in the User settings. Otherwise, on the next login to Render you will be forced to change your password again.

For a detailed description of the Users functionality, refer to the Architect User Manual (Account Settings > Users).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Edit User Details***.

5.2 Add Logo Images

In the Architect, you can add and set up the images that will be displayed as the header logo, the main logo, and the favicon (in the browser tab) in the Render.

These logo images will be defined in the Tenant.

To add your company logo, open Account Settings > Logo, select the type of logo, upload the image, and save the changes.

For a detailed description of the Logo functionality, refer to the Architect User Manual (Account Settings > Logo).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Add Logo Images***.

All images that were added as the Logo images are stored in the database. This database and other files used in the instruction can be found in the zip file (appmill_template > c_Codiac_FileStorage) at the top right corner of the page with video instructions.

5.3 Edit Tenant

The created Tenant will be assigned to the created users. It provides the Styles and Regional settings (language, currency, date format, etc.) that are further inherited by all assigned users.

In the Architect application, open Account Settings > Tenant, and edit the needed parameters by setting the data relevant for your organization.

- On the first tab, you can set up the logo uploaded previously on the Logo page and define the required information.
- On the second tab, you can set up language, currency, and date format. These settings will be inherited by all assigned users.
- On the third tab, you can set up styles and formatting for UI elements. Let’s keep the default button style for buttons and select the **M1: No Drop Menu** (no drop-down list) for the menu.

For a detailed description of the Tenant functionality, refer to the Architect User Manual (Account Settings > Tenant).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Edit Tenant***.

6 Set up Login Functionality

The Architect application provides the possibility to create and configure the Login functionality and additional options that can be used for Login procedure in the Render.

If the Login functionality is already set up in the Render and Architect applications after the Clients installation, the **Set up Login Functionality** section can be skipped. You can go to the **Example of Usage: Task Manager Tool** section to proceed with the Task Manager tool configuration.

If the Login functionality is not set up automatically after the Clients installation, we need to set up the user verification to the Render application via the Login functionality. That should be done as the access to the Render application should be provided only for users verified in the system.

In the Architect application, set up the Login functionality and define the Login data for the Render according to the steps below:

1. Create a **Login Group**.
2. Create a **Login User**.
3. Create a **Security Filter** that controls Security Level and sets the authentication method.
4. Create a **Menu**.
5. Create a **Login screen**.
6. Create a **Group screen**.
7. Set up the **Start screen**.
8. Configure a **Guest User** mode.
9. Log in to Render via Login Screen.

6.1 Create Login Group

The Login Group for users should be created in advance before a Login user as it is further used to create a Login user.

In the Architect application, open Account Settings > Groups and click the **Create** button.

Add values to the following fields:

- **Group Name** - enter **Login_Group**.
- **Group Description** - enter **Login Group**.

Leave the Start Screen empty so far until we create a Login screen several steps later.

Save the Login Group by clicking the Submit button.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Login Group***.

6.2 Create Login User

The Login User is a technical user that will be used for the authorization of the users in the Render. Also, the Login User is used to set up the Login functionality. I.e., the Login User will be automatically used by the system to verify and display the login screen. After that, users can use their user name and password to log in to the Render application.

In the Architect application, open Account Settings > Users and create a Login User.

Pay attention to the specific required fields:

- **User Name** - add a self-explanatory name, e.g., **LoginUser**.
- **Active Account** - select the check box to activate the **Login User**.
- **Account Type** - select the **Internal_User** account type.
- **Tenant Code** - select the tenant code from the drop-down list. In our case **1:Default**.
- **User Type** - select the **L: Login Account - Forced** type from the drop-down list.
- **Primary Group** - define the **Login Group** created preliminary. The Primary group defines the Start screen that will be opened for the user in the Render. In our case, it will be a Login screen that we set up further.
- **Group Area** - the group area will be selected automatically according to the Login Group defined in the Primary Group field.
- **Phone** - enter the phone number.
- **Email** - enter an email address.
- **Account Name** - select the account name from the drop-down list. In our case **LoginUser account**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Login User***.

6.3 Create Security Filter for Login User

In this step, we will create the Security Filter for the Login User. This should be done to manage the security levels for users in the Render application, restrict data accessibility and search results based on predefined parameters, and set authentication methods based on their security level.

In the Architect application, open Account Settings > Security Filters and create new Security Filter for Login User.

Pay attention to the specific required fields:

- **Tenant** - select the Tenant created previously.

- **Account Type** - select the Account type. In our case **Internal_User: UserType**.
- **User Type** - select the **L: Login Account - Forced** value from the drop-down list.
- **Description** - enter a short description. In our case **For Login**.
- **Session Lifetime** - let us set the time of session. The standard php value is 1200 seconds = 20 minutes.
- **Authorization Type** - for our example, we select **Login Authentication**.

Save the changes. The Security Filter for Login User has been created.

For a detailed description of the Security Filters functionality, refer to the Architect User Manual (Account Settings > Security Filters).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Security Filter for Login User***.

Let's go to the next step.

6.4 Create Login Menu

We need to create a Menu to further assign to this menu item the Login screen and group screen.

In the Architect application, open Screen Configurations > Menu and click the Create button.

Select the following options:

- **Group Name** - select the created **Login_Group** for users.
- **Menu Text** - enter the menu title, e.g., **LoginMenu**.
- **Menu Icon** - we do not need to define the menu icon as this menu will not be displayed anywhere on the UI.
- **Description** - add a short description, e.g., **Login Menu**.
- **Weight** - set weight to 0.

Save the menu by clicking the Submit button.

For a detailed description of the Menu functionality, refer to the Architect User Manual (Screen Configurations > Menu).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Login Menu***.

6.5 Create Login Screen

Now we can set up the Login screen. It will be displayed for users when they log in to Render.

In the Architect application, open Screen Configurations > Screens. Click the **Create with Builder** button.

On the first page of Screen builder, pay attention to the specific fields:

- **Add New Screen** - enter the self-explanatory title for the screen. In our case **LoginScreen**.
- **Screen Description** - add a short screen description. In our case **LoginScreen**.
- **Primary Table** - select the **c_user_accounts** DB table as we set up the login to the system.
- **Tab Weight** - enter **0**.
- **Tab Text** - enter a name. It will be displayed in the menu in the Render. In our case **Login Screen**.
- **Tab Description** - add a clear description for the screen. In our case **Login Screen**.
- **Screen State** - ensure that **View** screen state is selected.
- **Login Screen** - select the check box to activate the Login-specific settings for the screen.
- **Screen Layout** - select **1x1** format.

And go to the next page. The Login Screen has predefined tabs with different variants of settings.

Here, we will make a few changes.

Let us hide the buttons that we do not need to show now on the Login screen:

- Open the Registration button settings and select the Hidden check box to hide this button from the UI, as we do not need the Self-Registration feature now.
- Open the Password Reset button settings and select the Hidden check box to hide this button from the UI, as we do not have the Password reset screen set up.

Other fields on this tab should be configured:

- **Image** - you can use your logo to display it on the Login screen. Open Settings by clicking the corresponding icon.
- **Account Name** - the field will be used to enter the User account name. Click the Settings icon to configure the field.
  You can improve and change any setting you need, e.g., rename the label or edit the formatting of the font and size. But in our case:
  - On the **Primary** tab, we will leave the label name, change the **Label orientation**, and set it to **Inside**.
  - On the **Tab Order** tab, we will select the **Allow Tabs - Tab Order** parameter and set it to **10**.
- **Password** - the field will be used to enter the Password to log in to Render. Click the **Settings** icon to configure the field.
  In our case:
  - On the **Primary** tab, we will change the **Label orientation** to **Inside**.

    - On the **Tab Order** tab, we will select the **Allow Tabs - Tab Order** parameter and set it to **20**.

- **Login** button - the button will be used to log in to the application.
  Click the **Settings** icon to configure the button.

    - On the **Tab Order** tab, we will select the **Allow Tabs - Tab Order** parameter and set it to **30**.

Save the screen by clicking the **Create** button.

For a detailed description of the Login Screen configuration, refer to the Architect User Manual (Screen Configurations > Screens).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Login Screen***.

6.6 Create Group Screen for Login Screen

Now we need to link the created screen and User Group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens.

Set the specific parameters:

- **Add** - select the **Add** option.
- **Group Name** - select the **Login_Group** we have created previously.
- **Menu Name** - select the **LoginMenu** we have created previously.
- **Screen Name** - select the **LoginScreen** we have created previously.
- **Screen Text** - is a required field, and we have to fill it in. But this text will not be displayed anywhere on the Login Screen.
- **Weight** - let us set it to **0**.

Submit changes on the page.

For a detailed description of the Group Screen configuration, refer to the Architect User Manual (Screen Configurations > Group Screens).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Login Screen***.

6.7 Set up Start Screen

The last step in Login screen configuration is to set the Start screen for the user.

For this, in the Architect application, open Account Settings > Groups. Find and open the Login group we have created. Select the Login screen for the Start screen field.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Start Screen***.

6.8 Configure Guest User mode

The next operations and settings should be done in the Codiac **param.local.php** config file.

These settings should be done as part of the Login functionality settings to be able to use the Login Screen to log into the Render application.

For this, open the **params-local.php** file located under the following path: ***Program Files/AppMill/Clients/codiac/config***.

In the opened file, set the following parameters for guestUserMode:

```
'guestUserMode' => 'auto', //false || 'auto' || 'option'
```

If you use 'guestUserMode' => 'auto', then enter the username and password for the guest user.

```
'guestUserCredentials' => [
'username' => 'LoginUser',
'password' => 'Password1'
],
```

Use the username and password of the created LoginUser (you can check the data in Account Settings > Users > LoginUser).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Guest User mode***.

6.9 Log in to Render via Login Screen

After completing the previously described settings, open the Render application and reload the Login screen to log in for the first time. It will show the configured Login Screen.

On the Login Screen, enter the Administrator Name and Password that were created in the Service Configuration step during Service installation. In case the password was changed in the First Login to Render step, use that password.

To finish the login process, click the **Login** button. The Render application opens.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Log in via Login Screen***.

6.10 Update Login Screen

To adjust the Login Screen display in the Render application, we need to update the Login Screen in the Architect application.

For that, we will perform the following actions:

1. Log into the Architect application.
2. In the Architect application, open Screen Configurations > Screens and find the LoginScreen.
3. Open the LoginScreen in the Screen Builder
4. Click the Next button at the bottom right corner of the page, to open the second page of the LoginScreen
5. On the second page, we need to change field positions on the screen and fields' style. The style of the following fields will be corrected:
    a. **Account Name** - define the format of this field. Open the Account Name Field Settings pop-up window. Here, click the Formatting page at the bottom of the pop-up window. In the opened Formatting page, select the Text Color, Text Background Color, Font Family, and Font Size for Label and Field. Save changes and return to the screen editor.
    b. **Password** - define the format of this field. Open the Password Field Settings pop-up window. Here, click the Formatting page at the bottom of the pop-up window. In the opened Formatting page, select the Text Color, Text Background Color, Font Family, and Font Size for Label and Field. Save changes and return to the screen editor.
6. Open the **Section Settings** pop-up window and select the **Hide section header** check box.
7. Click the **Update** button at the bottom right corner of the page, to submit the Login Screen updates.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Login Screen***.

6.11 Update Styles for Login User

In case you need to update the Login Screen according to corporate styles, the Login Screen styles for the Login User can be edited.

In this step, we will show how to update the Login Screen styles in the Render application for Login User.

For that, in the Architect application, open Account Settings > Users and select the LoginUser account to be edited.

In the opened LoginUser account page, and perform the following steps to update styles for LoginUser:

- Open the Styles tab.
- In the Section for Login Screen section, update the parameters below:
    - **Section Border Size** - select the **1px** value from the drop-down list.
    - **Border color** - select the **#45818e** color for the border.

- **Use Images** - upload an image from the local computer. This image will be shown in the Render application on the Login Screen page.

- In the Body section, update the parameters below:
  - **Background color** - select the **#d9d9d9** color for the background.
  - **Use background linear gradient** - select the **Use gradient** option from the drop-down list.
  - **Gradient rotation** - enter **45**.
  - **First color** - select the **#76a5af** color as the first color of the gradient.
  - **Second color** - select the **#6aa84f** color as the first color of the gradient.

Submit changes on the page.

To check the updated Login Screen styles, we need to open the Render application.

In case the updated Login Screen styles are not displayed, clear web browser cookies.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Styles for Login User***.

7 Example of Usage: Task Manager Tool

Now, we are prepared for the setting up of any business logic. Let us set up the environment for the use case of Task Manager Tool.

7.1 User Roles

For our use case, we will create the HR user and two teams with Managers and Employees:

HR:

- HR user:
    - Kent Kim

Team A:

- Manager:
    - John Smith
- Employees:
    - Paul Powell
    - Diana Duke

Team B:

- Manager:
    - Jane Hanson
- Employees:
    - Harry Hagman
    - Anna Adams
    - Peter Preston

7.2 Screens

We will create the following screens for the HR user:

1. All Roles – the screen that displays all users' roles. The All Roles screen will be used as the start screen.
2. Add Roles – the screen that provides the possibility to add a role to the user.
3. Edit Roles – the screen that provides the possibility to edit user roles.

We will create the following screens for Team A and Team B managers:

1. All Tasks – the screen that displays all Manager and Employee users' tasks. The All Tasks screen will be used as the start screen for managers.
2. Create Task – the screen that provides the possibility to create a new task and assign it to the Manager or Employee user.
3. Edit Task – the Edit Task screen is the second step of the task creation process, and will appear when the Manager user submits a new task creation. Here, the Manager user will fill in the data for the just-created task.
4. Update Task – the screen that provides the possibility to update or delete the earlier-created task.

And for Team A and Team B employees:

1. All Tasks – the screen that displays all tasks assigned to the Employee user. The All Tasks screen will be used as the start screen for employees.
2. Create Task – the screen that provides the possibility to create a new task and assign it to the Employee user.
3. Edit Task – the Edit Task screen is the second step of the task creation process, and will appear when the Employee user submits a new task creation. Here, the Employee user will fill in the data for the just-created task.
4. Update Task – the screen that provides the possibility to update the earlier-created task.

7.2.1 Start Screen for HR User

When the HR user logs in to the Render. The Start Screen is displayed.

The Start Screen contains the following sections:

- The screen **Header** that contains the **Add** button.
- The screen **Body** that consists of the table with all users' roles.

  The table contains the following columns:

User	Manager	Description	Actions
			View / Edit

Also, you can open the Start Screen from the other screens via the All Roles menu and the All Roles menu item.

7.2.2 Start Screen for Manager Users

When the **Manager** user logs in to Render. The Start Screen is displayed.

Also, you can access the Start Screen from the other screens via the **Tasks** menu and the **All Tasks** menu item.

The **Start Screen** contains the following sections:

- The screen **Header** that contains the **Add Task** button.

- The screen **Body** that consists of:
    - the **Search** field that the Manager can use to filter the appropriate task.
    - the table with all tasks assigned to the Manager and Employee users.

      The table contains the following columns:

Task ID	Summary	Assigned to	Created on	Priority	Due Date	Actions
						View / Update

    - The list of tasks can be sorted by any table column except the Actions column.

7.2.3 Start Screen for Employee Users

When the **Employee** user logs in to Render. The **Start Screen** is displayed.
Also, you can access the Start Screen from the other screens via the **Tasks** menu and the **All Tasks** menu item.

The **Start Screen** contains the following sections:

- The screen **Header** that contains the **Add Task** button.
- The screen **Body** that consists of:
    - the **Search** field that the Employee can use to filter the appropriate task.
    - the table with all tasks assigned to the Manager and Employee users.

      The table contains the following columns:

Task ID	Summary	Assigned to	Created on	Priority	Due Date	Actions
						View / Update

    - The list of tasks can be sorted by any table column except the Actions column.

7.3 Use Cases

Let us look at the possible use cases that we can cover with the AppMill system.

7.3.1 Admin Use Cases List

Admin sets up all required environments for the Task Management Tool.

Use Case 1: The possibility to create users and user groups (HR, Managers, and Employees).

Use Case 2: The possibility to create and set up the database table, application menu, screens, group screens, security filters, custom queries, and auto sequence for the HR user to provide them the possibility to manage Manager and Employee users’ roles.

Use Case 3: The possibility to create and set up the table, menu, screens, group screen, security filters, custom queries, auto sequence, workflow, and alias relationships for Manager users to provide them access to manage and control the task creation and edition process, also, to assign tasks to Employee users.

Use Case 4: The possibility to create and set up the table, menu, screens, group screen, security filters, custom queries, and workflow for Employee users to provide them access to manage assigned to them tasks.

7.3.2 HR Use Cases

Use Case 1: The possibility to create users' roles.

Use Case 2: The possibility to add Manager and Employee roles for users.

Use Case 3: The possibility to edit users' roles.

Use Case 4: The possibility to manage a relationship between the Manager and Employee users.

7.3.3 Detailed HR Use Cases

1. The HR user has the possibility to assign the Manager to Employees for the users existing in the system.
    a. The HR user opens the All Roles screen.
    b. The HR user clicks the Add button to add a new role.
    c. In the opened screen, the HR user selects the User ID and Manager ID from the drop-down lists and fills in the Description field.
    d. The user will be added to the Manager's team as an Employee.
2. The HR user has the possibility to assign Manager roles to the users existing in the system.
    a. The HR user opens the All Roles screen.
    b. The HR user clicks the Add button to add a new role.
    c. In the opened screen, the HR user selects the same User ID and Manager ID from the drop-down lists and fills in the Description field.
    d. The user is defined as the Manager user.

7.3.4 Manager Use Cases

Use Case 1: The possibility to create tasks (to-do list) - add description, change task status, fill in fields (Due Date, Priority).

Use Case 2: The possibility to assign the task to an employee and un-assign it.

Use Case 3: The possibility to edit the task details.

Use Case 4: The possibility to delete the tasks.

Use Case 5: The possibility to view the list of all tasks, filter search results, and search for a task.

Use Case 6: The possibility to view the list of the tasks assigned to the selected employee.

Use Case 7: The possibility to view the workflow steps of the task.

7.3.5 Detailed Manager Use Cases

1. The Manager user has the possibility to create Task from the Start Screen:
    a. The Manager user clicks the **Add Task** button placed above the board.
    b. The **Create Task** screen is opened.
    c. In the screen Header, the Manager user can view the following parameter:
        i. Task ID - is generated automatically after saving.
    d. The Manager user fills in the optional fields in the body:
        i. Summary - text.
        ii. Priority - select box.
        iii. Created on - automatically updated based on the current date and time.
        iv. Due Date - automatically updated based on Priority.
    e. The Manager user clicks the Save button.
    f. Task is saved.
    g. The **Edit Task** screen is opened to proceed with the creation process.
    h. The Manager user edits the newly created Task.
        i. Assigned to - select box.
        ii. Status - select box.
        iii. Description - textarea.
        iv. Comments - textarea.
    i. The Manager user clicks the Save button.
    j. Task is created.
2. The Manager user can update the Task:
    a. The Manager user clicks the View / Update button.
    b. The **Update Task** screen is opened.
    c. The Manager user clicks the Edit button.
    d. The following fields can be edited: Status, Assigned to, Description, Comments.
    e. User edit the fields.

f. The Manager user clicks the Save button.

g. Task is updated.

3. The Manager user can delete Task:

a. The Manager user clicks the View / Update button.

b. The **Update Task** screen is opened.

c. The Manager user clicks the Delete button.

d. Confirmation pop-up window appears.

e. The Manager user clicks Ok to confirm the task deletion.

f. The Update Task screen field values are cleared.

g. The Manager user closes the Update Task screen to finish the task deletion.

4. The Manager user can view all tasks assigned to him and his Team:

a. The Manager user clicks the **Tasks** menu item.

b. The **All Tasks** screen is opened.

c. The screen contains the list of tasks (in table view):

Task ID | Summary | Assigned to | Priority | Due Date | Actions

View / Update

d. The Manager user can filter Tasks using the Search field or any value from the drop-down list of the table column.

e. The Manager user can sort Tasks by any table column except the Actions column.

f. The Manager user can open Task in view mode by clicking the View / Update button to display the task details.

g. The Manager user can open Task in edit mode by clicking the View / Update button and then the Edit button to update the task details.

7.3.6 Employee Use Cases

Use Case 1: The possibility to create tasks - add descriptions, fill in fields.

Use Case 2: The possibility to assign the task to an employee and un-assign it.

Use Case 3: The possibility to edit the task or change the task status.

Use Case 4: The possibility to view the list of assigned tasks (All Tasks), filter search results, and search for a task.

Use Case 5: The possibility to view the workflow steps of the task.

7.3.7 Detailed Employees Use Cases

1. The Employee user has the possibility to create Task from the Start Screen:
    a. The Employee user clicks the **Add Task** button placed above the board.
    b. The **Create Task** screen is opened.
    c. In the screen Header, the Employee user can view the following parameter:
        i. Task ID - is generated automatically after saving.
    d. The Employee user fills in the optional fields in the body:
        i. Summary - text.
        ii. Priority - select box.
        iii. Created on - automatically updated based on the current date and time.
        iv. Due Date - automatically updated based on Priority.
    e. The Employee user clicks the Save button.
    f. Task is saved.
    g. The **Edit Task** screen is opened to proceed with the creation process.
    h. The Employee user edits the newly created Task.
        i. Assigned to - select box.
        ii. Status - select box.
        iii. Description - textarea.
        iv. Comments - textarea.
    i. The Employee user clicks the Save button.
    j. Task is saved.
2. The Employee user can update the Task:
    a. The Employee user clicks the View / Update button.
    b. The **Update Task** screen is opened.
    c. The Employee user clicks the Edit button.
    d. The following fields can be edited: Status, Assigned to, Description, Comments.
    e. User edit needed fields.
    f. The Employee user clicks the Save button.
    g. Task is updated.

3. The Employee user can view the tasks assigned to him:
    a. The Employee user clicks the **Tasks** menu item.
    b. The **All Tasks** screen is opened.
    c. The screen contains the list of tasks (in table view) that are assigned to the current user:

        Task ID | Summary | Assigned to | Priority | Due Date | Actions

        View / Update

    d. The Employee user can filter Tasks using the Search field or any value from the drop-down list of the table column.
    e. The Employee user can sort Tasks by any table column except the Actions column.
    f. The Employee user can open Task in view mode by clicking the View / Update button to display the task details.
    g. The Employee user can open Task in edit mode by clicking the View / Update button and then the Edit button to update the task details.

8 Set up Environment for HR, Managers and Employees

To manage all users in the system, we need to introduce a new role - HR employee. This is a user from the HR department responsible for the distribution of user roles and responsibilities.

To set up the environment for the HR user, the Admin User has to execute the following steps:

8.1 Create Groups for HR, Managers and Employees

First, we need to create groups for HR, Manager, and Employee users.

In the Architect application, open Account settings > Group and create the following user groups:

- For HR user:
    - **Group Name** - enter a value. In our case **HR**.
    - **Group Description** - enter **HR**. But you can add more details if needed.

  Submit changes on the page.
- For Manager users:
    - **Group Name** - enter a value. In our case **Managers**.
    - **Group Description** - enter a value. In our case **Managers**.

  Submit changes on the page.
- For Employee users:
    - **Group Name** - enter a value. In our case **Employees**.
    - **Group Description** - enter a value. In our case **Employees**.

  Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create User Groups***.

8.2 Set up Security Filters

In this step, we will set up security filters for HR, Managers, and Employees, to manage security levels for users in the Render, restrict data accessibility and search results based on predefined parameters, and set authentication methods based on their security level.

8.2.1 Set up Security Filter for HR and Managers

For HR and Manager users we will use the Security Filter that is already created by the system.

In the Architect application, open Account Settings > Security Filters and update filters for HR and Manager users.

As the security filter that was created by the system has the following parameters:

- **Tenant** - enter a tenant name. In our case **1:Default**.
- **Account Type** - enter an account type. In our case **UserType.Internal_User**.
- **User Type** - enter a user type. In our case **U: Basic User**.
- **Allow change password** - select the check box.
- **Allow change settings** - select the check box.
- **Authorization Type** - select the **L: Login Authentication** authorization type.

We just need to update the security filter **Description**. For that, we enter a short description into the Description field. Enter the **Basic** value.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Security Filter for HR and Managers***.

8.2.2 Create Security Filter for Employee

For the Employee users, we need to create a new Security Filter.

In the Architect application, open Account Settings > Security Filters and click the Create button to create a new Security Filter.

Pay attention to the specific required fields:

- **Tenant** - select the Tenant created previously. In our case **1:Default**.
- **Account Type** - select the Account type. In our case **Internal_User: UserType**.
- **User Type** - select User Type added. In our case **I: Internal user**.
- **Description** - enter a short description. In our case **For Employee**.
- **Session Lifetime** - let us set the time of session. The standard php value is 1200 seconds = 20 minutes.
- **Authorization Type** - for our example, we select **Login Authentication**.
- **Allow change password** - select the check box.

Save the changes. The rule for Employee users has been created.

For a detailed description of the Security Filters functionality, refer to the Architect User Manual (Account Settings > Security Filters).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Security Filter for Employee***.

8.3 Create all Users

Let’s create the users that will be used in our system.

According to our use cases, we will create the following users and user groups:

- HR user.
- Team A with one Manager user and two Employee users.
- Team B with one Manager user and three Employee users.

In the Architect application, open Account Settings > Users and create users:

- Create a **HR User** named **Kent Kim**:
    - **Username** - enter the **KentKim** username.
    - **Password** - create a password for the HR user.
    - **Active Account** - select the check box.
    - **Account Type** - select the **Internal_User** from the drop-down list.
    - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
    - **User Type** - select the user type from the drop-down list. In our case **U: Basic User**.
    - **Primary Group** - select the **HR** group from the drop-down list.
    - **Group Area** - select the **HR** group from the drop-down list.
    - **Phone** - enter a phone number.
    - **Email** - enter an email address.
    - **Account Name** - enter the **Kent Kim** account name.

    Submit changes on the page.
- Create a **Manager** user named **John Smith** for **Team A**:
    - **User Name** - enter the **JohnSmith** user name.
    - **Password** - create a password for the Manager A user.
    - **Active Account** - select the check box.
    - **Account Type** - select the **Internal_User** from the drop-down list.
    - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
    - **User Type** - select the user type from the drop-down list. In our case **U: Basic User**.
    - **Primary Group** - select the **Managers** group from the drop-down list.
    - **Group Area** - select the **Managers** group from the drop-down list.

- **Phone** - enter a phone number.
- **Email** - enter an email address.
- **Account Name** - enter the **John Smith** account name.

Submit changes on the page.

- Create an **Employee** user named **Paul Powell** for **Team A**:
  - **Username** - enter the **PaulPowell** username.
  - **Password** - create a password for the Employee A1 user.
  - **Active Account** - select the check box.
  - **Account Type** - select the **Internal_User** from the drop-down list.
  - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
  - **User Type** - select the user type from the drop-down list. In our case **I: Internal user**.
  - **Primary Group** - select the **Employees** group from the drop-down list.
  - **Group Area** - select the **Employees** group from the drop-down list.
  - **Phone** - enter a phone number.
  - **Email** - enter an email address.
  - **Account Name** - enter the **Paul Powell** account name.

  Submit changes on the page.
- Create an **Employee** user named **Diana Duke** for **Team A**:
  - **Username** - enter the **DianaDuke** username.
  - **Password** - create a password for the Employee A2 user.
  - **Active Account** - select the check box.
  - **Account Type** - select the **Internal_User** from the drop-down list.
  - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
  - **User Type** - select the user type from the drop-down list. In our case **I: Internal user**.
  - **Primary Group** - select the **Employees** group from the drop-down list.
  - **Group Area** - select the **Employees** group from the drop-down list.
  - **Phone** - enter a phone number.
  - **Email** - enter an email address.
  - **Account Name** - enter the **Diana Duke** account name.

Submit changes on the page.

- Create a **Manager** user named **Jane Hanson** for **Team B**:
    - **Username** - enter the **JaneHanson** username.
    - **Password** - create a password for the Manager B user.
    - **Active Account** - select the check box.
    - **Account Type** - select the **Internal_User** from the drop-down list.
    - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
    - **User Type** - select the user type from the drop-down list. In our case **U: Basic User**.
    - **Primary Group** - select the **Managers** group from the drop-down list.
    - **Group Area** - select the **Managers** group from the drop-down list.
    - **Phone** - enter a phone number.
    - **Email** - enter an email address.
    - **Account Name** - enter the **Jane Hanson** account name.

  Submit changes on the page.
- Create an **Employee** user named **Harry Hagman** for **Team B**:
    - **Username** - enter the **HarryHagman** username.
    - **Password** - create a password for the Employee B1 user.
    - **Active Account** - select the check box.
    - **Account Type** - select the **Internal_User** from the drop-down list.
    - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
    - **User Type** - select the user type from the drop-down list. In our case **I: Internal user**.
    - **Primary Group** - select the **Employees** group from the drop-down list.
    - **Group Area** - select the **Employees** group from the drop-down list.
    - **Phone** - enter a phone number.
    - **Email** - enter an email address.
    - **Account Name** - enter the **Harry Hagman** account name.

  Submit changes on the page.
- Create an **Employee** user named **Anna Adams** for **Team B**:
    - **Username** - enter the **AnnaAdams** username.

- **Password** - create a password for the Employee B2 user.
- **Active Account** - select the check box.
- **Account Type** - select the **Internal_User** from the drop-down list.
- **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
- **User Type** - select the user type from the drop-down list. In our case **I: Internal user**.
- **Primary Group** - select the **Employees** group from the drop-down list.
- **Group Area** - select the **Employees** group from the drop-down list.
- **Phone** - enter a phone number.
- **Email** - enter an email address.
- **Account Name** - enter the **Anna Adams** account name.

Submit changes on the page.

- Create an **Employee** user named **Petr Preston** for **Team B**:
  - **Username** - enter the **PetrPreston** user username.
  - **Password** - create a password for the Employee B3 user.
  - **Active Account** - select the check box.
  - **Account Type** - select the **Internal_User** from the drop-down list.
  - **Tenant Code** - select the tenant code from the drop-down list. In our case **1: Default**.
  - **User Type** - select the user type from the drop-down list. In our case **I: Internal user**.
  - **Primary Group** - select the **Employees** group from the drop-down list.
  - **Group Area** - select the **Employees** group from the drop-down list.
  - **Phone** - enter a phone number.
  - **Email** - enter an email address.
  - **Account Name** - enter the **Petr Preston** account name.

  Submit changes on the page.

Pay attention to the user's Primary Group. You need to select the correct Primary Group, which the users belong to.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create all Users***.

8.4 Edit UI Styles for Users

Editing UI styles for users can significantly impact the usability, accessibility, and overall experience of the Render application.

Allowing users to customize themes or styles (e.g., dark mode, font sizes, menu types, etc.) can make the Render application more comfortable and enjoyable to use.

The UI style and theme according to corporate style or individual preferences can be specified:

- In the Render application:
    - on the User's Settings page (My Account > Settings) you can define the UI look and feel for the current user.
- In the Architect application:
    - in the User's settings (Account Settings > Users > User's Details > Styles tab) you can customize the individual Styles for the selected user.
    - in the Tenant settings (Account Settings > Tenant > Styles tab) you can set up the common Styles that will be applied to the users belonging to the Tenant.

Note if you change the Tenant value while editing a user, the style settings specified for a new selected Tenant will be applied.

To demonstrate how UI Styles can be updated for the appropriate user, we will describe the example below:

On the Settings page of the Render application

Log into the Render application as Jane Hanson.

In the Render application, on the Settings page (My Account > Settings), we will select the **M5: Menu with left navigation bar** menu type from the **Menu type** drop-down list and confirm changes. As we can see, the default menu style was changed for Jane Hanson.

To return to the previous menu type, we will select **M1: No Drop Menu** (is used by default) menu type from the **Menu type** drop-down list and confirm changes.

After that we can change Body styles. For that, in the Body section we will define the following parameters:

- **Border size** - select **1px** value from the drop-down list.
- **Border color** - select the **#00ffff** color.
- **Background color** - select the **#fff2cc** color.
- **Link color** - select the **#a61c00** color.
- **Text color** - select the **#cc4125** color.

Click the **Update** button at the bottom right corner of the Settings page, to apply style changes.

We can manage absolutely every Menu Format, Body, Section, Header, etc. parameters on the Settings page. To reset UI style changes, click the **Default settings** button.

On the Styles tab of the Users page in the Architect application

In the Architect application, on the Styles tab of the Users page (Account Settings > Users), we will update UI styles for the appropriate user. For that:

Log into the Architect application as an Admin user and select the JaneHanson user account.

Open the Styles tab of the Users page (**Account Settings > Users**). Here, we will not change the UI default styles and click the **Update** button at the bottom right corner of the Users page to apply the default style.

Go to the Render application, log out, and then log back to the Render. This should be done to apply style changes from the Architect application.

Please note that to apply some Styles, users need to log out and then log back into the Render application.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Edit UI Styles for Users***.

8.5 Create Tables to link Managers and Employees

This table is used to link users and managers. According to this relationship, the manager will be able to view all tasks (his/her own tasks and the tasks assigned to the team). Users will be able to view only their own tasks.

In the Architect application, open Features > Tables and create a new table for HR to link Managers and Employees.

Pay your attention to the Database type specific rules. It is recommended to specify a length for the VARCHAR and CHAR columns as it is required for MySQL and MSSQL databases. For example, the default row length for MySQL is 16,383. That’s why you need to specify the length of the VARCHAR and CHAR columns.

Create a new **users_roles** table with the following fields:

- Table Name - enter a table name. In our case **users_roles**.
- In the Fields/Column Details section define the following table parameters:

Name	**Type**	**Nullable**
roles_id	Integer	False
user_id	Integer	True
manager_id	Integer	True

description	Text	True

The first constraint type and name:

- Constraint Type - select the **Primary Key** constraint type.
- Constraint Name - enter the **users_roles_pk** constraint name.
- Columns - select the **roles_id** column.

The second constraint type and name:

- Constraint Type - select the **Unique Key** constraint type.
- Constraint Name - enter the **users_roles_uk** constraint name.
- Columns - select the **user_id** column.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Tables to link Managers and Employees***.

8.6 Set up Auto Sequence

Create a new Auto Sequence rule to create a unique role ID for every user.

In the Architect application, open Features > Auto Sequence and create a new Auto Sequence rule to create a unique role ID for Manager users.

- **Table Name** - select the **users_roles** table from the drop-down list.
- In the Fields Details section, define the following parameters:
    - **Field Name** - enter the **roles_id** field name.
    - **Increment by** - enter **1**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Auto Sequence***.

9 Set up Environment for HR Use Cases

To prepare the environment for HR's Use Cases, the Admin User needs to perform the steps described further.

9.1 Create Menu for HR

Create Menu for the HR user. This menu will provide access to the Render screen for the possibility to set up the associations between managers and employees.

In the Architect application, open Screen Configurations > Menu and create Menu for the HR user.

Create Menu with the following parameters:

- **Group Name** - select the **HR** group.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Roles**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. In our case **Roles**.
- **Weight** - enter **0**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Menu for HR***.

9.2 Create Screens required for the HR user in the Render

To cover all use cases of the HR user, we need to create 3 screens:

- **All Roles (All-Roles)** screen, where all roles that were created by the HR user will be displayed.
- **Add Role (Add-Roles)** screen, where the HR user can add the Manager or Employee roles to users.
- **Edit Role (Edit-Roles)** screen, where the HR user can edit the user role.

To use the data from the database Roles table data on the screens, we need to create custom queries in advance.

9.2.1 Create Custom Query

We need to create Custom Queries to get the required data (users) from the database tables. This setting is required to display the list of managers and employees on the HR screens.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query.

Create custom query for **Employees**:

- **Query Name** - enter the **Get-All-Employees** query name.
- **Query Value** - enter the following query value:

  select id as value, user_name as description, group_area from c_user_accounts where group_area like ‘%Managers%’ or group_area like ‘%Employee%’

  Please pay attention to the quotation marks in the queries: use a single quotation mark from the keyboard layout for the English language.

  Please note that the SQL query must not contain the line brakes as this may lead to wrong results when executing the query.
- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-All-Employees**.

Submit changes on the page.

Create custom query for **Managers**:

- **Query Name** - enter the **Get-All-Managers** query name.
- **Query Value** - enter the following query value:

  select id as value, user_name as description, group_area from c_user_accounts where group_area like ‘%Managers%’
- **Query Parameters** - enter the **:manager** query parameter.
- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-All-Managers**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query***.

9.2.2 Create Screen: Add Roles (Add-Roles)

In this step, we will create the Add Roles (Add-Roles) screen for the HR user.

In the Architect application, open Screen Configurations > Screen Builder and click the **Create with builder** button.

On the opened page of the screen builder, specify the following data:

- **Add New Screen** - add a unique screen name. In our case **Add-Roles**.
- **Screen Description** - add a short description. In our case **Add-Roles**.
- **Primary Table** - select the primary table from the drop-down list. In our case **users_roles**.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the simple search and select the **Search_users_roles: Universal DLL : Search_users_roles** search option.
  In the Simple Search Configuration, select **roles_id** for the "**PKs for alias framework**" and "**Parameter for search**".
- **Tab Text** - enter a name. It will be displayed in the menu in the Render. In our case **Add-Roles**.
- **Tab Description** - add a clear description for the screen. In our case **Add-Roles**.
- **Screen State** - select the **Insert** screen state from the drop-down list.
- **Screen Layout** - select **Heder 1x1**.

Submit changes on the page.

On the **second** page of the screen builder, screen configure the following:

On the **Header** level, add a section of the **Edit type**.

Add a field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.users_roles.roles_id**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Role ID**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read-only**

Add a button and set up the following parameters:

- **Value** - enter **Add** as it will be used for adding the data.
- **Button Action** - select the **Insert** action.

- **Full width for auxiliary buttons** - select the check box.

On the **Section** level, add a section of the **Edit type**.

Select the following check boxes to adjust section layout:

- **Hide section header**
- **Hide section pop-out**

Save the settings and proceed with the Section fields' configurations.

Add the first field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.users_roles.user_id**.
- **Label** - enter the text that will be displayed on UI for this field. In our case **User ID**.
- **Type** - select the **Relation datalist** type.
- **Custom Query** - select the **Get-All-Employees - Get-All-Employees** query from the list.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Required**

Add the second field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.users_roles.manager_id**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Manager ID**.
- **Type** - select the **Relation datalist** type.
- **Custom Query** - select the **Get-All-Managers - Get-All-Managers** query from the list.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Required**

Add the third field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.users_roles.description**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Description**.
- **Type** - select the **Textarea** type.
- **Num Rows** - enter **5**. You can adjust this value according to your needs.

- Select the following check boxes to adjust field layout:
    - **Always show field border**

Click the **Create** button to save changes on the page.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Add Roles Screen***.

9.2.3 Create Group Screen for Add Roles (Add-Roles) Screen

Now we need to link the created Add-Roles screen and HR group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screen and create a new Group screen

Set the specific parameters:

- **Add** - select the check box for the **Add** action.
- **Group Name** - select the **HR** group we have created previously.
- **Menu Name** - select the **Roles** we have created previously.
- **Screen Name** - enter a screen name: **Add-Roles**.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. In our case **Add-Roles**.
- **Weight** - let us set it to **0**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen For Add-Roles Screen***.

9.2.4 Check Add-Roles Screen in Render

After creating the Add-Roles screen and the Group Screen for the Add-Roles screen, let’s check how the screen is working in the Render application.

Open the Render application, the Login screen will be shown.

Log in to the Render application using the HR user credentials. In our case the user name is **KentKim**, and the user password is the password that was defined during HR user creation. Then, click the **Login** button to log in to the application.

If the Add-Roles screen and the Group Screen for the Add-Roles screen are set correctly, the Add Role menu will be displayed on the menu bar at the top of the page.

To open the Add-Roles screen, click the Add Role menu. Here, we can check the Add-Roles screen by selecting users and managers, adding descriptions, and creating new records.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Add-Roles Screen in Render***.

9.2.5 Create Screen: Edit Roles (Edit-Roles)

The Edit Roles screen can be created in an easier way - let's copy the previously created Add Roles screen and adjust it accordingly.

9.2.5.1 Copy Add Roles Screen

In this step, we need to create a copy of the Add Roles screen. For that, in the Architect application, open Screen Configurations > Screen Builder, find the Add Roles screen and click the Copy icon.

In the opened Update screen page, we will change the following fields' values:

- **Select existing screen** - change the **Add-Roles** to **- - Select - -** value.
- **Add new screen** - add a screen name. In our case **Edit-Roles**.
- **Screen Description** - change the **Add-Roles** to **Edit-Roles** value.
- **Tab Text** - change the tab text to the **Edit-Roles** value.
- **Tab Description** - change the tab description from the **Add-Roles** to **Edit-Roles** value.

Submit changes on the page.

9.2.5.2 Adjust Edit Roles Screen

After the **Add Roles** screen copy creation and renaming it to the **Edit Roles** screen, we will need to update proper parameters of the Edit Roles screen using the Screen Builder.

For that, in the Architect application, open Screen Configurations > Screen Builder, find and open the Edit Roles (Edit-Roles) screen.

On the **first** page of the screen builder, the data should be updated as follows:

- **Tab Weight** - enter **99** instead of **0**.
- **Screen State** - change the **Insert** to **View** screen state by selecting it from the drop-down list.

On the **second** page of the Edit-Roles screen builder page, make the following changes on the **Header** level:

Add a new button and set up the following parameters:

- **Value** - enter **Delete** as it will be used for deleting the data.
- **Button action** - select the **Delete** action.

Change the **Add** to **Edit** button and set up the following parameters:

- **Value** - change the **Add** to **Edit** button value as it will be used for editing the data.
- **Button action** - select the **Edit** action instead of the **Insert** action.

- **Full width for auxiliary buttons** - select the check box.

Pay attention to the fact that in the Render application, to perform the delete action, the HR user is required to:

1. Click the **Delete** button. The pop-up window with the delete confirmation message appears.
2. Confirm the role deletion. After that, the Edit Role screen fields' values will be cleared.
3. Close the Edit Role screen, to finish the role deletion process.

Update the screen.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Edit-Roles Screen***.

9.2.6 Create Group Screen for Edit Roles (Edit-Roles) screen

Now we need to link the created Edit-Roles screen and HR group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create new Group screens.

Set the specific parameters:

- Select the check boxes for the following action:
  - **Delete**
  - **Edit**
- **Group Name** - select the **HR** group we have created previously.
- **Menu Name** - select the **Roles** we have created previously.
- **Screen Name** - select the **Edit-Roles** we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Edit Roles** here.
- **Weight** - let us set it to **99**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Edit-Roles Screen***.

9.2.7 Create Custom Multi Fields

Custom Multi aliases will be used for setting up the fields on the screens.

In the Architect application, open Alias Management > Custom Multi and create a new entry for Custom Multi Fields for each field we need for the next screen - **role_id**, **user_id**, **manager_id** and **description** fields.

Let us create a Custom Multi for the **role_id** field.

Open the Create Custom Multi Fields page and specify the following parameters:

- **Database Table** - select the **users_roles**.
- **Database Field** - select the **role_id**.
- **SQL Statement** - enter the following SQL query: *select * from users_roles*
- Check the **Alias Code**, it should meet the requirements. In our case **Multi.usersRoles.roles_id**.

Submit the custom multi field.

Next, create Custom Multi for the **user_id** field.

In the Architect application, open the Create Custom Multi Fields page and specify the following parameters:

- **Database Table** - select the **users_roles**.
- **Database Field** - select the **user_id**.
- **SQL Statement** - enter the following SQL query: *select * from users_roles*
- Check the **Alias Code**, it should meet the requirements. In our case **Multi.usersRoles.user_id**.

Submit the custom multi field.

Then, create Custom Multi for the **manager_id** field.

In the Architect application, open the Create Custom Multi Fields page and specify the following parameters:

- **Database Table** - select the **users_roles**.
- **Database Field** - select the **manager_id**.
- **SQL Statement** - enter the following SQL query: *select * from users_roles*
- Check the **Alias Code**, it should meet the requirements. In our case **Multi.usersRoles.managerId**.

Submit the custom multi field.

And finally, create Custom Multi for the **description** field.

In the Architect application, open the Create Custom Multi Fields page and specify the following parameters:

- **Database Table** - select the **users_roles**.
- **Database Field** - select the **description**.

- **SQL Statement** - enter the following SQL query: *select * from users_roles*
- Check the **Alias Code**, it should meet the requirements. In our case **Multi.usersRoles.description**.

Submit the custom multi field.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Multi Aliases***.

These aliases are required to create the next screen - All Roles.

9.2.8 Create Screen: All Roles (All-Roles)

Let’s create the All Roles screen for the HR user. Using this screen, the HR user can manage the roles of the Manager and Employee users.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the first page of the screen builder, define the following data:

- **Add New Screen** - add a unique screen name. In our case **All-Roles**.
- **Screen Description** - add a short description. In our case **All-Roles**.
- **Primary Table** - select the table created for user accounts. In our case **c_user_accounts**.
- **Tab Weight** - enter **99**.
- **Hide search bar** - select the check box.
- **Simple Search** - activate the simple search and select the **Search_c_user_accounts** search option. In the Simple Search Configuration, select the check box for the **id** field for Primary Key field and **user_name** for the search parameter.
- **Pass Logged User** - select the check box.
- **Tab Text** - enter a name. It will be displayed in the menu in the Render. In our case **All-Roles**.
- **Tab Description** - add a clear description for the screen. In our case **All-Roles**.
- **Screen State** - select the **View** screen state from the drop-down list.
- **Screen Layout** - select **Heder 1x1**.

On the second page of the screen builder, screen configure the following:

On the Header level, add a section of the **Edit** type.

Add a button and set up the following parameters:

- **Value** - enter the **Add** value.
- **Button Action** - select the **Link** action from the drop-down list.
  The additional settings fields appear:

- **Type Link** - select the **Modal** value from the drop-down list.
- **Modal Size** - enter **80%**.
- **Menu** - select the **Roles - Roles** menu from the drop-down list.
- **Group Screen** - select the **Add Role** value from the drop-down list.
- **Screen** - select the **Add-Roles - Add-Roles** screen from the drop-down list.
- **Parameter** - select the **role_id** value from the drop-down list.
- **Refresh base page when exit** - select the check box.

Save the changes by clicking the **Add** button. The **Add** button will be placed on the Header level. You can adjust the button display on the page.

On the Section level, add a section with the following parameters:

- **Layout Label** - let's define the layout label as follows: **Relationships between employees and managers**.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Table type** value from the drop-down list.
  After that, the additional parameters appear. Set them up the following way:
  - **Readonly** - select the check box.
  - **Label Orientation** - select **Top** from the drop-down list.
  - **Show Type** - select **Scroll** from the drop-down list.
  - **Pixel Height** - specify **600**.

Add alias fields and the column labels for them to the table. Select the multi alias fields, that we created before:

Alias Field	Column Label
Multi.userRoles.rolesId	**Role ID**
Multi.usersRoles.userId	**User**
Multi.usersRoles.managerId	**Manager**
Multi.usersRoles.description	**Description**

Add the **Edit** button to the table column. For that, click the **Add Button Column** button and enter Actions in the appeared Column Label row.

Save the created table and close the pop-up window.

Now we can configure the table columns. For that, click the gear icon next to the column label.

For the **Role ID** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Numeric** type.
- Select the following check boxes:
    - **Always show field border**
    - **Read only**
    - **Hidden**

For the **User** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Relation datalist** type.
- **Custom Query** - select the **Get-All-Employees - Get-All-Employees** custom query.
- Select the following check boxes:
    - **Always show field border**
    - **Read only**

For the **Manager** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Relation datalist** type.
- **Custom Query** - select the **Get-All-Managers - Get-All-Managers** custom query.
- Select the following check boxes:
    - **Always show field border**
    - **Read only**

For the **Description** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Textarea** type.
- Select the following check boxes:
    - **Always show field border**
    - **Read only**

For the **Actions** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param Type** - select the **Button**.
- **Value** - enter **View / Edit**.
- **Button Action** - select the **Link** type.
- **Type Link** - select **Modal** from the drop-down list.

- **Refresh base page when exit** - select the check box.
- **Modal Size** - enter **80%**.
- **Menu** - select the **Roles - Roles** menu from the drop-down list.
- **Group Screen** - select the **Edit Role** value from the drop-down list.
- **Screen** - select the **Edit-Roles - Edit-Roles** value from the drop-down list.
- **Parameter** - select **role_id** value from the drop-down list.
- **Pass Through Parameter** - select **Multi.usersRoles.roleId**.

Save the changes and close the pop-up window.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create All-Roles Screen***.

9.2.9 Create Group Screen for All Roles (All-Roles) screen

Now we need to link the created All-Roles screen and HR group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- **Group Name** - select the **HR** group we have created previously.
- **Menu Name** - select the **Roles** we have created previously.
- **Screen Name** - select the **All-Roles** we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **All Roles** here.
- **Weight** - let us set it to **0**.

We can check the saved configurations in the Render.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for All-Roles Screen***.

9.3 Set up Start Screen for HR Group

The last step in screen configuration is to set the Start screen for the HR group.

In the Architect application, open Account Settings > Groups to set up the Start screen for the HR group.

Search for the HR group. Open the HR group in the update mode by clicking the **Edit** icon.

In the opened Update group page, for the HR group set the following parameters for the Start screen:

- **Start Screen** - select the **All-Roles** screen from the drop-down list.

Submit changes on the page.

We can check the saved Start Screen configurations for the HR group in the Render.

For that, open the Render application and use the HR user credentials to log in to the Render application.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Start Screen for HR Group***.

9.4 Update Screens and Group Screens Weight

After performing all above-mentioned settings, we need to update the weight parameters for screens and group screens. It should be done to adjust screens' layouts.

First, we need to update the weight parameters for screens. For that, in the Architect application, open Screen configurations > Screens and perform the following operations:

- For the **Add-Roles** screen:
    - **Tab Weight** - change the value from **0** to **99**.
    - **Hide search bar** - select check box.
- For the **Edit-Roles** screen:
    - **Hide search bar** - select check box.

Then, we will update the weight parameters for group screens. For that, in the Architect application, open Screen configurations > Group screens and perform the following operations:

- For the **Add-Roles** group screen:
    - **Weight** - change the value from **0** to **99**.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Screens and Group Screens Weight***.

9.5 Check Screens for HR User in Render

Let us show how the configured screens work in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

You will see the **All-Roles** screen as a start screen. Click the **Add** button at the top right corner of the screen. The **Add-Roles** screen opens.

On the **Add-Roles** screen, we will create a relationship between Employee and Manager users.

To update the particular record in the table, click the corresponding Edit button in the Action column. The **Edit-Roles** screen opens.

On the **Edit-Roles** screen, we can update the relationship between Manager and Employee. Here we can edit or delete the record.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Screens for HR User***.

10 Set up Environment for Manager Use Cases

To prepare the environment for the Manager's Use Cases, the Admin User needs to perform the following steps.

10.1 Create Menu for Managers

This menu will provide access to the Render screen for Manager users, allowing them to manage tasks for Manager and Employee users.

In the Architect application, open Screen Configurations > Menu and create Menu for the Manager user.

Create Menu with the following parameters:

- **Group Name** - select the **Managers** group.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Tasks**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. In our case **Tasks**.
- **Weight** - enter **0**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Menu for Managers***.

10.2 Create todo_tasks Table

In this step, we will create a new table for Manager users. This table provides the possibility to store and manage the tasks and screens for Managers and Employees.

In the Architect application, open Features > Tables and create a new table with the following fields:

- **Table Name** - enter a table name. In our case **todo_tasks**.
- In the **Fields/Column Details** section define the following table parameters:

Name	Type	Nullable
task_id	Integer	False
summary	Varchar	True
priority	Varchar	True
created_on	Date	True

updated_on	Date	True
due_date	Date	True
status	Varchar	True
description	Text (Postgres, MySQL, SQLite) or Varchar (for Oracle SQL)	True
comments	Text (Postgres, MySQL, SQLite) or Varchar (for Oracle SQL)	True
user_id	Integer	True
manager_id	Integer	True

- **Constraint Type** - select the **Primary Key** constraint type.
- **Constraint Name** - enter the **todo_tasks_pk** constraint name.
- **Columns** - select the **task_id** column.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create todo_tasks Table***.

10.3 Set up Auto Sequence

The next step is setting up a new Auto Sequence rule. Here, we will create a unique role ID for Manager users.

In the Architect application, open Features > Auto Sequence and create a new Auto Sequence rule:

- **Table Name** - select the **todo_tasks** table from the drop-down list.
- In the **Fields Details** section, define the following parameters:
    - **Field Name** - enter the **task_id** field name.
    - **Increment by** - enter **1**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Auto Sequence***.

10.4 Create Screens required in the Render for the Manager user

To proceed with setting up the environment for Manager users, we need to create the following screens for the Manager user:

- **Create Task (M-Create-Task)** - will be used to create a new task.
- **Edit Task (M-Edit-Task)** - will be used to edit the task, specifically define the task priority, and assign the task to the Employee. The **Edit Task** screen is the second step of the task creation process, and will appear when the Employee submits a new task creation.
- **Update Task (M-Update-Task)** - will be used to update the earlier-created task.
- **All Tasks (M-All-Tasks)** - will be used to display all tasks of the Manager and the users assigned to his/her team.

10.4.1 Create Custom Query

We need to create Custom Queries to get the required data from the database tables.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create custom query for Manager:

- **Query Name** - enter the **Get-Users-for-Manager** query name.
- **Query Value** - enter the following query value:

  select id as value, user_name as description from c_user_accounts where id in (select user_id from users_roles where manager_id= :manager)
- **Query Parameters** - enter the **:manager** query parameter.
- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-Users-for-Manager**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query***.

10.4.2 Copy Key Files and Templates for Writing Plugins

Currently, all Service components are required to be installed and the select components function is not available during the Service installation. I.e., all key files and templates are added with the installed Service components. If it is actual for your version of Installer, skip this section.

If you have the previous version of the AppMill Service installer with the possibility to select and deselect Service components, the instructions described below will be relevant to you.

After selecting the Service components in the Service installation step, the installer will:

- add **ModuleExamples** and **PluginExamples** folders with key files for modules and plugins accordingly,

  and

- check the existence of the **Modules/php** and **plugins/php** folders.

In case the **Modules/php** and **plugins/php** folders **exist**, the installer will not update them.

In case the **Modules/php** and **plugins/php** folders **do not exist**, the installer will create these folders.

We need to copy our key files and templates from the **ModuleExamples** and **PluginExamples** to **Modules/php** and **plugins/php** folders, accordingly, before writing the plugins. As the key files and templates should be in the folder with plugins.

The PHP plugin and extension function created in this step will be used both for the Manager and Employee screens.

10.4.3 Create Plugin File

Also, we will create the **ToDoTasks.php** file in the plugins > php directory. This file contains the extension function that will be used further in the screen creation step.

Please note that the syntax of SQL queries may have its own specifics for different databases.

Open the created **ToDoTasks.php** file and enter the following data:

```
include “system.php”;

class ToDoTasks extends CodiacModule implements ICodiacModule
{
   public function PreInsert ($json_params)
   {
         try {
                  $data = (array) $json_params[‘patch_json’];

                  //get the user ID.
                  $userId = $json_params[“lock_id”];
```

```
                //run an SQL query to get the manager ID
                $getManagerIdQuery = executeSQL("SELECT manager_id FROM users_roles WHERE
user_id=$userId");
                $managerIdResults = $getManagerIdQuery->fetchOne();
                if ($managerIdResults)
                        $managerId = $managerIdResults[0];

                //fill out the form data. this data will be used in the main creation function
                setAliasValue('Alias.todo_tasks.user_id', $userId);
                setAliasValue('Alias.todo_tasks.manager_id', $managerId);

                $info = 'successfully';
                $requestresult = 'successfully';
        } catch(Exception $e) {
                $info = $e->getMessage();
                $requestresult = 'unsuccessfully';
        }

        return ['resultbody'=> ['requestresult'=> $requestresult, 'extendedinfo'=>  $info]];

    }

}
```

Save the **ToDoTasks.php** file in the **<service_path>/plugins/<your_plugin_language>** folder. In our case we will save the **ToDoTasks.php** file to the **Service/plugins/php** folder.

The **ToDoTasks.php** file that was used here can be found in the zip file (appmill_template > service > plugin > php) at the top right corner of the page with video instructions.

10.4.4 Create Extension Function

In the Architect application, open System data > Extension functions and create a new extension function.

Define the following parameters:

- **Data Source Function** - enter a function value. In our case **ToDoTasks::PreInsert**.
- **Data Source Description** - enter a description. In our case **ToDoTasks::PreInsert**.
- **Primary Table** - select the **todo_tasks** table from the drop-down list.
- **Extension Library** - select the **PHP** library from the drop-down.
- **Extension Function** - enter an extension function value. In our case **ToDoTasks::PreInsert**.

Submit changes on the page.

10.4.5 Restart Service

In case users need to restart the AppMill Service on their device, please follow the steps below:

1. Enter **Services** to the search area in the Taskbar.
2. Open the **Services** application.
3. In the **Services** application window:
    a. select the **AppMill Service** application and
    b. click the **Restart** link at the top right corner of the window.

    As a result, the AppMill Service application will be restarted.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Copy Key Files, Create Plugin File and Extension Function, and Restart Service***.

10.4.6 Create Screen: Create Task (M-Create-Task)

This screen will be used as the first step of the task creation process, where the Manager can fill in the required fields to proceed with the creation process.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add New Screen** - enter the **M-Create-Task** screen name.
- **Screen Description** - enter the **M-Create-Task** screen description.
- **Primary Table** - select the **todo_tasks** table from the drop-down list.
- **Tab Weight** - enter **99**.
- **Hide Search Bar** - select the check box.
- **Simple Search** - activate the Simple Search and select **Search_todo_tasks: Universal DLL: Search_todo_tasks** from the drop-down list.

  In the Simple Search Configuration, select **task_id** for the primary key and search parameter.
- **Tab Text** - enter the **M-Create-Task** tab text.
- **Tab Description** - enter the **M-Create-Task** tab description.
- **Screen State** - select the **Insert** screen state.
- **Extension Function** - for the **Add** action define Extension Function - Pre. Turn on the function using the toggle switch on the right side of the function input field. Click the **Extensions** field. In

the opened Extensions Settings pop-up window, select and add the required extension function. In our case **ToDoTasks::PreInsert**.

- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **Task ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.task_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Task ID**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read-only**

Add the **User ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.user_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User ID**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read-only**
  - **Hidden**

Add the **Manager ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.manager_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Manager ID**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read-only**
  - **Hidden**

Add the **Add** button and set up the following parameters:

- **Value** - enter **Add** as it will be used for adding the data.
- **Button Action** - select the **Insert** action.
- **Full width for auxiliary buttons** - select the check box.

On the **Section** level, add a section of the **Edit type**.

Select the following check boxes to adjust section layout:

- **Hide section header**
- **Hide section pop-out**

Add the **Summary** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.summary**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Summary**.
- **Type** - select the **Text** type.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Required**

Add the **Priority** field and set up the following parameters:

- **Data Field** - select the Primary Table check box and select the **Alias.todo_tasks.priority**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Priority**.
- **Type** - select the **Datalist** type.
- **Values (Use the Delimiter ";")** - enter the **Low;Medium;High** value.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Required**

Add the **Created on** field and set up the following parameters:

- **Data field** - select the Primary Table check box and select the **Alias.todo_tasks.created_on**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Created on**.
- **Type** - select the **Text** type.

- **Format Type** - select the **Date/Time** format type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Use current date/time**
    - **Read only**
    - **Required**

Add the **Due Date** field and set up the following parameters:

- **Data field** - select the Primary Table check box and select the **Alias.todo_tasks.due_date**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Due Date**.
- **Type** - select the **With dependent field** type.
- **Format Type** - select the **Date/Time** format type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
    - **Required**
- **ID of dependent field** - enter ID of the **Created on** field: **GetAliasFramework--Alias-todo_tasks-created_on**. The ID can be found at the bottom left corner of the Created on Field Settings pop-up window.
  For the dependent field
    - **Lead time** - enter **3**. You can adjust this value according to your needs.
    - **Use field date** - select the check box.
- Add three rows with pivot fields with the following parameters:
    - The first row:
        - **ID of pivot field** - enter ID of the **Priority** field: **GetAliasFramework--Alias-todo_tasks-priority**. The ID can be found at the bottom left corner of the Priority Field Settings pop-up window.
        - **Field value** - enter the **Low** value.
        - **Lead time** - enter **5**. You can adjust this value according to your needs.
    - The second row:
        - **ID of pivot field** - enter ID of the Priority field. The ID can be found at the bottom left corner of the Priority Field Settings pop-up window.

- **Field value** - enter the **Medium** value.
- **Lead time** - enter **3**. You can adjust this value according to your needs.

- The third row:
  - **ID of pivot field** - enter the Priority field **ID**.The ID can be found at the bottom left corner of the Priority Field Settings pop-up window.
  - **Field value** - enter the **High** value.
  - **Lead time** - enter **2**. You can adjust this value according to your needs.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Create Task (M-Create-Task) Screen***.

10.4.7 Create Group Screen for Create Tasks (M-Create-Task) Screen

Now we need to link the created M-Create-Task screen and Manager group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Add** check box.
- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **M-Create-Task** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Create Task** here.
- **Weight** - let us set it to **0**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screens for Create Tasks (M-Create-Task) Screen***.

10.4.8 Check Create Task (M-Create-Task) Screen in Render

After the Create Task (M-Create-Task) screen is created, we need to check it in the Render application. For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the Create Tasks screen and the Group Screen for the Create Task screen are set correctly, the Create Task menu will be displayed on the menu bar at the top of the page.

To open the Create Task screen, click the Create Task menu. Here, we can check the Create Task screen by adding new tasks. To add a new task, define the task summary and priority. According to the priority value, the due date value will be changed. Click the **Save** button, to create the task.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Create Task (M-Create-Task) Screen in Render***.

10.4.9 Create Screen: Edit Task (M-Edit-Task)

This screen will be used as the second step of the task creation process, where the Manager can fill in the required fields to proceed with the creation process.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the first page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add new screen** - enter the **M-Edit-Task** screen name.
- **Screen Description** - enter the **M-Edit-Task** screen description.
- **Primary Table** - select the **todo_tasks** table from the drop-down list.
- **Tab Weight** - enter **99**.
- **Hide search bar** - select the check box.
- **Simple Search** - activate the Simple Search and select **Search_todo_tasks: Universal DLL: Search_todo_tasks** from the drop-down list.

  In the Simple Search Configuration, select **task_id** for the primary key and search parameter.
- **Tab Text** - enter the **M-Edit-Task** tab text.
- **Tab Description** - enter the **M-Edit-Task** tab description.
- **Screen State** - select the **Edit** screen state.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the second page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **Task ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.task_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Task ID**.

- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read-only**

Add the **Summary** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.summary**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Summary**.
- **Type** - select the **Text** type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Required**

Add the **Manager ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.manager_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Manager ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read-only**
    - **Hidden**

Add the **Priority** field and set up the following parameters:

- **Data Field** - select the Primary Table check box and select the **Alias.todo_tasks.priority**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Priority**.
- **Type** - select the **Text** type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Add the **Created on** field and set up the following parameters:

- **Data Field** - select the Primary Table check box and select the **Alias.todo_tasks.created_on**.

- **Label** - enter the text that will be displayed on UI for this field, in our case **Created on**.
- **Type** - select the **Text** type.
- **Format Type** - select the **Date/Time** format type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Add the **Updated on** field and set up the following parameters:

- **Data Field** - select the Primary Table check box and select the **Alias.todo_tasks.updated_on**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Updated on**.
- **Type** - select the **Text** type.
- **Format Type** - select the **Date/Time** format type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
    - **Use current date/time**

Add the **Due Date** field and set up the following parameters:

- **Data Field** - select the Primary Table check box and select the **Alias.todo_tasks.due_date**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Due Date**.
- **Type** - select the **Text** type.
- **Format Type** - select the **Date/Time** format type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Add an **Edit** button and set up the following parameters:

- **Value** - enter **Edit** as it will be used for adding the data.
- **Button Action** - select the **Edit** action.
- **Full width for auxiliary buttons** - select the check box.

On the **Section** level, add a section of the **Edit type**.

Select the following check boxes to adjust section layout:

- **Hide section header**
- **Hide section pop-out**

  Add the **User ID** field and set up the following parameters:

  - **Data Field** - select the **Primary Table** check box and select the **Alias.todo_tasks.user_id**.
  - **Label** - enter the text that will be displayed on UI for this field, in our case **Assigned to**.
  - **Type** - select the **Relation datalist** type.
  - Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Required**
  - **Custom Query** - select the **Get-Users-for-Manager - Get-Users-for-Manager** query from the drop-down list.
  - **:manager** - in the Field ID field enter the ID of the Manager ID field. The ID of the Manager ID field can be found at the bottom left corner of the Manager ID Field Settings pop-up window. Enter **GetAliasFramework--Alias-todo_tasks-manager_id**.

  Add the **Status** field and set up the following parameters:

  - **Data field** - select the **Primary Table** check box and select the **Alias.todo_tasks.status**.
  - **Label** - enter the text that will be displayed on UI for this field, in our case **Status**.
  - **Type** - select the **Datalist** type.
  - **Values (Use the Delimiter ";")** - enter the **to_do;in_progress;approved;done** value.
  - Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Required**

  Add the **Description** field and set up the following parameters:

  - **Data field** - select the **Primary Table** check box and select the **Alias.todo_tasks.description**.
  - **Label** - enter the text that will be displayed on UI for this field, in our case **Description**.
  - **Type** - select the **Textarea** type.
  - **Num Rows** - enter 5. You can adjust this value according to your needs.

- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Comments** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.todo_tasks.comments**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Comments**.
- **Type** - select the **Textarea** type.
- **Num Rows** - enter 5. You can adjust this value according to your needs.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

You can adjust the layout on the fields of the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Edit Task (M-Edit-Task) Screen***.

10.4.10 Create Group Screen for Edit Task (M-Edit-Task) Screen

Now we need to link the created M-Edit-Task screen and Manager group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Edit** check box.
- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **M-Edit-Task** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Edit Task** value here.
- **Weight** - let us set it to **99**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Edit Task (M-Edit-Task) Screen***.

10.4.11 Create Manager_Task_Creation Workflow

To split the Task creation process into steps for Manager users, we need to incorporate the Workflow into it. Let us make an easy example.

The task can be created sequentially.

The diagram for our case can be simple and consists of 3 steps:

Start workflow step > **Process** workflow step > **End** workflow step.

On the **Start** workflow step (called Create), the user should fill in the required information.

On the **Process** workflow step (called Details), the user should add all additional details.

On the **End** workflow step, the final overview step will be displayed.

To create a new Workflow Diagram, in the Architect application, open Workflow > Workflow Builder. Click the Create button to open the Workflow Diagram creation page. Here, fill in the following parameters:

- **Workflow Diagram Name** - enter a **Manager_Task_Creation** value.
- **Task Description** - enter a **Manager_Task_Creation** description.
- **Menu** - select the **Managers - Tasks** menu from the drop-down list.
- **Auto Advance** - select the check box.

Add the steps below to the Workflow diagram drawing area and fill in the step parameters:

- **Start** step
    - **Group** - select the **Managers: Managers** group from the drop-down list.
    - **Screen** - select the **M-Create-Task** screen from the drop-down list.
    - **Step Type** - select the **Start** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Create**.
- **Process** step
    - **Name** - enter the **Fill Info** step name.
    - **Group** - select the **Managers: Managers** group from the drop-down list.
    - **Screen** - select the **M-Edit-Task** screen from the drop-down list.
    - **Step Type** - select the **Process** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Fill Info**.
    - **Order** - enter **2**.
- **End** step
    - **Label** - enter a label value. In our case **Done**.

Connect the workflow diagram process steps to each other via arrows as follows:

Start step > **Process** step > **End** step.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Manager_Task_Creation Workflow***.

10.4.12 Check Manager_Task_Creation Workflow in Render

After the **Manager_Task_Creation** workflow is set up, we need to check it in the Render application. For that, open the Render application and use the Manager user credentials to log in to the Render application. At the top left corner of the page, click the Create Task menu item to open the Create Task screen. Here, the workflow diagram is displayed under the menu bar.

If the Manager_Task_Creation workflow is set correctly, the workflow steps will be performed one by one according to the steps of the task creation process:

- The **Start** step named **Create** is for the **Create Task** screen.
- The **Process** step named **Fill Info** is for the **Edit Task** screen.
- The **End** step named **Done** appears when the Manager user saves the changes on the Edit Task screen.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Manager_Task_Creation Workflow in Render***.

10.4.13 Create Screen: Update Task (M-Update-Task)

This screen will provide the possibility to update the details of the Task. It will be available when clicking the View / Update button for the corresponding task displayed in the task table on the All Tasks screen.

The Update Task screen can be created in an easier way - let’s copy the previously created Edit Task screen and adjust it accordingly.

10.4.13.1 Copy Edit Task Screen

In this step, we need to create a copy of the Edit Task screen. For that, in the Architect application, open Screen Configurations > Screen Builder, find the Edit Task screen and click the copy icon.

In the opened Update screen page, we will change the following fields’ values:

- **Select existing screen** - change the **M-Edit-Task** to **- - Select - -** value.
- **Add new screen** - add a screen name. In our case **M-Update-Task**.
- **Screen Description** - change the **M-Edit-Task** to **M-Update-Task** value.
- **Tab Text** - change the tab text to the **M-Update-Task** value.
- **Tab Description** - change the tab description from the **M-Edit-Task** to **M-Update-Task** value.

Submit changes on the page.

10.4.13.2 Adjust Update Task (M-Update-Task) Screen

After the **Edit Task** screen copy creation and renaming it to the **Update Task** screen, we will need to update proper parameters of the Update Task screen using the Screen Builder.

For that, in the Architect application, open Screen Configurations > Screen Builder, find and open the Update Task (M-Edit-Task) screen.

On the **first** page of the screen builder, the data should be updated as follows:

- **Screen State** - change the **Insert** to **View** screen state by selecting it from the drop-down list.

On the **second** page of the Edit-Roles screen builder page, make the following changes on the **Header** level:

Add a new button and set up the following parameters:

- **Value** - enter **Delete** as it will be used for deleting the data.
- **Button Action** - select the **Delete** action.

Pay attention to the fact that in the Render application, to perform the delete action, Manager users are required to:

1. Click the **Delete** button. The pop-up window with the delete confirmation message appears.
2. Confirm the task deletion. After that, the Update Task screen fields' values will be cleared.
3. Close the Update Task screen, to finish the task deletion process.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Update Task (M-Update-Task) Screen***.

10.4.14 Create Group Screen for Update Tasks (M-Update-Task) Screen

Now we need to link the created M-Update-Task screen and Manager group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the check boxes for the following actions**:**
  - **Delete**
  - **Edit**
- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **M-Update-Task** screen we have created previously.

- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Update Task** value here.
- **Weight** - let us set it to **99**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Update Tasks (M-Update-Task) Screen***.

10.4.15 Create Screen: All Tasks (M-All-Tasks)

In this step, we will create the screen that provides an overview of all tasks that are available for the Manager user.

These are tasks assigned to the Manager themself and tasks assigned to the Employee users from the linked team.

The screen represents the table with the tasks and related details.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, define the following data:

- **Add new screen** - enter the **M-All-Tasks** screen name.
- **Screen Description** - enter the **M-All-Tasks** screen description.
- **Primary Table** - select the **c_user_accounts** table from the drop-down list.
- **Tab Weight** - enter **99**.
- **Hide search bar** - select the check box.
- **Simple Search** - activate the Simple Search and select **Search_c_user_accounts: Universal DLL: Search_c_user_accounts** from the drop-down list.

  In the Simple Search Configuration, select:

  - **id** for the primary key and
  - **user_name** for the search parameter.
- **Pass Logged User** - select the check box.
- **Tab Text** - enter the **M-All-Tasks** tab text.
- **Tab Description** - enter the **M-All-Tasks** tab description.
- **Screen State** - select the **View** screen state.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, screen configure the following:

On the **Header** level, add a section of the **Edit type**.

Add the **Account ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.c_user_accounts.id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Account ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read-only**
    - **Hidden**

Add the **User ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.users_roles.user_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read-only**
    - **Hidden**

Add the **Manager ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_roles.manager_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Manager ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read-only**
    - **Hidden**

Add the **Add Task** button and set up the following parameters:

- **Value** - enter **Add Task** as it will be used for adding tasks.

- **Button Action** - select the **Link** action.
- **Type Link** - select the Modal link type.
- **Modal Size** - enter 80%.
- **Refresh base page when exit** - select the check box.
- **Menu** - select the **Tasks - Tasks** menu from the drop-down list.
- **Group Screen** - select the **Create Task** group screen from the drop-down list.
- **Screen** - select the **M-Create-Task - M-Create-Task** screen from the drop-down list.
- **Parameter** - select the **task_id** parameter from the drop-down list.

On the **Section** level, set the following parameters:

- **Hide section header** - select the check box.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Table type** for the section.
- **Readonly** - select the check box.
- **Include column filter** - select the check box.
- **Include extended search** - select the check box.
- **Label Orientation** - select the **Top** orientation.
- **Show Type** - select the **Scroll** type.
- **Pixel Height** - enter **600**. You can adjust this value according to your needs.
- Add the following alias fields with column labels:

Alias field	Column label
Array.todo_tasks.task_id	Task ID
Array.todo_tasks.summary	Summary
Array.todo_tasks.user_id	Assigned to
Array.todo_tasks.priority	Priority
Array.todo_tasks.due_date	Due Date

- **Add button column** - click the **Add button column** button and enter the **Actions** column label.

Save changes to the **Section** settings.

Now we need to configure the columns.

For the **Task ID** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param Type** - select the **Numeric** type.
- **Read only** - select the check box.

Save the changes and close the pop-up window.

For the **Summary** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param Type** - select the **Text** type.
- **Read only** - select the check box.

Save the changes and close the pop-up window.

For the **Assigned to** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Relation datalist** type.
- **Custom query** - select the **Get-All-Employees - Get-All-Employees** query from the drop-down list.
- **Always show field border** - select the check box.
- **Read only** - select the check box.

Save the changes and close the pop-up window.

For the **Priority** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param Type** - select the **Text** type.
- **Always show field border** - select the check box.

Save the changes and close the pop-up window.

For the **Due Date** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Text** type.
- **Format type** - select the **Date/Time** format type.
- **Always show field border** - select the check box.
- **Read only** - select the check box.

Save the changes and close the pop-up window.

For the **Actions** column, click the gear wheel icon, and in the opened pop-up window set up the following:

- **Param type** - select the **Button** type.
- **Button action** - select the **Link** type.
- **Value** - enter the **View / Update** button value.
- **Type Link** - select the **Modal** link type from the drop-down list.
- **Refresh base page when exit** - select the check box.
- **Modal size** - enter **80%**.
- **Menu** - select the **Tasks - Tasks** menu from the drop-down list.
- **Group Screen** - select the **Update Task** menu from the drop-down list.
- **Screen** - select the **M-Update-Task - M-Update-Task** menu from the drop-down list.
- **Parameter** - select the **task_id** parameter.
- **Pass Through Parameter** - select the **Array.todo_tasks.task_id** parameter.

Submit changes and close the pop-up window.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create All Tasks (M-All-Tasks) Screen***.

10.4.16 Create Group Screen for All Tasks (M-All-Tasks) Screen

Now we need to link the created All-Tasks screen and Manager group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Inquire** check box.
- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **M-All-Tasks** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Tasks** here.
- **Weight** - let us set it to **0**.

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for All Tasks (M-All-Tasks) Screen***.

10.4.17 Create Alias Relationship for All Tasks Group Screen

To proceed with the set up environment for the Manager users process, we will create the Alias Relationships for the All Tasks group screen.

The created Alias Relationship provides the possibility to oversee the management, creation, and updating of dependencies between two Relationship-type database fields. To establish new dependencies, simply select the parent and child tables, then specify the respective parent and child fields.

In the Architect application, open Alias Management > Manage Alias Relationship and create a new Alias Relationship.

Create the first Alias Relationship with the following parameters:

- **Group screen name** - select the **M-All-Tasks** name from the drop-down list.
- **Parent table** - sselect the **c_user_accounts** table from the drop-down list.
- **Parent field** - select the **id** field from the drop-down list.
- **Child table** - select the **todo_tasks** table from the drop-down list.
- **Child field** - select the **manager_id** field from the drop-down list.
- **Active** - select the check box.

Submit changes on the page.

Create the second Alias Relationship with the following parameters:

- **Group screen name** - select the **M-All-Tasks** name from the drop-down list.
- **Parent table** - select the **c_user_accounts** table from the drop-down list.
- **Parent field** - select the **id** field from the drop-down list.
- **Child table** - select the **users_roles** table from the drop-down list.
- **Child field** - select the **user_id** field from the drop-down list.
- **Active** - select the check box.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Alias Relationship for All Tasks Group Screen***.

10.5 Set up Start Screen for Managers Group

The last step in screen configuration is to set the Start screen for the Managers group.

In the Architect application, open Account Settings > Groups to set up the Start screen for Manager users.

Search for the Managers group. Open the Managers group in the update mode by clicking the **Edit** icon.

In the opened Update group page, for the Managers group set the following parameters for the Start screen:

- **Start Screen** - select the **M-All-Tasks** screen from the drop-down list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Start Screen for Managers Group***.

10.6 Create Task Priority (TaskPriority) List

In this step, we need to create the Task Priority list to organize and store the data in a structured way and further retrieve the data on the Render screens as a list of options.

For that, in the Architect application, open System data > Lists and create a list.

To create a list with the first list option, fill in the following parameters:

- **List Name** - enter the **TaskPriority** value.
- **Description** - enter a short description that will be displayed as a list option in the Render. In our case **Low**.
- **Entry Name** - enter the **Low** entry name that will be stored in the database.
- **Weight** - enter **0**.

Submit changes on the page.

After the list creation we will add more list options via the **Update Bulk** functionality. For that, on the Lists Overview page we need to search the TaskPriority list and perform the update bulk operation.

On the opened Update Bulk page for the TaskPriority list, click the **Add more** button to add the priority options to the list.

Add the second list option and fill in the following parameters:

- **Description** - enter the **Medium** value.
- **Entry Name** - enter the **Medium** value.
- **Weight** - enter **1**.

Add the third list option and fill in the following parameters:

- **Description** - enter the **High** value.
- **Entry Name** - enter the **High** value.
- **Weight** - enter **3**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Task Priority (TaskPriority) List***.

10.7 Update Screens, Group Screens, and todo_tasks Table Parameters

After performing all above-mentioned settings, we need to update some parameters for screens, group screens, and the todo_tasks table. It should be done to adjust screens' layouts.

First, we need to update the weight parameters for screens. For that, in the Architect application, open Screen configurations > Screens and perform the following operations:

- For the **All Tasks (M-All-Tasks)** screen:
    - Update the following **Add Task** button parameters:
        - **Type Link** - change the **Modal** to **Self** type.
        - **Refresh base page when exit** - deselect the check box.
    - Update the following **Priority** column parameters:
        - **Param Type** - change the **Text** to **List** type.
        - **Pull up** - click the **Pull up** button and select the **TaskPriority** list from the List Params pop-up window.

  Submit changes on the page.
- For the **Create Task (M-Create-Task)** screen:
    - Update the following **Priority** field parameters:
        - **Type** - change the **Datalist** to **List** type.
        - **Pull up** - click the **Pull up** button and select the **TaskPriority** list from the List Params pop-up window.
    - Update the following **Due Date** field parameters:
        - **Field Value** - change the field values: **low** > **Low**, **normal** > **Medium**, **high** > **High**.

  Submit changes on the page.
- For the **Edit Task (M-Edit-Task)** screen:
    - Update the following **Priority** field parameters:
        - **Type** - change the **Text** to **List** type.

        - **Pull up** - click the **Pull up** button and select the **TaskPriority** list from the List Params pop-up window.

    Submit changes on the page.

- For the **Update Task (M-Update-Task)** screen:
    - Update the following **Priority** field parameters:
        - **Type** - change the **Text** to **List** type.
        - **Pull up** - click the **Pull up** button and select the **TaskPriority** list from the List Params pop-up window.

    Submit changes on the page.

Then, we will update the weight parameters for group screens. For that, open Screen configurations > Group screens and perform the following operations:

- For the **Create Task** group screen:
    - **Weight** - change the value from **0** to **99**.

The last update step is the **todo_tasks** table edition. For that, open Features > Tables and change the **Date** to **Timestamp** type for the following parameters:

- **created_on**
- **updated_on**
- **due_date**

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Screens, Group Screens, and todo_tasks Table Parameters***.

10.8 Check Screens for Manager Users in Render

Let us show how the configured screens work in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned environment for Manager users is set up correctly, we will see the All Tasks screen as the start screen and be able to create, edit, update and delete tasks.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Screens for Manager Users in Render***.

11 Set up Environment for Employee Use Cases

To prepare the environment for the Employee's Use Cases, the Admin User needs to perform the following steps.

11.1 Create Menu for Employee

First, we will create a menu for the Employee users. The created menu will provide access to the Render screen for Employee users, allowing them to manage tasks that were assigned by Manager users.

In the Architect application, open Screen Configurations > Menu and create a Menu for the Employee user.

Create Menu with the following parameters:

- **Group Name** - select the **Employees** group.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Tasks**.
- **Description** - enter the **Tasks** description. It will be displayed when hovering over the menu name on the UI in the Render.
- **Weight** - enter **0**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Menu for Employee***.

11.2 Create Screens required in the Render for the Employee user

Let's create the following screens for the Employee users, to set up the environment for the Employee users:

- **Create Task (E-Create-Task)** - will be used to create a new task.
- **Edit Task (E-Edit-Task)** - will be used to edit the task, specifically define the task priority, and assign the task to the Employee. The **Edit Task** screen is the second step of the task creation process, and will appear when the Employee submits a new task creation.
- **Update Task (E-Update-Task)** - will be used to update the earlier-created task.
- **All Tasks (E-All-Tasks)** - will be used to display all tasks that were assigned to the Employee by the Manager user.

11.2.1 Create Custom Query

We need to create Custom Queries to get the required data from the database tables.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create a custom query for Employees:

- **Query Name** - enter the **Get-Self-and-Manager** query name.
- **Query Value** - enter the following query value:

  select id as value, user_name as description from c_user_accounts where id in ((select user_id from users_roles where user_id= :user) , (select manager_id from users_roles where user_id = :user))
- **Query Parameters** - enter the **:user** query parameter.
- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-Self-and-Manager**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query***.

11.2.2 Create Screen: Create Task (E-Create-Task)

This screen will be used as the first step of the task creation process, where the Employee can fill in the required fields to proceed with the creation process.

In the Architect application, open Screen Configurations > Screen Builder.

We have two options on how to create a new screen:

1. Make a copy from the existing screen and adjust it.
2. Create a new screen using the Create with builder functionality

In our example, we will create the Create Task (E-Create-Task) screen from a copy of an existing screen.

11.2.2.1 Create E-Create-Task Screen from M-Create-Task Screen Copy

To create a Create Task (E-Create-Task) screen, we will copy the M-Create-Task screen, change the screen name to the E-Create-Task, and adjust the screen.

In the Architect application, open Screen Configurations > Screen Builder, find the M-Create-Task screen and click the copy icon for the entry.

In the opened Update screen page, we will change the following fields' values:

- **Select existing screen** - change the **M-Create-Task** to **- - Select - -** value.
- **Add new screen** - add a screen name. In our case **E-Create-Task**.
- **Screen Description** - change the **M-Create-Task** to **E-Create-Task** value.
- **Tab Text** - change the tab text to the **E-Create-Task** value.
- **Tab Description** - change the tab description from the **M-Create-Task** to **E-Create-Task** value.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Create Task (E-Create-Task) Screen***.

11.2.3 Create Group Screen for Create Tasks (E-Create-Task) Screen

Now we need to link the created E-Create-Task screen and Employee group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Add** check box.
- **Group Name** - select the **Employees** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **E-Create-Task** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Create Task** here.
- **Weight** - let us set it to **99**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Create Tasks (E-Create-Task) Screen***.

11.2.4 Create Screen: Edit Task (E-Edit-Task)

This screen will be used as the second step of the task creation process, where the Employee can fill in the required fields to proceed with the creation process.

In the Architect application, open Screen Configurations > Screen Builder.

We have two options on how to create a new screen:

1. Make a copy from the existing screen and adjust it.
2. Create a new screen using the Create with builder functionality

In our example, we will create the Edit Task (E-Edit-Task) screen from a copy of an existing screen.

11.2.5 Create E-Edit-Task Screen from M-Edit-Task Screen Copy

To create a Create Task (E-Edit-Task) screen, we will copy the M-Edit-Task screen, change the screen name to the E-Edit-Task, and adjust the screen.

In the Architect application, open Screen Configurations > Screen Builder, find the M-Edit-Task screen and click the copy icon for the entry.

In the opened Update screen page, we will change the following fields' values:

- **Select existing screen** - change the **M-Edit-Task** to **- - Select - -** value.
- **Add new screen** - add a screen name. In our case **E-Edit-Task**.
- **Screen Description** - change the **M-Edit-Task** to **E-Edit-Task** value.
- **Tab Text** - change the tab text to the **E-Edit-Task** value.
- **Tab Description** - change the tab description from the **M-Edit-Task** to **E-Edit-Task** value.

Submit changes on the page.

After updating the Edit Task screen, we need to adjust some screen parameters. For that, find the Edit Task screen and perform the changes below.

On the **second** page of the Edit Task screen builder page, make the following changes on the **Section** level:

- For the **Assigned to** field:
    - **Custom Query** - change the **Get-Users-for-Manager - Get-Users-for-Manager** to **Get-Self-and-Manager - Get-Self-and-Manager** query.
    - **:user** - in the Field ID field enter the ID of the Assigned to field. The ID of the Assigned to field can be found at the bottom left corner of the Assigned to Field Settings pop-up window. Enter **GetAliasFramework--Alias-todo_tasks-user_id**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Edit Task (E-Edit-Task) Screen***.

11.2.6 Create Group Screens for Edit Task (E-Edit-Task) Screen

Now we need to link the created E-Edit-Task screen and Employee group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Edit** check box**.**

- **Group Name** - select the **Employees** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **E-Edit-Task** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Edit Task** value here.
- **Weight** - let us set it to **99**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screens for Edit Task (E-Edit-Task) Screen***.

11.2.7 Create Employee_Task_Creation Workflow

To split the Task creation process into steps for Employee users, we need to incorporate the Workflow into it. Let us make an easy example.

The task can be created sequentially.

The diagram for our case can be simple and consists of 3 steps:

Start workflow step > **Process** workflow step > **End** workflow step

On the **Start** workflow step (called Create), the user should fill in the required information.

On the **Process** workflow step (called Details), the user should add all additional details.

On the **End** workflow step, the final overview step will be displayed.

In the Architect application, open Workflow > Workflow Builder and create a new Workflow Diagram.

Fill in the following parameters:

- **Workflow Diagram Name** - enter the **Employee_Task_Creation** value.
- **Task Description** - enter a description. In our case **Employee_Task_Creation**.
- **Menu** - select the **Employees - Tasks** menu from the drop-down list.
- **Auto Advance** - select the check box.

Add the steps below to the Workflow diagram drawing area and fill in the step parameters:

- **Start** step
    - **Group** - select the **Employees: Employees** group from the drop-down list.
    - **Screen** - select the **E-Create-Task** screen from the drop-down list.
    - **Step Type** - select the **Start** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Create**.

- **Process** step
    - **Name** - enter the **Fill Info** step name.
    - **Group** - select the **Employees: Employees** group from the drop-down list.
    - **Screen** - select the **E-Edit-Task** screen from the drop-down list.
    - **Step Type** - select the **Process** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Fill Info**.
    - **Order** - enter **2**.
- **End** step
    - **Label** - enter a label value. In our case **Done**.

Connect the workflow diagram process steps to each other via arrows as follows:

Start step > **Process** step > **End** step.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Employee_Task_Creation Workflow***.

11.2.8 Create Screen: Update Task (E-Update-Task)

This screen will provide the possibility to update the details of the Task. It will be available when clicking the Edit button for the corresponding task displayed in the task table on the All Tasks screen.

In the Architect application, open Screen Configurations > Screen Builder.

We have two options on how to create a new screen:

1. Make a copy from the existing screen and adjust it.
2. Create a new screen using the Create with builder functionality

In our example, we will create the Update Task (E-Update-Task) screen from a copy of an existing screen.

11.2.8.1 Create E-Update-Task Screen from M-Update-Task Screen Copy

To create an Update Task (E-Update-Task) screen, we will copy the M-Update-Task screen, change the copy name to the E-Update-Task, and adjust the screen.

In the Architect application, open Screen Configurations > Screen Builder, find the M-Update-Task screen and click the copy icon for the entry.

In the opened Update screen page, we will change the following fields' values:

- **Select existing screen** - change the **M-Update-Task** to **- - Select - -** value.
- **Add new screen** - add a screen name. In our case **E-Update-Task**.
- **Screen Description** - change the **M-Update-Task** to **E-Update-Task** value.

- **Tab Text** - change the tab text to the **E-Update-Task** value.
- **Tab Description** - change the tab description from the **M-Update-Task** to **E-Update-Task** value.

Submit changes on the page.

After updating the Update Task screen, we need to adjust some screen parameters. For that, find the Update Task screen and perform the changes below.

On the second page of the Update Task screen builder page, make the following changes:

On the **Header** level, delete the **Delete** button.

On the **Section** level:

- For the **Assigned to** field:
    - **Custom Query** - change the **Get-Users-for-Manager - Get-Users-for-Manager** to **Get-Self-and-Manager - Get-Self-and-Manager** query.
    - **:user** - in the Field ID field enter the ID of the Assigned to field. The ID of the Assigned to field can be found at the bottom left corner of the Assigned to Field Settings pop-up window. Enter **GetAliasFramework--Alias-todo_tasks-user_id**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Update Task (E-Update-Task) Screen***.

11.2.9 Create Group Screen for Update Task (E-Update-Task) Screen

Now we need to link the created E-Update-Task screen and Employee group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Edit** check box.
- **Group Name** - select the **Employees** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **E-Update-Task** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Update Task** value here.
- **Weight** - let us set it to **99**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Update Task (E-Update-Task) Screen***.

11.2.10 Create Screen: All Tasks (E-All-Tasks)

In this step, we will create a screen that provides the possibility to overview all tasks that are assigned to the Employee user.

The screen represents the table with the tasks and related details.

In the Architect application, open Screen Configurations > Screen Builder.

We have two options on how to create a new screen:

1. Make a copy from the existing screen and adjust it.
2. Create a new screen using the Create with builder functionality

In our example, we will create the All Tasks (E-All-Tasks) screen from a copy of an existing screen.

11.2.10.1 Create E-All-Tasks Screen from M-All-Tasks Screen Copy

To create an Update Task (E-All-Tasks) screen, we will copy the M-All-Tasks screen, change the copy name to the E-All-Tasks, and adjust the screen.

In the Architect application, open Screen Configurations > Screen Builder, find the M-Update-Task screen and click the copy icon for the entry.

In the opened Update screen page, we will change the following fields' values:

- **Select existing screen** - change the **M-All-Tasks** to **- - Select - -** value.
- **Add new screen** - add a screen name. In our case **E-All-Tasks**.
- **Screen Description** - change the **M-All-Tasks** to **E-All-Tasks** value.
- **Tab Text** - change the tab text to the **E-All-Tasks** value.
- **Tab Description** - change the tab description from the **M-All-Tasks** to **E-All-Tasks** value.

Submit changes on the page.

After updating the All Tasks screen, we need to adjust some screen parameters. For that, find the All Tasks screen and perform the changes below.

On the **second** page of the All Tasks screen builder page, make the following changes:

On the **Header** level:

- For the Add Task button:
    - **Menu** - change the Manager's **Tasks - Tasks** menu to the Employee's **Tasks - Tasks** menu.
    - **Group Screen** - select the Employee's **Create Task** group screen from the drop-down list.
    - **Screen** - select the **E-Create-Task - E-Create-Task** screen from the drop-down list.
    - **Parameter** - select the **task_id** parameter from the drop-down list.

On the **Section** level:

- For the **Actions** column:
    - **Group Screen** - change the Manager's **Update Task** group screen to the Employee's **Update Task** group screen.
    - **Screen** - select the **E-Update-Task - E-Update-Task** screen from the drop-down list.
    - **Parameter** - select the **task_id** parameter from the drop-down list.
    - **Pass through parameter** - select the **Array.todo_tasks.task_id** parameter from the drop-down list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create All Tasks (E-All-Tasks) screen***.

11.2.11 Create Group Screen for All Tasks (E-All-Tasks) Screen

Now we need to link the created All-Tasks screen and Employee group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the **Inquire** check box.
- **Group Name** - select the **Employees** group we have created previously.
- **Menu Name** - select the **Tasks** menu we have created previously.
- **Screen Name** - select the **E-All-Tasks** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Tasks** here.
- **Weight** - let us set it to **0**.

Save changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for All Tasks (E-All-Tasks) Screen***.

11.2.12 Create Alias Relationship for All Tasks Group Screen

To proceed with the set up environment for the Employee users process, we will create the Alias Relationships for the All Tasks group screen.

In the Architect application, open Alias Management > Manage Alias Relationship and create a new Alias Relationship.

Create the first Alias Relationship with the following parameters:

- **Group Screen Name** - select the **E-All-Tasks** name from the drop-down list.
- **Parent Table** - select the **c_user_accounts** table from the drop-down list.
- **Parent Field** - select the **id** field from the drop-down list.
- **Child Table** - select the **users_roles** table from the drop-down list.
- **Child Field** - select the **user_id** field from the drop-down list.
- **Active** - select the check box.

Submit changes on the page.

Create the second Alias Relationship with the following parameters:

- **Group Screen Name** - select the **E-All-Tasks** name from the drop-down list.
- **Parent Table** - select the **c_user_accounts** table from the drop-down list.
- **Parent Field** - select the **id** field from the drop-down list.
- **Child Table** - select the **todo_tasks** table from the drop-down list.
- **Child Field** - select the **user_id** field from the drop-down list.
- **Active** - select the check box.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Alias Relationship for All Tasks Group Screen***.

11.3 Set up Start Screen for Employees Group

The last step in screen configuration is to set the Start screen for the Employees group.

In the Architect application, open Account Settings > Groups to set up the Start screen for Employee users.

Search for the Employees group. Open the Employees group in the update mode by clicking the **edit** icon.

In the opened Update group page, for the Employees group set the following parameters for the Start screen:

- **Start Screen** - select the **E-All-Tasks** screen from the drop-down list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Start Screen for Employees Group***.

11.4 Check Screens for Employee Users in Render

Let us show how the configured screens work in the Render application.

For that, open the Render application and use the Employee user credentials to log in to the Render application.

If the above-mentioned environment for Employee users is set up correctly, we will see the All Tasks screen as the start screen and be able to create, edit, and update tasks.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Screens for Employee Users in Render***.

12 User Onboarding Process

Automating and streamlining the onboarding process for new hires is essential for creating a positive and efficient start to their journey with the company. By leveraging technology and well-structured workflows, you can ensure new hires feel welcomed, informed, and prepared, leading to higher satisfaction, better integration, and improved long-term retention.

The use cases related to the users' onboarding process will include the regular onboarding procedures, such as:

- filling in the user card with personal information
- providing the documents proving the qualification
- procurement of the workstation equipment
- acknowledgement of corporate standards and organizational guidelines.

So, we will extend the Task Manager Tool with the functionality assisting HR specialists in the Onboarding procedure.

Let's shortly describe the use cases for the Onboarding process:

1. The HR user opens the **User Information** tab and fills in the initial data (Personal and Job details) for a new user on the User Info tab.
2. The HR user adds the information about the required training and certifications completed by the user on the **Certification** tab.
3. The HR user assigns the equipment for setting up the workstation for a new employee on the **Equipment** tab:
    a. This step includes a hold step when it is further assigned to the Manager of the user's department
    b. Manager checks the equipment configuration and the related details, updates the equipment details, if needed, and reassigns the step back to the HR user
4. The HR user familiarizes a new employee with the corporate standards and organizational guidelines, rules and instructions and marks the reviewed policies on the **Instructions** tab.
5. Once all Onboarding steps are completed, the Onboarding process can be closed.
6. Email notifications will be sent to the involved users.

12.1 Presetting

Before setting up the functions to execute Onboarding process use cases, we need to set up the necessary environment and create required data in the system. These settings and data will be used in the configuration of the screens and workflow of the Onboarding process.

12.1.1 Set up Architect Screen Access

To set up the environment for the Onboarding process, we create access rules that define what menu items and pages are available for the HR and Manager users in the Architect application.

The HR and Manager users are the main characters of the Onboarding process and can act as system administrators. They will be able to create a new employee profile, add the user information in the database, check if the new employee has certifications or not, and so on. For that, the HR and Manager users should be granted more access to the Architect application.

To provide access to the menu items of the Architect application for HR and Manager users, we need to create a new Architect Screen Access.

In the Architect application, open System Data > Architect Screen Access and create new Architect Screen Access.

Create a new Architect Screen Access for the **HR** group with the following parameters:

- **Group Name** - select the **HR** group from the drop-down list.
- **Menu Name** - select the check boxes of the following menus to access them to the HR group:
    - **Groups**
    - **Users**
    - **Tasks**
    - **Task History**
    - **Task Notes**

Submit changes on the page.

Create a new Architect Screen Access for the **Manager** group with the following parameters:

- **Group Name** - select the **Manager** group from the drop-down list.
- **Menu Name** - select the check boxes of the following menus to access them to the Manager group:
    - **Tasks**
    - **Task History**
    - **Task Notes**

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Architect Screen Access***.

12.1.2 Add Admin Group Rights to Users

To provide users with more rights to access the screens, we need to add admin group rights to users.

In the Architect application, open Account Settings > Users and select the appropriate account. Here, we will add admin group rights to the following users:

- HR
    - Kent Kim
- Managers
    - John Smith
    - Jane Hanson

12.1.2.1 Add Admin Group Rights to HR

To add the admin rights for the HR user (Kent Kim), first, we need to activate and display the **Admin Group Rights** field on Kent Kim's user account page. For that, in the Architect application, go to Account Settings > Users, select and open the Kent Kim's user account page, and update the following parameter:

- **Group Area** - select the **Admin_Group** option in the user's groups list.

Submit changes on the page.

After submitting, we need to select and re-open Kent Kim's user account page again. In the **Admin Group Rights** field on the user account page, select the **HR** group for the HR user (Kent Kim).

12.1.2.2 Add Admin Group Rights to Managers

To add the admin rights for the Manager user (Jane Hanson), first, we need to activate and display the **Admin Group Rights** field on Jane Hanson's user account page. For that, in the Architect application, go to Account Settings > Users, select and open the Jane Hanson's user account page, and update the following parameter:

- **Group Area** - select the **Admin_Group** option in the user's groups list.

Submit changes on the page.

After submitting, we need to select and re-open Jane Hanson's user account page. In the **Admin Group Rights** field on the user account page, select the **Manager** group for the Manager user (Jane Hanson).

To add the admin rights for the Manager user (John Smith), first, we need to activate and display the **Admin Group Rights** field on John Smith's user account page. For that, in the Architect application, go to Account Settings > Users, select and open the John Smith's user account page, and update the following parameter:

- **Group Area** - select the **Admin_Group** option in the user's groups list.

Submit changes on the page.

After submitting, we need to select and re-open John Smith's user account page. In the appeared **Admin Group Rights** field on the user account page, select the **Manager** group for the Manager user (John Smith).

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Add Admin Group Rights to Users***.

12.1.3 Check New Admin Users Access to Architect

We need to check and ensure the HR user (Kent Kim) and Manager users (Jane Hanson and John Smith) have access to the Architect application. After setting Architect Screen Access and adding Admin Group Rights for the HR and Manager groups, they will see the functionalities available according to their admin rights.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check New Admin Users Access to Architect***.

12.1.4 Create Database Table: user_info

To connect the data from the database and the data that will be added to the screens in the Render application, we create a new database table to store the users' information. The user_info table data will be used on the created screens, in the notification settings, and in the report generation process.

In the Architect application, open Features > Tables and create a new table with the following fields:

- **Table Name** - enter a table name. In our case **user_info**.
- In the **Fields/Column Details** section define the following table parameters:

Name	Type	Nullable	Scale	Length
user_info_id	Integer	False	0	-
first_name	Varchar	True	-	-
second_name	Varchar	True	-	-
email	Varchar	True	-	-
phone	Varchar	True	-	-
identify	Varchar	True	-	-
birthday	Date	True	-	-
tax_percent	Numeric	True	2	4
tax_remission_percent	Numeric	True	2	4

job_title	Varchar	True	-	-
employment_type	Varchar	True	-	-
working_days	Varchar	True	-	-
start_work_time	Varchar	True	-	-
end_work_time	Varchar	True	-	-
start_contract	Date	True	-	-
end_contract	Date	True	-	-
document_id	Integer	True	-	-
referral	Varchar	True	-	-
friends	Text (Postgres, MySQL, SQLite) or Varchar (for Oracle SQL)	True	-	-

- **Constraint Type** - select the **Primary Key** constraint type.
- **Constraint Name** - enter the **user_info_pk** constraint name.
- **Columns** - select the **user_info_id** column.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create user_info Table***.

12.1.5 Create Document Family and Document Group

In sections below, we will create a new document family and document groups to work with documents (certifications and recommendation letters) in the Render application and manage document access rights for the HR user (Kent Kim) and Manager users (Jane Hanson and John Smith).

12.1.5.1 Create Document Family

To manage groups of associated documents, we need to create a new document family that contains such document groups.

In the Architect application, open Document Access > Document Family and click the **Create** button.

To create a new document family, fill in the following parameters:

- **Family Name** - enter a document family name. Here, we define the **Documents** value.
- **Family Description** - enter a short description. In our case, we will use the same family description as the family name - **Documents**.
- **Device ID** - select the **C_Codiac_FileStorage** value from the drop-down list.
- **Category** - enter **main** value.
- **Description** - enter **main** value.
- **KP.1** - click the KP.1 field. The pop-up window with the key part list appears. Select the **Alias.user_info.user_info_id** value.

Submit changes on the page.

12.1.5.2 Create Document Group

To manage and control the access permissions and rights allowing to perform actions on documents (files) uploaded on the Render screens, we need to create a new document group.

In the Architect application, open Document Access > Document Groups and click the **Create** button.

To create a new document group, fill in the following parameters:

- **Group Name** - enter a document group name. In our example, we will enter the **Documents_main_full** value.
- **Group Description** - enter a short document group description. In our case, we will use the same group description as the group name - **Documents_main_full**.
- **Document Family** - select **Documents - Documents** from the drop-down list.
- **Document Category** - select the **main** parameter from the list.
- **Access Rights** - select the **Full Access** option. This will provide the possibility for users to perform all the actions (read, write, modify, delete) on documents.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Document Family and Document Group***.

12.1.6 Add Document Group to Users

To provide access to document groups to Admin, HR, and Manager users, we need to add the document groups created in the previous step to users' account pages.

In the Architect application, open Account Settings > Users.

To add document group to Admin, HR, and Manager users, we will need to update the following parameter for every user separately:

- **Document Group** - select the **Document_main_full** group from the list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Add Document Group to Users***.

12.1.7 Create List of Employment Types (EmploymentType)

To demonstrate the possibility to organize and store data in a structured way and further retrieve the data on screens in the Render application as a list of options, we need to create lists. In our Onboarding process example, we create a list containing employment types. The list with employment types will be added to the User Info tab and used as a list of options in the Render application.

For that, in the Architect application, open System data > Lists and create a list.

Fill in the following parameters:

- **List Name** - enter the **EmploymentType** value.
- **Description** - enter a short description that will be displayed as a list option in the Render. In our case **Contract**.
- **Entry Name** - enter the **contract** entry name that will be stored in the database.
- **Weight** - enter **3**.

Submit changes on the page.

After the list is created, we add more list options via the **Update Bulk** functionality. Using the Update Bulk functionality, we can review all the value options belonging to the EmploymentType list listed out on the page and create a new value option for the selected list.

To add new value options to the EmploymentType list in bulk, on the Lists Overview page, select the EmploymentType list and click the **Update Bulk** button at the bottom right corner of the page.

On the opened Update Bulk page for the EmploymentType list, click the **Add more** button to add the priority options to the list.

Add the second list option and fill in the following parameters:

- **Description** - enter the **Part-time** value.
- **Entry Name** - enter the **part-time** value.
- **Weight** - enter **2**.

Add the third list option and fill in the following parameters:

- **Description** - enter the **Full-time** value.
- **Entry Name** - enter the **full-time** value.
- **Weight** - enter **1**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Employment Type (EmploymentType) List***.

12.1.8 Create Days List

Next, we need to create a list that contains weekdays. This list will be added to the User Info screen and used as a list of options in the Render application.

For that, in the Architect application, open System data > Lists and create a list.

Fill in the following parameters:

- **List Name** - enter the **Days** value.
- **Description** - enter a short description that will be displayed as a list option in the Render. In our case **Monday**.
- **Entry Name** - enter the **Monday** entry name that will be stored in the database.
- **Weight** - enter **1**.

Submit changes on the page.

After the list creation we can add remaining options via the **Update Bulk** functionality. For that, on the Lists Overview page we need to search the Days list and perform the update bulk operation.

On the opened Update Bulk page for the Days list, click the **Add more** button to add the priority options to the list.

Add the second list option and fill in the following parameters:

- **Description** - enter the **Tuesday** value.
- **Entry Name** - enter the **Tuesday** value.
- **Weight** - enter **2**.

Add the third list option and fill in the following parameters:

- **Description** - enter the **Wednesday** value.
- **Entry Name** - enter the **Wednesday** value.
- **Weight** - enter **3**.

Add the fourth list option and fill in the following parameters:

- **Description** - enter the **Thursday** value.
- **Entry Name** - enter the **Thursday** value.
- **Weight** - enter **4**.

Add the fifth list option and fill in the following parameters:

- **Description** - enter the **Friday** value.
- **Entry Name** - enter the **Friday** value.
- **Weight** - enter **5**.

Add the sixth list option and fill in the following parameters:

- **Description** - enter the **Saturday** value.
- **Entry Name** - enter the **Saturday** value.
- **Weight** - enter **6**.

Add the seventh list option and fill in the following parameters:

- **Description** - enter the **Sunday** value.
- **Entry Name** - enter the **Sunday** value.
- **Weight** - enter **7**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Days List***.

12.1.9 Create Custom Query for Job Title (JobTitle)

Now, we will create a custom query for job titles. This should be done to get the required data about job titles from the database tables.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create a custom query for Job Title.

Fill in the following parameters:

- **Query Name** - enter the **Get-JobTitle-list** query name.
- **Query Value** - enter the following query value:

select LOWER(group_name) as value, group_name as description from c_group where group_name not in ('Admin_Group', 'Login_Group')

- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-JobTitle-list**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query for Job Title (JobTitle)***.

12.1.10 Create Custom Query for User Names (Usernames)

Then, we will create a custom query for user names. This query will provide the required user names from the database tables.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select "id" as "value", "user_name" as "description", "group_area" from "c_user_accounts" where "group_area" like '%Managers%' or "group_area" like '%Employees%'
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create a custom query for user names.

Fill in the following parameters:

- **Query Name** - enter the **Get-All-Usernames** query name.
- **Query Value** - enter the following query value:

  select user_name as username from c_user_accounts where user_name like '%:username%'
- **Query Params** - enter **:username** parameter.
- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-All-Usernames-for-text-field-and-textarea**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query for User Names (usernames)***.

12.1.11 Create Custom Query for user_info_id

The next custom query that we create is for user_info_id. With this query, we will get the required user_info_id from the database tables.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create a custom query for users.

Fill in the following parameters:

- **Query Name** - enter the **Get-All-Users** query name.
- **Query Value** - enter the following query value:

  select id as value, user_name as description from c_user_accounts
- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-All-Users**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query for user_info_id***.

13 Set up User Information Tab

In this section, we will provide a description of the whole process of the User Info tab creation.

According to our use case, the User Info tab will include details such as user personal information (e.g. ID, first and second names, employment type, contract details (start and end date), working days, job position details, email address, phone number, birthday date, social security number, tax rate, preferences, and specific notes).

To set up the User Info tab, we need to perform the following steps:

1. Create a User Info screen.
2. Create a User Info menu.
3. Create a Group Screens for User Info screen.

Below we will describe every step in detail.

13.1 Create Screen: User Info

This screen will be used as the first step of the user onboarding process, where the HR user can fill in the fields and add information about a new user. For example, user ID, first and second names, employment type, working days, email address, phone number, and so on.

Here, we will create a new **User Info** screen with Screen Builder functionality. For that, open Screen Configurations > Screens and click the **Create with builder** button.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add New Screen** - enter the **UserInfo** screen name.
- **Screen Description** - enter the **UserInfo** screen description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the Simple Search and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

  In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.
- **Tab Text** - enter the **UserInfo** tab text.
- **Tab Description** - enter the **UserInfo** tab description.
- **Screen State** - select the **View** screen state.
- In the **Screen Layout** section select the **Header 2x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen.

On the **Header** level add a section of the **Edit type**.

On the **Section** level we have 2 subsections:

- **The first subsection** - contains the details about user employment type and working conditions. Here, we need to fill in the Default tab and add two more depth tabs by clicking the plus icon to the right of the Default tab:
    - **Default (first) tab** - on the default tab named **Full-time**, users can add the working conditions details for users that are working full-time. Fill in the following parameters for the Full-time tab:
        - **Layout Label** - enter **Full-time** value.
        - **Hide section pop-up** - select the check box.
        - **Section Type** - select the **Edit Type** from the drop-down list.
    - **The second tab** - on the second tab named **Part-time**, users can add the working conditions details for users that are working part-time:
        - **Layout Label** - enter **Part-time** value.
        - **Hide section pop-up** - select the check box.
        - **Section Type** - select the **Edit Type** from the drop-down list.
    - **The third tab** - on the third tab named **Contract**, users can add the working conditions details for users that are working by contract:
        - **Layout Label** - enter **Contract** value.
        - **Hide section pop-up** - select the check box.
        - **Section Type** - select the **Edit Type** from the drop-down list.
- **The second subsection** - contains the details about the user's job title, email address, phone number, birthday, tax range, social security number, and recommendations from the previous job:
    - **Hide section header** - select the check box.
    - **Hide section pop-up** - select the check box.
    - **Section Type** - select the **Edit Type** from the drop-down list.

Click the **Create** button to save changes and create a **UserInfo** screen.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create User Info Screen Layout***.

13.1.1 User Info Screen Header Configuration

To configure the User Info screen header, perform the following operations on the **second** page of the screen builder.

On the **Header** level, add the following fields:

Add the **ID** field that will provide the possibility in the Render application to select the user from the list and add new information to his/her profile details. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the user ID will be available to be selected from the users list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **ID**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only - Except insert**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

Add the **First Name** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the First Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **First Name**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **First Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only - Except insert**
- For the **First Name** field, we need to set up the JavaScript functionality. It should be done to demonstrate the possibilities of the JavaScript functionality as part of field customization.
  To add the JavaScript functionality to the field, we will perform the following actions:
    - In the opened **First Name** field settings pop-up window, click the **Edit JavaScript** button. The **Custom Java Script** pop-up window opens.

- In the opened **Custom Java Script** pop-up window, go to the **Common javascript** tab.
- On the **Common javascript** tab, go to the **onFocusOut** tab and select the **To Upper case** template from the drop-down list.

Add the **Second Name** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the Second Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Second Name**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Second Name**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read only - Except insert**
- For the **Second Name** field, we need to set up the JavaScript functionality. It should be done to demonstrate the possibilities of the JavaScript functionality as part of field customization.
  To add the JavaScript functionality to the field, we will perform the following actions:
  - In the opened **Second Name** field settings pop-up window, click the **Edit JavaScript** button. The **Custom Java Script** pop-up window opens.
  - In the opened **Custom Java Script** pop-up window, go to the **Common javascript** tab.
  - In the **Common javascript** tab, go to the **onFocusOut** tab and select the **To Upper case** template from the drop-down list.

Add the **Employment Type** field that will contain the list with employee types. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.employment_type**.
- **Type** - select the **List** type. The List field type allows users to select an option from the list in the Render application. In our case, users will be able to select the employee type from the drop-down list.
  By selecting the List field type, the **Pull up** button appears. Click the **Pull up** button to open the List Params pop-up window.
- **List Params** - select the **EmploymentType** list from the list.

- **Label** - enter the text that will be displayed on UI for this field, in our case **Employment Type**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Employment Type**.
- **Section refresh** - select the check box to access the additional **Section to Refresh** field.
- **Section to Refresh** - select **Section 1** from the drop-down list.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Click the **Update** button to save changes of the User Info screen.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***User Info Screen Header Configuration***.

13.1.2 User Info Screen Section Configuration

Then, we will define the **first subsection** parameters of the User Info screen.

On the **second** page of the screen builder, add the following fields to the **first subsection** of the Section level:

- On the **Full-time (Default)** tab:

Add the **Remission Percent** field. In the Render application, this field allows adding the calculated remission percent. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.tax_remission_percent**.
- **Type** - select the **Numeric** type. This field type provides the possibility to add a numeric value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Remission Percent**.
- **Format Type** - select the **Decimal - Static** type from the drop-down list.
- **Max Length** - enter **4**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Remission Percent**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Precision** - enter **2**.

- On the **Part-time** tab:

Add the **Working Days** field. This field will be set as the Working Days list with available days (options) to be selected as user working days. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.working_days**.
- **Type** - select the **Multi-select list** type. This field type allows users to select more than one option from the list. In our case, in the Render application, users can select more than one day as a working day.
  By selecting the Multi-select list field type, the **Pull up** button appears. Click the **Pull up** button to access the List Params.
- **List Params** - select the **Days** list from the list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Working Days**.
- **Num Rows** - enter **8**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Working Days**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**

Add the **Start Time** field that will define the user's working time start. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.start_work_time**.
- **Type** - select the **Text** type. This field type allows adding the text value to the field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Start Time**.
- **Format Type** - select the **Custom** type from the drop-down list. This field format type restricts the field to the format template defined in the Format Template field. In our case, the text value of the Start Time field will be added using the numeric characters and has the nn:nn view defined in the Format Template field.
  By selecting the Custom type, the additional Format Template field appears.
- **Format Template** - define the **nn:nn** format type, where **n** is for numeric character input only.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Start Time**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**

Add the **End Time** field that will define the user's working time end. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.end_work_time**.
- **Type** - select the **Text** type. This field type allows adding the text value to the field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **End Time**.
- **Format Type** - select the **Custom** type from the drop-down list. This field format type restricts the field to the format template defined in the Format Template field. In our case, the text value of the End Time field will be added using the numeric characters and has the nn:nn view defined in the Format Template field.
  By selecting the Custom type, the additional Format Template field appears.
- **Format Template** - define the **nn:nn** format type, where **n** is for numeric character input only.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **End Time**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**

- On the **Contract** tab:

Add the **Start Contract** field that will provide the possibility to add the user's start contract date. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.start_contract**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Start Contract**.
- **Format Type** - select the **Date** type from the drop-down list. This field format type allows using a calendar to define the start contract date in the Render application.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Start Contract**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
- Click the **Formatting** button to customize the field look and feel. In the opened Formatting pop-up window, define the **#93c47d** color as a **Text Background Color** parameter for the field.

Add the **End Contract** field that will provide the possibility to add the user's end contract date. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.end_contract**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **End Contract**.
- **Format Type** - select the **Date** type from the drop-down list. This field format type allows using a calendar to define the end contract date in the Render application.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **End Contract**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- Click the **Formatting** button to customize the field look and feel. In the opened Formatting pop-up window, define the **#e06666** color as a **Text Background Color** parameter for the field.

After that, we need to update every tab of the **first subsection**. To update tabs, click the section settings icon on the corresponding tab. In the opened Section Settings pop-up window, define the following parameters:

- **Linked Field** - select the **Alias.user_info.employment_type** field to link the selected section body state to a field value from the header. This parameter can be edited once on the **Full-time (Default)** tab.
- **Active Value** - this parameter should be updated for every tab correspondingly. I.e.:
    - on the **Full-time (Default)** tab, select the **full-time** value from the drop-down list.
    - on the **Part-time** tab, select the **part-time** value from the drop-down list.
    - on the **Contract** tab, select the **contract** value from the drop-down list.

Then, we will move on to the settings for the **second subsection** of the Section level.

Add the **Phone** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.phone**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Phone**.
- **Format Type** - select the **Phone** type from the drop-down list.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Phone**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Email** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.email**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Email**.
- **Format Type** - select the **Email** type from the drop-down list.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Email**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Job Title** field that will be used to add the user's job title. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.job_title**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the job title will be available to be selected from the Job Title list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Job Title**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Job Title**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the **Get-JobTitle-list - Get-JobTitle-list** query from the drop-down list.

Add the **Tax** field. This is a numeric type field that will be used to add the tax value. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.tax_percent**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Tax**.
- **Format Type** - select the **Decimal - Dynamic** type from the drop-down list.
- **Max Length** - enter **4**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Tax**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Precision** - enter **2**.

Add the **Social Security Number** field. This field will be used in the Render application to add the user's social security number in a specific way. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.identify**.
- **Type** - select the **Text** type. This field type allows adding the text value to the field. In our case, we will add the user's social security number to the field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Social Security Number**.
- **Format Type** - select the **Custom** type from the drop-down list. This field format type restricts the field to the format template defined in the Format Template field. In our case, the text value of the Social Security Number field will be added using the text and numeric characters and has the xnn-nnn-nnn view defined in the Format Template field.
  By selecting the Custom type, the additional Format Template field appears.
- **Format Template** - define the **xnn-nnn-nnn** format type, where **x** is for any English text and numeric character and **n** is for numeric character input only.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Social Security Number**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**

Add the **Birthday** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.birthday**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Birthday**.
- **Format Type** - select the **Date** type from the drop-down list. This field format type allows using a calendar to define the birthday date.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Birthday**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**

Add the **Referral** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.referral**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Referral**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Referral**.

- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the **Get-All-usernames - Get-All-usernames-fortext-field-and-textarea** query from the drop-down list.

Add the **Family/Friends** field that will be used to add information about the user's family members and friends. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.friends**.
- **Type** - select the **Textarea** type. This field type allows adding the user's family members and friends data in text format in the Render application.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Family/Friends**.
- **Num Rows** - enter 4.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Family/Friends**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the **Get-All-usernames - Get-All-usernames-for-text-field-and-textarea** query from the drop-down list.
  In the field that appears to the right, the **:username** option will be defined automatically.

Add the field with **Document** type that allows users to upload and download documents on the screen in the Render application. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.document_id**.
- **Type** - select the **Document** type.
- **Document Family** - select the **Documents - Documents** value from the drop-down list. The **Document Category** field appears.
- **Document Category** - select the **main - main** value from the drop-down list.

Add the **Recommendations from the previous job** label that will be used as the label for the field with the Document type. For that, set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this label, in our case **Recommendations from the previous job**.

You can adjust the layout on the fields of the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the appropriate link:

- *The first Subsection Configuration of the User Info Screen*
- *The second Subsection Configuration of the User Info Screen*

13.2 Create User Info Menu

To access the User Info screen created above, which is displayed in the Render application as a User Info tab, we need to create a menu containing this screen.

To create the User Info menu, in the Architect application, open Screen Configurations > Menu and click the **Create** button on the Menu Overview page.

In the opened Update Menu page, fill in the following parameters:

- **Group Name** - select the **Admin_Group** group from the drop-down list.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **User Info**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. Here, we enter the **User Info** value.
- **Weight** - enter **30**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: *Create User Info Menu*.

13.3 Create Group Screens for User Info Screen

Now, we need to link the created User Info screen and Admin_Group group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the following check boxes:
    - **Add**
    - **Delete**
    - **Edit**
    - **Show Add button**
    - **Show Delete button**
    - **Show Edit button**
- **Group Name** - select the **Admin_Group** group we have created previously.
- **Menu Name** - select the **User_Info** menu we have created previously.
- **Screen Name** - select the **UserInfo** screen we have created previously.

- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **User Info** here.
- **Weight** - let us set it to **30**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screens for User Info Screen***.

13.4 Check User Info Screen in Render

Let us show how the configured User Info screen works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned environment for HR and Manager users is set up correctly, we will see the User Info menu on the menu board and have access to the User Info screen. On the User Info screen, we will be able to add, edit, and delete user data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check User Info Screen in Render***.

14 Set up Certificates Tab

The next tab that we will create is named Certificates. The Certificates tab will provide for the HR user (Kent Kim) and Manager users (Jane Hanson and John Smith) the possibility to add the qualification certificates and documents of new employees in the Render application.

The Certification tab set-up process contains the following steps:

1. Create a certificates table.
2. Create a Certificates screen.
3. Set up the certificate assignment to users.

Below we will describe every step in detail.

14.1 Create certificates Table

In this step, we will create a new table for Admin_Group users. This table provides the possibility to store and manage the certificates and screens for Admin_Group users.

In the Architect application, open Features > Tables and create a new table with the following fields:

- **Table Name** - enter a table name. In our case **certificates**.
- In the **Fields/Column Details** section define the following table parameters:

Name	Type	Nullable
user_info_id	Integer	False
name	Varchar	True
description	Varchar	True
image	Text	True

- **Constraint Type** - select the **Primary Key** constraint type.
- **Constraint Name** - enter the **certificates_pk** constraint name.
- **Columns** - select **user_info_id** and **name** columns.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create certificates Table***.

14.2 Create Screen: Certificates

The Certificates screen should be created to provide the possibility for HR and Manager users to upload and store the certificates and documents confirming the employee's qualifications and compliance with the position.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Select existing screen** - as the Certificates screen will be accessed from the User Info screen, we need to select the **UserInfo** screen from the drop-down list.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **10**.
- **Tab Text** - enter the **Certificates** tab text.
- **Tab Description** - enter the **Certificates** tab description.
- **Screen State** - select the **View** state from the drop-down list.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **ID**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Hidden**

Add the **First Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the First Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **First Name**.

- **Tooltip** - enter the text that will be displayed on UI for this field as a tooltip, in our case **First Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Add the **Second Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the Second Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Second Name**.
- **Tooltip** - enter the text that will be displayed on UI for this field as a tooltip, in our case **Second Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

On the **Section** level, add a section of the **Table type**.

Select the following check boxes to adjust section layout:

- **Hide section header**
- **Hide section pop-out**

The next part of the parameters that can be set up in the Section Settings pop-up window is connected to the general table settings.

- **Alias Framework PK Configuration** - this feature is used in special cases when the tables are connected via a composite key.
- **Key parts** - select the **Key parts** to define the fields taken from the tables of different levels (to fill in values from parent to child tables). Here, we will select the **Alias.user_info.user_info_id** option from the drop-down list.
- **Label Orientation** - select the **Top** value from the drop-down list to display the table header in the top row.
- **Show Type** - select the **Scroll** value from the drop-down list.
- **Pixel Height** - enter **500**.

After that, we need to set the table columns, labels, and aliases that should be used to receive the values. For that, add the following alias fields:

- **Array.certificates.name** - enter the **Name** value to be displayed as the column name.
- **Array.certificates.description** - enter the **Description** value to be displayed as the column name.
- **Array.certificates.image** - enter the **Image** value to be displayed as the column name.

Save the changes on the page.

Now, we need to set every column separately. It can be done directly in the table.

To open a column-specific setting, click the **Gear** icon next to the column title. The **Config Column** pop-up window appears. Here, set up the parameters for the following columns:

- **Name**
    - **Param Type** - select the **Text** type from the drop-down list. In the Render application, users will be able to add the certificate name in text format.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
- **Description**
    - **Param Type** - select the **Textarea** type from the drop-down list. In the Render application, users will be able to add the certificate description in text format to the textarea.
    - **Num Rows** - enter **4**.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
- **Image**
    - **Param Type** - select the **DImage** type from the drop-down list. In the Render application, users will be able to upload and download certificates in image formats.
    - **Field Width** - enter **200px**.
    - Select the following check boxes to adjust field layout:
        - **Allow full size modal popup**

Submit changes on the page.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Certificates Screen***.

14.3 Set up Certificate Assignment to Users

To set up the certificate assignment to a new employee, we need to create a dependency between the certificate and user data. According to the created dependency, the HR user (Kent Kim) and Manager users (Jane Hanson and John Smith) will get the possibility to assign a certificate to a new employee in the Certificates tab in the Render application.

In the Architect application, open **Alias Management** > **Alias Dependency for Database Aliases** to manage dependencies for database aliases.

Fill in the following parameters:

- **Alias Type** - select the **Array** type from the drop-down list.
- **Alias Table** - select the **certificates** table from the drop-down list.
- **Request Table** - select the **user_info** table from the drop-down list.
- **Group Screen Name** - select the **DEF** name from the drop-down list.
- **Dependency Type** - select the **PKS** type from the drop-down list.
- **Dependency Alias Type** - select the **Single** alias type from the drop-down list.
- **Dependents on** - select the **Alias.user_info.user_info_id** alias from the drop-down list. Click the Add to List button to add the **Alias.user_info.user_info_id** alias to the dependency list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Certificate Assignment to Users***.

14.4 Check Certificates Screen in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned screen for HR and Manager users is set up correctly, we will see the Certificates screen as a part of the User Info screen displayed as a tab. Here, we will be able to create, edit, and delete table data. Also, in the edit mode, we will be able to upload the certificates and documents in image format.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Certificates Screen in Render***.

15 Set up Equipment Tab

The next tab that we will create is named Equipment. The Equipment tab will provide the possibility for managers or HRs to request for new equipment required for the user's workstation.

To set up the Equipment tab, we need to perform the following steps:

1. Create an equipment table.
2. Create an Equipment screen.
3. Set up the equipment assignment to users.

Below we will describe every step in detail.

15.1 Create equipment Table

In this step, we will create a new table for Admin_Group users. This table provides the possibility to manage equipment and screens for Admin_Group users.

In the Architect application, open Features > Tables and create a new table with the following fields:

- **Table Name** - enter a table name. In our case **equipment**.
- In the **Fields/Column Details** section define the following table parameters:

Name	Type	Nullable	Scale	Length
user_info_id	Integer	False	-	-
type	Varchar	False	-	-
brand	Varchar	True	-	-
price	Numeric	True	2	7
tax	Numeric	True	2	7
year	Integer	True	-	-
status	Varchar	True	-	-

- **Constraint Type** - select the **Primary Key** constraint type.
- **Constraint Name** - enter the **equipment_pk** constraint name.
- **Columns** - select **user_info_id** and **type** columns.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create equipment Table***.

15.2 Create Screen: Equipment

The Equipment screen should be created to provide the possibility for users to request for new equipment.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Select existing screen** - as the Equipment screen will be accessed from the User Info screen, we need to select the **UserInfo** screen from the drop-down list.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **20**.
- **Tab Text** - enter the **Equipment** tab text.
- **Tab Description** - enter the **Equipment** tab description.
- **Screen State** - select the **View** state from the drop-down list.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **ID**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read only**
  - **Hidden**

Add the **First Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**.

- **Type** - select the **Text** type. This field type allows users to enter the text value to the First Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **First Name**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **First Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Add the **Second Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the Second Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Second Name**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Second Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

On the **Section** level, add a section of the **Table type**.

Select the following check boxes to adjust section layout:

- **Hide section header**
- **Hide section pop-out**

The next part of the parameters that can be set up in the Section Settings pop-up window is connected to the general table settings.

- **Alias Framework PK Configuration** - this feature is used in special cases when the tables are connected via a composite key.
- **Key parts** - select the **Key parts** to define the fields taken from the tables of different levels (to fill in values from parent to child tables). Here, we will select the **Alias.user_info.user_info_id** option from the drop-down list.
- **Label Orientation** - select the **Top** value from the drop-down list to display the table header in the top row.

- **Show Type** - select the **Scroll** value from the drop-down list.
- **Pixel Height** - enter **500**.

After that, we need to set the table columns, labels, and aliases that should be used to receive the values. For that, add the following alias fields:

- **Array.equipment.type** - enter the **Type** value to be displayed as the column name.
- **Array.equipment.brand** - enter the **Brand** value to be displayed as the column name.
- **Array.equipment.status** - enter the **Status** value to be displayed as the column name.
- **Array.equipment.year** - enter the **Year** value to be displayed as the column name.
- **Array.equipment.price** - enter the **Price** value to be displayed as the column name.
- **Array.equipment.tax** - enter the **Tax** value to be displayed as the column name.

Save the changes on the page.

Now, we need to set every column separately. It can be done directly in the table.

To open a column-specific setting, click the **Gear** icon next to the column title. The **Config Column** pop-up appears. Here, set up the parameters for the following columns:

- **Type**
    - **Param Type** - select the **Text** type from the drop-down list. In the Render application, users will be able to add the equipment type in text format to the Type column.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
- **Brand**
    - **Param Type** - select the **Text** type from the drop-down list. In the Render application, users will be able to add the equipment brand in text format to the Brand column.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
- **Status**
    - **Param Type** - select the **Datalist** type from the drop-down list. In the Render application, users will be able to select the equipment request status from the list.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
    - **Values (use the delimiter “;”)** - enter **waiting;returned;assigned** value.

- **Year**
    - **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add a year value in numeric format to the Year column.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
    - Click the **Edit JavaScript** button and define the JavaScript functionality to the **Year** column. For that, perform the following steps:
        - In the opened Custom JavaScript pop-up window, go to the Common javascript tab.
        - On the **Common javascript** tab, go to the **onFocusOut** tab and select the **Check max value** template from the drop-down list.
        - In the appeared **Check max size** form, define the:
            - **Max Value** - enter **2050** value.
            - **Error Message** - enter the message to be displayed as an error message in case the entered value in the Render will be greater than the value defined in the Max Value field.
        - After that, select the **Check min size** from the drop-down list.
        - In the appeared **Check min size** form, define the:
            - **Min Value** - enter **1970** value.
            - **Error Message** - enter the message to be displayed as an error message in case the entered value in the Render will be less than the value defined in the Min Value field.
- **Price**
    - **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add a price value in numeric format to the Price column.
    - **Format Type** - select the **Currency - Static** option from the drop-down list. I.e., that in the Render application, the Price column cannot contain fractional symbols. For the entered number, the fractional part is always filled in. For example, for currency you need to type “150” to get “1.50” in the field.
    - **Max Length** - enter **9**.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
- **Tax**

- **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add a tax value in numeric format to the Tax column.
- **Format Type** - select the **Currency - Dynamic** option from the drop-down list. I.e., that in the Render application, the tax column can contain fractional symbols. If the fractional part is missing, it will be automatically filled with zeros. For example, for currency you need to type “1.5” to get “1.50” in the field.
- **Max Length** - enter **9**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Submit changes on the page.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Equipment Screen***.

15.3 Set up Equipment Assignment to Users

To set up the equipment assignment to a new employee, we need to create a dependency between equipment and user data. According to the created dependency, the HR user (Kent Kim) and Manager users (Jane Hanson and John Smith) will get the possibility to assign equipment to a new employee in the Equipment tab in the Render application.

In the Architect application, open Alias Management > Alias Dependency for Database Aliases to manage dependencies for database aliases.

Fill in the following parameters:

- **Alias Type** - select the **Array** type from the drop-down list.
- **Alias Table** - select the **equipment** table from the drop-down list.
- **Request Table** - select the **user_info** table from the drop-down list.
- **Group Screen Name** - select the **DEF** name from the drop-down list.
- **Dependency Type** - select the **PKS** type from the drop-down list.
- **Dependency Alias Type** - select the **Single** alias type from the drop-down list.
- **Dependents on** - select the **Alias.user_info.user_info_id** alias from the drop-down list. Click the Add to List button to add the **Alias.user_info.user_info_id** alias to the dependency list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Equipment Assignment to Users***.

15.4 Check Equipment Screen in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned screen for HR and Manager users is set up correctly, we will see the Equipment screen as a part of the User Info screen displayed as a tab. Here, we will be able to create, edit, and delete equipment data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Equipment Screens in Render***.

16 Set up Instructions Tab

The last step of a new user onboarding process is reviewing and acknowledging the company policies. For that, we need to create one more tab named Instructions where HR user can:

- upload, edit, and download documents and certificates;
- schedule the instruction time for a new employee by selecting the corresponding radio button of the workweek;
- define if a new employee has reviewed and acknowledged the company policies;
- mark the reviewed policies by a new employee by selecting the corresponding check boxes.

To set up the Instructions tab, we need to perform the following steps:

1. Update the user_info table with new fields to be used on the Instructions screen.
2. Create an Instructions screen.
3. Create a custom query for the Instructions screen.
4. Add custom queries to the Instructions screen.

Below we will describe every step in detail.

16.1 Add Fields to user_info Table

To proceed with our onboarding process settings, we need to add new fields to the user_info table. New table data will be used during the Instructions screen creation.

In the Architect application, open Features > Tables and select the user_info table.

To add new fields to the user_info table, in the **Fields/Column Details** section define the following fields:

Name	Type	Nullable	Scale	Length
mentor_eq_type	Varchar	True	-	-
mentor_eq_brand	Varchar	True	-	-
mentor	Varchar	True	-	-
rule1	Char	True	-	-
rule2	Char	True	-	-
rule3	Char	True	-	-

rule4	Char	True	-	-
weekdays_4_3	Char	True	-	-
weekdays_5_2	Char	True	-	-
weekdays_6_1	Char	True	-	-

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Add Fields to user_info Table***.

16.2 Create Screen: Instructions

The Instructions screen will provide the possibility for HR users to organize the transparent procedure for familiarizing new employees with the corporate standards and organizational guidelines, rules and instructions, including defining practical employment norms and outlining written guidelines for internal cooperation.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Select existing screen** - as the Instructions screen will be accessed from the User Info screen, we need to select the **UserInfo** screen from the drop-down list.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **30**.
- **Tab Text** - enter the **Instructions** tab text.
- **Tab Description** - enter the **Instructions** tab description.
- **Screen State** - select the **View** state from the drop-down list.
- In the **Screen Layout** section select the **Header 2x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **ID** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.

- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Hidden**

Add the **First Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the First Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **First Name**.
- **Tooltip** - enter the text that will be displayed on UI for this field as a tooltip, in our case **First Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Add the **Second Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**.
- **Type** - select the **Text** type. This field type allows users to enter the text value to the Second Name field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Second Name**.
- **Tooltip** - enter the text that will be displayed on UI for this field as a tooltip, in our case **Second Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

On the **Section** level we have 2 subsections. For the **first subsection** we need to set parameters as follows:

- **Hide section header** - select the check box.
- **Hide section pop-up** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

After that, we need to add the following fields to the **first subsection** of the Section level:

Add the **Mentor** field. In the Render application, this field will contain the mentor names list. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.mentor**.
- **Type** - select the **List** type. The List field type allows users to select an option from the list in the Render application. In our case, users will be able to select a mentor from the mentor names drop-down list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Mentor**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Mentor**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Mentor Equipment Brand** field. In the Render application, this field will contain the mentor equipment brand list. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.mentor_eq_brand**.
- **Type** - select the **Multi-select list** type. This field type allows users to select more than one mentor equipment brand from the list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Mentor Equipment Brand**.
- **Num Rows** - enter **4**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Mentor Equipment Brand**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Mentor Equipment Type** field. In the Render application, this field will contain the mentor equipment type list. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.mentor_eq_type**.
- **Type** - select the **Inline search** type. This field type allows users to search for the appropriate mentor equipment type from the list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Mentor Equipment Type**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Mentor Equipment type**.

- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Certificates from additional educational institutions** label and set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this field, in our case **Certificates from additional educational institutions label**.

Add the **Certificates** button. In the Render application, clicking this button provides the possibility to upload, view, edit, and download certificates and other documents that can prove the user's qualifications. For that, set up the following parameters:

- **Value** - enter the **Certificates** value as it will be used for uploading certificates and other documents.
- **Button Action** - select the **Document** action from the drop-down list. This button action allows users to manage the documents in the Render application.
- **Document Family** - select the **Documents - Documents** option from the drop-down list.
- **Document Category** - select the **main - main** option from the drop-down list.

After that, we will set up the **second subsection**.

For the **second subsection** we need to set parameters as follows:

- **Hide section header** - select the check box.
- **Hide section pop-up** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

To proceed with the **second subsection** configuration, we need to add the following fields:

Add the **Internal Rules:** label and set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this field, in our case **Internal Rules:** value.

Add the **Rule 1** field that can be selected when users finish reviewing and acknowledging the company policy. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule1**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for Rule 1 as reviewed.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Rule 1**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Rule 1**.

- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Rule 2** field that can be selected when users finish reviewing and acknowledging the company policy. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule2**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for Rule 2 as reviewed.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Rule 2**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Rule 2**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Information security:** label and set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this field, in our case **Information security:** value.

Add the **Local+Web** field that defines in what format the instructions were provided. Set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule3**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for the instructions that were provided in both local and web formats.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Local+Web**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Local+Web**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - enter the **inform_security** value.

Add the **Local** field that defines in what format the instructions were provided. Set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule4**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for the instructions that were provided in local format.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Local**.

- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Local**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - enter the **inform_security** value.

Add the **Workweek:** label and set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this field, in our case **Workweek:** value.

Add the **4+3** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.weekdays_4_3**.
- **Type** - select the **Radio** type. This field type allows users to select one option from the group.
- **Label** - enter the text that will be displayed on UI for this field, in our case **4+3**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **4+3**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - enter the **workweek** value.

Add the **5+2** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.weekdays_5_2**.
- **Type** - select the **Radio** type. This field type allows users to select one option from the group.
- **Label** - enter the text that will be displayed on UI for this field, in our case **5+2**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **5+2**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - enter the **workweek** value.

Add the **6+1** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.weekdays_6_1**.
- **Type** - select the **Radio** type. This field type allows users to select one option from the group.
- **Label** - enter the text that will be displayed on UI for this field, in our case **6+1**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **6+1**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
- **Group Name** - enter the **workweek** value.

Submit changes on the page.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Instructions Screen***.

16.3 Create Custom Query for Instructions Screen

Next, we need to create custom queries to be used on the Instructions screen for fields with the Linked List type.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create the following custom queries:

- **Get-Equipment-Type** - to get the list with equipment types.
- **Get-Equipment-Brand** - to get the list with equipment brands
- **Get-Usernames-By-Equipment-Brand-and-Type** - to get the users list according to the selected equipment type and brand

The first custom query (**Get-Equipment-Type**) will be used to display the list with equipment types on the Instructions screen in the Render application. Fill in the parameters as follows:

- **Query Name** - enter the query name. In our example, we enter **Get-Equipment-Type**.
- **Query Value** - enter the following query value:

  select DISTINCT(type) as "equipType" from equipment where type like '%:equipType%'

- **Query Params** - enter the :**equipType** parameter.
- **Query PKS** - enter the **user_info_id;type** primary key.
- **Description** - enter the following description: **Get-Equipment-Type-for-inline-search**.

Submit changes on the page.

The second custom query (**Get-Equipment-Brand**) will be used to display the list with equipment brands on the Instructions screen in the Render application. Fill in the parameters as follows:

- **Query Name** - enter the query name. In our example, we enter **Get-Equipment-Brand**.
- **Query Value** - enter the following query value:

  select DISTINCT(brand) as "mentor_eq_type" from equipment where type=':mentor_eq_type'

- **Query Params** - enter the **:mentor_eq_type** parameter.
- **Query PKS** - enter the **user_info_id;type** primary key.
- **Description** - enter the following description: **Get-Equipment-Brand-for-linked-list**.

Submit changes on the page.

The third custom query (**Get-Usernames-By-Equipment-Brand-and-Type**) will be used to display the users list according to the selected equipment type and brand on the Instructions screen in the Render application. Fill in the parameters as follows:

- **Query Name** - enter the query name. In our example, we enter **Get-Usernames-By-Equipment-Brand-and-Type**.
- **Query Value** - enter the following query value:

select user_name as "equipBrand" from c_user_accounts where id IN (select user_info_id from equipment where brand IN (:equipBrand) and (':mentor_eq_type'='' OR type=':mentor_eq_type'))

  Query Params - enter the **:equipBrand;:mentor_eq_type** parameter.

- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-Usernames-By-Equipment-Brand-and-Type-for-linked-list**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Custom Query for Instructions Screen***.

16.4 Add Custom Queries to Instructions Screen

Now, we need to update the Instructions screen using the custom queries created in the step above.

In the Architect application, open Screen Configurations > Screen Builder and select the Instructions screen to be updated.

On the **second** page of the screen builder, update the following fields:

- **Mentor Equipment Type**
    - **Linked List** - select the check box to access the Linked List functionality and set the relation between different fields. By selecting this check box, the additional fields appear:
        - **Field ID to Monitor** - select the **Alias.user_info.mentor_eq_brand** option from the drop-down list.
        - **Query to Execute** - select the **Get-Equipment-Brand - Get-Equipment-Brand-for-linked-list** value from the drop-down list.
    - **Custom Query** - select the **Get-Equipment-Type - Get-Equipment-Type-for-inline-search** value from the drop-down list. In the field that appears to the right, the **:equipType** option will be defined automatically.
- **Mentor Equipment Brand**
    - **Linked List** - select the check box to access the Linked List functionality and set the relation between different fields. By selecting this check box, the additional fields appear:
        - **Field ID to Monitor** - select the **Alias.user_info.mentor** option from the drop-down list.
        - **Query to Execute** - select the **Get-Usernames-By-Equipment-Brand-and-Type - Get-Usernames-By-Equipment-Brand-and-Type-for-linked-list** value from the drop-down list.
        - **Additional Field to Send** - select the **Alias.user_info.mentor_eq_type** value from the list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Add Custom Queries to Instructions Screen***.

16.5 Check Instructions Screen in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned screen for Admin_Group users is set up correctly, we will see the Instructions screen as a part of the User Info screen displayed as a tab. Here, we will be able to add, edit, and delete data, and also select the check box(es) and radio button. In the view mode, we will be able to manage (add, edit, view, and delete) certificates and documents by clicking the Certificates button.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Instructions Screen in Render***.

Also, by clicking the Certificates button, you can access the PDF annotation editing if available. Using the Annotation functionality, users that have Admin Rights can leave their comments during the time when the PDF annotation is available.

The time when the PDF annotation is available for editing depends on the Document Category and can be set on the Document Family page (Document Access > Document Family) of the Architect application. For more details, refer to the Document Family page of the AppMill Architect User Manual.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check PDF Annotation***.

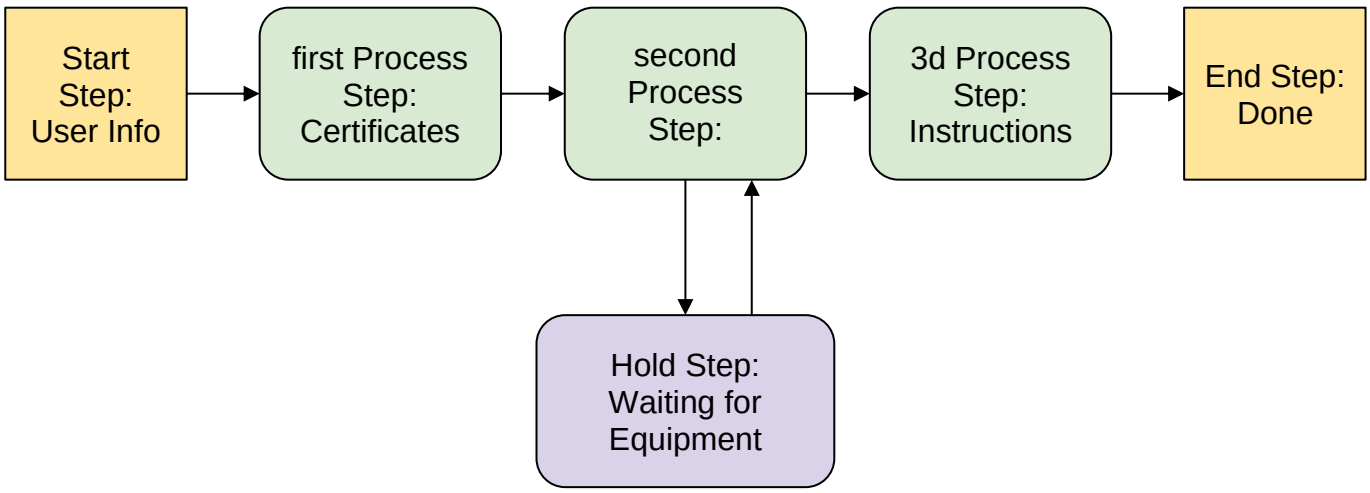
17 Set up Onboarding Process Workflow

The Onboarding Process Workflow demonstrates how the onboarding process is completed step by step from the start point to the end point.

Below we describe the creation of the workflow for the onboarding process in detail.

17.1 Create Workflow Diagram for Onboarding Process

The workflow diagram for Onboarding process includes the workflow steps that will be performed one by one according to the user onboarding process.



The diagram for our case consists of the following steps:

- **Start** step (User Info tab): the HR user adds a new employee's personal information and job details.
- **Process** step (Certificates tab): the HR user adds all additional details relative to the new employee's certifications and qualification.
- **Process** steps (Equipment tab): the HR user assigns the equipment to the new employee.
- **Hold** workflow step (Equipment tab, Waiting step): the HR user waits for the Manager user to approve the requested equipment. The Manager user checks the availability of the requested equipment. To approve the workflow step, the Manager user goes to the Architect application (Workflow > Tasks) and changes the workflow step from Hold to Equipment.
- **Process** steps (Instructions tab): the HR user works with the new employee's additional documents, schedules the instruction time, and the new employee reviews the company policies, instructions, internal rules.
- **End** step: the final overview step will be displayed.

In the Architect application, open Workflow > Workflow Builder and create a new Workflow Diagram.

Fill in the following parameters:

- **Workflow Diagram Name** - enter the workflow diagram name. In this example, we enter the **User_Info_WF** value.
- **Task Description** - enter a description. In our case **User Info Workflow**.
- **Menu** - select the **Admin_Group - User Info** menu from the drop-down list.
- **Auto Advance** - ensure that the check box is disabled.

Add the steps below to the Workflow diagram drawing area and fill in the step parameters:

- **Start** step
    - **Group** - select the **Admin_Group: Admin_Group** group from the drop-down list.
    - **Screen** - select the **UserInfo** screen from the drop-down list.
    - **Step Type** - select the **Start** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Add User Info**.
- **The first Process** step
    - **Name** - enter the **Certificates** step name.
    - **Group** - select the **Admin_Group: Admin_Group** group from the drop-down list.
    - **Screen** - select the **Certificates** screen from the drop-down list.
    - **Step Type** - select the **Process** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Certificates**.
    - **Order** - enter **10**.
- **The second Process** step
    - **Name** - enter the **Equipment** step name.
    - **Group** - select the **Admin_Group: Admin_Group** group from the drop-down list.
    - **Screen** - select the **Equipment** screen from the drop-down list.
    - **Step Type** - select the **Process** step type from the drop-down list.
    - **Label** - enter a label value. In our case **Equipment**.
    - **Order** - enter **20**.
- **Hold** step
    - **Name** - the **Hold** step name is predefined.
    - **Group** - select the **Admin_Group: Admin_Group** group from the drop-down list.
    - **Screen** - select the **Equipment** screen from the drop-down list.

- **Step Type** - select the **Hold** step type from the drop-down list.
- **Label** - enter a label value. In our case **Waiting**.
- **Order** - enter **30**.

- **The 3d Process** step
  - **Name** - enter the **Instructions** step name.
  - **Group** - select the **Admin_Group: Admin_Group** group from the drop-down list.
  - **Screen** - select the **Instructions** screen from the drop-down list.
  - **Step Type** - select the **Process** step type from the drop-down list.
  - **Label** - enter a label value. In our case **Instructions**.
  - **Order** - enter **40**.

- **End** step
  - **Label** - enter a label value. In our case **Done**.

Connect the workflow diagram process steps to each other via arrows.

Save changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Workflow Diagram for Onboarding Process***.

17.2 Check User Info Workflow in Render

After the **User Info** workflow is set up, we need to check it in the Render application. For that, open the Render application and use the HR user credentials to log in to the Render application.

At the top left corner of the page, click the User Info menu item to open the User Info screen. Here, the workflow diagram is displayed under the menu bar.

If the User Info workflow is configured correctly, each step of the user onboarding process will be performed sequentially.

For example:

- In the **Start** step named **Add User Info** on the **User Info** tab, the HR user adds new user data to the form (profile). After that, on the Taskbar, the HR user assigns the next step (Certificates) to the Manager user. I.e., the HR user requests for certificates or other documents from the Manager user.
- In the first **Process** step named **Certificates** on the **Certificates** tab, the Manager user gets the request for certificates from the HR user. On the Certificates tab, the Manager user uploads the certificates to the table. After that, on the Taskbar, he/her assigns the next step (Equipment) to the HR user.

- In the second **Process** step named **Equipment** on the **Equipment** tab, the HR user receives the response from the Manager user and can proceed with the request for equipment for a new user to provide him/her with the company policy and instructions. After filling in the equipment parameters on the Equipment tab, the HR user sends the request to another Manager user for equipment approval.
- In the **Hold** step named **Waiting** on the **Equipment** tab, the Manager user receives the request for equipment approval from the HR user and checks the availability of the requested equipment. After that, the Manager user needs to go to the Architect application (Workflow > Tasks) and change the workflow step from Hold to Equipment manually. This should be done to access the possibility to change the Waiting step to the Equipment step in the Render application. Then, the HR user receives the equipment approval from the Manager user and can assign the next workflow step to the Manager user that will provide the instructions to a new user.
- The 3d **Process** step named **Instructions** is for the **Instructions** tab.
- The **End** step named **Done** appears when the HR user saves the changes on the **Instructions** tab.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check User Info Workflow in Render***.

18 Set up Notifications

The Notification functionality allows you to create and configure notifications that will be sent to users via email depending on the configured settings at specific points within workflows and processes executed on screens (e.g., field value updatings, date editing, etc.) in the Render application.

To show how the email notifications can be sent in connection with the data operations performed on the screen in the Render application, we will configure the notification for the **Birthday**, **Email**, and **Social Security Number** fields of the **User Info** screen.

To configure the Notification functionality, perform the following steps:

1. Set up an email server for Tenant.
2. Update the email address for the Manager user.
   The notifications will be sent to the updated email address. In our example, we update the email address for the Manager user named Jane Hanson.
3. Create notification template for different recipient types:
   - users in the Manager group
   - the custom recipient
   - the corresponding users from the Recipient List
4. Configure the possibility to send notifications according to the changes made in fields (Birthday, Email, and Social Security Number) on the User Info screen.
5. Configure the possibility to send notifications according to the workflow step changes.

Below we will describe every step in detail.

18.1 Set up Email Server for Tenant

The first step of the notification settings is adding an Email Server for the Tenant. We need to use the Email Server as a mail transfer agent that handles receiving incoming email messages and forwarding outgoing ones.

In the Architect application, open Account Setting > Tenant and set up an email server.

For that, open the Default tenant in the edit mode and go to the Email server tab. Here, fill in the following parameters:

- **Email Server** - enter a mail transfer agent responsible for receiving incoming email messages and forwarding outgoing ones.
- **Email Account** - enter the email address that should be used for the Tenant.
- **Email Password** - enter the password used to secure the email account.

Submit changes on the page.

Please note that most fields are hidden in the Tenant Update mode. To view and check these fields, click the **Show hidden fields** button and enter your account password in the pop-up window that appears.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Email Server for Tenant***.

18.2 Update Email for Recipient (Manager User)

To proceed with notification settings, we need to update the email address with the relevant one for the Manager user. This new email address will be used further to receive notifications. As an example, we update the email address for the Manager user named Jane Hanson.

In the Architect application, open Account Settings > Users and select the JaneHanson user account to be updated.

In the opened page with the Jane Hanson user information, enter a new email address to the **Email** field. In our case, we enter **temporaryuser@appmill.com**. Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Email for Manager User to Receive the Notifications***.

18.3 Create Notification Templates

Next, we need to create a notification template that will be sent to the users in case the value in the screen fields is changed.

Here, we set up notifications for the different recipient types:

- **Group**
- **Custom**
- **User**

The notification settings will be different for each of the recipient types. Below we will look closer to each case.

18.3.1 Notifications for Group

We have defined that if the data in the Birthday field of the User Info screen is changed, then the users in the Manager group will receive notifications via email. For that, we need to perform the operations described below.

In the Architect application, open System Data > Notifications and create a new notification template.

On the Create notification page, fill in the following parameters:

- **Notify Name** - enter a name for a notification template. Here, we enter **Update_field_group** name.
- **Notify Type** - select the **Email Notification** type from the drop-down list.

- **Description** - enter a short description. Here, we enter **Update_field_group** description.
- **Recipient Type** - select the **G: Group** recipient type from the drop-down list. According to the selected recipient type, the recipients in the Recipient List will be different. In our case, we have selected the G: Group recipient type, which defines that the notification will be sent to the user group(s) selected in the Recipient List field.
- **Recipient List** - select the **Manager** group from the list.
- **Parameters** - enter the parameters that will be taken into account for sending the notification. Here, we enter **:field_name;:new_value**. I.e., if the value of the Birthday field is changed, a notification will be sent to the users in the Manager group.
- **Subject** - enter the subject of the email or message. To show in the notification subject which field was updated, the Subject field should contain the field name in the format ":field_name" and the text. The subject of the sent email displays the field name according to the field with updated data. E.g., if the value in the Birthday field is updated, the email subject will include the field name and additional text explaining the notification reason. In our case, the email subject will contain the sending notification reason: **Field ":field_name" has been changed**.
- **Body** - here, the visual editor can be used to create the notification body. You can add the notification body according to your preferences. In our example, we use the Source code functionality (View > Source code) to add the notification body in the code format. To ensure that our notification body is displayed correctly, we can use a Preview mode (View > Preview).

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Notification Template for Group***.

18.3.2 Notifications for Custom Recipient

In this step, we need to create a notification template with the Custom recipient type. This notification will be sent to the user only under the specific conditions. To show this functionality, we will use two fields on the User Info screen:

- **Email** - the field with an Email format field type, to which the notification will be sent.
- **Security Social Number** - the field changing of which will initiate the notification sending.

In the Architect application, open System Data > Notifications and create a new notification template.

On the Create notification page, fill in the following parameters:

- **Notify Name** - enter a name for a notification template. Here, we enter **Update_field_custom** name.
- **Notify Type** - select the **Email Notification** type from the drop-down list.
- **Description** - enter a short description. Here, we enter **Update_field_custom** description.

- **Recipient Type** - select the recipient type from the drop-down list. In this case, we have selected the **C: Custom** recipient type to customize the notification sending to users.
- **Parameters** - enter the parameters that will be taken into account for sending the notification. Here, we enter **:field_name;:new_value**. I.e., if the value in fields (Email and Social Security Number) is changed, a notification will be sent to the email address entered into the Email field of the User Info screen in the Render application.
- **Subject** - enter the subject of the email or message. To show in the notification subject which field was updated, the Subject field should contain the field name in the format ":field_name" and text. According to the field with updated data, the subject displays the corresponding field name. For example, if the value in the Social Security Number field is updated, the email subject will include the field name and additional text explaining the notification reason. In our case, the subject will contain the sending notification reason: **Field ":field_name" has been changed.**
- **Body** - here, the visual editor can be used to create the notification body. You can add the notification body according to your preferences. In our example, we use the Source code functionality (View > Source code) to add the notification body in the code format. To ensure that our notification body is displayed correctly, we can use a Preview mode (View > Preview).

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Notification Template for Custom Recipient***.

18.3.3 Notifications for Users

In this step, we create a notification template that will be sent to the corresponding user(s) in case a new workflow starts or the workflow step changes.

In the Architect application, open System Data > Notifications and create a new notification template.

On the Create notification page, fill in the following parameters:

- **Notify Name** - enter a name for a notification template. Here, we enter **Start_workflow_users** name.
- **Notify Type** - select the **Email Notification** type from the drop-down list.
- **Description** - enter a short description. Here, we enter **Start_workflow_users** description.
- **Recipient Type** - select the recipient type from the drop-down list. In this case, we have selected the **U: Users** recipient type, which defines that the notification will be sent to the user(s) selected in the Recipient List field.
- **Recipient List** - select the users to be notified from the list: **Admin** and **JaneHanson**.
- **Parameters** - enter the parameters that will be taken into account for sending the notification. Here, we enter **:wf_name**. I.e., if a new workflow is started, a notification will be sent to the selected users from the Recipient List. E.g, when the HR user (Kent Kim) adds new employee details on the User Info screen in the Render application, he starts a new workflow process. And the notification about it will be sent to the user selected from the Recipient List field.

- **Subject** - enter the subject of the email or message. In our case, the subject will contain the sending notification reason: **New workflow has been started.**

- **Body** - here, the visual editor can be used to create the notification body. You can add the notification body according to your preferences. In our example, we use the Source code functionality (View > Source code) to add the notification body in the code format. To ensure that our notification body is displayed correctly, we can use a Preview mode (View > Preview).

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Notification Template for Users***.

18.4 Configure Notifications for User Info Screen

In this step, we configure the notification to be sent if the field value changes on the User Info screen in the Render, in accordance with our use case.

For that, in the Architect application, open Screen Configurations > Screens and select the User Info screen to be updated.

Go to the **second** page of the screen builder and open the corresponding field settings:

- **Birthday**
  On the **Notifications** tab, click the **Add notification** button, an entry for configuring the notification conditions appears. Fill in the required fields:

  - **Notification** - select the notification template to be sent via email from the drop-down list. In this example, we select **Update_field_group**.

  - **Action** - select the action that specifies the state of the field when notification should be sent. In our case, we select the Update value. I.e., if the field value is edited, the notification will be sent to users in the Manager group.

  - **:field_name** - select the **Field name** value from the drop-down list to connect and send the field name in the notification.

  - **:new_value** - select the **New value** from the drop-down list to connect and send the new value added to the field in the notification.

- **Email** and **Social Security Number**
  Open the **Email** field settings and copy the field **ID** at the bottom left corner of the pop-up window.
  Open the **Social Security Number** field settings and go to the **Notifications** tab. Here, click the **Add notification** button, an entry for configuring the notification conditions appears. Define the following parameters:

  - **Notification** - select the notification template to be sent via email from the drop-down list. In this example, we select **Update_field_custom**

- **Action** - select the action that specifies the state of the field when notification should be sent. In our case, we select the **Update** value. I.e., if the field value is edited, the notification will be sent to the currently logged in user.
- **Recipient Field ID** - enter ID of the Email field: **GetAliasFramework--Alias-user_info-email** I.e., an ID of the field that contains the email address to which the notification (about the current field condition changing) will be sent.
- **:field_name** - select the **Field name** value from the drop-down list to connect and send the field name in the notification.
- **:new_value** - select the **New value** from the drop-down list to connect and send the new value added to the field in the notification.

Save changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link:

- ***Configure Notifications for User Info Screen (Birthday field)***
- ***Configure Notifications for User Info Screen (Email and Social Security Number fields)***

18.5 Configure Notifications for Workflow Process

In this step, we set up the notifications to be sent in the Render application when the workflow steps change.

In the Architect application, open Workflow > Workflow Builder and update the Onboarding workflow settings.

Perform the following operations to add notifications to the workflow link between the **Start step (User Info)** and the **first Process step (Certificates)**:

1. Select and open the **User_Info_WF** workflow.
2. Click the **link** between the **Start step (User Info)** and the **first Process step (Certificates)** and then click the **Settings (wheel gear)** icon.
3. In the section opened on the right side of the drawing area, switch the **Use Notifications** mode on.
4. In the Notification section, fill in the following parameters:
   - **Notify** - select the **email** notification type from the drop-down list.
   - **Notify Template** - select the **Start_workflow_users** notification template from the drop-down list.
   - **Param Name** - select the **:wf_name** parameter from the drop-down list.
   - **Param Value** - enter the **User_Info_WF** workflow diagram name.

Save changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Notifications for Workflow Process***.

18.6 Check Notification in Render

Let us show how the configured notification works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

18.6.1 Check Notification for Group

First, we need to check how the notification works for the Group.

To verify if the notification functionality has been configured correctly, let's update the John Smith user's personal information. This action should initiate the notification sending via email to the Manager user (Jane Hanson).

Go to the User Info screen and perform the following steps:

1. select the user named John Smith.
2. update the birthday date in the Birthday field.
3. save changes.

If the above-mentioned configurations are set up correctly, the Manager user named Jane Hanson will receive a notification about Birthday field value changes via email.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Notification for Group in Render***.

18.6.2 Check Notification for Custom Recipient

Then, we need to check how the notification will be sent to the email address entered in the Email field on the User Info screen in the Render application.

Go to the User Info screen and perform the following steps:

1. select the user named John Smith.
2. update the value in the Email and Social Security Number fields.
3. save changes.

If the above-mentioned configurations are set up correctly, the notification informing that the Social Security Number field value is changed will be sent to the email address entered in the Email field on the User Info screen in the Render application.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Notification for Custom Recipient in Render***.

18.6.3 Check Notification for Users

And the last one is the notification check for users. In case a new workflow starts, the users selected from the Recipient List should receive the notification via email.

Go to the User Info screen and perform the following steps:

1. Start a new workflow process by adding a new user. To start the workflow process, we need to add the Paul Powell user's personal information to the fields on the User Info screen in the Render application.
2. Fill in all fields in the form.
3. Save the user details.

As a result, the workflow process will be changed from the Start step to the first Process step.

If the above-mentioned configurations are set up correctly, the user named Jane Hanson will receive a notification about the new workflow process starting via email.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Notification for Users in Render***.

18.7 Set up Extension Functions for Notification

Also, we can use the plugin file and extension function functionality to set up the notifications' sending to users.

To set up the notification sending using extension functions, we need to perform the following steps:

1. Create an extension function to be used for sending notifications.
2. Add the extension function to the workflow link.
3. Create extension functions for the execute screen.
4. Create a list with extension functions.
5. Update the user_info table with a new field.
6. Create an Execute screen.
7. Create an Execute menu.
8. Create the group screens for the Execute screen.

Below we will describe every step in detail.

18.7.1 Create Extension Function for Notification

To create new extension functions for notifications, we need to open the **ToDoTasks.php** file in the plugins > php directory and enter the following data:

```
    public function SendEmailToAdmin($json_params)
```

```
{
print("-------- START ToDoTasks::SendEmailToAdmin --------\n");

try {
$userEmail = "temporaryuser@appmill.com";
sendEmail($userEmail, "go to next step", "message body");

$info = 'successfully';
$requestresult = 'successfully';
} catch(Exception $e) {
$info = $e->getMessage();
$requestresult = 'unsuccessfully';

var_dump($info);
}

print("-------- STOP ToDoTasks::SendEmailToAdmin --------\n");

return ['resultbody'=> ['requestresult'=> $requestresult, 'extendedinfo'=>  $info]];
}
```

The above-mentioned code contains the notification template (subject and message body) and the recipient email address.

The **ToDoTasks.php** file that was used here can be found in the zip file (appmill_template > service > plugin > php) at the top right corner of the page with video instructions.

This extension function will be used further in the send notification during the workflow process.

In the Architect application, open System data > Extension functions and create a new extension function.

Define the following parameters:

- **Data Source Function** - enter a function value. In our case **ToDoTasks::SendEmailToAdmin**.
- **Data Source Description** - enter a description. In our case **ToDoTasks::SendEmailToAdmin**.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Extension Library** - select the **PHP** library from the drop-down.
- **Extension Function** - enter an extension function value. In our case **ToDoTasks::SendEmailToAdmin**.

Submit changes on the page.

After creating a new extension function, we recommend restarting the AppMill Service application to avoid incorrect work.

For that, follow the steps below:

1. Enter **Services** to the search area in the Taskbar.

2. Open the **Services** application.
3. In the **Services** application window:
    a. select the **AppMill Service** application and
    b. click the **Restart** link at the top right corner of the window.

    As a result, the AppMill Service application will be restarted.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Extension Function for Notification***.

18.7.2 Add Extension Function to Workflow Link

In this step, we need to add a new extension function to the workflow link between the **first Process step (Certificates)** and the **second Process step (Equipment)** of the workflow process. According to this use case, when the user is going from the Certificates screen to the Equipment screen, an email notification will be sent.

In the Architect application, open Workflow > Workflow Builder and update settings of the workflow link between the **first Process step (Certificates)** and the **second Process step (Equipment)** of the workflow process.

To add notifications to the workflow link, perform the following operations:

1. Select and open the **User_Info_WF** workflow.
2. Click the **link** between the **first Process step (Certificates)** and the **second Process step (Equipment)** and then click the **Settings (wheel gear)** icon.
3. In the section opened on the right side of the drawing area, select the **CodiacSDK.PhpWrapper | ToDoTasks::SendEmailToAdmin** value from the **Step Code** drop-down list.

Save changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Add Extension Function to Workflow Link***.

18.7.3 Check Extension Function for Workflow Link in Render

Let us show how the configured extension function works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned extension function is set up correctly, we will receive the notification via email when the HR user is going from the first Process step (Certificates) and the second Process step (Equipment) of the workflow process.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Extension Function for Workflow Link in Render***.

18.7.4 Create Extension Functions for Execute Screen

Before we create the extension functions, we need to add new data (notification templates) to the **ToDoTasks.php** file in the plugins > php directory. These extension functions will be further used on the Execute screen in the Render application to provide the possibility of sending notifications via email.

The notification templates in the **ToDoTasks.php** file contain data in code format, including the subject, message body, and the recipient email address. For example:

```
public function FirstFunction ($json_params)

        {
        print("-------- START ToDoTasks::FirstFunction --------\n");

        try {
        $data = (array) $json_params['patch_json'];
        $userEmail = $json_params["user_email"];
        $header = "First Function";
        $message = "Body of First Function";
        sendEmail($userEmail, $header, $message);

        $info = 'successfully';
        $requestresult = 'successfully';
        } catch(Exception $e) {
        $info = $e->getMessage();
        $requestresult = 'unsuccessfully';

        var_dump($info);
        }

        print("-------- STOP ToDoTasks::FirstFunction --------\n");

        return ['resultbody'=> ['requestresult'=> $requestresult, 'extendedinfo'=>  $info]];
        }

        public function SecondFunction ($json_params)
        {
        print("-------- START ToDoTasks::SecondFunction --------\n");

        try {
        $data = (array) $json_params['patch_json'];
        $userEmail = $json_params["user_email"];
        sendEmail($userEmail, "Second Function", "Body of Second Function");

        $info = 'successfully';
        $requestresult = 'successfully';
        } catch(Exception $e) {
        $info = $e->getMessage();
```

```
$requestresult = ‘unsuccessfully’;

var_dump($info);
}

print(“-------- STOP ToDoTasks::SecondFunction --------\n”);

return [‘resultbody’=> [‘requestresult’=> $requestresult, ‘extendedinfo’=>  $info]];
}

public function ThirdFunction ($json_params)
{
print(“-------- START ToDoTasks::ThirdFunction --------\n”);

try {
$data = (array) $json_params[‘patch_json’];
$userEmail = $json_params[“user_email”];
sendEmail($userEmail, “Third Function”, “Body of Third Function”);

$info = ‘successfully’;
$requestresult = ‘successfully’;
} catch(Exception $e) {
$info = $e->getMessage();
$requestresult = ‘unsuccessfully’;

var_dump($info);
}

print(“-------- STOP ToDoTasks::ThirdFunction --------\n”);

return [‘resultbody’=> [‘requestresult’=> $requestresult, ‘extendedinfo’=>  $info]];
}
```

After adding new notification templates to the **ToDoTasks.php** file, we need to create new extension functions. For example, we create new extension functions named the first Function, the second Function, and the third Function.

The **ToDoTasks.php** file that was used here can be found in the zip file (appmill_template > service > plugin > php) at the top right corner of the page with video instructions.

To create new extension functions named the first Function, the second Function, and the third Function one-by-one in the Architect application, open System Data > Extension Functions, click the **Create** button, and fill in the following parameters for each extension function correspondingly:

- The **first Function**
    - **Data Source Function** - enter a function value. In our case **ToDoTasks::FirstFunction**.
    - **Data Source Description** - enter a description. In our case **ToDoTasks::FirstFunction**.

- **Primary Table** - select the **user_info** table from the drop-down list.
- **Extension Library** - select the **PHP** library from the drop-down.
- **Extension Function** - enter an extension function value. In our case **ToDoTasks::FirstFunction**.

Submit changes on the page.

- The **second Function**
  - **Data Source Function** - enter a function value. In our case **ToDoTasks::SecondFunction**.
  - **Data Source Description** - enter a description. In our case **ToDoTasks::SecondFunction**.
  - **Primary Table** - select the **user_info** table from the drop-down list.
  - **Extension Library** - select the **PHP** library from the drop-down.
  - **Extension Function** - enter an extension function value. In our case **ToDoTasks::SecondFunction**.

  Submit changes on the page.
- The **third Function**
  - **Data Source Function** - enter a function value. In our case **ToDoTasks::ThirdFunction**.
  - **Data Source Description** - enter a description. In our case **ToDoTasks:ThirdFunction**.
  - **Primary Table** - select the **user_info** table from the drop-down list.
  - **Extension Library** - select the **PHP** library from the drop-down.
  - **Extension Function** - enter an extension function value. In our case **ToDoTasks::ThirdFunction**.

  Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Extension Functions for Execute Screen***.

18.7.5 Create List with Extension Functions

Next, we create a list that contains extension functions created in the previous step. This list will be added to the Execute screen as an example of how the notifications can be sent according to the selected option from the list in the Render application.

For that, in the Architect application, open System data > Lists and create a list with extension functions.

To create the list with the first list option, fill in the following parameters:

- **List Name** - enter the **EmailTemplates** value.

- **Description** - enter a short description that will be displayed as a list option in the Render. In our case **first**.
- **Entry Name** - enter the entry name that will be stored in the database. In our case **first**.
- **Extension Function** - select the **CodiacSCD.PhpWrapper - ToDoTasks::FirstFunction - ToDoTasks::FirstFunction** value from the drop-down list.
- **Weight** - enter **1**.

Submit changes on the page.

After the list creation, we can add more list options via the **Update Bulk** functionality. For that, on the Lists Overview page we need to search the EmailTemplates list and perform the update bulk operation.

On the opened Update Bulk page for the EmailTemplates list, click the **Add more** button to add the priority options to the list.

Add the second list option and fill in the following parameters:

- **Description** - enter a short description that will be displayed as a list option in the Render. In our case **second**.
- **Entry Name** - enter the entry name that will be stored in the database. In our case **second**.
- **Extension Function** - select the **CodiacSCD.PhpWrapper - ToDoTasks::SecondFunction - ToDoTasks::SecondFunction** value from the drop-down list.
- **Weight** - enter **2**.

Add the third list option and fill in the following parameters:

- **Description** - enter a short description that will be displayed as a list option in the Render. In our case **third**.
- **Entry Name** - enter the entry name that will be stored in the database. In our case **third**.
- **Extension Function** - select the **CodiacSCD.PhpWrapper - ToDoTasks::ThirdFunction - ToDoTasks::ThirdFunction** value from the drop-down list.
- **Weight** - enter **3**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create List with Extension Functions***.

18.7.6 Update user_info Table

It is necessary to add a new table field to the user_info table. This new table field will be further used to create an Execute screen. Let us add this field.

In the Architect application, open Features > Tables and select the **user_info** table to be updated.

Add a new table field with the following parameters:

Name	Type	Nullable	Scale	Length
execute_function	Varchar	True	-	-

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update user_info Table***.

18.8 Create Screen: Execute

The Execute screen will be used to show the possibilities of sending notifications from the screen in the Render application.

Here, we will create a new **Execute** screen with Screen Builder functionality. For that, open Screen Configurations > Screens and click the **Create with builder** button.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add New Screen** - enter the **Execute** screen name.
- **Screen Description** - enter the **Execute** screen description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the Simple Search and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

  In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.
- **Tab Text** - enter the **Execute** tab text.
- **Tab Description** - enter the **Execute** tab description.
- **Screen State** - select the **View** screen state.
- **Execute** - click the **Function** button to add a new extension function as a main function to the Execute action. In the opened Extensions Settings pop-up window, define the extension function that will be executed when the user clicks the **Execute** button in the main actions' menu on the screen in the Render application. In our case as an example, we select the **ToDoTasks::SecondFunction** option from the Extensions drop-down list and add it to the **Execute** action.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen.

On the **Header** level add a section of the **Edit type**.

Add the **ID** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **ID**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip for this field, in our case **ID**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

On the **Section** level we need to set up the following parameters:

- **Hide section header** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

After that, we need to add the following fields to the Section level:

Add the **Email Template** field. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.execute_function**.
- **Type** - select the **List With Extension Function** type. The **List With Extension Function** field type allows users to select an option from the list in the Render application. In our case, users will be able to select an extension function from the extension function drop-down list.
- **Pull up** - click the **Pull up** button and select the **EmailTemplates** list from the List Params pop-up window.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Email Template**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Email Template**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Send first template** button and set up the following parameters:

- **Value** - enter **Send first template** as it will be used for sending the first notification template.
- **Button Action** - select the **Execute** action.
- **Function** - add a new extension function. In the opened Extensions Settings pop-up window, define the extension function that will be executed when the user clicks the **Send first template** button on the screen in the Render application. In our case as an example, we select the **ToDoTasks::FirstFunction** option from the Extensions drop-down list and click the **Add** button.

Add the **Send second template** button and set up the following parameters:

- **Value** - enter **Send second template** as it will be used for sending the first notification template.
- **Button Action** - select the **Execute** action.
- **Function** - add a new extension function. In the opened Extensions Settings pop-up window, define the extension function that will be executed when the user clicks the **Send second template** button on the screen in the Render application. In our case as an example, we select the **ToDoTasks::SecondFunction** option from the Extensions drop-down list and click the **Add** button. After that, make sure that the **ToDoTasks::SecondFunction** is the only one extension function in the list. If not, delete an extra one. In our example, we need to delete the **ToDoTasks::FirstFunction** extension function from the list to avoid conflicts while sending the notifications.

Add the **Send third template** button and set up the following parameters:

- **Value** - enter **Send third template** as it will be used for sending the first notification template.
- **Button Action** - select the **Execute** action.
- **Function** - add a new extension function. In the opened Extensions Settings pop-up window, define the extension function that will be executed when the user clicks the **Send third template** button on the screen in the Render application. In our case as an example, we select the **ToDoTasks::ThirdFunction** option from the Extensions drop-down list and click the **Add** button. After that, make sure that the **ToDoTasks::ThirdFunction** is the only one extension function in the list. If not, delete an extra one. In our example, we need to delete the **ToDoTasks::SecondFunction** extension function from the list to avoid conflicts while sending the notifications.

Submit changes on the page.

You can adjust the layout of the fields on the screen according to your preferences.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Screen: Execute***.

18.8.1 Create Execute Menu

To access the **Execute** screen created above in the Render application, we need to create a menu containing this screen.

To create the Execute menu, in the Architect application, open Screen Configurations > Menu and click the **Create** button on the Menu Overview page.

In the opened Update Menu page, fill in the following parameters:

- **Group Name** - select the Admin_Group group from the drop-down list.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Execute**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. Here, we enter the **Execute** value.
- **Weight** - enter **35**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Execute Menu***.

18.8.2 Create Group Screens for Execute Screen

Now, we need to link the created Execute screen and Admin_Group group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the following check boxes:
  - **Edit**
  - **Execute**
  - **Show Edit button**
  - **Show Execute button**
- **Group Name** - select the **Admin_Group** group we have created previously.
- **Menu Name** - select the **Execute** menu we have created previously.
- **Screen Name** - select the **Execute** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Execute** here.
- **Weight** - let us set it to **0**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screens for Execute Screen***.

18.8.3 Check Execute Screen in Render

Let us show how the configured Execute screen works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned environment for the HR user (Kent Kim) and Manager users (Jane Hanson and John Smith) is set up correctly, we will see the Execute menu on the menu board and have access to the Execute screen. On the Execute screen, we will be able to edit the data on the Execute screen, execute extension functions, and receive notifications via email.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Execute Screen in Render***.

19 Configure Reports

The AppMill solution provides the possibility to aggregate any data and information from the database. The aggregated data can be displayed in the PDF form. The AppMill Reports functionality provides the possibility to generate the data in PDF format.

In scope of the current instruction, we will see the following report types and ways to configure report templates:

- **System Reports:** this report type refers to the report templates created via a designer application (QtRPT application). That means the report templates are created via a third-party tool (QtRPT application) integrated with the AppMill Reports solution. You can select these templates for the report generation directly in the Architect application.
- **PDF Form**: this report template can be prepared via any PDF editor and then uploaded into Architect as a PDF file. You can use the uploaded PDF templates for the report generation in the Architect application.
- **Editable:** this report template is created directly in the Architect application using all the facilities provided by the Architect Report Builder feature. You just need to configure the required parameters and the data to be displayed in the report.

It is important to consider that all the reports are generated based on the pre-created report templates. To be sure that you get the needed report and data in the result of report generation, you need to ensure that the required report template exists in the system and the custom aliases are set up correctly.

To guide you through the Reports functionality, we will describe the report types and the way how to create the required report templates mentioned above accordingly. We will use the data from the Task Manager and Onboarding usage example.

19.1 Prerequisites

And before going directly to the explanation of the report configuration process, we need to make the following preparation:

- Set up Reports folder on your computer where the reports will be temporarily stored during the generation process.
- Set up custom aliases. These aliases will be used to link the report parameters with the data from the database. If the aliases are defined correctly, the data displayed in the report will be correct.

See further the details on how to make the required pre-settings for our example.

19.1.1 Set up Reports Folder

Let us set up or check the existing folder on your computer where the reports will be temporarily stored during the generation process. I.e., in the Render application, when the report generation is started, the report will temporarily appear in this reports folder. When the report generation is finished, the report will be automatically transferred to the folder defined in the Document Device.

Ensure that the folder path defined in the Default tenant matches the path to the available folder on your device:

- Check the path to your device folder

To check that the temporal report folder exists, in the Architect application, go to Account Settings > Tenant and open the Default tenant. Here, ensure that the **File Report Path** field is defined and contains the path to the existing directory or folder on your computer. The generated reports will be temporarily placed into this folder. You can define the path to any local directory or folder on your device.

In our case, the **File Report Path** field contains the following path to the **ReportTemp** folder: **C:\Program Files\AppMill\ReportTemp**.

- Check the folder on your device

After that, ensure that the **ReportTemp** folder exists on your device. If this folder exists, then we can use it as a repository for the reports in the generation process. If the **ReportTemp** folder does not exist, you need to create a new folder with the name equal to the name defined in the Default tenant.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Reports Folder***.

19.1.2 Set up Custom Aliases

To proceed with the report generation settings, we need to create custom aliases that will be used in the report template.

In the report templates, the fields will contain aliases to get the required data from the database. In our example, we will display the user's First and Second Names and show the values of check boxes on the Instructions screen in the generated report. In the generated reports, the fields with the aliases will display the user's data from the database.

In the Architect application, open Alias Management > Custom Generated and create a new custom alias.

Define the following parameters for the value of the Rule 1 check box alias:

- **Alias Code** - in this field, we enter the custom alias code according to the naming convention. The alias code contains 3 parts:
    - first part - automatically filled in with the **Custom** value, as we are creating the alias with the Custom Generated type.
    - second part - a table name.
    - third part - a table column name.

Every part of the alias code is separated by the period. For example, we enter the **Custom.user_info.rule1_checkbox** value.

- **Description** - enter a description for the alias. Here, we enter the **Custom.user_info.rule1_checkbox** value.

- **Type** - this field is automatically filled in with the **Custom Generated** value, as we are creating the alias with the Custom Generated type.
- **SQL Statement** - note that the SQL query may vary for different databases. In this field, we enter the following SQL query:

SELECT CASE WHEN rule1='1' THEN 'V' ELSE 'X' END AS "value" FROM user_info WHERE user_info_id=:Alias.user_info.user_info_id

Define the following parameters for the value of the Rule 2 check box alias:

- **Alias Code** - in this field, we enter the custom alias code according to the naming convention. The alias code contains 3 parts:
    - first part - automatically filled in with the **Custom** value, as we are creating the alias with the Custom Generated type.
    - second part - a table name.
    - 3d part - a table column name.

Every part of the alias code is separated by the period. For example, we enter the **Custom.user_info.rule2_checkbox** value.

- **Description** - enter a description for the alias. Here, we enter the **Custom.user_info.rule2_checkbox** value.
- **Type** - this field is automatically filled in with the **Custom Generated** value, as we are creating the alias with the Custom Generated type.
- **SQL Statement** - note that the SQL query may vary for different databases. In this field, we enter the following SQL query:

SELECT CASE WHEN rule2='1' THEN 'V' ELSE 'X' END AS "value" FROM user_info WHERE user_info_id=:Alias.user_info.user_info_id

Define the following parameters for the value of the Rule 3 check box alias:

- **Alias Code** - in this field, we enter the custom alias code according to the naming convention. The alias code contains 3 parts:
    - first part - automatically filled in with the **Custom** value, as we are creating the alias with the Custom Generated type.
    - second part - a table name.
    - 3d part - a table column name.

Every part of the alias code is separated by the period. For example, we enter the **Custom.user_info.rule3_checkbox** value.

- **Description** - enter a description for the alias. Here, we enter the **Custom.user_info.rule3_checkbox** value.

- **Type** - this field is automatically filled in with the **Custom Generated** value, as we are creating the alias with the Custom Generated type.
- **SQL Statement** - note that the SQL query may vary for different databases. In this field, we enter the following SQL query:

SELECT CASE WHEN rule3='1' THEN 'V' ELSE 'X' END AS "value" FROM user_info WHERE user_info_id=:Alias.user_info.user_info_id

The alias reload is not required here. It is performed automatically when creating the aliases. But if you need to do this, go to **Features** > **Management** and click the **Reload aliases** button.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Custom Aliases***.

19.2 Create System Report

The first report type that we want to introduce to you is a **System** report.

The System report template type refers to pre-existing report templates within the system. Here, we can use the templates that we will create using QtRPT web or desktop applications. The report templates can be selected and used for report generation on the Report Builder page in the Architect application.

19.2.1 Create Report Template via QtRPT application

The QtRPT application web and desktop versions are effective tools to prepare the report templates for generating reports. We will show both web and desktop versions of the QtRPT application. You can work on one report template using both the web and desktop versions of the QtRPT application. For example, you can create a new report template in the web QtRPT application and then update it in the desktop QtRPT application, and vice versa.

With QtRPT, we can create custom reports that can include tables, images, charts, and more. It allows for dynamic content generation based on the application's data.

19.2.1.1 Create Report Template via QtRPT web application

The QtRPT web application can be accessed by the link. For example, **localhost:88/architect/webdesigner/**.

For our example, log in to the QtRPT web application using the credential of the Manager or HR user.

As a report template, we will use an insurance policy document. This type of document may include the user's personal information, such as their first and last names, email address, phone number, etc. Using this document, we can demonstrate the possibilities of the report generation functionality - from creating the report templates to generating the final report in the Render application. And when you click the Report Generation button on the Reports screen, the insurance policy document will be automatically populated with the corresponding user data.

To create a new report template via QtRPT web application, open the QtRPT web application by link and perform the following steps:

1. **Add a PDF file that will be used as a report template**

Select the **PDF -> Report** icon from the Toolbar at the top of the QtRPT web application page. In the opened **PDF -> Report** pop-up window, select the PDF file for the template located on your local computer.

2. **Reduce the file size**

Reduce the report template size to have access to the free space on the page. This should be done to insert a new report template element into the existing one.

3. **Add aliases**

Add aliases to the report template. For that, click the **Add new text** icon from the Toolbar on the left.

In the opened **Field editor** pop-up window, search the alias in the **Aliases** field. Select the corresponding alias from the search results.

The alias name should be in the square brackets. Otherwise, the alias will not apply the correct data and will be displayed as regular text. If there is any text outside the square brackets, just delete it. Ensure that the text area contains only the alias in the square brackets. In our example, we search and add every alias separately:

Add the aliases listed below to the field that should contain the user's First and Second names:

- **Alias.user_info.first_name** - user's First Name.
- **Alias.user_info.second_name** - user's Second Name.

Add the aliases listed below to display the marked by the user check boxes on the Instructions screen in the Render application:

- **Custom.user_info.rule1_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 1 check box.
- **Custom.user_info.rule2_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 2 check box.
- **Custom.user_info.rule3_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 3 check box.

4. **Define a primary table**

Select the **Table** icon from the Toolbar at the top of the page. In the opened **Tables** pop-up window define the following parameters:

- **Request primary table** - select the **user_info** table from the drop-down list.
- **Name of alias** - click the plus icon next to the Action column name and select the **Alias.user_info.user_info_id** alias from the drop-down list.
- **Value of alias** - enter the Jane Hanson user ID. It is 6.

5. **Save a template**

To save the report template, perform the following steps:

- open the **Templates directory** (Service > Plugins > Templates directory).
- add a new report template name to the **Template saving** field.
- click the **Save as new** icon.

If you click the **Save** icon from the Toolbar at the top left corner of the page, that will download/save the XML template of the report you are working on.

Note that clicking the **Close** button will save the report template and close the **Templates** pop-up window. Clicking the **Close** icon at the top right corner of the pop-up window will cancel the data saving and close the **Templates** pop-up window.

6. Review a configured report

Before adding the configured report template to the Render application, it is recommended to check the created template. Click the **Preview** icon at the top left corner of the page. Ensure that the report will be generated with the Jane Hanson user data that contains the First and Second names and marked check boxes for the Rules.

Note: to have the report template preview option enabled in the designer application, the following parameter values should be set up on the Document Groups page (Document Access > Document Groups) in the Architect application:

- Document Family: system
- Document Category: system
- Access Rights: Full Access

7. Check the report generation for other users

To check the report generation for the user named John Smith, the following actions should be done:

- In the **Tables** pop-up window, enter **5** to the **Value of alias** field.
- In the **Templates** pop-up window, save a new report template with the same name and delete the old report template by clicking the **minus** icon. The old report template is placed higher than the new one.
- Click the **Preview** icon at the top left corner of the page. The report will be generated with the John Smith user data that contains the First and Second names and marked check boxes for the Rules.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template via QtRPT web application***.

19.2.1.2 Update Report Template via QtRPT web application

In the example below, we show the possibility to update the report template using the QtRPT web application.

To update the report template using the QtRPT web application, perform the following actions:

- Open the report template to be updated. For that, click the **Templates** icon from the Toolbar at the top of the page. In the opened **Templates** pop-up window, select the corresponding report template.
- Change the report template and click the **Save** icon.

To check the updated report template, click the **Preview** icon at the top left corner of the page. The report will be generated with the updated data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Report Template via QtRPT web application***.

19.2.1.3 Create Report Template via QtRPT desktop application

There is the possibility to create a new report template using the desktop version of QtRPT application.

To create a new report template using the QtRPT desktop application, you need to perform the actions described below:

- **Open the QtRPT desktop application**

Run the **QtRptDesigner.exe** file from the QtRptDesigner folder. In our case, the path to the execution is as follows: **C:/Program Files/AppMill/Clients/QtRptDesigner/QtRptDesigner.exe**.

- **Connect to the Service using the Configuration**

In the application login screen, click the **Configuration** button. In the opened **Configuration** pop-up window, define the following parameters:

  - **Service and port** - enter the service and port values. In our example, **localhost:34567**.
  - **Request key** - is a key that will be used for encrypting messages with the service. Enter any value to this field. Note that the field cannot be empty. In our example, we enter the **no** value.

Save the changes. The data added to the **Configuration** pop-up window will be saved and can be used for further login.

- **Login to the QtRPT desktop application**

Log in to the QtRPT desktop application using the Admin user credentials.

- **Add a new report template**

To add a new report template, perform the following actions:

  - Select the report template from the **PDF2QtRPT utility** (Service > Plugins > PDF2QtRPT utility).
  - In the opened **PDF2QtRPT converting tool** pop-up window, click the **Select PDF** button to upload the PDF file to be used as a report template.

In the opened report template, we can add new text elements. For our case, we need to add aliases to the report template.

Click the **New text** icon from the Toolbar on the left to add a new alias. Note that the aliases should have the round number for the left and top placement values defined on the panel to the left from the report template. Otherwise, the user data and check boxes will be displayed incorrectly in the generated report.

Requirements to the alias name: The alias name should be in the square brackets. Otherwise, the alias will not apply the correct data and will be displayed as regular text. If there is any text outside the square brackets, just delete it. Ensure that the text area contains only the alias in the square brackets. In our example, we search and add every alias separately:

Add the aliases to display the user's First and Second names:

- **Alias.user_info.first_name** - user's First Name.
- **Alias.user_info.second_name** - user's Second Name.

Add the aliases to display the marked by the user check boxes on the Instructions screen in the Render application:

- **Custom.user_info.rule1_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 1 check box.
- **Custom.user_info.rule2_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 2 check box.
- **Custom.user_info.rule3_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 3 check box.

- **Edit input parameters for the report template**

Also, we need to set up the primary table. For that, open Service > Plugins > Edit input parameters and perform the following actions:

- **Request primary table** - select the **user_info** table from the drop-down list.
- **Add Alias** - click this button to add the alias connected to the **user_info** table. Clicking this button access the following parameters:
  - **Name of alias** - select the **Alias.user_info.user_info_id** alias from the drop-down list.
  - **Value of alias** - enter 6. It is the user ID of Jane Hanson.

Note that clicking the **Close** button will save the report template and close the **Input parameters editing** pop-up window. Clicking the **Close** icon at the top right corner of the window will cancel the data saving and close the **Input parameters editing** pop-up window.

- **Preview a new report template**

Before adding the configured report template to the Render application, it is recommended to check the created template. Click the **Preview** icon at the top left corner of the page. Ensure that the report will be

generated with the Jane Hanson user data that contains the First and Second names and marked check boxes for the Rules.

Note: to have the report template preview option enabled in the designer application, the following parameter values should be set up on the Document Groups page (Document Access > Document Groups) in the Architect application:

- Document Family: system
- Document Category: system
- Access Rights: Full Access
- Save a new report template.

To save the report template, perform the following steps:

- open the **Templates directory** (Service > Plugins > Templates directory).
- add a new report template name to the **Template saving** field.

Click the **Save as new** icon.

To check the updated report template, click the **Preview** icon at the top left corner of the page. The report will be generated with the updated data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template via QtRPT desktop application***.

19.2.1.4 Update Report Template via QtRPT desktop application

In this step, we show how to update the report template using the QtRPT desktop application.

To update the existing report template using the QtRPT desktop application, we need to perform the actions described below:

- **Open the QtRPT desktop application**

Open the **QtRptDesigner.exe** application in the QtRptDesigner folder. In our case, the path to the application is as follows: **C:/Program Files/AppMill/Clients/QtRptDesigner/QtRptDesigner.exe**.

- **Connect to the Service using the Configuration**

In the application login screen, click the **Configuration** button. In the opened **Configuration** pop-up window, ensure that the following parameters are predefined with the data we entered at the first login to the desktop application:

- **Service and port - localhost:34567**.
- **Request key - no**.

Click the **Save** button to save data and close the **Configuration** pop-up window.

- **Login to the QtRPT desktop application**

Log in to the QtRPT desktop application using the Admin user credentials.

- **Update the report template**

To open the report template to be updated, perform the following actions:

- select the report template from the **Templates directory** (Service > Plugins > Templates directory).
- click the **open** icon from the **Action** column and close the **Templates directory** field.

In the opened report template, we can update the already existing aliases.

Note that the aliases for the check boxes should have the integer left and top placement values. Otherwise, the check boxes will be displayed incorrectly in the generated report. The left and top placement values are displayed on the panel to the left from the report template.

- **Save changes in the report template**

To save changes in the report template, perform the following steps:

- open the **Templates directory** (Service > Plugins > Templates directory).
- select the updated report template.
- click the **Save** icon.

To check the updated report template, click the **Preview** icon at the top left corner of the page. The report will be generated with the updated data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Report Template via QtRPT desktop application***.

19.2.1.5 Create Environment for Reports Generation

The Report Generation functionality can be added to any screen that we have created before, but to show you this functionality more widely and in detail, we will create a new screen. The settings on the Report screen will provide the possibility to demonstrate the different report generation use cases.

This screen will be used to demonstrate

Before we create the report templates, there are important pre-settings that should be done.

19.2.1.6 Set up Document Settings

Next, we need to update the Document Family and Document Groups parameters to be able to set up the report generation.

To set up the document family and document group for reports, we need to perform the steps below:

- **Document Family**

First, we add a new category to the document family.

For that, in the Architect application, go to Document Access > Document Family and update the **Documents** family.

Click the **Add category** button at the bottom left corner of the page. In the appeared row, define the following parameters:

- **Device ID** - enter the folder or the directory where the generated reports will be stored on your device.
- **Category** - enter a new category name. In this example, we add the **report** value.
- **Description** - enter the **report** value to the description field.
- **Retention Time (in days)** - to save space on your device, it is recommended to define the time range in days during which the generated reports will be stored in the folder or the directory defined in the **Device ID** field. Otherwise, the previously generated reports will be stored there until the folder is full. But such previously generated reports cannot be opened or downloaded from the Render application.

Submit changes on the page.

- **Document Groups**

Second, we add the early-created new category to the document family.

For that, in the Architect application, go to Document Access > Document Groups and update the following parameter for the **Documents_main_full** group:

- **Document Category** - select the **report** category from the list.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Document Family and Document Groups for Reports***.

19.2.1.7 Set up System Report

To use the reports with the System type in the Render application, we need to create it in the Architect application first.

For that, in the Architect application, go to Features > Report Builder and create a new report with the following parameters:

- **Library -** is always the same for our use case: **CodiacSDK.Universal**. This value can be changed for special scenarios.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Simple Search** - when the user clicks the **Generate Report** button on the screen in the Render application, a search will be initiated based on the configuration specified here. The search results will determine the content displayed in the pop-up window. The users will need to search for data on the screen in the Render application, and the report will be generated based on the selected data.

Activate the **Simple Search** and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, add the **Second Name** to the **Search Field Label** field and select **user_info_id** for the primary key and **second_name** for the search parameter.

- **Tenant** - enter the tenant that is connected to the report. In this example, we enter **1.**
- **Report Name** - enter the name for the report to be generated. Here, we add the **W9** value.
- **Description** - enter a short description for the report: **W9 form**.
- **Document Family** - select the **Documents - Documents** document family from the drop-down list.
- **Document Category** - select the **report - report** document category from the drop-down list
- **Template Type** - select the **System** template type that will be used for report generating.
    - **Template Type** - this additional field appears after selecting the **System** template type in the **Template Type** field above and contains the report templates list. Select the corresponding report template from the list. As an example, we select the **W9_web** report template created via the QtRPT web application.

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create System Report***.

19.2.1.8 Create Reports Menu for HR and Managers

The Reports menu will provide access to the screen in the Render application that we will create further.

In the Architect application, open Screen Configurations > Menu and create Menu for Reports.

Create Menu for HR with the following parameters:

- **Group Name** - select the **HR** group.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Reports**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. In our case **Reports**.
- **Weight** - enter **40**.

Submit changes on the page.

Create Menu for Managers with the following parameters:

- **Group Name** - select the **Managers** group.

- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Reports**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. In our case **Reports**.
- **Weight** - enter **40**.

Submit changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Menu for Reports***.

19.2.1.9 Create Screen: Report

This screen will be used to demonstrate the work of the Reports functionality.

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add New Screen** - enter the **Report** screen name.
- **Screen Description** - enter the **Report** screen description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the Simple Search and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for the search parameter.

- **Tab Text** - enter the **Report** tab text.
- **Tab Description** - enter the **Report** tab description.
- **Screen State** - select the **View** screen state.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **First Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**.

- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **First Name**.
- Select the following check boxes to adjust field layout:
    - **Read-only**

Add the **Second Name** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**.
- **Type** - select the **Text** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Second Name**.
- Select the following check boxes to adjust field layout:
    - **Read-only**

On the **Section** level, add a section of the **Edit type**.

Select the following check boxes to adjust section layout:

- **Hide section header**
- **Hide section pop-out**

Add the **Generate Report** button. In the Render application, clicking this button provides access to the manual report generation. I.e., we can select the user data to generate the report directly in the Generate report functionality. Set up the following parameters:

- **Value** - enter **Generate Report** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **W9 form** report template from the drop-down list.

Add the **Simple Report** button. In the Render application, we need to use the search functionality on the screen and click this button to automatically generate the report with search data. Set up the following parameters:

- **Value** - enter **Simple Report** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **W9 form** report template from the drop-down list.
- Select the following check boxes to adjust button layout:
    - **Passthru - No Input**

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Screen: Reports***.

19.2.1.10 Create Group Screen for Reports

Now we need to link the created Reports screen and HR and Managers groups and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters for the **HR** group:

- **Group Name** - select the **HR** group we have created previously.
- **Menu Name** - select the **Reports** menu we have created previously.
- **Screen Name** - select the **Report** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Report** value here.
- **Weight** - let us set it to **40**.

Save the changes on the page.

Set the specific parameters for the **Managers** group:

- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Reports** menu we have created previously.
- **Screen Name** - select the **Report** screen we have created previously.
- **Screen Text** - is a required field and we must fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Report** value here.
- **Weight** - let us set it to **40**.

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Group Screen for Reports***.

19.2.2 Check System Report Generation in Render

Let us show how the System report generation works in the Render application.

For that, open the Render application and use the HR user credentials to log in to the Render application.

If the above-mentioned environment for System report generation is set up correctly, we will see the Report menu on the menu board and have access to the Report screen. On the Report screen, we will be able to generate reports using two ways:

- **Generate Report**

This report generation type is set up to be used when you need to select user data manually to be added to the generated report. Click the **Generate Report** button to access the report generation functionality. To generate the report, perform the step below:

- **Select record** - select the user's Second Name from the drop-down list.
- **Generate report** - click this button to generate the report.
- **Download from API server -** click this button to download and open the generated report. The downloaded report opens in the separate tab.

Note that in the Generate report pop-up window, the report can be generated just once. To generate another report, the Generate report pop-up window should be closed and reopened.

- **Simple Report**

Use the search functionality on the screen to search for the corresponding user and click the **Simple Report** button. The report will be generated automatically with the searched user data. The generated report opens in the separate tab.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check System Report Generation in Render***.

19.3 Create PDF Form Report Type

Another type of report template is a PDF Form. It is a report template created using the online or desktop PDF applications. Users can then utilize the created report template in PDF format for generating reports.

19.3.1 Create Report Template with PDF Form via Online PDF Editor

In the example below, we describe the creation of a report template using one of the online PDF editors.

The report template with PDF Form type can be created using any online PDF editor. So, you can use any online PDF editor to work with a report template.

To create a new report template, perform the following steps:

- Open any PDF editor.
- Upload and open the PDF file to be used as a report template. As an example, we use the police insurance document in the PDF format. Such a document can contain information about the user's personal data (first and second names, email, phone, address, etc.).
- To connect the data in the report with the data from the database table, we need to add the aliases to the report template. As an example, we use the following aliases:
    - **Alias.user_info.first_name** - user's First Name. This alias can be added to the field or area that contains the user's First Name.
    - **Alias.user_info.second_name** - user's Second Name. This alias can be added to the field or area that contains the user's Second Name.

- **Custom.user_info.rule1_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 1 check box.
- **Custom.user_info.rule2_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 2 check box.
- **Custom.user_info.rule3_checkbox** - this check box will be marked in the generated report, if the user selected the Rule 3 check box.

To add aliases in the list above, we perform the following steps for every alias:

- Add the **Textbox** or **field** to the corresponding part of the report template. For example, the textbox or field that contains the alias with the user's First Name should be added to the report template field where the user's First Name will be displayed.
- Resize the textbox to make it fit the field in the report template.
- Add the alias name to the textbox using textbox properties > tooltip. Note that the alias name should be in the square brackets. Otherwise, it will be displayed as a regular text. Also, you can enter the other text in the Tooltip field in addition to the alias in the square brackets.

- Download the created report template. Further, this report template can be used to generate reports in the Render application.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template with PDF Form via Online PDF Editor***.

19.3.2 Create Report Template with PDF Form via Desktop PDF Editor

Now, we describe the report template creation using the desktop PDF editor.

You can use any PDF desktop application to create the report template in PDF format to be used for report generation. In our example, we will use one of them to demonstrate how the report template can be created.

To create a new report template, we need to perform the following actions:

- Upload the PDF file to the desktop PDF application. As an example, we use the police insurance document in the PDF format. A document like this may include personal information such as the user's First and Last name, email address, phone number, etc. that can be included in the generated report.
- To connect the data in the report with the data from the database table, we need to add the aliases to the report template. As an example, we use the following aliases:
  - **Alias.user_info.second_name** - user's Second Name. Add this alias to the user's Second Name field.
  - **Alias.user_info.email** - user's email address. Add this alias to the email address field.

To add aliases in the list above, perform the steps below for every alias:

- Using the **Text Field** > **Properties** option, fill in the **Tooltip** field with the alias name. Note that the alias name should be in the square brackets. Otherwise, it will be displayed as a regular text. Also, you can enter the other text in the Tooltip field in addition to the alias in the square brackets.
- Save the report template. Further, this report template can be used to generate reports in the Render application.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template with PDF Form via Desktop PDF Editor***.

19.3.3 Create PDF Form Report

Let us show you the creation of the report with a PDF Form type.

First, we create a new report for the report template created in the online PDF editor.

In the Architect application, go to Features > Report Builder and click the **Create** button.

Define the following parameters:

- **Library -** is always the same for our use case: **CodiacSDK.Universal**. This value can be changed for special scenarios.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Simple Search** - when the user clicks the button to generate a report on the screen in the Render application, a search will be initiated based on the configuration specified here. The search results will determine the content displayed in the pop-up window. The users will need to search for data on the screen in the Render application, and the report will be generated based on the selected data.

Activate the **Simple Search** and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, add the **Second Name** to the **Search Field Label** field and select **user_info_id** for the primary key and **second_name** for the search parameter.

- **Tenant** - enter the tenant that is connected to the report. In this example, we enter **1.**
- **Report Name** - enter the name for the report to be generated. Here, we add the **W9_pdf_form_1** value.
- **Description** - enter a short description for the report: **W9_pdf_form_1**.
- **Document Family** - select the **Documents - Documents** document family from the drop-down list.
- **Document Category** - select the **report - report** document category from the drop-down list
- **Template Type** - select the **PDF Form** template type that will be used for report generating. In the appeared **Select Form** section fill in the following parameters:
    - **Upload PDF Form** - select this parameter to activate the Upload PDF Form function.

- **File select** - click this button to upload the report template in PDF format.

Save the changes on the page.

For the report template created in the desktop PDF editor, we create a new report.

In the Architect application, go to Features > Report Builder and click the **Create** button.

Define the following parameters:

- **Library -** is always the same for our use case: **CodiacSDK.Universal**. This value can be changed for special scenarios.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Simple Search** - when the user clicks the button to generate a report on the screen in the Render application, a search will be initiated based on the configuration specified here. The search results will determine the content displayed in the pop-up window. The users will need to search for data on the screen in the Render application, and the report will be generated based on the selected data.

Activate the **Simple Search** and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, add the **Second Name** to the **Search Field Label** field and select **user_info_id** for the primary key and **second_name** for the search parameter.

- **Tenant** - enter the tenant that is connected to the report. In this example, we enter **1.**
- **Report Name** - enter the name for the report to be generated. Here, we add the **W9_pdf_form_2** value.
- **Description** - enter a short description for the report: **W9_pdf_form_2**.
- **Document Family** - select the **Documents - Documents** document family from the drop-down list.
- **Document Category** - select the **report - report** document category from the drop-down list
- **Template Type** - select the **PDF Form** template type that will be used for report generating. The appeared the **Select Form** section fill in the following parameters:
    - **Available Form** - select this parameter to activate the Available Form function.
    - Select the report template in PDF format from the drop-down list. All templates from the drop-down list that are uploaded by users and placed in the **AppMill\Service\ReportTemplatePDFFormTemplateFolder** folder.

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create PDR Form Report***.

19.3.4 Update Report Screen

To show how the above-created report templates in PDF format can be used in the Render application, we need to update the Report screen.

In the Architect application, open the Screen Configurations > Screen Builder and select the Report screen.

On the **second** page of the screen builder, we need to update the **Section** of the screen. Specifically, we add the possibility to use report templates created in PDF editors to generate a report in the Render application.

In the Section level, we need to add new labels and buttons. For that, we will perform the following operations:

Add the **System Reports** label to clarify which buttons on the screen relate to the generation of the reports with System type. Set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this label, in our case **System Reports**.
- **Type** - select the **Label** type.

Add the **PDF Form Reports** label to clarify which buttons on the screen relate to the generation of the reports with PDF Form type. Set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this label, in our case **PDF Form Reports**.
- **Type** - select the **Label** type.

Add the **Report type 1** button. In the Render application, clicking this button provides access to the manual report generation. I.e., we can select the user data to generate the report directly in the Generate report functionality. Here, we use the report template created via an online PDF editor. Set up the following parameters:

- **Value** - enter **Generate Report** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **W9_pdf_form_1** report template from the drop-down list.

Add the **Report type 2** button. In the Render application, clicking this button provides access to the manual report generation. I.e., we can select the user data to generate the report directly in the Generate report functionality. Here, we use the report template created via a desktop PDF editor. Set up the following parameters:

- **Value** - enter **Generate Report** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **W9_pdf_form_2** report template from the drop-down list.

Save changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Report Screen***.

19.3.5 Check PDF Form in Render

Let us show how the PDF Form report generation works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned environment for PDF Form report generation is set up correctly, we will see new labels and buttons on the Report screen. Here, we will be able to generate reports in PDF format using two reports templates:

- **Report type 1** - clicking this button will generate a report based on the report template created in the online PDF editor.
- **Report type 2** - clicking this button will generate a report based on the report template created in the desktop PDF editor.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check PDF Form Report Generation in Render***.

19.4 Create Editable Report

In the sections below, we describe another way to create report and report templates. The Editable report type provides the possibility to create or edit report templates directly in the Architect application using the Report Builder functionality.

We will demonstrate the creation of two report templates types:

- Form Letter
- Reports

19.4.1 Create Form Letter Template

The Editable Letter report template type contains the letter title and body. The letter body can include aliases and text in free form. We use this report template type to generate reports containing the user profile data such as first and second names, email address, and phone number.

To create a new report with Editable Letter type, in the Architect application, open Features > Report Builder and click the **Create** button.

Define the following parameters:

- **Library -** is always the same for our use case: **CodiacSDK.Universal**. This value can be changed for special scenarios.
- **Primary Table** - select the **user_info** table from the drop-down list.

- **Simple Search** - when the user clicks the button to generate a report on the screen in the Render application, a search will be initiated based on the configuration specified here. The search results will determine the content displayed in the pop-up window. The users will need to search for data on the screen in the Render application, and the report will be generated based on the selected data.

Activate the **Simple Search** and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, add the **Second Name** to the **Search Field Label** field and select **user_info_id** for the primary key and **second_name** for the search parameter.

- **Tenant** - enter the tenant that is connected to the report. In this example, we enter **1.**
- **Report Name** - enter the name for the report to be generated. Here, we add the **Edditable_Letter** value.
- **Description** - enter a short description for the report: **letter**.
- **Document Family** - select the **Documents - Documents** document family from the drop-down list.
- **Document Category** - select the **report - report** document category from the drop-down list
- **Template Type** - select the **Editable** template type that will be used for report generating. In the appeared **Select Template** section fill in the following parameters:
    - **Document Type** - select the **From Letter** document type from the drop-down list.
    - **Available Template** - select the **1 Page Freeform.xml** template from the drop-down list. Such a report template contains 1 page with the free filling form.

All templates from the drop-down list are uploaded by the system during the AppMill Service installation process and placed in the **AppMill\Service\ReportTemplate\FormTemplateFolder** folder.

According to the selected template, the section below will contain different parameters. In the example we are describing, the Enter Report Details section appears with the following parameters:

- **Image Title** - select and upload an image from your local device. By default, the image uploaded to the first Image Title area will be displayed at the top left corner of the report.
- **Image Title** - select and upload an image from your local device. By default, the image uploaded to the second Image Title area will be displayed at the top right corner of the report.
- **Free Form Section** - enter a letter body containing text and aliases. To connect the report template to the data from the database table, we add the aliases to the letter body. In this example, we add the following aliases in the square brackets:
    - **Alias.user_info.first_name -** user's First Name.
    - **Alias.user_info.second_name -** user's Second Name.
    - **Alias.user_info.email -** user's email address.

Alias.user_info.phone - user's phone number. Note that the aliases should be in the square brackets. Otherwise, it will be displayed as a regular text.

Save the changes on the page.

Click the **Generate Preview** button at the bottom left corner of the page to generate and preview the report template. The report template opens in a new tab.

The created report template can be used to generate reports in the Render application.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template with Editable Letter Type***.

19.4.2 Create Reports Template

According to the report templates with an Editable Report type created here, we can generate reports containing details about equipment used in the Onboarding process. The report template will include the equipment details sorted and displayed in the table view in the generated report.

To create a new report with Editable Report type, in the Architect application, open Features > Report Builder and click the **Create** button.

Define the following parameters:

- **Library -** and for our use case: **CodiacSDK.Universal**. This value can be changed for special scenarios.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Simple Search** - when the user clicks the button to generate a report on the screen in the Render application, a search will be initiated based on the configuration specified here. The search results will determine the content displayed in the pop-up window. The users will need to search for data on the screen in the Render application, and the report will be generated based on the selected data.

Activate the **Simple Search** and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, add the **Second Name** to the **Search Field Label** field and select **user_info_id** for the primary key and **second_name** for the search parameter.

- **Tenant** - enter the tenant that is connected to the report. In this example, we enter **1.**
- **Report Name** - enter the name for the report to be generated. Here, we add the **Edditable_Report** value.
- **Description** - enter a short description for the report: **report**.
- **Document Family** - select the **Documents - Documents** document family from the drop-down list.
- **Document Category** - select the **report - report** document category from the drop-down list

- **Template Type** - select the **Editable** template type that will be used for report generating. In the appeared **Select Template** section fill in the following parameters:
    - **Document Type** - select the **Reports** document type from the drop-down list.
    - **Available Template** - select the **StandardTemplate3NoGroupsOneTotals.xml** template from the drop-down list.

All templates from the drop-down list are uploaded by the system during the AppMill Service installation process and placed in the **AppMill\Service\ReportTemplate\ReportTemplateFolder** folder.

According to the selected template, the section below will contain different parameters. In the example we are describing, the **Enter Report Details** section appears with the following parameters:

- **Report Title** - enter the title of the report to be displayed at the top of the generated report page. By default, the Report Title field contains the value according to the selected template in the Available Template field. Here, we delete the default value and add the **Equipment list for [Alias.user_info.first_name] [Alias.user_info.second_name]** title that contains the aliases to connect the report template to the database table.
- **Image Title** - select and upload an image from your local device.
- **Image Width** - define the image width.
- **Image Height** - define the image height.

- In the **Enter Column Details** section we can define the following column parameters:
    - **Data Source** - enter the parent table identifier alias: **Alias.user_info.user_info_id**. By default, the field is predefined with DS1 value. The value entered in the Data Source field defines the alias preview style. Note that if the parent table identifier alias is used as a Data Source in the report template, the data added in the **Preview and inline data** section for the preview will be ignored.
    - **Name** - enter column names. As we use this report template to display the equipment details, the following names will be defined for the columns:
        - Column 1 - enter the **Type** value for the equipment type column.
        - Column 2 - enter the **Brand** value for the equipment brand column.
        - Column 3 - enter the **Price** value for the equipment price column.
    - **Head Bold** - select this parameter for all columns to put the column headers in bold.
    - **Data Format** - select the data format for the columns:
        - Column 1 - select the **String** from the drop-down list for the **Type** column.
        - Column 2 - select the **String** from the drop-down list for the **Brand** column.
        - Column 3 - select the **Decimal** from the drop-down list for the **Price** column.

- **Alias Name** - enter the alias name in the square brackets for every column to get the data from the database table to be displayed in the generated report.
  - Column 1 - enter the **[Array.equipment.type]** alias for the **Type** column.
  - Column 2 - enter the **[Array.equipment.brand]** alias for the **Brand** column.
  - Column 3 - enter the **[Array.equipment.price]** alias for the **Price** column.

- **Preview and inline data** - click the link to preview data and check the entered aliases to the report template.
- **Generate Preview** - click the button at the bottom right corner of the page to generate and open the report template preview in the separate tab.

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template with Editable Report Type***.

19.4.3 Create Report Template with Editable Report (DS1) Type

Based on the report templates created with the Editable Report (DS1) type, we can generate reports that provide detailed information about the equipment used during the Onboarding process. These reports will feature the equipment details organized and displayed in a table format.

The main differences of the report template with the Editable Report (DS1) type from the report template with the Editable Report type is:

Differences	**Editable Report**	**Editable Report (DS1)**
Data in generated reports	The generated report will contain the data defined in the report template.	In the Architect application, make sure to select the Override Inline option (Features > Report Builder) to ensure that the correct data from the database appears in the report generated by the Render application.
Generate Preview button in the Report Builder	Clicking the Generate Preview button in the Report Builder displays the created report template containing the empty form without any test data.	Clicking the Generate Preview button in the Report Builder displays the created report template containing the empty form with test data added in the Preview and inline data section.
Alias name	The alias name consists of 3 parts: Array.table_name.field_name.	The alias name consists of 4 parts: DS1.Array.table_name.field_name.

To create a new report with Editable Report type, in the Architect application, open Features > Report Builder and click the **Create** button.

Define the following parameters:

- **Library -** and for our use case: CodiacSDK.Universal. This value can be changed for special scenarios.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Simple Search** - when the user clicks the button to generate a report on the screen in the Render application, a search will be initiated based on the configuration specified here. The search results will determine the content displayed in the pop-up window. The users will need to search for data on the screen in the Render application, and the report will be generated based on the selected data.

Activate the **Simple Search** and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, add the **Second Name** to the **Search Field Label** field and select **user_info_id** for the primary key and **second_name** for the search parameter.

- **Tenant** - enter the tenant that is connected to the report. In this example, we enter **1.**
- **Report Name** - enter the name for the report to be generated. Here, we add the **Edditable_Report_DS1** value.
- **Description** - enter a short description for the report: **Report_DS1**.
- **Document Family** - select the **Documents - Documents** document family from the drop-down list.
- **Document Category** - select the **report - report** document category from the drop-down list
- **Template Type** - select the **Editable** template type that will be used for report generating. In the appeared **Select Template** section fill in the following parameters:
    - **Document Type** - select the **Reports** document type from the drop-down list.
    - **Available Template** - select the **StandardTemplate3NoGroupsOneTotals.xml** template from the drop-down list.

All templates from the drop-down list are uploaded by the system during the AppMill Service installation process and placed in the **AppMill\Service\ReportTemplate\ReportTemplateFolder** folder.

According to the selected template, the section below will contain different parameters. In the example we are describing, the **Enter Report Details** section appears with the following parameters:

- **Report Title** - enter the title of the report to be displayed at the top of the generated report page. By default, the Report Title field contains the value according to the selected template in the Available Template field. Here, we delete the default value and add the **Equipment list for [Alias.user_info.first_name] [Alias.user_info.second_name]** title that contains the aliases to connect the report template to the database table.
- **Underline** - select the check box. In the generated report, the report title will be underlined.

- **Image Title** - select and upload an image from your local device.
- **Image Width** - define the image width.
- **Image Height** - define the image height.

- In the **Enter Column Details** section we can define the following column parameters:
  - **Data Source** - by default, the field is predefined with **DS1** value. The value entered in the Data Source field defines the alias preview style.
  - **Override Inline** - select the check box. If this check box is selected, the report generated in the Render application will contain the correct data from the database. Otherwise, the report will display the data added in the **Preview and inline data** section as an example for the preview.
  - **Name** - enter column names. As we use this report template to display the equipment details, the following names will be defined for the columns:
    - Column 1 - enter the **Type** value for the equipment type column.
    - Column 2 - enter the **Brand** value for the equipment brand column.
    - Column 3 - enter the **Price** value for the equipment price column.
  - **Head Bold** - select this parameter for all columns to put the column headers in bold.
  - **Data Format** - select the data format for the columns:
    - Column 1 - select the **String** from the drop-down list for the **Type** column.
    - Column 2 - select the **String** from the drop-down list for the **Brand** column.
    - Column 3 - select the **Decimal** from the drop-down list for the **Price** column.
  - **Alias Name** - enter the alias name in the square brackets for every column to get the data from the database table to be displayed in the generated report. If the DS1 value is entered in the Data Source field, the alias names will contain the DS1 value as one of the alias name parts:
    - Column 1 - enter the **[DS1.Array.equipment.type]** alias for the **Type** column.
    - Column 2 - enter the **[DS1.Array.equipment.brand]** alias for the **Brand** column.
    - Column 3 - enter the **[DS1.Array.equipment.price]** alias for the **Price** column.
- **Preview and inline data** - click the link to preview data and check the entered aliases to the report template.
- **Add Row** - click the button at the bottom left corner of the page to add a new table row with the data to be displayed in the generated report preview in the separate tab.
- **Generate Preview** - click the button at the bottom right corner of the page to generate and open the report template preview in the separate tab.

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Report Template with Editable Report (DS1) Type***.

19.4.4 Update Report Screen

To demonstrate how to use the above-created report templates with editable letter and report types in the Render application, we need to update the Report screen.

In the Architect application, open the Screen Configurations > Screen Builder and select the Report screen.

On the **second** page of the screen builder, we need to update the **Section** of the screen. Specifically, we add the possibility to use report templates with Editable Letter and Editable Report types to generate a report in the Render application.

In the Section level, we need to add new labels and buttons. For that, we will perform the following operations:

Add the **Editable Reports** label to clarify which buttons on the screen relate to the generation of the reports with Editable Letter and Editable Report types. Set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this label, in our case **Editable Reports**.
- **Type** - select the **Label** type.

Add the **Simple Letter** button to generate reports with the Editable Letter type. Set up the following parameters:

- **Value** - enter **Simple Letter** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **letter** option from the drop-down list.
- Select the following check boxes to adjust button layout:
    - **Passthru - No Input**

Add the **Simple Report** button to generate reports with the Editable Report type. Set up the following parameters:

- **Value** - enter **Simple Report** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **report** option from the drop-down list.
- Select the following check boxes to adjust button layout:
    - **Passthru - No Input**

Add the **Report_DS1** button to generate reports with the Editable Report (DS1) type. Set up the following parameters:

- **Value** - enter **Report_DS1** as it will be used for generating the report.
- **Button Action** - select the **Generate report** action.
- **Report Template** - select the **Report_DS1** option from the drop-down list.
- Select the following check boxes to adjust button layout:

Save the changes on the page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Update Report Screen***.

19.4.5 Check Editable Report Generation in Render

Let us show how the Editable report generation works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned environment for Editable report generation is set up correctly, we will see new labels and buttons on the Report screen. Here, we will be able to generate reports in PDF format using two reports templates. Perform the following steps to generate reports with the Editable type:

- **Search for the corresponding user**

For example, we need to generate the report with the data of the user named John Smith, then we enter the John Smith name in the search field.

- **Simple Letter** - click this button to generate a report with the Editable Letter report template type.

or

- **Simple Report** - click this button to generate a report with the Editable Report report template type.
- **Report_DS1** - click this button to generate a report with the Editable Report (DS1) report template type.

This report generation type is set up to be used when you need to select user data manually to be added to the generated report. Click the **Report_DS1** button to access the report generation functionality. To generate the report, perform the step below:

- **Select record** - select the user's Second Name from the drop-down list.
- **Generate report** - click this button to generate the report.
- **Download from API server** - click this button to download and open the generated report. The downloaded report opens in the separate tab.

Note that in the Generate report pop-up window, the report can be generated just once. To generate another report, the Generate report pop-up window should be closed and reopened.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Check Editable Report Generation in Render***.

20 Create Screen with all Field Types

As you know from the instructions presented previously, the AppMill solution provides a wide range of possibilities to build and manage screens, tabs, and different content there. We demonstrated how quickly and easy you can configure all possible operations and actions and set up data that build the business logic and processes.

Now we would like to summarize and demonstrate how we set up different field types. For clarity and ease of information perception, we will gather all possible types of fields on one screen:

- text
- text area
- numeric
- list
- multi-select list
- check boxes
- radio buttons
- document uploads
- inline searches
- label
- button
- datalist
- relational datalist
- dependent fields
- linked list
- tooltips

20.1 Presetting

Before we proceed directly to the creation of the fields, we need to do following preparations:

1. Create a new menu to access the required screen
2. Create a new screen to demonstrate the field types
3. Create a new group screen to provide an access for users
4. Enhance the database table (user_info) with new necessary data
5. Create a new custom query to use it in the data demonstration

After these preparations, we will proceed directly to the creation and configuration of the fields (including required buttons, images, and labels).

20.1.1 Create Demo Menu

In the Architect application, open Screen Configurations > Menu and create Menu.

Create **Demo** Menu with the following parameters:

- **Group Name** - select the **Manager** group.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Demo**.
- **Menu Icon** - select the menu icon from the drop-down list.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. In our case **Demo**.
- **Weight** - enter **0**.

Submit changes on the page.

Create **Execute** Menu with the following parameters:

- **Group Name** - select the **Manager** group.
- **Menu Text** - enter a menu name. It will be displayed on the UI in the Render. In our case **Execute**.
- **Description** - enter a menu description. It will be displayed when hovering over the menu name on the UI in the Render. In our case **Execute**.
- **Weight** - enter **40**.

Submit changes on the page.

20.1.2 Create Screen: Field Types

In the Architect application, open Screen Configurations > Screen Builder and click the **Create with builder** button.

On the opened page of the screen builder, specify the following data:

- **Add New Screen** - add a unique screen name. In our case **Field Types**.
- **Screen Description** - add a short description. In our case **Field Types**.
- **Primary Table** - select the primary table from the drop-down list. In our case **user_info**.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the simple search and select the **Search_user_info: Universal DLL : Search_user_info** search option.

In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.

- **Tab Text** - enter a name. It will be displayed in the menu in the Render. In our case **Field Types**.
- **Tab Description** - add a clear description for the screen. In our case **Field Types**.
- **Screen State** - select the **View** screen state from the drop-down list.
- **Screen Layout** - select **Header 2x3**.

On the second page of the screen builder, we need to configure the **Header** and **Section** of the screen.

For all sections (header and sections) on the screen define the Edit type: click the gear icon of each section and select the **Edit Type from** the **Section Type** drop-down list.

Submit changes on the page by clicking the Create button.

20.1.3 Create Group Screen for Field Types Screen

To assign the Fields Type screen to a particular menu item, define the available actions and set up the visibility of the buttons on the screen in the Render, we need to use the Group Screen functionality.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Create a new Group screen for the **Demo** menu. Click the Create button and set the following parameters on the creation screen:

- **Available actions** - define available actions on the screen by selecting the following check boxes:
  - **Add**
  - **Delete**
  - **Edit**
  - **Copy**
  - **Execute**
- **Visibility of the buttons** - define the visibility of the buttons on the screen by selecting the following check boxes:
  - **Show Add Button**
  - **Show Delete Button**
  - **Show Edit Button**
  - **Show Copy Button**
  - **Show Execute Button**
- **Group Name** - select the **Managers** group we have created previously.

- **Menu Name** - select the **Demo** menu we have created previously.
- **Screen Name** - select the **Field Types** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Field Types** value here.
- **Weight** - let us set it to **0**.

Submit changes on the page.

Create a new Group screen for the **Execute** menu. Click the Create button and set the following parameters on the creation screen:

- **Available actions** - define available actions on the screen by selecting the following check boxes:
  - **Edit**
  - **Execute**
- **Visibility of the buttons** - define the visibility of the buttons on the screen by selecting the following check boxes:
  - **Show Edit Button**
  - **Show Execute Button**
- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Execute** menu we have created previously.
- **Screen Name** - select the **Execute** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter the **Execute** value here.
- **Weight** - let us set it to **35**.

Submit changes on the page.

20.1.4 Update User Info Table

To demonstrate more demo data on the Field Types screen, we need to add more fields to the **user_info** table.

In the Architect application, open Features > Tables and select the **user_info** table.

Add new columns with the following parameters to the current table:

Name	Type	Nullable	Scale	Length
currency_dynamic	Numeric	True	2	9

currency_static	Numeric	True	2	9
decimal_dynamic	Numeric	True	3	7
decimal_static	Numeric	True	3	7
datalist	Varchar	True	-	-
relation_datalist	Varchar	True	-	-
relation_datalist_multi	Text (Postgres, MySQL, SQLite) or Varchar (for Oracle SQL)	True	-	-
create_on	Timestamp	True	-	-
due_date	Timestamp	True	-	-
priority	Varchar	True	-	-

Submit changes on the page.

20.1.5 Create Custom Query

To further use the Relation Datalist field type, we need to prepare the custom query that will find all users for the field value.

When creating or updating a custom query, please note that the syntax of the query may vary depending on the database. For example:

- For Oracle, field and table names must be in double quotes:
  select “id” as “value”, “user_name” as “description”, “group_area” from “c_user_accounts” where “group_area” like ‘%Managers%’ or “group_area” like ‘%Employees%’
- SQL query must not contain the line breaks, otherwise it may lead to incorrect results when executing the query.
- SQL queries are sensitive to the type of quotes used. Usually, double quotes are used to protect field and table names from being converted to the default format, while single quotes are used to denote string values.

In the Architect application, open Features > Custom Query and create a new custom query with the following parameters:

- **Query Name** - enter the query name. In our example, we enter **Get-All-Users-for-multi**.

- **Query Value** - enter the following query value:

select user_name from c_user_accounts

- **Query PKS** - enter the **id** primary key.
- **Description** - enter the following description: **Get-All-Users-for-multi**.

Save the changes on the page.

20.1.6 Check in Render

Let us show how the configured screen works in the Render application.

For that, log in to the Render application using the Manager user credentials.

If the above-mentioned screen is configured correctly for Manager users (Jane Hanson and John Smith), we will see the Field Types menu and the Field Types screen displayed as a tab. On the Field Types screen, we will be able to add, edit, delete, copy, and execute data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Environment for Field Types Screen***.

20.2 Configure Header Level of Field Types Screen

First, we will configure the fields on the Header level, afterwards on the Sections level.

On the second page on the Screen Builder for the Field Types, we will configure the data of different types. To make the demonstration easier for understanding we will use clear naming to mirror the type and sense of the data or field. We will follow the order when placing the data on the screen and describe the configuration of these fields accordingly.

In the **Header** section of the screen, we add the following fields:

Insert Action Image

Add the **Insert Image**. On the Field Types screen in the Render application, clicking on the Insert Image, users can perform the insert action. Set up the following parameters for the image:

- **Value** - enter the image name. We enter the **Insert Action Image** name.
- **Static Image Action** - select the **Insert** action from the drop-down list.
- Upload file to be used as the insert button.
- Select the following check box:
    - **Full width for auxiliary buttons**

Edit Action Image

Add the **Edit Image**. On the Field Types screen in the Render application, clicking on the Edit Image, users can perform the edit action. Set up the following parameters for the image:

- **Value -** enter the image name. We enter the **Edit Action Image** name.
- **Static Image Action** - select the **Edit** action from the drop-down list.
- Upload file to be used as the edit button.
- Select the following check box:
    - **Full width for auxiliary buttons**
    - **Use auxiliary buttons with text**

Insert Action Button

Add the **Insert** button. On the Field Types screen in the Render application, clicking on the Insert button, users can perform the insert action. Set up the following parameters for the button:

- **Value -** enter the button name. We enter the **Insert Action Button** name.
- **Button Action -** select the **Insert** action from the drop-down list.
- Select the following check box:
    - **Full width for auxiliary buttons**

Edit Action Button

Add the **Edit** button. On the Field Types screen in the Render application, clicking on the Edit button, users can perform the edit action. Set up the following parameters for the button:

- **Value -** enter the button name. We enter the **Edit Action Button** name.
- **Button Action -** select the **Edit** action from the drop-down list.
- Select the following check box:
    - **Full width for auxiliary buttons**
    - **Use auxiliary buttons with text**

Label for Button and Image Action

Add the **Button and Image Action** label to be displayed in the Render application as a description for the buttons and image actions. Set up the following parameters for the label:

- **Label -** enter the label name. We enter the **Button and Image Action label** name.
- **Formatting**

We can demonstrate how to format the **Button and Image Action** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Color** - select the text color from the drop-down list.

- **Font Family** - select the font family from the drop-down list. In our example, we select **Comic Sans MS**.
- **Text Align** - select the **Center** option from the drop-down list.

Labels for Or

Add the **or** label to be placed between the **Insert Image** and the **Insert Action Button**. For that, set up the following parameters:

- **Label -** enter the label name. We enter the **or** name.
- **Formatting**

Format the **or** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Align** - select the **Center** option from the drop-down list.

We need to add the second **or** label that will be placed between the **Edit Image** and the **Edit Action Button**. For that, set up the following parameters:

- **Label -** enter the label name. We enter the **or** name.
- **Formatting**

Format the **or** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Align** - select the **Center** option from the drop-down list.

Field of Relation Datalist type

Add the **Relation Datalist** field that will provide the possibility in the Render application to select the user from the list and add new information to his/her profile details. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the user ID will be available to be selected from the users list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Relation Datalist**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Relation Datalist**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only - Except insert**
    - **Required**

- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

Field of Numeric type with a Link

Add the **Numeric Link** field to be used as a link to another screen in the Render application. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Numeric** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Numeric Link**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Numeric Link**.
- Select the following check box to adjust field layout:
    - **Link Type** - to assign a link to the field. The additional parameters appear:
        - **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Modal** option from the drop-down list, to open a linked page in a modal (pop-up) window.
        - **Refresh base page when exit** - select the check box.
        - **Modal Size - 80%.**
        - **Menu** - select the menu that the required screen assigned to from the drop-down list. We select the **Demo - Demo** option from the drop-down list.
        - **Group Screen** - select the group screen that the required screen assigned to from the drop-down list. We select the **Field Types** option from the drop-down list.
        - **Screen** - select the required screen that will be opened by link from the drop-down list. We select the **FieldTypes - FieldTypes** option from the drop-down list.
        - **Parameter** - specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.
        - **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.user_info.user_info_id** option from the drop-down list.

Field of Text type with input mask

Add the **NOFIELD input Mask** field. This field type is used to hide the entered value of the field on UI in the Render application. Set up the following parameters for the field:

- **Data field** - select the **NOFIELD.password**.
- **Type** - select the **Text** type.

- **Label** - enter the text that will be displayed on UI for this field, in our case **NOFIELD input Mask**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **This field value will be sent to the service but is not in the database. The value of this field can be processed inside the plugin**.
- Select the following check box to adjust field layout:
    - **Always show field border**
    - **Apply input mask**
    - **Nofield**

Clarification Label

Add the **Example of setting up extension functions on a separate screen:** label that will be displayed in the Render application as a name for the button and image actions with extension function. Set up the following parameters for the label:

- **Label -** enter the label name. We enter the **Example of setting up extension functions on a separate screen:** name.
- **Formatting**

Format the **Example of setting up extension functions on a separate screen:** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Color** - select the text color from the drop-down list.
- **Text Align** - select the **Left** option from the drop-down list.

Link Action Button

Add the **Button Link (Go to Main Extension Function Screen)** button. On the Field Types screen in the Render application, clicking on the **Button Link (Go to Main Extension Function Screen)** button, users can open the linked screen. Set up the following parameters for the button:

- **Value -** enter the button name. We enter the **Button Link (Go to Main Extension Function)** name.
- **Button Action** - select the **Link** action from the drop-down list. The additional parameters appear:
    - **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Blank** option from the drop-down list, to open a linked page in a new tab.
    - **Menu** - select the menu that the required screen assigned to from the drop-down list. We select the **Execute - Execute** option from the drop-down list.
    - **Group Screen** - select the group screen that the required screen assigned to from the drop-down list. We select the **Execute** option from the drop-down list.

- **Screen** - select the required screen that will be opened by link from the drop-down list. We select the **Execute - Execute** option from the drop-down list.
- **Parameter** - specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.
- **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.user_info.user_info_id** option from the drop-down list.

Link of Image type

Add the **Link Image**. On the Field Types screen in the Render application, clicking on the Link Image, users can open the linked Execute screen. Set up the following parameters for the image:

- **Value** - enter the image name. We enter the **Link Image** name.
- **Static Image Action** - select the **Link** action from the drop-down list. The additional parameters appear:
  - **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Blank** option from the drop-down list, to open a linked page in a new tab.
  - **Menu** - select the menu that the required screen assigned to from the drop-down list. We select the **Execute - Execute** option from the drop-down list.
  - **Group Screen** - select the group screen that the required screen assigned to from the drop-down list. We select the **Execute** option from the drop-down list.
  - **Screen** - select the required screen that will be opened by link from the drop-down list. We select the **Execute - Execute** option from the drop-down list.
  - **Parameter** - specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.
  - **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.user_info.user_info_id** option from the drop-down list.
- Upload file to be used as the image.

Clarification Link

Add the **Example of setting up reports on a separate screen:** label to be displayed in the Render application as a name for the button and image actions with extension function. Set up the following parameters for the label:

- **Label** - enter the label name. We enter the **Example of setting up reports on a separate screen:** name.

Format the **Example of setting up reports on a separate screen:** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Color** - select the text color from the drop-down list.
- **Text Align** - select the **Left** option from the drop-down list.

Button of Link type

Add the **Button Link (Go to Main Reports Screen)** button. On the Field Types screen in the Render application, clicking on the **Button Link (Go to Main Reports Screen)** button, users can open the linked screen. Set up the following parameters for the button:

- **Value -** enter the button name. We enter the **Button Link (Go to Main Reports Screen)** name.
- **Button Action -** select the **Link** action from the drop-down list. The additional parameters appear:
    - **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Blank** option from the drop-down list, to open a linked page in a new tab.
    - **Menu -** select the menu that the required screen assigned to from the drop-down list. We select the **Reports - Reports** option from the drop-down list.
    - **Group Screen -** select the group screen that the required screen assigned to from the drop-down list. We select the **Reports** option from the drop-down list.
    - **Screen -** select the required screen that will be opened by link from the drop-down list. We select the **Reports - Reports** option from the drop-down list.
    - **Parameter -** specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.
    - **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.user_info.user_info_id** option from the drop-down list.

Image of Link type

Add the **Link Image**. On the Field Types screen in the Render application, clicking on the Link Image, users can open the linked Reports screen. Set up the following parameters for the image:

- **Value -** enter the image name. We enter the **Link Image** name.
- **Static Image Action** - select the **Link** action from the drop-down list. The additional parameters appear:
    - **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Blank** option from the drop-down list, to open a linked page in a new tab.
    - **Menu** - select the menu that the required screen assigned to from the drop-down list. We select the **Reports - Reports** option from the drop-down list.
    - **Group Screen** - select the group screen that the required screen assigned to from the drop-down list. We select the **Reports** option from the drop-down list.

- **Screen** - select the required screen that will be opened by link from the drop-down list. We select the **Reports - Reports** option from the drop-down list.
- **Parameter** - specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.
- **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.user_info.user_info_id** option from the drop-down list.

- Upload file to be used as the insert image.

Field of List type

Add the **List to Switch Sections** field. On the Field Types screen in the Render application, changing the value in this field will switch tabs in the section. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.employment_type**.
- **Type** - select the **List** type.
- **Pull up** - click the **Pull up** button and select the **EmploymentType** list from the List Params pop-up window.
- **Label** - enter the text that will be displayed on UI for this field, in our case **List to Switch Sections**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **List to Switch Sections**.
- Select the following check box to adjust field layout:
  - **Section Refresh** - to switch the section display in the Render application when selecting the section name from the list. The additional **Section to Refresh** field appears:
    - **Section to Refresh** - select the section from the drop-down list to be refreshed in the Render application. We select the Section 1 option from the drop-down list.
  - **Always show the field border**

Label of Link type

Add the **Link Label (Link to the Third Party Resource)** label. On the Field types screen in the Render application, clicking on the link, users will open the linked screen. Set up the following parameters for the label:

- **Label** - enter the label name. We enter the **Link Label (Link to the Third-Party Resource)** name.
- **Link Type** - select the check box. The additional parameters appear:
  - **URL** - enter the URL address. We enter the **https://appmill.com** value.

- **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Blank** option from the drop-down list, to open a linked page in a new tab.

Submit the changes on the page.

20.2.1 Check Field Types Screen Header in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned screen is set up correctly for Manager users (Jane Hanson and John Smith), we will see the configured Header section displayed on the Field Types screen. In the Header section of the Field Types screen, we will be able to add, edit, delete, copy, execute data, and open screens by clicking the linked buttons, images, and fields.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen Header and Check in Render***.

20.3 Configure Sections Level of the Field Types Screen

Thereafter, we will configure the data on the Sections level.

20.3.1 Set up Sections' Switch

On the Header level we configured the **List to Switch Sections** field so that the sections on the section level will be switched based on the selected **Employment Type** value in this field. There are 3 values of the Employment Type:

- **Full-time**
- **Part-time**
- **Contract**

According to these values we need to configure 3 sections (3 tabs of the first section) that can be switched by the user in the Render.

20.3.1.1 Configure tab for Full-time Employment Type

Click the **Gear (Settings)** icon of the first section and set up the following parameters:

- **Layout Label** - enter the section name. Here, we add the first tab name: **Switched Section - Full-time**.
- **Linked Field** - select the linked field alias from the drop-down list. In our case, we select the **List to Switch Sections** linked field alias from the drop-down list: **Alias.user_info.employment_type**.
- **Active Value** - enter the value that will be linked with the section. Here, we add the **full-time** value.

- **Section Type** - select the section type from the drop-down list. For this section, we select the **Edit type** from the drop-down list.

Save the changes and close the pop-up window.

After configuring all required tabs, we will need to return to this first tab and set up the **Linked Field** value once more to apply this value to all assigned tabs.

20.3.1.2 Configure Switched Section - Part-time tab

Click the **Plus (Add Tab)** icon next to Default tab. A new tab named **Link 1** will appear next to the first Default tab. Click the Gear icon and set up the following parameters the newly added tab:

- **Layout Label** - enter the section name. Here, we add the first tab name: **Switched Section - Part-time**.
- **Hide section pop-out** - select the check box.
- ***Linked Field*** *- the linked field alias will be predefined according to the selected option in the* ***first tab****. In our case, the* ***Alias.user_info.employment_type*** *will be predefined.*
- **Active Value** - enter the value that will be linked with the section. Here, we add the **part-time** value.
- **Section Type** - select the section type from the drop-down list. For this section, we select the **Edit type** from the drop-down list.

20.3.1.3 Configure Switched Section - Contract tab

Click the **Plus (Add Tab)** icon next to Default tab. A new tab named **Link 2** will appear next to the first Default tab. Click the Gear icon and set up the following parameters the newly added tab:

- **Layout Label** - enter the section name. Here, we add the first tab name: **Switched Section - Contract**.
- **Hide section pop-out** - select the check box.
- ***Linked Field*** *- the linked field alias will be predefined according to the selected option in the* ***first tab****. In our case, the* ***Alias.user_info.employment_type*** *will be predefined.*
- **Active Value** - enter the value that will be linked with the section. Here, we add the **contract** value.
- **Section Type** - select the section type from the drop-down list. For this section, we select the **Edit type** from the drop-down list.

When all three tabs are configured and saved, we must return to the first (Default) tab and set up the **Linked Field** value. Open the Setting for the Default tab and select the **Linked Field** alias from the drop-down list. In our case, we select **Alias.user_info.employment_type**.

Click the **Update** button at the bottom of the Screen Builder page to save changes.

20.3.1.4 Check Sections' Switch in Render

Let us show how the configured sections' switch works on the screen in the Render application.

For that, use the Manager user credentials to log in the Render application.

In case the above-mentioned settings are correct, for Manager users (Jane Hanson and John Smith), we will see the configured first section displayed on the Field Types screen and will be able to switch sections. According to the selected option in the List to Switch Sections field on the Header level, on the Section level, the sections displayed in the first section will be switched. For example, if you select the **Full-time** option from the **List to Switch Sections** on header level, the **Switched Section - Full-time** section appears below. That will work for the **Part-time** and **Contract** option accordingly.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Set up Sections' Switch***.

20.3.2 Configure the Switched Sections

Then, we will add data to every tab of the first section:

- **Switched Section - Contract**
- **Switched Section - Full-time**
- **Switched Section - Part-time**

Configure Switched Section - Contract tab

On the **Switched Section - Contract tab**, we add the following parameters:

Text / Textarea Type Fields

Text Email (js focusout lowercase) field

Add the **Text Email (js focusout lowercase)** field. In the Render application, this field type is used to add the data in text format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.email**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Top** option from the drop-down list.
- **Type** - select the **Text** type. This field type provides the possibility to add a text value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Email (js focusout lowercase)**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Text Email (js focusout lowercase)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **JavaScript** function to the **Text Email (js focusout lowercase)** field. Click the **Edit JavaScript** button at the bottom of the **Field Settings** pop-up window. In the **Common javascript** tab of the opened **Custom JavaScript** pop-up window, define the JavaScript function to the OnFocusOut action.

- **Select JS template** - select the existing JavaScript template from the drop-down list. In our case, we select the **To Lower case** template to define that the entered data in the field must be written in small letters. I.e., in case the entered data in the Render application is written in capital letters, such letters will be replaced with small letters.

Text Date field

Add the **Text Date** field. In the Render application, this field allows defining the date in the calendar. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.birthday**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Text** type. This field type provides the possibility to add a text value.
- **Format Type** - select the field format type from the drop-down list. We select the **Date** option from the drop-down list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Date**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Text Date**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Text Phone field

Add the **Text Phone** field. In the Render application, this field allows adding the phone number. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.phone**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Text** type. This field type provides the possibility to add a text value.
- **Format Type** - select the field format type from the drop-down list. We select the **Phone** option from the drop-down list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Phone**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Text Phone**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Text Custom (js focusout Uppercase)

Add the **Text Custom (js focusout Uppercase)** field. In the Render application, this field allows adding the data in custom format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.identify**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Top** option from the drop-down list.
- **Type** - select the **Text** type. This field type provides the possibility to add a text value.
- **Format Type** - select the field format type from the drop-down list. We select the **Custom** option from the drop-down list to add the information in the custom format. The additional **Format Template** field appears:
    - **Format Template** - using the **x**, **n**, **-**, **+** characters enter the custom field format that will be shown as a placeholder of the field on the screen in the Render application. We enter the **xnn-nnn-nnn** value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Custom (js focusout Uppercase)**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Text Custom (js focusout Uppercase)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **JavaScript** function to the **Text Custom (js focusout Uppercase)** field. Click the **Edit JavaScript** button at the bottom of the **Field Settings** pop-up window. In the **Common javascript** tab of the opened **Custom JavaScript** pop-up window, define the JavaScript function to the OnFocusOut action.

- **Select JS template** - select the existing JavaScript template from the drop-down list. In our case, we select the **To Upper case** template to define that the entered data in the field must be written in capital letters. I.e., in case the entered data in the Render application is written in small letters, such letters will be replaced with capital letters.

Text (with custom query)

Add the **Text (with custom query)** field. In the Render application, this field allows adding the data in text format with the custom query. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.referral**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Text** type. This field type provides the possibility to add a text value.

- **Label** - enter the text that will be displayed on UI for this field, in our case **Text (with custom query)**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **All characters after "[" and up to the end of the line will be used for the search**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the custom query from the drop-down list. In this example, we select the **Get-All-usernames - Get-All-usernames-for-text-field-and-textarea** option from the drop-down list. In the field that appears to the right, the **:username** option will be defined automatically.

Textarea (with custom query)

Add the **Textarea (with custom query)** field. In the Render application, this field is displayed as the text area and allows adding the data in text format with custom query. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.friends**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Textarea** type. This field type provides the possibility to add a text value in the textarea in the Render application.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Textarea (with custom query)**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **All characters after "[" and up to the end of the line will be used for the search**.
- **Num Rows** - specify the quantity of rows displayed for the field. We enter the **4** value.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the custom query from the drop-down list. In this example, we select the **Get-All-usernames - Get-All-usernames-for-text-field-and-textarea** option from the drop-down list. In the field that appears to the right, the **:username** option will be defined automatically.

Numeric Type Fields

Numeric Decimal Static (4 / 2)

Add the **Numeric Decimal Static (4 / 2)** field. In the Render application, this field allows adding the data in numeric format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.tax_remission_percent**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Numeric** type. This field type provides the possibility to add a numeric value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Numeric Decimal Static (4 / 2)**.
- **Format Type** - select the **Decimal - Static** type from the drop-down list.
- **Max Length** - enter **4**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Numeric Decimal Static (4 / 2)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Precision** - enter **2**.

Numeric Currency Dynamic (9) field

Add the **Numeric Currency Dynamic (9)** field. In the Render application, this field allows adding the data in numeric format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.currency_dynamic**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Numeric** type. This field type provides the possibility to add a numeric value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Numeric Currency Dynamic (9)**.
- **Format Type** - select the **Currency - Dynamic** type from the drop-down list.
- **Max Length** - enter **9**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Numeric Currency Dynamic (9)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Numeric Currency Static (9) field

Add the **Numeric Currency Static (9)** field. In the Render application, this field allows adding the data in numeric format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.currency_static**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Numeric** type. This field type provides the possibility to add a numeric value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Numeric Currency Static (9)**.
- **Format Type** - select the **Currency - Static** type from the drop-down list.
- **Max Length** - enter **9**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Numeric Currency Static (9)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Numeric Decimal Dynamic (7 / 3) field

Add the **Numeric Decimal Dynamic (7 / 3)** field. In the Render application, this field allows adding the data in numeric format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.decimal_dymanic**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Numeric** type. This field type provides the possibility to add a numeric value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Numeric Decimal Dynamic (7 / 3)**.
- **Format Type** - select the **Decimal - Dynamic** type from the drop-down list.
- **Max Length** - enter **7**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Numeric Decimal Dynamic (7 / 3)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Precision** - enter **3**.

Add the **Numeric Decimal Static (7 / 3)** field. In the Render application, this field allows adding the data in numeric format. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.decimal_static**.
- **Label Orientation** - select the label orientation from the drop-down list. We select the **Left** option from the drop-down list.
- **Type** - select the **Numeric** type. This field type provides the possibility to add a numeric value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Numeric Decimal Static (7 / 3)**.
- **Format Type** - select the **Decimal - Static** type from the drop-down list.
- **Max Length** - enter **7**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Numeric Decimal Static (7 / 3)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Precision** - enter **3**.

For the **Switched Section - Contract tab**, these are all the fields.

Configure Switched Section - Full-time (Default) tab

On the **Switched Section - Full-time (Default) tab**, we add the following parameters:

Label Type Field

Add the label. For that, set up the following parameters:

- **Label** - enter the label name. In this example, we enter the **A large list of fields is located in the "Switched Section - Contract"** value.

Multi-select list Type Field

Add the **Multiselect List** field. This field will be set as the Working Days list with available days (options) to be selected as user working days. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.working_days**.
- **Type** - select the **Multi-select list** type. This field type allows users to select more than one option from the list. In our case, in the Render application, users can select more than one day as a working day.
  By selecting the Multi-select list field type, the **Pull up** button appears. Click the **Pull up** button to access the List Params.
- **List Params** - select the **Days** list from the list.

- **Label** - enter the text that will be displayed on UI for this field, in our case **Multiselect List**.
- **Num Rows** - enter **8**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Multiselect List**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

For the **Switched Section - Full-time (Default)**, these are all the fields.

Configure Switched Section - Part-time tab

On the **Switched Section - Part-time tab**, we add the following parameters:

Label Type Field

- **Label** - enter the label name. In this example, we enter the **A large list of fields located in the "Switched Section - Contract"** value.

Text Type Fields

Text Custom field

Add the **Text Custom** field that will define the user's working time end. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.end_work_time**.
- **Type** - select the **Text** type. This field type allows adding the text value to the field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Custom**.
- **Format Type** - select the **Custom** type from the drop-down list. This field format type restricts the field to the format template defined in the Format Template field. In our case, the text value of the End Time field will be added using the numeric characters and has the nn:nn view defined in the Format Template field.
  By selecting the Custom type, the additional Format Template field appears.
- **Format Template** - define the **nn:nn** format type, where **n** is for numeric character input only.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Text Custom**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Text Custom Field

Add the **Text Custom** field that will define the user's working time start. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.start_work_time**.

- **Type** - select the **Text** type. This field type allows adding the text value to the field.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Custom**.
- **Format Type** - select the **Custom** type from the drop-down list. This field format type restricts the field to the format template defined in the Format Template field. In our case, the text value of the Start Time field will be added using the numeric characters and has the nn:nn view defined in the Format Template field.
  By selecting the Custom type, the additional Format Template field appears.
- **Format Template** - define the **nn:nn** format type, where **n** is for numeric character input only.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Text Custom**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**

For the **Switched Section - Part-time (Default)**, these are all the fields.

Click the **Update** button to save changes of the Field Types screen.

You can adjust the layout on the fields of the screen according to your preferences.

20.3.2.1 Check the Switched Sections in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application using the Manager user credentials.

If the above-mentioned sections are set up correctly for Manager users (Jane Hanson and John Smith), we will see the configured first subsection displayed on the Field Types screen. In the first subsection of the Field Types screen, we will be able to add, edit, and delete user data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen first Subsection and Check in Render***.

20.3.3 Configure the second section

Further, on the second page of the screen builder of the Field Types screen, we will add and configure new parameters for the second section on the section level.

First, configure the section settings:

- **Hide section header** - select the check box.
- **Hide section pop-up** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

After that, we can add the following fields to the section:

List Type field

Add the **List** field. In the Render application, this field is displayed as the list with options. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.execute_function**.
- **Type** - select the **List** type. The **List** field type allows users to select an option from the list in the Render application.
- **Pull up** - click the **Pull up** button and select the **EmailTemplates** list from the List Params pop-up window.
- **Label** - enter the text that will be displayed on UI for this field, in our case **List**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **List**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Relation Datalist Type fields

Add the **Relation Datalist** field. In the Render application, this field is displayed as the list with options. Selecting the option from this relation datalist will affect the data display on the screen. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.relation_datalist**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Relation Datalist**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Relation Datalist**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the **Get-All-Users-for-multi - Get-All-Users-for-multi** query from the drop-down list.

Add the **Relation Datalist Multi** field. In the Render application, this field is displayed as the multi select list with options. Selecting the option from this relation datalist will affect the data display on the screen. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.relation_datalist_multi**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Relation Datalist Multi**.

- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Relation Datalist Multi**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Multi select**
- **Custom Query** - select the **Get-All-Users-for-multi - Get-All-Users-for-multi** query from the drop-down list.

Datalist Type field

Add the **Datalist** field. In the Render application, this field is displayed as the multi select list with options. The list options can be added to the datalist in the Field Settings pop-up window. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.datalist**.
- **Type** - select the **Datalist** type. This field type allows users to select options from the list that are defined with custom query.
- **Values (Use the Delimiter ";")** - enter the datalist values. In our case, we enter **First option;Second option;Third option** values.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Datalist**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Datalist**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Multi select**

Checkbox Type fields

Add the **Check box (without group)** label. For that, set up the following parameters:

- **Label** - enter the label name. In this example, we enter the **Check box (without group)** value.

Add the first check box and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule1**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for Rule 1 as reviewed.
- **Label** - enter the text that will be displayed on UI for this field, in our case **1**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Check box**.

- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the second check box and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule2**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for Rule 2 as reviewed.
- **Label** - enter the text that will be displayed on UI for this field, in our case **2**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Check box**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Check box (with group)** label for the set of check boxes that build one check box group in case they have the same Group Name specified. Note that for the check box group, the selection is restricted to one option only. For that, set up the following parameters:

- **Label** - enter the label name. In this example, we enter the **Check box (with group)** value.

Add the third check box and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule3**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for Rule 3 as reviewed.
- **Label** - enter the text that will be displayed on UI for this field, in our case **3**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Check box**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - enter the check box group name. In this example, we enter **checkbox_group_1** value.

Add the fourth check box and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.rule4**.
- **Type** - select the **Checkbox** type. This field type allows users to check the box for Rule 4 as reviewed.

- **Label** - enter the text that will be displayed on UI for this field, in our case **4**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Check box**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - select the existing check box group name from the drop-down list. In this example, we select **checkbox_group_1** value.

Radio Type fields

Add the **Radio (without group)** label. For that, set up the following parameters:

- **Label** - enter the label name. In this example, we enter the **Radio (without group)** value.

Add the **4+3** field that will be displayed as a radio button and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.weekdays_4_3**.
- **Type** - select the **Radio** type. This field type allows users to select one option from the group.
- **Label** - enter the text that will be displayed on UI for this field, in our case **4+3**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Radio**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **Radio (with group)** label. For that, set up the following parameters:

- **Label** - enter the label name. In this example, we enter the **Radio (with group)** value.

Add the **5+2** field that will be displayed as a radio button and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.weekdays_5_2**.
- **Type** - select the **Radio** type. This field type allows users to select one option from the group.
- **Label** - enter the text that will be displayed on UI for this field, in our case **5+2**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Radio**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - enter the radio group name. In this example, we enter the **radio_group_1** value.

Add the **6+1** field that will be displayed as a radio button and can be selected as one of the options in the Render application. Set up the following parameters for the field:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.weekdays_6_1**.
- **Type** - select the **Radio** type. This field type allows users to select one option from the group.
- **Label** - enter the text that will be displayed on UI for this field, in our case **6+1**.
- Tooltip - enter the text that will be displayed on UI for this field as a tooltip, in our case **Radio**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Group Name** - select the existing radio group name from the drop-down list. In this example, we select the **radio_group_1** value.

Label Type field

Add the **Image Map for picture (map element coordinates are taken in pixels from the original image)** label. For that, set up the following parameters:

- **Label** - enter the label name. In this example, we enter the **Image Map for picture (map element coordinates are taken in pixels from the original image)** value.

Format the **Image Map for picture (map element coordinates are taken in pixels from the original image)** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Color** - select the text color from the drop-down list.
- **Text Align** - select the **Center** option from the drop-down list.

Static Image

Add the **Image Map** image. In the Render application, this type of image is used as a map with sectors containing the linked pages. Set up the following parameters for the image:

- **Value** - enter the image name. We enter the **Image Map** value.
- Upload file to be used as the image map.
- **Use image map** - select the check box to use the image as the map.
- **Area Parameters** section - define the following parameters for the first area:
    - **Shape** - select the shape type from the drop-down list. We select the **Rect** option from the drop-down list.
    - **Coords** - enter the sector coordinates on the image map. We enter the **0,0,88,45** value.
    - **Target** - select the display mode for the sector linked page from the drop-down list. In this example, we select the **Blank** value to open the linked page in a new tab.
    - **Title** - enter the sector name. We enter the **Home Page** value.

- **Href** - enter the link to the internal or external data to be opened by clicking the sector in the Render. We enter the **https://appmill.com** value.

- **Area Parameters** section - define the following parameters for the second sector:
  - **Shape** - select the shape type from the drop-down list. We select the **Rect** option from the drop-down list.
  - **Coords** - enter the sector coordinates on the image map. We enter the **88,0,195,45** value.
  - **Target** - select the display mode for the sector linked page from the drop-down list. In this example, we select the **Blank** value to open the linked page in a new tab.
  - **Title** - enter the sector name. We enter the **Platform** value.
  - **Href** - enter the link to the internal or external data to be opened by clicking the sector in the Render. We enter the **https://appmill.com/platform/** value.
- **Area Parameters** section - define the following parameters for the third sector:
  - **Shape** - select the shape type from the drop-down list. We select the **Rect** option from the drop-down list.
  - **Coords** - enter the sector coordinates on the image map. We enter the **195,0,307,45** value.
  - **Target** - select the display mode for the sector linked page from the drop-down list. In this example, we select the **Blank** value to open the linked page in a new tab.
  - **Title** - enter the sector name. We enter the **Why AppMill?** value.
  - **Href** - enter the link to the internal or external data to be opened by clicking the sector in the Render. We enter the **https://appmill.com/why-appmill/** value.

Save the changes and close the pop-up.

Click the **Update** button at the bottom of the Screen Builder to save the changes.

You can adjust the layout on the fields of the screen according to your preferences.

20.3.3.1 Check the second section in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned screen is set up correctly for Manager users (Jane Hanson and John Smith), we will see the configured second subsection displayed on the Field Types screen. In the second subsection of the Field Types screen, we will be able to add, edit, delete user data, select the options from the lists, check the boxes and radio buttons, and by clicking the sector on the image map be redirected to the corresponding pages.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen second Subsection and Check in Render***.

20.3.4 Configure the third section: Linked List and With Dependent Field

Further, on the second page of the screen builder of the Field Types screen, we will add and configure new parameters for the third section on the section level.

First, configure the section settings:

- **Layout Label -** enter the subsection name. In our case, we enter the **Linked List Elements and With Dependent Field** value.
- **Hide section pop-up** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

Then, we will add the following fields to the **Linked List Elements and With Dependent Field** section:

Inline Search Type field

Add the **Inline Search (with Linked List to multiselect)** field. This field type allows users to see search results without redirecting to a search results page. In the Render application, the field with the inline search option can be displayed as a multi select list with options. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.mentor_eq_type**.
- **Type -** select the **Inline search** type. This field type allows users to search for the appropriate mentor equipment type from the list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Inline Search (with Linked List to multiselect)**.
- **Label Orientation** - select the **Top** option from the drop-down list.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Inline Search (with Linked List to multiselect)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **Custom Query** - select the **Get-Equipment-Type - Get-Equipment-Type-for-inline-search** option from the drop-down list. In the field that appears to the right, the **:equipType** option will be defined automatically.

Multi-select list Type field

Add the **Multiselect (with linked list to List)** field. This field type allows users to select multiple values from a drop-down list of field options in the Render application. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.mentor_eq_brand**.
- **Type** - select the **Multi-select list** type. This field type allows users to select more than one mentor equipment brand from the list.
- **Label Orientation** - select the **Top** option from the drop-down list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Multiselect (with linked list to List)**.
- **Num Rows** - enter **5**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case**Multiselect (with linked list to List)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

List Type field

Add the **List (depends on Inline Search and Multiselect fields)** field. In the Render application, this field will contain the list option. Set up the following parameters for the field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.mentor**.
- **Type** - select the **List** type. The List field type allows users to select an option from the list in the Render application. In our case, users will be able to select a mentor from the mentor names drop-down list.
- **Label Orientation** - select the **Top** option from the drop-down list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **List (depends on Inline Search and Multiselect fields)**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **List (depends on Inline Search and Multiselect fields)**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Save the changes to apply the added fields in the list of reference fields. After that, the list options in the **Field ID monitor** field will be available.

Update the **Inline Search (with Linked List to multiselect)** field:

- **Linked List** - select the check box to set the relation between different fields/values.
- **Field ID monitor** - select the field ID from the drop-down list. We select the **Alias.user_info.mentor_eq_brand** value.

- **Query to Execute** - select the query to execute from the drop-down list. In this case, **Get-Equipment-Band - Get-Equipment-Band-for-linked-list** value from the drop-down list.

Update the **Multiselect (with linked list to List)** field:

- **Linked List** - select the check box to set the relation between different fields/values.
- **Field ID monitor** - select the field ID from the drop-down list. We select the **Alias.user_info.mentor** value.
- **Query to Execute** - select the query to execute from the drop-down list. In this case, **Get-Usernames-By-Equipment-Band-and-Type - Get-Usernames-By-Equipment-Band-and-Type-for-linked-list** value from the drop-down list.
- **Additional Field to Send** - select the **Alias.user_info.mentor_eq_type** field from the list.

Text Type field

Add the **Text Datetime (base field)** field. This field type allows adding the information about date and time in the format of Date and Time in the Render application. Set up the following parameters for the field:

- **Data field** - select the Primary Table check box and select the **Alias.user_info.create_on**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Text Datetime (base field)**.
- **Type** - select the **Text** type.
- **Format Type** - select the **Date/Time** format type.
- **Tooltip** - enter the text that will be displayed on UI for this field, in our case **Base field for With dependent field**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Add the **List (pivot condition)** field and set up the following parameters:

- **Data field** - select the Primary Table check box and select the **Alias.user_info.priority**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **List (pivot condition)**.
- **Type** - select the **List** type.
- **Pull up** - click the **Pull up** button and select the **TaskPriority** list from the List Params pop-up window.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

With dependent field Type field

Add the **With Dependent Field** field. This field type allows users to define the conditions for the case when the value of one date field should be calculated based on the values of other fields (dependent fields). Set up the following parameters for the field:

- **Data field** - select the Primary Table check box and select the **Alias.user_info.due_date**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **With Dependent Field**.
- **Type** - select the **With dependent field** type.
- **Format Type** - select the **Date/Time** format type.
- **Tooltip** - enter the text that will be displayed on UI for this field, in our case **With Dependent Field**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
- **ID of dependent field** - enter the **Text Datetime (base field)** field **ID**. The ID can be found at the bottom left corner of the Created on Field Settings pop-up window.
  For the dependent field
    - **Lead time** - enter **5**. You can adjust this value according to your needs.
    - **Use field date** - select the check box.
- Add three rows with pivot fields with the following parameters:
    - The first row:
        - **ID of pivot field** - enter ID of the **List (pivot condition)** field. The ID can be found at the bottom left corner of the Priority Field Settings pop-up window.
        - **Field value** - enter the **Low** value.
        - **Lead time** - enter **7**. You can adjust this value according to your needs.
    - The second row:
        - **ID of pivot field** - enter ID of the **List (pivot condition)** field. The ID can be found at the bottom left corner of the Priority Field Settings pop-up window.
        - **Field value** - enter the **Medium** value.
        - **Lead time** - enter **3**. You can adjust this value according to your needs.
    - The third row:
        - **ID of pivot field** - enter ID of the **List (pivot condition)** field. The ID can be found at the bottom left corner of the Priority Field Settings pop-up window.
        - **Field value** - enter the **High** value.

- **Lead time** - enter **2**. You can adjust this value according to your needs.

Click the **Update** button to save changes of the Field Types screen.

You can adjust the layout on the fields of the screen according to your preferences.

20.3.4.1 Check the Linked List and With Dependent Field Section in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned screen is set up correctly for Manager users (Jane Hanson and John Smith), we will see the configured third subsection displayed on the Field Types screen. In the third subsection of the Field Types screen, we will be able to add, edit, delete user data.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen third Subsection and Check in Render***.

20.3.5 Configure the fourth section: Label Iframe

Further, on the second page of the screen builder of the Field Types screen, we will add and configure new parameters for the fourth section on the section level.

First, configure the section settings:

- **Layout Label -** enter the subsection name. In our case, we enter the **Label Iframe** value.
- **Hide section pop-up** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

After that, we can add the following fields to the **Label Iframe** section:

Iframe Label Type

Add the **Iframe** label. In the Render application, the label with the Iframe type is containing a widget or a preview of the linked page. Set up the following parameters for the label:

- **Label** - enter the label name. In this example, we enter the **Iframe** value.
- **Link Type** - select the check box.
- **URL** - enter the URL address. In this example, we enter the https://appmill.com/ value.
- **Type Link** - select the type link from the drop-down list. We select the Iframe value from the drop-down list.

Save the changes on the page.

20.3.5.1 Check Label Iframe Section in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned section is configured correctly for Manager users (Jane Hanson and John Smith), we will see and access the AppMill website from the section.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen 4th Subsection and Check in Render***.

20.3.6 Configure the fifth section (Edit Type): Document Types

Then, we will define the section to demonstrate the fields of Document Type.

First, configure the section settings:

- **Layout Label -** enter the subsection name. In our case, we enter the **Document Types** value.
- **Hide section pop-up** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

After that, we add the following fields to the Document Types **s**ection:

Document Type field

Add the **Document Type Field** that allows users to upload and download documents on the screen in the Render application. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.document_id**.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Document Type Field**.
- **Label Orientation** - select the label orientation from the drop-down list. In this example, we select the **Top** value.
- **Type** - select the **Document** type.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **Document Type Field**.
- **Document Family** - select the **Documents - Documents** value from the drop-down list. The **Document Category** field appears.
- **Document Category** - select the **main - main** value from the drop-down list.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Label Type

Add the **Please note! We will use this document for the Document Types section. Therefore, we need to use txt or pdf files.** label that will be used as the label for the field with the Document type. For that, set up the following parameters:

- **Label** - enter the text that will be displayed on UI for this label, in our case **Please note! We will use this document for the Document Types section. Therefore, we need to use txt or pdf files.**

Format the **Please note! We will use this document for the Document Types section. Therefore, we need to use txt or pdf files.** label. In the Label Settings pop-up window, click the **Formatting** button. The Formatting pop-up window opens. Define the following parameters for the label:

- **Text Color** - select the text color from the drop-down list.
- **Text Align** - select the **Center** option from the drop-down list.

Document Type Button

Add the **Document Type Button** that allows users to upload and download documents on the screen in the Render application. For that, set up the following parameters:

- **Value -** enter the button name. We enter the **Document Type Button** name.
- **Button Action -** select the **Document** action from the drop-down list.
- **Document Family** - select the document family from the drop-down list. We select the **Documents - Documents** option. The **Document Category** field appears.
- **Document Category** - select the **main - main** value from the drop-down list.

Static Image

Add the **Document Type Image** that allows users to upload and download documents on the screen in the Render application. For that, set up the following parameters:

- **Value -** enter the image name. We enter the **Document Type Image** name.
- **Static Image Action** - select the **Document** action from the drop-down list.
- Upload file to be displayed as the image and used to upload documents in the Render application.
- **Document Family** - select the document family from the drop-down list. We select the **Documents - Documents** option. The **Document Category** field appears.
- **Document Category** - select the **main - main** value from the drop-down list.

Submit changes on the page.

20.3.6.1 Check Document Types Section in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned section is set up correctly for Manager users (Jane Hanson and John Smith), we will see the configured section and be able to upload, download, view, and edit the documents.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen 5th Subsection and Check in Render***.

20.3.7 Configure the sixth section (Document Type): Document Types Section

Then, we will configure the section of the Document Type.

First, configure the section settings:

- **Layout Label -** enter the subsection name. In our case, we enter the **Document Types Section** value.
- **Hide section pop-up** - select the check box.
- **Section Type -** select the **Document Type** from the drop-down list. The **Function Name for Getting List** field appears.
- **Function Name for Getting List -** select the function from the drop-down list. In our case, we select the **GetList_user_info: Universal DLL : GetList_user_info** function.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Document Type Section. We add the **document_id** alias field.

Please be aware that changing the section size with the Document Type may reset previously set parameters for the section.

20.3.7.1 Check Document Types Section in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned section is set up correctly for Manager users (Jane Hanson and John Smith), we will be able to preview the uploaded document in txt or pdf format in this section.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure Field Types Screen 6th Subsection and Check in Render***.

20.4 Create Screen: Table Type

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, define the following data:

- **Select Existing Screen** - as the **Tables Type** screen will be accessed from the Field Types screen, we need to select the **FieldTypes** screen from the drop-down list.
- **Screen Description** - enter the **TablesType** screen description.

- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **10**.
- **Tab Text** - enter the **TablesType** tab text.
- **Tab Description** - enter the **TablesType** tab description.
- **Screen State** - select the **View** state from the drop-down list.
- In the **Screen Layout** section select the **Header 2x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen.

On the **Header** level, add a section of the **Edit type** and the following fields:

Add the **User** field and set up the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User**.
- **Tooltip** - enter the text that will be displayed on UI for this field, in our case **User**.
- Select the following check boxes to adjust field layout:
  - **Always show field border**
  - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

On the **Section** level we have 2 subsections:

- **first subsection**
  - **Section Type** - select the **Edit Type** from the drop-down list.
- **second subsection**
  - **Section Type** - select the **Edit Type** from the drop-down list.

20.4.1 Check Table Type Screen in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned screen for Manager users (Jane Hanson and John Smith) is set up correctly, we will see the Tables Type screen as a part of the Field Types screen displayed as a tab.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create and Check Tables Type Screen in Render***.

20.5 Configure the first Table section

Next, we configure the Table section of the Tables Type screen with the **Table** section type to display the data in table format in the Render application.

In the Architect application, Screen Configuration > Screens, open the FieldsType screen via the Screen builder option.

On the second page of the screen builder, open the Section setting by clicking the Gear icon and fill in the following fields:

- **Layout Label -** enter the subsection name. In our case, we enter the **Table** value.
- **CSV Download** - select the check box.
- **Section Type** - select the **Table Type** from the drop-down list.

The next part of the parameters that can be set up in the Section Settings pop-up window is connected to the general table settings:

- **Alias Framework PK Configuration** - this feature is used in special cases when the tables are connected via a composite key.
- **Key parts -** select the **Key parts** to define the fields taken from the tables of different levels (to fill in values from parent to child tables). Here, we do not select anything or select the “-- Select --”option from the drop-down list.
- **Include Column Filter** - select the check box to add the filter to the table columns.
- **Label Orientation** - select the **Top** value from the drop-down list to display the table header in the top row.
- **Show Type** - select the **Scroll** value from the drop-down list.
- **Pixel Height** - enter **500**.

After that, we need to set the table columns and aliases that should be used to receive the values. For that, add the following **alias fields** and enter the corresponding **column labels** that will be displayed as a column name:

Alias Field	Column Label
Array.certificates.user_info_id	Numeric
Array.certificates.name	Text Link
Array.certificates.description	Textarea
Array.certificates.image	DImage

- **Add Button Column** - click the button to add the button column and enter **Execute Button** as a column name.
- **Add Static Image Column** - click the button to add the column containing the static images and enter **Image Link** as a column name.

Save the changes in the pop-up window and close it.

Then, on the second of the screen builder we can proceed with configuring every column separately directly in the table.

To open a column-specific setting, click the **Gear** icon next to the column title. The **Config Column** pop-up window appears. Here, set up the parameters for every column:

- **Numeric:**
  - **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, the column can be filled in with the numeric data.
  - Select the following check boxes to adjust field layout:
    - **Required**
- **Text Link:**
  - **Param Type** - select the **Text** type from the drop-down list. In the Render application, users will be able to add the data in text format.
  - Select the following check boxes to adjust field layout:
    - **Required**
    - **Link Type**
      After selecting the Link Type check box, the additional parameters appear:
      - **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Modal** option from the drop-down list, to open a linked page in a modal (pop-up) window.
      - **Refresh base page when exit** - select the check box.
      - **Modal Size - 80%.**
      - **Menu** - select the menu that the required screen assigned to from the drop-down list. We select the **Demo - Demo** option from the drop-down list.
      - **Group Screen** - select the group screen that the required screen assigned to from the drop-down list. We select the **Field Types** option from the drop-down list.
      - **Screen** - select the required screen that will be opened by link from the drop-down list. We select the **FieldTypes - FieldTypes** option from the drop-down list.
      - **Parameter** - specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.

    - **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.certificates.user_info_id** option from the drop-down list.

- **Textarea:**
    - **Param Type** - select the **Textarea** type from the drop-down list. In the Render application, users will be able to add the description in text format to the textarea.
    - **Num Rows** - enter **4**.
    - Select the following check boxes to adjust field layout:
        - **Always show field border**
- **DImage:**
    - **Param Type** - select the **DImage** type from the drop-down list. In the Render application, users will be able to upload and download documents in image formats.
    - **Field Width** - enter the field width. We enter the **100px** value.
    - Select the following check boxes to adjust field layout:
        - **Allow full size modal popup**
- **Execute Button:**
    - **Param Type** - select the **Button** type from the drop-down list. In the Render application, users will be able to perform the execute action by clicking the **Execute Button** in the table column.
    - **Button Action** - select the button action from the drop-down list. In our example, we select the **Execute** action from the drop-down list.
    - **Value** - enter the button name. We enter the **Execute** name.
    - **Function** - add a new extension function. In the opened Extensions Settings pop-up window, define the extension function that will be executed when the user clicks the **Execute** button on the screen in the Render application. In our case as an example, we select the **ToDoTasks::ThirdFunction** option from the Extensions drop-down list and click the **Add** button.
- **Image Link:**
    - **Param Type** - select the **Image** type from the drop-down list. In the Render application, users will be able to be redirected to the linked page by clicking the image.
    - **Field Width** - enter the field width. We enter the **80px** value.
    - **Image Action** - select the image action from the drop-down list. In this example, we select the **Link** action from the drop-down list. The additional parameters appear:

- **Type Link** - select the type link display mode for the linked page from the drop-down list. We select the **Blank** option from the drop-down list, to open a linked page in a new tab.
- **Menu** - select the menu that the required screen assigned to from the drop-down list. We select the **Demo - Demo** option from the drop-down list.
- **Group Screen** - select the group screen that the required screen assigned to from the drop-down list. We select the **Field Types** option from the drop-down list.
- **Screen** - select the required screen that will be opened by link from the drop-down list. We select the **FieldTypes - FieldTypes** option from the drop-down list.
- **Parameter** - specify the required parameter that will be used for searching on the screen from the drop-down list. We select the **user_info_id** option from the drop-down list.
- **Pass through parameter** - select the parameter from the drop-down list that will be passed by link to the defined screen. We select the **Alias.certificates.user_info_id** option from the drop-down list.

- **Upload Image** - select and upload the image.

You can adjust the layout of the fields on the screen according to your preferences.

Save the changes on the page by clicking the Update button at the bottom of the page.

20.5.1 Check the Table section in Render

Let us show how the configured screen works in the Render application.

Open the Render application and use the Manager user credentials to log in to the Render application.

Then, Fields Type menu item > Tables Type tab:

If the above-mentioned screen for Manager users (Jane Hanson and John Smith) is configured correctly, we will see the Table section displayed in the Tables Type tab of the Field Types screen.

You will be able to add, delete, and update the table data, upload, and download the image, execute actions, and to be redirected to the linked page.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure the Table section of Tables Type Screen and check n in Render***.

20.6 Configure the second Table section

Next, we configure the second Table section of the Tables Type screen with the Table section type to display the data in table format in the Render application.

In the Architect application, Screen Configuration > Screens, open the FieldsType screen via the Screen builder option.

On the second page of the screen builder, open the Section settings by clicking the Gear icon and fill in the following fields:

- **Layout Label -** enter the subsection name. In our case, we enter the **Table_2** value.
- **Hide section pop-out** - select the check box.
- **CSV Download** - select the check box.
- **Section Type** - select the **Table Type** from the drop-down list.

The next part of the parameters that can be set up in the Section Settings pop-up window is connected to the general table settings:

- **Alias Framework PK Configuration** - this feature is used in special cases when the tables are connected via a composite key.
- **Key parts -** select the **Key parts** to define the fields taken from the tables of different levels (to fill in values from parent to child tables). Here, we will select the **Alias.user_info.user_info_id** option from the drop-down list.
- **Total for the Column** - select the check box to add the calculation for the columns.
- **Total for the Row** - select the check box to add the calculation for the rows.
- **Include Column Filter** - select the check box to add the filter to the table columns.
- **Label Orientation** - select the **Top** value from the drop-down list to display the table header in the top row.
- **Show Type** - select the **Scroll** value from the drop-down list.
- **Pixel Height** - enter **500**.

After that, we need to set the table columns, labels, and aliases that should be used to receive the values. For that, add the following **alias fields** and enter the corresponding **column labels** that will be displayed as a column name and reflect the field type:

Alias Field	Column Label
Array.equipment.type	Text
Array.equipment.brand	Text field
Array.equipment.price	Numeric Currency Static
Array.equipment.tax	Numeric Currency Dynamic
Array.equipment.year	Numeric (with js min - max)

Save the changes in the pop-up window.

Now, we need to set every column separately. It can be done directly in the table.

To open a column-specific setting, click the **Gear** icon next to the column title. The **Config Column** pop-up appears. Here, set up the parameters for the following columns:

To open a column-specific setting, click the Gear icon next to the column title. The Config Column pop-up appears. Here, set up the parameters for the corresponding columns:

Text:

- **Param Type** - select the Text type from the drop-down list. In the Render application, users will be able to add the data in text format to the Text column.

Text field:

- **Param Type** - select the Text type from the drop-down list. In the Render application, users will be able to add the data in text format to the Text field column.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

Numeric Currency Static:

- **Param Type** - select the Numeric type from the drop-down list. In the Render application, users will be able to add the data in numeric format to the Numeric Currency Static column.
- **Format Type** - select the format of the column data from the drop-down list. We select the **Currency - Static** type.
- **Max Length** - define the entered data max length. We enter the 7 value.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Use to calculate totals**

Numeric Currency Dynamic:

- **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add the data in numeric format to the **Numeric Currency Dynamic** column.
- **Format Type** - select the format of the column data from the drop-down list. We select the **Currency - Dynamic** type.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Use to calculate totals**

Numeric (with js min - max):

- **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add the data in numeric format to the **Numeric (with js min - max)** column.
- Select the following check boxes to adjust field layout:
    - **Always show field border**

- **Edit JavaScript -** click the button and define the JavaScript functionality to the **Numeric (with js min - max)** column. For that, perform the following steps:
    - In the opened Custom JavaScript pop-up window, go to the Common javascript tab.
    - On the **Common javascript** tab, go to the **onFocusOut** tab and select the **Check max value** template from the drop-down list.
    - In the appeared **Check max size** form, define the:
        - **Max Value** - enter **2050** value.
        - **Error Message** - enter the message to be displayed as an error message in case the entered value in the Render will be greater than the value defined in the Max Value field.
    - After that, select the **Check min size** from the drop-down list.
    - In the appeared **Check min size** form, define the:
        - **Min Value** - enter **1970** value.
        - **Error Message** - enter the message to be displayed as an error message in case the entered value in the Render will be less than the value defined in the Min Value field.
        - **Use Confirm Modal** - select the check box to enable the pop-up window containing the confirmation message in the Render application.

You can adjust the layout of the fields on the screen according to your preferences.

Save the changes by clicking the Update button.

20.6.1 Check the second Table Section in Render

Let us show how the configured screen works in the Render application.

Open the Render application and use the Manager user credentials to log in to the Render application.

Then, Fields Type menu item > Tables Type tab:

If the above-mentioned screen for Manager users (Jane Hanson and John Smith) is set up correctly, we will see the Table section displayed in the Tables Type tab of the Field Types screen. You will be able to see the data in the table format, add, delete, and edit the column data, and calculate the numeric data for the columns and rows.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Configure second Subsection of Tables Type Screen and Check in Render***.

21 Create Environment for Orders Screen

Now, we will create the environment containing the following settings for a new Orders screen:

- Create user_order table
- Set up a new auto sequence
- Create a new screen named Orders
- Create a new group screen
- Check alias dependencies and relationship

Below, we describe every setting in detail.

21.1 Create user_oder Table

In this step, we will create a new table for Manager users. This table provides the possibility to manage orders and screens for Manager users.

In the Architect application, open Features > Tables and create a new table with the following fields:

- **Table Name** - enter a table name. In our case **user_order**.
- In the **Fields/Column Details** section define the following table parameters:

Name	Type	Nullable	Scale	Length
user_order_id	Integer	False	-	-
user_info_id	Integer	True	-	-
cost_of_goods	Numeric	True	2	9
cost_of_service	Numeric	True	2	9
total_cost	Numeric	True	2	9
created_on	Date	True	-	-

- **Constraint Type** - select the **Primary Key** constraint type.
- **Constraint Name** - enter the **user_order_pk** constraint name.
- **Columns** - select **user_order_id** column.

Submit changes on the page.

21.2 Set up Auto Sequence

The next step is setting up a new Auto Sequence rule. Here, we will create a unique order ID for Manager users.

In the Architect application, open Features > Auto Sequence and create a new Auto Sequence rule:

- **Table Name** - select the **user_order** table from the drop-down list.
- In the **Fields Details** section, define the following parameters:
    - **Field Name** - enter the **user_order_id** field name.
    - **Increment by** - enter **1**.

Submit changes on the page.

21.3 Create Screen: Orders

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add New Screen** - enter the **Orders** screen name.
- **Screen Description** - enter the **Orders** screen description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **99**.
- **Hide Search Bar** - select the check box.
- **Simple Search** - activate the Simple Search and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.

- **Tab Text** - enter the **Orders** tab text.
- **Tab Description** - enter the **Orders** description.
- **Screen State** - select the **View** screen state.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen. For that, we will perform the following operations:

On the **Header** level, add a section of the **Edit type**.

Add the **User** field that will provide the possibility in the Render application to select the user from the list and add new information to his/her profile details. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the user ID will be available to be selected from the users list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **User**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

On the **Section** level, add a section of the **Table type**.

- **Layout Label** - enter the subsection name. In our case, we enter the **Orders** value.
- **Hide section pop-out** - select the check box.
- **CSV Download** - select the check box.
- **Section Type** - select the **Table Type** from the drop-down list.

The next part of the parameters that can be set up in the Section Settings pop-up window is connected to the general table settings.

- **Alias Framework PK Configuration** - this feature is used in special cases when the tables are connected via a composite key.
- **Total for the Column** - select the check box to add the calculation for the columns.
- **Label Orientation** - select the **Top** value from the drop-down list to display the table header in the top row.
- **Show Type** - select the **Scroll** value from the drop-down list.
- **Pixel Height** - enter **500**.

After that, we need to set the table columns, labels, and aliases that should be used to receive the values. For that, add the following alias fields:

- **Array.user_order.user_order_id** - enter the **Order ID (hidden)** value to be displayed as the column name.
- **Array.user_order.user_info_id** - enter the **User ID (hidden)** value to be displayed as the column name.

- **Array.user_order.created_on** - enter the **Date** value to be displayed as the column name.
- **Array.user_order.cost_of_goods** - enter the **Goods (js for total)** value to be displayed as the column name.
- **Array.user_order.cost_of_service** - enter the **Service (js for total)** value to be displayed as the column name.
- **Array.user_order.total_cost** - enter the **Total** value to be displayed as the column name.

Save the changes on the page.

Now, we need to set every column separately. It can be done directly in the table.

To open a column-specific setting, click the **Gear** icon next to the column title. The **Config Column** pop-up window appears. Here, set up the parameters for the following columns:

- **Order ID (hidden)**
  - **Param Type** - select the **Text** type from the drop-down list. In the Render application, the column can be filled in with the data in text format.
  - Select the following check boxes to adjust field layout:
    - **Hidden**
- **User ID (hidden)**
  - **Param Type** - select the **Text** type from the drop-down list. In the Render application, users will be able to add the data in text format.
  - Select the following check boxes to adjust field layout:
    - **Hidden**
- **Date**
  - **Param Type** - select the **Text** type from the drop-down list. In the Render application, users will be able to add the data in text format.
  - **Format Type** - select the **Date** format type from the drop-down list.
  - Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Use current date/time**
- **Total**
  - **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add the data in numeric format.
  - **Format Type** - select the **Decimal - Dynamic** format type from the drop-down list.

- **Max Length** - enter the max length for the data entered in the **Total** field in the Render application. We enter the **9** value.
- **Precision** - enter the precision value. In this example, we enter the 2 value.
- Select the following check boxes to adjust field layout:
  - **Read only**

- **Goods (js for total)**
  - **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add the data in numeric format.
  - **Format Type** - select the **Decimal - Dynamic** format type from the drop-down list.
  - **Max Length** - enter the max length for the data entered in the **Goods (js for total)** field in the Render application. We enter the **9** value.
  - Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Use to calculate totals**
  - **Precision** - enter the precision value. In this example, we enter the **2** value**.**
  - **Edit JavaScript -** click the button and define the JavaScript functionality to the **Goods (js for total)** column. For that, in the opened Custom JavaScript pop-up window, go to the **onFocusOut** tab in the **Custom javascript** tab and add the script:

```
var $goods = $(this);
var row = $goods.closest('tr');
var $service = row.find('input[name="Array.user_order.cost_of_service"]');
var total = row.find('input[name="Array.user_order.total_cost"]');

var g_cost = $goods.val() ? parseFloat($goods.val()) : parseFloat(0);
var s_cost = $service.val() ? parseFloat($service.val()) : parseFloat(0);

total.val(parseFloat(g_cost + s_cost));
```

- **Service (js for total)**
  - **Param Type** - select the **Numeric** type from the drop-down list. In the Render application, users will be able to add the data in numeric format.
  - **Format Type** - select the **Decimal - Dynamic** format type from the drop-down list.
  - **Max Length** - enter the max length for the data entered in the **Service (js for total)** field in the Render application. We enter the **9** value.
  - Select the following check boxes to adjust field layout:

- **Always show field border**
- **Use to calculate totals**

- **Precision** - enter the precision value. In this example, we enter the **2** value**.**
- **Edit JavaScript -** click the button and define the JavaScript functionality to the **Service (js for total)** column. For that, in the opened Custom JavaScript pop-up window, go to the **onFocusOut** tab in the **Custom javascript** tab and add the script:

```
var $service = $(this);
var row = $service.closest('tr');
var $goods = row.find('input[name="Array.user_order.cost_of_goods"]');
var total = row.find('input[name="Array.user_order.total_cost"]');

var g_cost = $goods.val() ? parseFloat($goods.val()) : parseFloat(0);
var s_cost = $service.val() ? parseFloat($service.val()) : parseFloat(0);

total.val(parseFloat(g_cost + s_cost));
```

Submit changes on the page.

21.4 Create Group Screen for Orders Screen

Now, we need to link the created Orders screen and Managers group and grant permissions for the actions on the screen.

In the Architect application, open Screen Configurations > Group Screens and create a new Group screen.

Set the specific parameters:

- Select the following check boxes:
  - **Edit**
  - **Show Edit button**
- **Group Name** - select the **Managers** group we have created previously.
- **Menu Name** - select the **Demo** menu we have created previously.
- **Screen Name** - select the **Orders** screen we have created previously.
- **Screen Text** - is a required field and we have to fill in it. But this text will not be displayed anywhere for the Login screen. Enter **Orders** here.
- **Weight** - let us set it to **80**.

Submit changes on the page.

21.5 Check the Alias Dependencies and Relationship

In the Architect application, open Alias Management > Alias Dependency for Database Aliases to manage dependencies for database aliases.

In the search area, to check if the alias dependencies are existing, type the **user_oder** alias name to the Alias Table.

In the Architect application, open Alias Management > Manage Alias Relationship to manage relationships for database aliases.

In the Manage Alias Relationship Overview page, check if the **user_order** child table has the relationships with the **user_info** parent table.

21.6 Check Orders Screen in Render

Let us show how the configured screen works in the Render application.

For that, open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned screen for Manager users (Jane Hanson and John Smith) is set up correctly, we will see the Orders menu and can access the Orders screen. In the Orders screen, we will be able to edit the data in the table, select data, add, update, and delete the data in the columns, and get total calculations for the table columns.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Environment for Orders Screen***.

22 Create Chart Screens

To demonstrate possible types of charts that can be used in the application, we will create 2 screens (Chart_1 and Chart_2) containing different chart types.

On the Chart screens we will demonstrate how to configure the following chart types accordingly:

- Chart_1 screen:
    - Chart-Pie Type
    - Chart-Doughnut
    - Chart-Line Type
    - Chart-Time-Series Type
- Chart_2 screen:
    - Chart-Bar-Horizontal
    - Chart-Bar-Vertical
    - Chart-Scatter-With-Linear-Regression Type (data will be sorted in ascending order for the field selected for the x-axis)
    - Chart-Scatter-With-Linear-Regression Type (to disable sorting, the values along the x-axis must be the same)

22.1 Create Screen: Chart_1

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** creation page of the screen builder, define the following data:

- **Select Existing Screen** - as the **Chart_1** screen will be accessed from the Field Types screen, we need to select the **FieldTypes** screen from the drop-down list.
- **Screen Description** - enter the **Chart_1** screen description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **20**.
- **Tab Text** - enter the **Chart_1** tab text.
- **Tab Description** - enter the **Chart_1** tab description.
- **Screen State** - select the **View** state from the drop-down list.
- In the **Screen Layout** section select the **Header 2x2** screen layout.

On the **second** creation page of the screen builder, we need to configure the **Header** and **Section** of the screen.

On the **Header** level add a section of the **Edit type** and the following fields:

On the Header, add the **User** field. Click the Field icon in the right top corner of the header section and define the following parameters in the opened pop-up window:

- **Data field** - activate the **Primary Table** check box and select the **Alias.user_info.user_info_id** from the drop-down list.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the user ID will be available to be selected from the users list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **User**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

Click the Add button to save the changes and close the modal window.

On the Section level we have 4 subsections according to the layout that we have selected on the first page of the Screen Builder.

On the Section level, we will configure the following chart types:

- **Chart-Pie Type**
- **Chart-Doughnut**
- **Chart-Line Type**
- **Chart-Time-Series Type**

Let us further configure these charts.

22.1.1 Chart-Pie Type

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Pie Type** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Pie Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWork: Universal DLL : GetAliasFrameWork** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to display data in the Chart-Pie type section. We add the following alias fields:

Alias Field	Column Label	Format Type
Alias.user_info.tax_percent	Tax	Numeric
Alias.user_info.tax_remission_percent	Remission	Numeric

Save the changes by clicking the Save button and close the pop-up window.

22.1.2 Chart-Doughnut Type

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Doughnut Type** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Doughnut Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWork: Universal DLL : GetAliasFrameWork** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Doughnut Type Section.

We add the following alias fields:

Alias Field	Column Label	Format Type
Alias.user_info.tax_percent	Tax	Numeric
Alias.user_info.tax_remission_percent	Remission	Numeric

Save the changes by clicking the Save button and close the pop-up window.

22.1.3 Chart-Line Type

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Line Type** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Line Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWorkMV: Universal DLL : GetAliasFrameWorkMV** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Line Type Section.

For the first table (Y-axis), we add the following alias fields:

Alias Type	Column Label	Format Type
Alias.user_order.cost_of_goods	Goods	Numeric
Alias.user_order.cost_of_service	Service	Numeric
Alias.user_order.total_cost	Total	Currency

For the 2st table (X-axis), we add the following alias fields:

Alias Type	Format Type
Alias.user_order.created_on	Date

Save the changes by clicking the Save button and close the pop-up window.

22.1.4 Chart-Time-Series Type

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Time-Series Type** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Time-Series Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWorkMV: Universal DLL : GetAliasFrameWorkMV** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Time-Series Type Section.

For the first table (Y-axis), we add the following alias fields:

Alias Type	Column Label	Format Type
Alias.user_order.cost_of_goods	Goods	Numeric
Alias.user_order.cost_of_service	Service	Numeric
Alias.user_order.total_cost	Total	Currency

For the first table (X-axis), we add the following alias fields:

Alias Type	Format Type
Alias.user_order.created_on	Text / Numeric / or leave it empty

As the Alias.user_order.created_on is of the Date type, we can decide not to select the Format type or use Text or Numeric values to avoid the double conversion to date format.

Save the changes by clicking the Save button and close the pop-up window.

You can adjust the layout of the fields on the screen according to your preferences.

After configuring the sections, save the changes by clicking the Update button at the bottom of the page.

22.2 Check Chart_1 Screen in Render

Let us show how the configured screen works in the Render application.

Open the Render application and use the Manager user credentials to log in to the Render application.

Click the Field Types menu item, you will see the Chart_1 tab on the Field Types screen, if the screen for the Manager users (Jane Hanson and John Smith) is set up correctly.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create and Check Chart_1 Screen in Render***.

22.3 Create Screen: Chart_2

In the Architect application, open the Screen Configurations > Screen Builder and create a new screen.

On the **first** page of the screen builder, define the following data:

- **Select Existing Screen** - as the **Chart_2** screen will be accessed from the Field Types screen, we need to select the **FieldTypes** screen from the drop-down list.
- **Screen Description** - enter the **Chart_1** screen description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **30**.
- **Tab Text** - enter the **Chart_2** tab text.
- **Tab Description** - enter the **Chart_2** tab description.
- **Screen State** - select the **View** state from the drop-down list.
- In the **Screen Layout** section select the **Header 2x2** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Section** of the screen.

On the **Header** level add a section of the **Edit type** and the following fields:

On the Header, add the **User** field. Click the Field icon in the right top corner of the header section and define the following parameters in the opened pop-up:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.

- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the user ID will be available to be selected from the users list.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User**.
- **Tooltip** - enter the text that will be displayed on UI as a tooltip, in our case **User**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

Click the Add button to save the changes and close the modal window.

On the Section level we have 4 subsections according to the layout that we have selected on the first page of the Screen Builder.

On the Section level, we will configure the following chart types:

- **Chart-Bar-Horizontal**
- **Chart-Bar-Vertical**
- **Chart-Scatter-With-Linear-Regression Type (data will be sorted in ascending order for the field selected for the x-axis)**
- **Chart-Scatter-With-Linear-Regression Type (to disable sorting, the values along the x-axis must be the same)**

Let us further configure these charts.

22.3.1 Chart-Bar-Horizontal Type

Open the Section Setting and set up the following parameters:

- **Layout label -** enter the subsection name. In our case, we enter the **Chart-Bar-Horizontal Type** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Bar-Horizontal Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWorkMV: Universal DLL : GetAliasFrameWorkMV** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Bar-Horizontal Type Section.

For the first table (Y-axis), we add the following alias fields:

Alias Field	Column Label	Format Type

Alias.user_order.cost_of_goods	Goods	Numeric
Alias.user_order.cost_of_service	Service	Numeric
Alias.user_order.total_cost	Total	Currency

For the 2st table (X-axis), we add the following alias fields:

Alias Field	**Format Type**
Alias.user_order.created_on	Date

Save the changes by clicking the Save button and close the pop-up window.

22.3.2 Chart-Bar-Vertical Type

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Bar-Vertical Type** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Bar-Vertical Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWorkMV: Universal DLL : GetAliasFrameWorkMV** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Bar-Vertical Type Section. We add the following alias fields:

For the first table (Y-axis), we add the following alias fields:

Alias Field	**Column Label**	**Format Type**
Alias.user_order.cost_of_goods	Goods	Numeric
Alias.user_order.cost_of_service	Service	Numeric
Alias.user_order.total_cost	Total	Currency

For the 2st table (X-axis), we add the following alias fields:

Alias Field	**Format Type**

Alias.user_order.created_on	Date

Save the changes by clicking the Save button and close the pop-up window.

22.3.3 Chart-Scatter-With-Linear-Regression Type (sort by x-axis)

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Scatter-With-Linear-Regression Type (data will be sorted in ascending order for the field selected for the x-axis)** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Scatter-With-Linear-Regression Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWorkMV: Universal DLL : GetAliasFrameWorkMV** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Scatter-With-Linear-Regression Type Section. We add the following alias fields:

For the first table (Y-axis), we add the following alias fields:

Alias Field	**Column Label**	**Format Type**
Alias.user_order.cost_of_goods	Goods	Numeric
Alias.user_order.cost_of_service	Service	Numeric
Alias.user_order.total_cost	Total	Currency

For the 2st table (X-axis), we add the following alias fields:

Alias Field	**Format Type**
Alias.user_order.created_on	Date

Save the changes by clicking the Save button and close the pop-up window.

22.3.4 Chart-Scatter-With-Linear-Regression Type (without sort by x-axis)

Open the Section Setting and set up the following parameters:

- **Layout label** - enter the subsection name. In our case, we enter the **Chart-Scatter-With-Linear-Regression Type (to disable sorting, the values along the x-axis must be the same)** value.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Chart-Scatter-With-Linear-Regression Type** from the drop-down list.
- **Function Name for Getting List** - select the function from the drop-down list. In this example, we select **GetAliasFrameWorkMV: Universal DLL : GetAliasFrameWorkMV** function from the drop-down list.
- **Add alias field** - type the alias field name and click the **Add alias field** button to add the alias field to the Chart-Scatter-With-Linear-Regression Type Section. We add the following alias fields:

For the first table (Y-axis), we add the following alias fields:

Alias Field	Column Label	Format Type
Alias.user_order.cost_of_goods	Goods	Numeric
Alias.user_order.cost_of_service	Service	Numeric
Alias.user_order.total_cost	Total	Currency

For the 2st table (X-axis), we add the following alias fields:

Alias Field	Format Type
Alias.user_order.user_info_id	Numeric

Save the changes by clicking the Save button and close the pop-up window.

22.4 Check Chart_2 Screen in Render

Let us show how the configured screen works in the Render application.

Open the Render application and use the Manager user credentials to log in to the Render application.

If the above-mentioned screen for Manager users (Jane Hanson and John Smith) is set up correctly, we will see the Chart_2 screen displayed in a separate tab.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create and Check Chart_2 Screen in Render***.

23 Part of Form Functionality

In scope of the current instruction, we also would like to demonstrate the possibility to build the screen in the way that the changes of any screen data (including the modal window data) are applied and saved by clicking the Save button for the whole screen in Render. In comparison to the regular case, when the changes on data from the modal windows must be saved additionally in the modal window, the Part of Form functionality can be very useful. We will demonstrate it in the use case in scope of the Onboarding process.

To set up the Part of Form feature correctly and demonstrate the use case, we should perform the following steps:

1. Create a new screen (Part of Form) where the use case will be demonstrated
2. Create separate section for modal windows within this screen
3. Set up these sections to display the section data in a modal window
4. Create a Group Screen
5. At the end we will show the result in the Render

23.1 Create Screen: Part of Form

The Part of Form screen will be used to show the possibilities of processing the data in the modal windows in the Render application.

Open Screen Configurations > Screens and click the **Create with builder** button.

23.1.1 General Screen Data

On the first page of the screen builder, we need to specify the data describing the screen and configure the screen-related properties.

For that, define the following data:

- **Add New Screen** - enter the **PartofForm** screen name.
- **Screen Description** -let's repeat the **s**creen name (**PartofForm**) for the description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the Simple Search and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.

- **Tab Text** - enter the **PartofForm** tab text.
- **Tab Description** - enter the **PartofForm** tab description.

- **Screen State** - select the **View** screen state.
- In the **Screen Layout** section select the **Header 1x1** screen layout.

On the **second** page of the screen builder, we need to configure the **Header** and **Sections** of the screen.

23.1.2 Header level

On the **Header** level add a section of the **Edit type**. In this section, we will add the following fields:

- **User** - for the user ID
- **Name** - for the first name
- **Surname** - for the second name

Select the Field icon to add the **User** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User**.
- **Tooltip** - you can enter the text that will be displayed on UI as a tooltip for this field, in our case **User**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list to be able further to get the list of all users.

Select the Field icon to add the **Name** field and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**
- **Label** - enter the text that will be displayed on UI for this field, in our case **Name**.
- **Tooltip** - you can enter the text that will be displayed on UI as a tooltip for this field, in our case **Name** or **First Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Click the **Add** button to save the field.

Add the next field for the **Second Name** and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**

- **Label** - enter the text that will be displayed on UI for this field, in our case **Surname**.
- **Tooltip** - you can enter the text that will be displayed on UI as a tooltip for this field, in our case **Surname** or **Second Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Click the **Add** button to save the field.

23.1.3 Section level

On the **Section** level, click the Settings icon set up the section:

- **Layout Label** - enter the **Main Section** value.
- **Hide section header** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

Save the section settings.

Add the **Job Title** field. For that, set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.job_title** value.
- **Label** - enter the text that will be displayed on UI for this field, in our case **Job Title**.
- **Type** - select the **Relation Datalist**.
- **Custom Query** - select the **Get–JobTitle-list-Get-JobTitle-list** from the List Params pop-up window.
- **Always show field border** - select the following check boxes to adjust field layout:

Close the Setup pop-up window.

23.2 Create sections for modal window

To demo the use case, we need to add two sections, click twice on the Plus icon next to the Default section.

We will set up these sections to display the User’s Contacts and Personal Data.

In first of the appeared sections (Contact information) open the section settings and define the following parameters:

- **Layout Label** - according to our use case, enter **Contacts** value.
- **Hide section pop-up** - select the check box

- **Use as part of form** - select this check box to specify that the data on this section will be considered as part of the whole screen.
- **Section identifier** - this field appears when you select the **Use as part of form** check box. Define here the identifier of the section. You can enter any value but note that the value must be unique. We enter the **user_contacts** value for the case.

Save the data and close the pop-up window.

Add the following fields to the **Contacts** section:

- **Phone**

Define the parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.phone** value.
- **Label** - enter the text that will be displayed as a field name on UI, in our case **Phone**.
- **Tooltip** - enter the text that will be displayed as a tooltip for this field on UI, in our case **Phone**.
- **Type** - select the **Text**.
- **Format Type** - select the **Phone**.
- **Always show field border** - select the check box to display the field border.

Save the changes for the field.

- **Email**

Define the parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.email** value.
- **Label** - enter the text that will be displayed as a field name on UI, in our case **Email**.
- **Tooltip** - enter the text that will be displayed as a tooltip for this field on UI, in our case **Email**.
- **Type** - select the **Text** type.
- **Format Type** - select the **Email**.
- **Always show field border** - select the check box to display the field border.

Save the changes for the field.

The next section will contain Personal Data of the Users. Set up the following section parameters:

- **Layout Label** - according to our use case, enter **Personal Data** value.
- **Hide section pop-up** - select the check box
- **Use as part of form** - select this check box to specify that the data on this section will be considered as part of the whole screen.

- **Section identifier** - this field appears when you select the **Use as part of form** check box. Define here the identifier of the section. You can enter any value but note that the value must be unique. We enter the **personal_data** value for the case.

Close the Section Settings pop-up.

Add the following fields to the section:

- **Birthday**

Define the parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.birthday** value.
- **Label** - enter the text that will be displayed as a field name on UI, in our case **Birthday**.
- **Tooltip** - enter the text that will be displayed as a tooltip for this field on UI, in our case **Birthday**.
- **Type** - select the **Text**
- **Format Type** - select the **Date**.
- **Always show field border** - select the check box to display the field border.

Save the changes for the field.

- **Identifier**

Define the parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.identify** value.
- **Label** - enter the text that will be displayed as a field name on UI, in our case **Identifier**.
- **Tooltip** - enter the text that will be displayed as a tooltip for this field on UI, in our case **Identifier**.
- **Type** - select the **Text** type.
- **Format Type** - select the **Custom**.
- **Format Template** - enter the following template we use: **xnn-nnn-nnn**
- **Always show field border** - select the check box to display the field border.

Save the changes for the field. Adjust the layout of the fields on the section.

23.3 Set up sections to display in a modal window

So, we have already set up the Main section and the Contacts and Personal Data sections that can be opened from the Main section. Now we will add the possibility to open the Contacts and Personal Data sections by clicking the corresponding buttons on the Main section.

On the Main section, add the following buttons:

- **Contacts** - to open the Contacts section
- **Personal Data** - to open the Personal Data section.

For the **Contacts** button define the following parameters:

- **Value** - enter the button name, in our case **Contacts**.
- **Button Action** - select the **Part of Form** action.
- **Section Identifier** - enter here the value you previously set up in the Section Settings: **user_contacts**

Click the **Add** button to save the parameters and to close the pop-up window.

For the **Personal Data** button define the following parameters:

- **Value** - enter the button name, in our case **Personal Data**.
- **Button Action** - select the **Part of Form** action.
- **Section Identifier** - enter here the value you previously set up in the Section Settings: **personal_data**

Click the **Add** button to save the parameters and to close the pop-up window.

23.4 Create Group Screen

To add required permissions to the Part of Form screen so that the users of the Managers user group can have access and execute actions on the screen.

Open **Screen Configurations > Group Screens** and create a new Group Screen.

Set up the following parameters:

- **Edit** - select the check box to grant users **Edit** permissions on the screen.
- **Show Edit Button** - select the check box to make the **Edit** button visible on the screen.
- **Group Name** - select the **Admin_Group** name.
- **Menu Name** - select the **Demo** option.
- **Screen Name** - select the **PartOfForm** screen.
- **Screen Text** - enter the **Part Of Form** value.
- **Weight** - enter the **0** value.

Save the changes.

23.5 Check Part of Form Feature in Render

Let us show how the configured feature works in the Render application.

Open the Render application using the Manager user credentials to log in to the Render application.

Select the Part of Form menu item. Input any user name into the search field on the opened screen to open the details. Click the Edit button to open the screen in the Edit mode.

Click the Contacts button. In the opened pop-up window, change the value of any field (Phone or Email).

Save the screen. If the settings are set up correctly, the changes will be considered as a part of the whole screen and will be saved.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create Screen with Part_of_Form Sections***.

24 API Access from Screen

There is a possibility to display the data on the screens in the Render that are received not only from the AppMill service, but also via API from the external third-party services and applications. This feature is available for the lists type of data (the List for API field type).

Currently, the external data can be displayed on the screens of the **Edit** and **Table types**. We will see both cases. But before the use case demonstration, we need to make necessary presetting.

24.1 Create Guest Users

We recommend you to create a separate user (User Type = G: Guest User Account) to set the list for work with AppMill API. A guest user can have multiple connections to the service, which will allow several users to receive the list at the same time.

To create a Guest User Account, open Account Settings > Users.

24.1.1 Create User 1 (GuestUser)

As we have users in the system, we can use the data already set and just copy the existing user account. Let’s copy the user account of John Smith.

In the opened editor, fill in the following data:

- **User Name** - enter the **GuestUser** value.
- **Password** - enter the password for the account. Pay attention to the identifier determining the password strength.
- **Account Type** - select **Interal_Type** value.
- **Tenant Code** - select **1:Default** value.
- **User Type** - select **G: Guest User Account** value.
- **Primary Group** - select **Managers** value.
  Please be aware that for security reasons, the guest user should not belong to the Admin_Group group.

- **Group Area** - select the **Managers** value.
- **Document Group** - clear the field and keep it **empty**.
- **Account Name** - enter a unique account name - **GuestUser acc**.

Save the changes by clicking the Submit button.

24.1.2 Create User 2 (GuestSystemUser)

To create the next Guest user, let us copy the created Guest User. In the User Overview list select the Guest User and copy.

In the opened editor, fill in the following data:

- **User Name** - enter the **GuestSystemUser** value.
- **Password** - enter the password for the account. Pay attention to the identifier determining the password strength.
- **Account Type** - select **Interal_Type** value.
- **Tenant Code** - select **1:Default** value.
- **User Type** - select **G: Guest User Account** value.
- **Primary Group** - select **Managers** value.
  Please be aware that for security reasons, the guest user should not belong to the Admin_Group group.
- **Group Area** - select the **Managers** value.
- **Document Group** - clear the field and keep it **empty**.
- **Account Name** - enter a unique account name - **GuestSystemUser acc**.

Save the changes by clicking the Submit button.

24.2 Create Security Filters

Open Account Settings > Security Filters and create a new entry for Security Filter.

Define the following parameters:

- **Tenant** - select the **1:Default** value.
- **Account Type** - select the **Internal_User.UserType** value.
- **User Type** - select the **G: Guest User Account** value.
- **Description** - enter the **Guest** value.
- **Authorization Type** - select the **L: Login Authentication** option.

Save the changes by clicking the Submit button.

24.3 Create Screen: GuestApi (without user authorization)

The next step is to create and configure a screen for the API access without the user authorization.

Open **Screen Configurations > Screens**. To create a new screen, we can create it from scratch (via Create with Builder option) or copy any existing screen and then edit it. Let us save the time and make a copy from an existing screen (e.g., TablesTypes screen).

24.3.1 Copy Screen

On the Screens page, find the **TablesTypes** screen and click the Copy icon for the screen entry.

On the screen creation page, we need to leave appropriate data and edit the following parameter:

- **Add New Screen** - add a screen name: **GuestApi**
- **Screen Description** - add a short description: **GuestApi**
- **Tab Text** - enter a name: **GuestApi.** It will be displayed in the menu in the Render.
- **Tab Description** - add a description for the screen: **GuestApi**

Click the **Submit** button to save the changes and create a screen.

24.3.2 Edit GuestApi Screen

On the Screens page, find a newly created screen (GuestApi) and click the Update with Builder icon.

On the second page of the screen builder, pay your attention to the following parameters:

- **Pass_thru** - make sure that the check box is selected. If the Pass_thru check box is enabled, the screen will be available for API access.
- **Pass Thru Guest User** - select the check box.
  The drop-down list containing the Guest Users will be available. Select the **GuestSystemUser acc** user.

Further, we can edit the following fields according to our requirements:

- **Weight** - change the value to 0.
- **Simple Search** - activate the check box for the simple search and select the **Search_user_info: Universal DLL : Search_user_info** search option.
  In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.

Go to the second screen builder page and save the changes by clicking the Update button.

24.4 Create Group Screen

To add required permissions to the **GuestApi** screen so that the users of the Managers user group can have access and execute actions on the screen.

Open **Screen Configurations > Group Screens** and create a new Group Screen.

Set up the following parameters:

- **Inquire** - select the check box to grant users **Inquire** permissions on the screen.
- **Group Name** - select the **Managers** name.
- **Menu Name** - select the **Demo** option.
- **Screen Name** - select the **GuestApi** screen.
- **Screen Text** - enter the **GuestApi** value.
- **Weight** - enter the **99** value.

Save the changes.

24.5 Check GuestApi screen (without user authorization)

To check the use case we set up, you need to use an API platform that provides the possibility to design, build, test, and collaborate on APIs. For testing the use case, we will use the Postman application.

Pay attention to the following parameters:

- **screen_name** - use the **GuestApi** value. It is taken from the Select Existing Screen field of the first Screen builder page.
- **screen_tab_name** - use the **GuestApi** value. It is taken from the Tab Name field of the first Screen builder page.
- **user_info_id** - use the value: 6

24.6 Create Screen: UserApi (with user authorization)

The next step is to create and configure a screen for the API access using the user authorization.

Open **Screen Configurations > Screens**. To create a new screen, we can create it from scratch (via Create with Builder option) or copy any existing screen and then edit it. Let us save the time and make a copy from a newly created screen (GuestApi).

24.6.1 Create UserApi Screen from GuestApi Screen Copy

On the Screens page, find the GuestApi screen and click the Copy icon for the screen entry.

On the screen creation page, we need to leave appropriate data and edit the following parameter:

- **Add New Screen** - add a screen name: **UserApi**
- **Screen Description** - add a short description: **UserApi**
- **Pass_thru** - make sure that the check box is selected. If the Pass_thru check box is enabled, the screen will be available for API access.

- **Pass Thru Guest User** - deselect the check box.
- **Tab Text** - enter a name: **UserApi**. It will be displayed in the menu in the Render.
- **Tab Description** - add a description for the screen: **UserApi**

Click the **Submit** button to save the changes and create a screen.

24.6.2 Create UserApi Group Screen

To add required permissions to the **UserApi** screen so that the users of the Managers user group can have access and execute actions on the screen.

Open **Screen Configurations** > **Group Screens** and create a new Group Screen.

Set up the following parameters:

- **Inquire** - select the check box to grant users **Inquire** permissions on the screen.
- **Group Name** - select the **Managers** name.
- **Menu Name** - select the **Demo** option.
- **Screen Name** - select the **UserApi** screen.
- **Screen Text** - enter the **UserApi** value.
- **Weight** - enter the **99** value.

Save the changes.

24.7 Check UserApi screen (with user authorization)

As for the first use case, we will use the Postman application for testing purposes.

Pay attention to the following parameters:

- **screen_name** - use the **UserApi** value. It is taken from the Select Existing Screen field of the first Screen builder page.
- **screen_tab_name** - use the **UserApi** value. It is taken from the Tab Name field of the first Screen builder page.
- **user_info_id** - use the value: 6

To watch the video instructions for the above-described settings and configurations, follow the link: ***Create and test Inquire request for AppMill API***.

25 Screen (Edit type) with Fields Filled in via API

The possibility to display the values on the screens in the Render that are received from the external services and applications is available for the List type of data (the List for API field type) for the screen of Edit type.

Let us see the demonstration of this feature for the Edit type screen.

25.1 Create Basic Screen for API usage demonstration

Open Screen Configurations > Screens and click the **Create with builder** button to create a new screen.

25.1.1 General Screen Data

On the first page of the screen builder, specify the basic screen data and configure the screen properties.

For that, define the following data:

- **Add New Screen** - enter the **ApiLists** screen name.
- **Screen Description** - use the **ApiLists** screen name for the description.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Tab Weight** - enter **0**.
- **Simple Search** - activate the Simple Search and select **Search_user_info: Universal DLL: Search_user_info** from the drop-down list.

In the Simple Search Configuration, select **user_info_id** for the primary key and **first_name** and **second_name** for search parameters.

- **Tab Text** - enter the **ApiLists** tab text.
- **Tab Description** - enter the **ApiLists** tab description.
- **Screen State** - select the **View** screen state.
- **Screen Layout** - select the **Header 2x1** screen layout in the section.

On the **second** page of the screen builder, configure the **Header** and **Sections** of the screen.

25.1.2 Header level

On the **Header** level add a section of the **Edit type**. In this section, we will add the following fields:

- **User** - for the user ID
- **Name** - for the first name
- **Surname** - for the second name

Add the following fields by selecting the Field icon:

- **User**
- **Name**
- **Surname**

Set up the following parameters for the **User** field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type.
- **Label** - enter the text that will be displayed on UI for this field, in our case **User**.
- **Tooltip** - you can enter the text that will be displayed on UI as a tooltip for this field, in our case **User**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list to be able further to get the list of all users.

Close the modal window to save the current field and add the next one.

Set up the following parameters for the **Name** field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.first_name**
- **Label** - enter the text that will be displayed on UI for this field, in our case **Name**.
- **Type** - select the **Text** type.
- **Tooltip** - you can enter the text that will be displayed on UI as a tooltip for this field, in our case **Name** or **First Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Click the **Add** button to save the field.

Add the next field for the **Surname** and set up the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.second_name**
- **Label** - enter the text that will be displayed on UI for this field, in our case **Surname**.
- **Tooltip** - you can enter the text that will be displayed on UI as a tooltip for this field, in our case **Surname** or **Second Name**.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only**

Click the **Add** button to save the field.

25.1.3 Section level

On the **Section** level, we have two sections.

For the first section, click the Settings icon and set up the following parameters:

- **Hide section header** - select the check box.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

Save the section settings.

Set up the same parameters for the first section:

- **Hide section header** - select the check box.
- **Hide section pop-out** - select the check box.
- **Section Type** - select the **Edit Type** from the drop-down list.

Save the section settings.

Save the screen by clicking the Create button.

25.2 Create Group Screen

To add required permissions to the **ApiLists** screen so that the users of the Managers user group can have access and execute required actions on the screen.

Open **Screen Configurations** > **Group Screens** and create a new Group Screen.

Set up the following parameters:

- **Edit** - select the check box to grant users **Edit** permissions on the screen.
- **Show Edit Button** - select the check box to make the **Edit** button visible on the screen.
- **Group Name** - select the **Admin_Group** name.
- **Menu Name** - select the **Demo** option.
- **Screen Name** - select the **ApiLists** screen.
- **Screen Text** - enter the **ApiLists** value.
- **Weight** - enter the **0** value.

Save the changes.

25.3 Check ApiLists Screen in Render

Let us check how the configured feature works in the Render application.

Open the Render application using the Manager user credentials to log in to the Render application.

Select the **ApiLists** menu item. Input any username of the Managers group (e.g. Jane Hanson) into the search field on the screen.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Create basic screen for lists filled in via API***.

25.4 Configure Edit Type Section

In the scope of this description, we configure the **Edit** type section of the **ApiLists** screen to fill in the list fields with the values received from the external systems.

To set up this behavior, we will perform the following steps:

1. Add demo data to the user_info database table
2. Add the fields to be filled in using the REST API
3. Add the fields to be filled in using the AppMill API

25.5 Add Demo Data to the user_info Table

To demonstrate the use case and have enough demo data, we need to add some additional fields to the **user_info** table.

In the Architect application, open **Features** > **Tables**, find the **user_info** table and add the following fields:

Name	Type	Nullable
t1	Varchar	True
t2	Varchar	True
t3	Varchar	True
t4	Varchar	True
api_list_1	Varchar	True
api_list_2	Varchar	True
api_list_3	Varchar	True

api_list_4	Varchar	True
api_list_5	Varchar	True
api_list_6	Varchar	True

Save the changes by clicking the Submit button.

25.6 Add fields to be filled in via API

The next step is to add the list fields to the screen using both **REST API** and **AppMill API** types.

Open **Screen Configurations > Screens**, find the **ApiLists** screen and open with the **Update with builder** option.

25.6.1 Add fields to be filled in via REST API

On the second page of the screen builder, we will add the following data to the **first section** on the section level:

- Label: **REST without params**
- Fields: **List 1, List 2, list, list 3, datalist, List 4**

For the label, define only the Label name (**REST without params**).

To demonstrate more possibilities of the field configuration, we will add the dependencies between the fields. E.g. the values displayed in the List 2 will depend on the value selected in the List 1 field, the field list 3 will depend on the list field, the List 4 will depend on the datalist field, the Guest List will depend on the test 3 field and the User List will depend on the test 4 field.

The setting up of the fields are provided further.

Add the **List 1** field with the following parameters:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.api_list_1**
- **Label** - enter the text that will be displayed on UI for this field, in our case **List 1**.
- **Type (data type)** - select the **List from API** type.
- **Always show field border** - select the check box to adjust the field layout.
- **Type (API type)** - select the REST API option.
  Specify the appeared additional fields as follows:
  - **URL to REST LIST** - enter the URL to the REST API. In our case: https://appmill.com/rest_api/objects
  - **Use for Option Value** - enter the parameter that will be transferred: **id**

- **Use for Option Description -** enter the parameter that will be transferred as value for the field: **name**

Save the field by clicking the **Add** button.

Add the **List 2** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.api_list_2**
- **Label** - enter the text that will be displayed on UI for this field, in our case **List 2**.
- **Label Orientation** - select the **Top** value.
- **Type (data type)** - select the **List from API** type.
- **Num Rows** - enter the quantity, e.g. **5**
- **Always show field border -** select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
  - **URL to REST LIST** - enter the URL to the REST API. In our case: https://appmill.com/rest_api/objects
  - **Use for Option Value** - enter the parameter that will be transferred: **id**
  - **Use for Option Description -** enter the parameter that will be transferred as value for the field: **data->color**

Save the field by clicking the **Add** button.

Add the **list** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.t1**
- **Label** - enter the text that will be displayed on UI for this field: **list**.
- **Type (data type)** - select the **Multi-select List** type.
  Click on the Pull up button and select the parameter to use: **FUNDS**.
- **Num Rows** - enter the quantity, e.g. **5**
- **Always show field border -** select the check box to adjust the field layout.

Save the field by clicking the **Add** button.

Add the **list 3** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.api_list_3**
- **Label** - enter the text that will be displayed on UI for this field, in our case **list 3**.
- **Label Orientation** - select the **Top** value.

- **Type (data type)** - select the **List from API** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **5**
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
    - **URL to REST LIST** - enter the URL to the REST API. In our case: https://appmill.com/rest_api/objects
    - **Parameter Field ID** - enter the ID of the field configured for the Primary Key on the screen: **GetAliasFramework--Alias-user_info-t1**
      You can check and copy the Field ID value displayed in the corresponding **list** Field Settings modal window in the Screen Builder.
    - **Use for Option Value** - enter the parameter that will be transferred: **id**
    - **Use for Option Description -** enter the parameter that will be transferred as value for the field: **name**

Save the field by clicking the **Add** button.

Add the **datalist** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.t2**
- **Label** - enter the text that will be displayed on UI for this field: **datalist**.
- **Type (data type)** - select the **Datalist** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **5**
- **Values (Use the Delimiter ";")** - enter the values: 1;2;3;4;5;6;7;8;9
- **Field ID** - pay attention to the Field ID displayed in the lower left corner of the pop-up window: **GetAliasFramework--Alias-user_info-t2**. This value will be used further.

Save the field by clicking the **Add** button.

Add the **List 4** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.api_list_4**
- **Label** - enter the text that will be displayed on UI for this field, in our case **List 4**.
- **Label Orientation** - select the **Top** value.

- **Type (data type)** - select the **List from API** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **6**
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
  - **URL to REST LIST** - enter the URL to the REST API. In our case: https://appmill.com/rest_api/objects
  - **Parameter Field ID** - enter the ID of the field configured for the Primary Key on the screen: **GetAliasFramework--Alias-user_info-t2**
    You can copy the Field ID value displayed in the corresponding **datalist** Field Settings modal window in the Screen Builder.
  - **Use for Option Value** - enter the parameter that will be transferred: **id**
  - **Use for Option Description -** enter the parameter that will be transferred as value for the field: **name**

Save the field by clicking the **Add** button.

25.6.2 Add fields to be filled in via AppMill API

On the second page of the screen builder, we will add examples on how to set up fields using two types of AppMill API:

- basic AppMill API
- Current Service

The settings of the fields are provided further.

On the second page of the **ApiLists** screen builder, we will add the following data to the **first section** on the section level:

- Label: **AppMill API**
- Fields: **test 3, Guest List (AppMill API), test 4, User List (Current Service)**

For the label, define only the Label name (**AppMill API**).

The settings of the fields are provided further.

Add the **test 3** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.t3**
- **Label** - enter the text that will be displayed on UI for this field, in our case **test 3**.
- **Type (data type)** - select the **Relation Datalist** type.

- **Always show field border** - select the check box to adjust the field layout.
- **Custom Query** - select the custom query: **Get-All-Users - Get-All-Users**
- **Field ID** - pay attention to the Field ID displayed in the lower left corner of the pop-up window: **GetAliasFramework--Alias-user_info-t3**. This value will be used further in the **Guest List** field.

Save the field by clicking the **Add** button.

Add the **Guest List** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.api_list_5**
- **Label** - enter the text that will be displayed on UI for this field, in our case **Guest List**.
- **Label Orientation** - select the **Top** value.
- **Type (data type)** - select the **List from API** type.
- **Always show field border** - select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **6**
- **Type (API type)** - select the **AppMill API** option.
  Specify the appeared additional fields as follows:
  - **URL to REST LIST** - enter the URL to the REST API. In our case: **localhost:34567**
  - **Screen Name** - enter the value: **GuestApi**
  - **Screen Tab Name** - enter the value: **GuestApi**
  - **Parameter Name** - enter the value: **user_info_id**
  - **Parameter Field ID** - enter the ID of the field configured for the Primary Key on the screen: **GetAliasFramework--Alias-user_info-t3**
    You can copy the Field ID value displayed in the corresponding **test 3** Field Settings modal window in the Screen Builder.
  - **Result Block** - enter the parameters that will be displayed as a result: **Array.equipment**
  - **Alias for Value** - enter the parameter that will be transferred: **Array.equipment.type**
  - **Alias for Description** - enter the parameter that will be transferred as value for the field: **Array.equipment.brand**

Save the field by clicking the **Add** button.

Add the **test 4** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.t4**
- **Label** - enter the text that will be displayed on UI for this field, in our case **test 4**.

- **Type (data type)** - select the **Relation Datalist** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Custom Query** - select the custom query: **Get-All-Users - Get-All-Users**
- **Field ID** - pay attention to the Field ID displayed in the lower left corner of the pop-up window: **GetAliasFramework--Alias-user_info-t4**. This value will be used further in the **User List** field.

Save the field by clicking the **Add** button.

Add the **User List** field with the following parameters:

- **Data Field** - select the **Primary Table** check box and select the **Alias.user_info.api_list_6**
- **Label** - enter the text that will be displayed on UI for this field, in our case **User List**.
- **Label Orientation** - select the **Top** value.
- **Type (data type)** - select the **List from API** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **6**
- **Type (API type)** - select the **Current Service** option.
  Specify the appeared additional fields as follows:
  - **Username for Service** - enter the Username that is defined for the Service: **GuestUser**
  - **Password for Service** - enter the Password that is defined for the Service.
  - **URL to REST LIST** - enter the URL to the REST API. In our case: **localhost:34567**
  - **Screen Name** - enter the value: **UserApi**
  - **Screen Tab Name** - enter the value: **UserApi**
  - **Parameter Name** - enter the value: **user_info_id**
  - **Parameter Field ID** - enter the ID of the field configured for the Primary Key on the screen: **GetAliasFramework--Alias-user_info-t4**
    You can copy the Field ID value displayed in the corresponding **test 4** Field Settings modal window in the Screen Builder.
  - **Result Block -** enter the parameters that will be displayed as a result: **Array.equipment**
  - **Alias for Value** - enter the parameter that will be transferred: **Array.equipment.type**
  - **Alias for Description -** enter the parameter that will be transferred as value for the field: **Array.equipment.brand**

Save the field by clicking the **Add** button.

Save the changes on the screen by clicking the **Update** button.

25.7 Check Screen in Render

Let us check how the configured feature works in the Render application.

Open the Render application using the Manager user credentials to log in to the Render application.

Select the **ApiLists** menu item. Input any username of the Managers group (e.g. Jane Hanson) into the search field on the screen. You will see all the fields we set up above. Open the screen in the Edit mode by clicking the Edit icon. All the values are received from the external service via API. You can also check the dependencies between the fields.

To watch the video instructions for the above-mentioned settings and how to check them in the Render, follow the link: ***Configure the fields filled in via API for the Edit Type section***.

26 Screen (Table type) with Fields Filled in via API

The possibility to display the values on the screens in the Render that are received from the external services and applications is available for the List type of data (the List for API field type) for the screen of Edit type.

Let us see the demonstration of this feature for the Table type screen.

26.1 Add Demo Data to the equipment Table

To demonstrate the use case for the Table type and have enough demo data, we need to add additional fields to the **equipment** table.

In the Architect application, open **Features > Tables**, find the **equipment** table and add the following fields:

Name	Type	Nullable
t1	Varchar	True
t2	Varchar	True
t3	Varchar	True
t4	Varchar	True
api_list_1	Varchar	True
api_list_2	Varchar	True

api_list_3	Varchar	True
api_list_4	Varchar	True
api_list_5	Varchar	True
api_list_6	Varchar	True

Save the changes by clicking the Submit button.

26.2 Add fields to be filled in via API

The next step is to add the list fields to the screen using both **REST API** and **AppMill API** types.

Open **Screen Configurations > Screens**, find the **ApiLists** screen and open with the **Update with builder** option.

26.2.1 Configure Table Type Section

In the scope of this description, we configure the **Table** type section of the **ApiLists** screen to fill in the list fields with the values received from the external systems.

On the second page of the **ApiLists** screen builder, open settings of the **second section** on the section level. Set up the following parameters:

- **Section Type** - select the **Table type**.
- **Alias Framework PK Configuration**:
    - **Key Part** - select the value from the drop-down: **Alias.user_infor.user_info_id**
- **Label Orientation** - select the Top option.
- **Pixel Height** - enter the **500** value.

Add the following data to the table block:

alias_field	**column_label**
Array.equipment.type	type
Array.equipment.brand	brand
Array.equipment.api_list_1	api_list_1
Array.equipment.api_list_2	api_list_2

Array.equipment.t1	t1
Array.equipment.api_list_3	api_list_3
Array.equipment.t2	t2
Array.equipment.api_list_4	api_list_4
Array.equipment.t3	t3
Array.equipment.api_list_5	api_list_5
Array.equipment.t4	t4
Array.equipment.api_list_6	api_list_6

Save the changes and close the modal window.

26.2.2 Configure Columns

The next step is to configure the table columns by clicking the corresponding Gear icon. Configure the following columns:

- **Type** - define the following parameters:
  - **Param Type** - select the Text option.
- **Brand** - define the following parameters:
  - **Param Type** - select the Text option.
  - **Always show field border** - select the check box.

26.2.3 Configure Fields to be Filled in via REST API

Then we need to configure the fields that will be filled with the values from the external system using REST API. Click the corresponding Gear icon to configure the columns and fields.

To demonstrate more possibilities of the field configuration, we will add the dependencies between the fields. E.g. the values displayed in the api_list_3 will depend on the value selected in the t1 field, the field api_list_4 will depend on the t2 field, the api_list_ 5 will depend on the t3 field and the api_list_6 will depend on the t4 field.

The setting up of the fields are provided further.

For the **api_list_1** field, configure the following parameters:

- **Param Type (data type)** - select the **List from API** type.

- **Always show field border** - select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **5**
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
    - **URL to REST LIST** - enter the URL to the REST API. In our case: **https://appmill.com/rest_api/objects**
    - **Use for Option Value** - enter the parameter that will be transferred: **id**
    - **Use for Option Description** - enter the parameter that will be transferred as value for the field: **name**

Save the field by clicking the **Save** button.

For the **api_list_2** field, configure the following parameters:

- **Param Type (data type)** - select the **List from API** type.
- **Always show field border** - select the check box to adjust the field layout.
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
    - **URL to REST LIST** - enter the URL to the REST API. In our case: https://appmill.com/rest_api/objects
    - **Use for Option Value** - enter the parameter that will be transferred: **id**
    - **Use for Option Description** - enter the parameter that will be transferred as value for the field: **data->color**

Save the field by clicking the **Save** button.

For the **t1** field, configure the following parameters:

- **Param Type (data type)** - select the **Multi-select List** type.
  Click on the Pull up button and select the parameter to use: **FUNDS**.
- **Num Rows** - enter the quantity, e.g. **5**
- **Always show field border** - select the check box to adjust the field layout.

Save the field by clicking the **Save** button.

For the **api_list_3** field, configure the following parameters:

- **Param Type (data type)** - select the **List from API** type.
- **Always show field border** - select the check box to adjust the field layout.

- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **5**
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
    - **URL to REST LIST** - enter the URL to the **REST API**. In our case: https://appmill.com/rest_api/objects
    - **Alias for Parameter** - enter the alias for a field that is linked with the field:: **Array.equipment.t1**
    - **Use for Option Value** - enter the parameter that will be transferred: **id**
    - **Use for Option Description** - enter the parameter that will be transferred as value for the field: **name**

Save the field by clicking the **Save** button.

For the **t2** field, configure the following parameters:

- **Param Type (data type)** - select the **Datalist** type.
- **Always show field border** - select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Values (Use the Delimiter ";")** - enter the values: 1;2;3;4;5;6;7;8;9

Save the field by clicking the **Save** button.

For the **api_list_4** field, configure the following parameters:

- **Param Type (data type)** - select the **List from API** type.
- **Always show field border** - select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Type (API type)** - select the **REST API** option.
  Specify the appeared additional fields as follows:
    - **URL to REST LIST** - enter the URL to the REST API. In our case: https://appmill.com/rest_api/objects
    - **Alias for Parameter** - enter the alias for a field that is linked with the field: **Array.equipment.t2**
    - **Use for Option Value** - enter the parameter that will be transferred: **id**
    - **Use for Option Description** - enter the parameter that will be transferred as value for the field: **name**

Save the field by clicking the **Save** button.

26.2.4 Configure Fields to be Filled in via AppMill API

On the second page of the screen builder, we will set up fields using two types of AppMill API:

- AppMill API
- Current Service

For the **t3** field, configure the following parameters:

- **Param Type (data type)** - select the **Relation Datalist** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Custom Query** - select the custom query: **Get-All-Users - Get-All-Users**

Save the field by clicking the **Save** button.

For the **api_list_5** field, with the following parameters:

- **Param Type (data type)** - select the **List from API** type.
- **Always show field border -** select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **6**
- **Type (API type)** - select the **AppMill API** option.
  Specify the appeared additional fields as follows:
  - **Service IP** - enter the URL to the REST API. In our case we enter: **localhost:34567**
  - **Screen Name** - enter the value: **GuestApi**
  - **Screen Tab Name** - enter the value: **GuestApi**
  - **Parameter Name** - enter the value: **user_info_id**
  - **Alias for Parameter** - enter the value: **Array.equipment.t3**
  - **Result Block** - enter the parameters that will be displayed as a result: **Array.equipment**
  - **Alias for Value** - enter the parameter that will be transferred: **Array.equipment.type**
  - **Alias for Description -** enter the parameter that will be transferred as value for the field: **Array.equipment.type**

Save the field by clicking the **Save** button.

Add the **t4** field with the following parameters:

- **Param Type (data type)** - select the **Relation Datalist** type.
- **Always show field border -** select the check box to adjust the field layout.

- **Custom Query** - select the custom query: **Get-All-Users - Get-All-Users**

Save the field by clicking the **Save** button.

Add the **api_list_6** field with the following parameters:

- **Param Type (data type)** - select the **List from API** type.
- **Always show field border** - select the check box to adjust the field layout.
- **Multiselect** - select the check box to allow the multi-selection.
- **Num Rows** - enter the quantity, e.g. **5**
- **Type (API type)** - select the **Current Service** option.
  Specify the appeared additional fields as follows:
  - **Username for Service** - enter the Username that is defined for the Service: **GuestUser**
  - **Password for Service** - enter the Password that is defined for the Service.
  - **Screen Name** - enter the value: **UserApi**
  - **Screen Tab Name** - enter the value: **UserApi**
  - **Parameter Name** - enter the value: **user_info_id**
  - **Alias for Parameter** - enter the alias for parameter: **Array.equipment.t4**
  - **Result Block** - enter the parameters that will be displayed as a result: **Array.equipment**
  - **Alias for Value** - enter the parameter that will be transferred: **Array.equipment.type**
  - **Alias for Description** - enter the parameter that will be transferred as value for the field: **Array.equipment.type**

Save the field by clicking the **Save** button.

Save the changes on the screen by clicking the **Update** button.

26.3 Check Screen in Render

Let us check how the configured feature works in the Render application.

Open the Render application using the Manager user credentials to log in to the Render application.

Select the **ApiLists** menu item. Input any username of the Managers group (e.g. Jane Hanson) into the search field on the screen. You will see all the fields we set up above. Open the screen in the Edit mode by clicking the Edit icon.

All the values are received from the external service via API. You can also check the dependencies between the fields.

To watch the video instructions for the above-mentioned settings and how to check them in the Render, follow the link: ***Configure the fields filled in via API for the Table Type section***.

27 External (third-party) plugins and libraries

The AppMill platform gives the possibility to cover any business logic and add any business-specific solution to the application using the external (third-party) libraries and plugins. Let us demonstrate the usage of such plugins via standard package managers.

27.1 Prerequisites

To install a third-party php library, we need to make the following preparations:

1. Update the **Documents** document family.
2. Update the **Documents_main_full** document group.
3. Update the **Reports** screen.

Detailed instructions for these pre-settings are provided below.

27.1.1 Update Document Family

Let us add a new document category to the **Documents** document family.

In the Architect application, open Document Access > Document Family and select the **Documents** document family from the list.

Click the **Add category** button at the bottom left corner of the page. In the appeared row, define the following parameters:

- **Category** - enter a new category name. In this example, we add the **plugin_report** value.
- **Description** - enter the **plugin_report** value to the description field.
- **KP.1** - click the KP.1 field. The pop-up window with the key part list appears. Select the **Alias.user_info.user_info_id** value.

Submit changes on the page.

27.1.2 Update Document Group

In the Architect application, open Document Access > Document Groups and select the **Documents_main_full** document group to be updated with the following parameters:

- **Document Category** - select **main**, **plugin_report**, **and report** parameters from the list.

Submit changes on the page.

27.1.3 Update Report Screen

Now we need to add a new button to the **Reports** screen.

In the Architect application, open Screen configurations > Screens and perform the following operations for the **Reports** screen:

On the **second** page of the screen builder on the **Header** level:

Add the **User** field:

- **Data field** - select the **Primary Table** check box and select the **Alias.user_info.user_info_id**.
- **Type** - select the **Relation datalist** type. This field type allows users to select options from the list that are defined with custom query. In our case, the user ID will be available to be selected from the users list.
- Select the following check boxes to adjust field layout:
    - **Always show field border**
    - **Read only - Except insert**
- **Custom Query** - select the **Get-All-Users - Get-All-Users** query from the drop-down list.

Click the **Update** button to save changes of the User Info screen.

To watch the video instructions for the above-mentioned settings and configurations, follow the link: ***Prerequisites Actions***.

28 Third-Party Plugins Usage

In the AppMill application, you can use external **PHP, JavaScript, Python** libraries.

More information on how to set up and use **PHP**, **JavaScript**, **Python** libraries, can be found in the following video-instructions:

- ***Using a third-party PHP library***
- ***Using a third-party JavaScript library in a plugin***
- ***Using a third-party Python library in a plugin***

Below, we will describe in detail the usage of PHP Library.

28.1 Use Third-Party PHP Library in Plugin

To enable a third-party php library, we need to perform the following actions:

1. Add a Composer for local use.
2. Install Spreadsheet Library via Composer.
3. Create a plugin file and auxiliary folder.
4. Create an extension function.

Below we will describe every action in detail.

28.1.1 Add Composer for Local Use

We will use a Composer to simplify and enhance management of dependencies in PHP projects.

To install Composer, perform the following steps:

1. Download **composer.phar** from Composer's official website.
2. Copy or move the **composer.phar** to the AppMill > Service > plugins > php directory.

When installing the Clients applications the Tomcat Web server installs PHP 7.3.29 on your computer. We will use PHP to run Composer and install the required library.

3. Verify the PHP version installed with Tomcat Web Server (PHP 7.3.29 in this case) by running the command: **php -v**
4. Run the following command to initiate Composer: **php composer.phar**

28.1.2 Install Spreadsheet Library via Composer

As a third-party php library we will use a Spreadsheet library. You can view the Spreadsheet library installation instructions on the official site.

To install the Spreadsheet library into our project, we need to run the Command Prompt application as administrator.

In the Command Prompt application, navigate to the AppMill > Service > plugins > php directory and run the following commands:

1. **php -v**
2. **php composer.phar require phpoffice/phpspreadsheet**

As a result, in the AppMill > Service > plugins > php > vendor directory, the **composer** folder appears.

28.1.3 Create Plugin File and Auxiliary Folder

To connect the Spreadsheet library with our project, we need to create a new plugin file named, e.g. **UseSpreadsheetLibrary.php**, and place it in the AppMill > Service > plugins > php directory.

The **UseSpreadsheetLibrary.php** plugin file contains the data related to the equipment information to be displayed in reports.

Then, create a new folder named, e.g. **_tmp_files**, where the tmp files will be placed, in the AppMill > Service > plugins > php directory.

After adding a new plugin file, the Service should be restarted.

To restart the AppMill Service on their device, please follow the steps below:

1. Enter Services to the search area in the Taskbar.
2. Open the Services application.
3. In the Services application window:

a. select the AppMill Service application and

b. click the Restart link at the top right corner of the window.

As a result, the AppMill Service application will be restarted.

28.1.4 Create Extension Function

Now we need to create a new extension function.

In the Architect application, open System data > Extension functions and create a new extension function.

Define the following parameters:

- **Data Source Function** - enter a function value. In our case **UseSpreadsheetLibrary::PluginReport**.
- **Data Source Description** - enter a description. In our case **UseSpreadsheetLibrary::PluginReport**.
- **Primary Table** - select the **user_info** table from the drop-down list.
- **Extension Library** - select the **PHP** library from the drop-down list.
- **Extension Function** - enter an extension function value. In our case **UseSpreadsheetLibrary::PluginReport**.

Submit changes on the page.

28.1.5 Add Button with Execute (possible document return) Action

If all previous settings are finished, we can add a new button with an Execute (possible document return) action to the **Reports** screen.

In the Architect application, open Screen Configurations > Screens and select the **Reports** screen to be updated.

On the **second** page of the screen builder on the Sections level:

Add the **Report by PHP plugin** button and set up the following parameters:

- **Value** - enter **Report by PHP plugin** as it will be used for sending the first notification template.
- **Button Action** - select the **Execute (possible document return)** action.
- **Function** - add a new extension function. In the opened Extensions Settings pop-up window, define the extension function that will be executed when the user clicks the **Report by PHP plugin** button on the screen in the Render application. In our case as an example, we select the **UseSpreadsheetLibrary::PluginReport** option from the Extensions drop-down list and click the **Add** button.
- **Document Family** - select the **Documents - Documents** option from the drop-down list.
- **Document Category** - select the **plugin_report - plugin_report** option from the drop-down list.

Save the changes on the page.

28.2 Check PHP Library Usage in Render

Let us show how the configured screen works in the Render application.

For that, log in to the Render application using the Manager user credentials.

If the above-mentioned configurations are set up correctly for Manager users (Jane Hanson and John Smith), we will see the **Report by PHP plugin** button on the Reports screen. Click the **Report by PHP plugin** button, you will receive the report created using the third-party PHP library in the plugin.

To watch the video instructions for PHP library usage, follow the link: ***Use Third-Party PHP Library in Plugin***.

29 Files Used in this Instruction

All AppMill documents' templates that we used during the Task Manager environment creation for the HR, Manager, and Employee users and setting up the Onboarding process can be found in the archive at the top right corner of any page with video content.

The .zip archive contains the files we used in the current instruction. Please pay your attention to the _README.txt file that contains additional details.