|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

FPT UNIVERSITY

**Capstone Project Document**

**Motorbike Accessories Inventory Management**

|  |  |
| --- | --- |
| **Group 7** | |
| **Group members** | Phạm Minh Hoàng – SE62769 (Leader)  Nguyễn Trường Thịnh – SE61887  Đỗ Trung Hiếu – SE61847  ~~Nguyễn Huy Hoàng – SE62442~~ |
| **Supervisor** | Lại Đức Hùng |
| **Ext. Supervisor** | N/A |
| **Project Code** | RFIM |

– **Ho Chi Minh City, 13 May, 2019** –

Report 4 – Software Design Description

# Table of Contents

[Table of Contents 3](#_Toc17065057)

[D. Software Design Description 4](#_Toc17065058)

[1. Design Overview 4](#_Toc17065059)

[2. System Architectural Design 5](#_Toc17065060)

[2.1. Web Application Architecture Description 5](#_Toc17065061)

[2.2. Mobile Application Architecture Description 6](#_Toc17065062)

[3. Component Diagram 7](#_Toc17065063)

[4. Detail Description 7](#_Toc17065064)

[4.1. Class Diagram 7](#_Toc17065065)

[4.2. Class Diagram Explanation 9](#_Toc17065066)

[4.3. Sequence Diagram 20](#_Toc17065067)

[4.4. Activity Diagram 39](#_Toc17065068)

[5. User Interface Design 46](#_Toc17065069)

[5.1. Component Interface 46](#_Toc17065070)

[5.2. User Interface Design 49](#_Toc17065071)

[6. Database Design 111](#_Toc17065072)

[6.1. Entity Relationship Diagram (ERD) 111](#_Toc17065073)

[6.2. Entity Dictionary 112](#_Toc17065074)

[7. Algorithms 112](#_Toc17065075)

[7.1. Find box base on the longest stock in date and closest warehouse’s gate 112](#_Toc17065076)

[7.2. Convert shelves’ coordinates to matrix algorithm 114](#_Toc17065077)

[7.3. Dijkstra algorithms 116](#_Toc17065078)

# D. Software Design Description

## Design Overview

* This document describes the technical and user interface design of RFIM system. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
  + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagram that describes the connection and integration of the system.
  + Section 4: gives the detail design description, which includes class diagram, class explanation and sequence diagram to detail the application functions.
  + Section 5: describes a fully attributed Entity Relationship Diagram.

## System Architectural Design

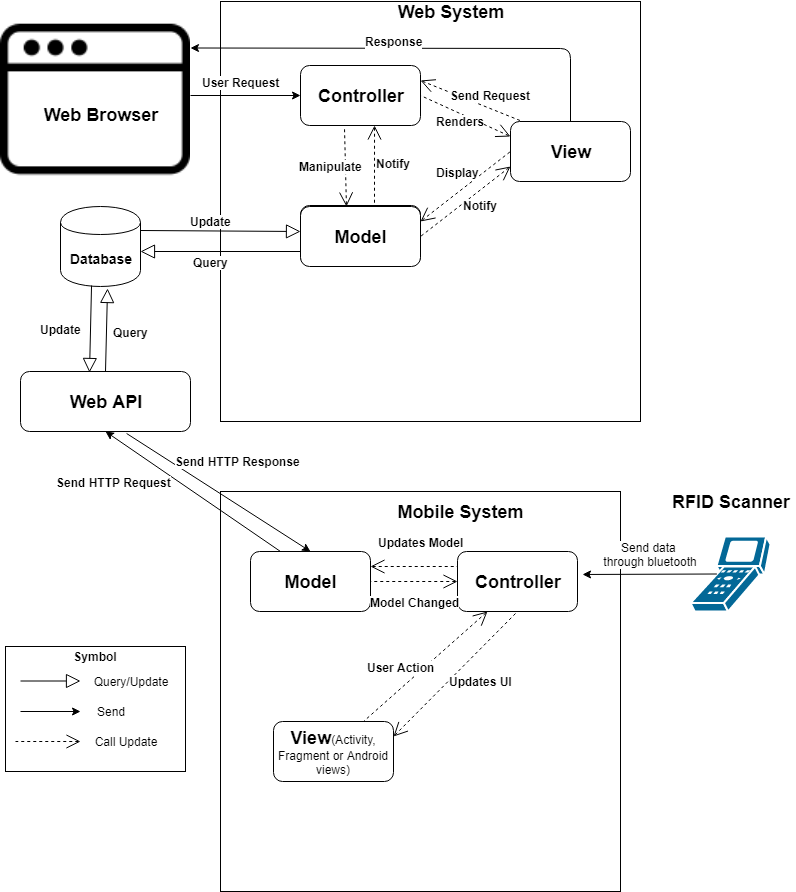


Figure 45 - System Architectural Design

### Web Application Architecture Description

In web application, our system is developed with ASP.NET Core MVC architecture. We decide to choose this architect for web application because of following advantages:

* ASP.NET Core is a cross-platform, high-performance, open-source framework for building application.
* Application is built-in dependency injection.
* ASP.NET Core has some technologies like entity framework to connect database.
* Support to develop and run on Windows, macOS, and Linux.
* Support to develop Web API and easily to build Web UI.
* ASP.NET Core is open-source and community-focused.

The MVC pattern (Model-View-Controller) separates application into 3 main components: Model-View-Controller. MVC patter helps you to build application more testable and easier. MVC-based app contains:

* Controller: is a part of application like a handler browser to handle and response user interaction and input. Controller handles route data and call repository’s method to get business data for handling bussines logic. Then select view to render to web browser for user.
* View: is a part of application for displaying the app’s user interface. It get data from model and specifies how that data should be presented. It updates UI when model changes.
* Model: is a part of application for represents the business data. The model classes use validation logic to enforce business data. The model notifies views when it changes and lets view query the model.

(Reference: [4])

### Mobile Application Architecture Description

We decided to choose Android and Arduino to implement our system because of the following benefits:

* MVC is a popular architecture which can help device code in to Model, View and Controller.
* We focus on the user who has to work in warehouse and has to move too much. Therefore we choose Android mobile device to develop our system.
* Android is the popular handheld operating system in the word so that user can install easily and use our system. In addition we can develop Android application easily on Window, Mac OS and Linux.
* Compare with React Native and other hybrid framework on mobile, Android is more stable and faster.
* Arduino framework is integrated a lot of modules like RFID scanner, Bluetooth, wifi, etc… which are affordable.
* Bluetooth technology helps transfer data between Android handheld and RFID scanner without using cable.

## Component Diagram

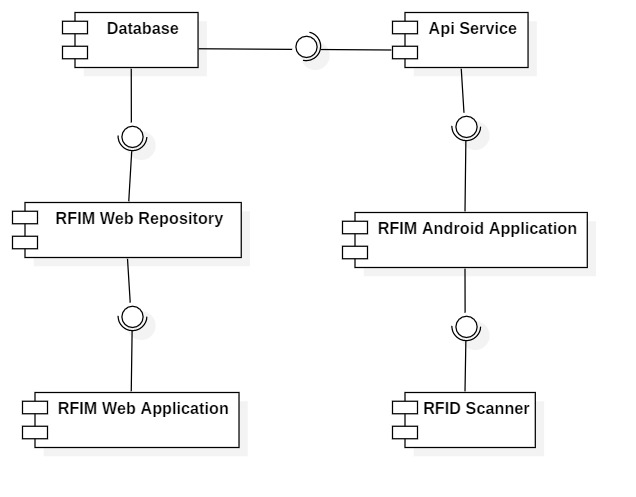


Figure 46 – Component Diagram

|  |  |
| --- | --- |
| COMPONENT DICTIONARY: DESCRIBES COMPONENTS | |
| Component name | **Description** |
| RFID Scanner | RFID scanner device |
| API Service | Provide API for Android application. Interact with Database |
| RFIM Android Application | Receive data from RFID scanner. Interact with API |
| RFIM Web Application | Handle business |
| RFIM Web Repository | Interact with Database |
| Database | Store data |

Table 15 - Component Dictionary

## Detail Description

### Class Diagram

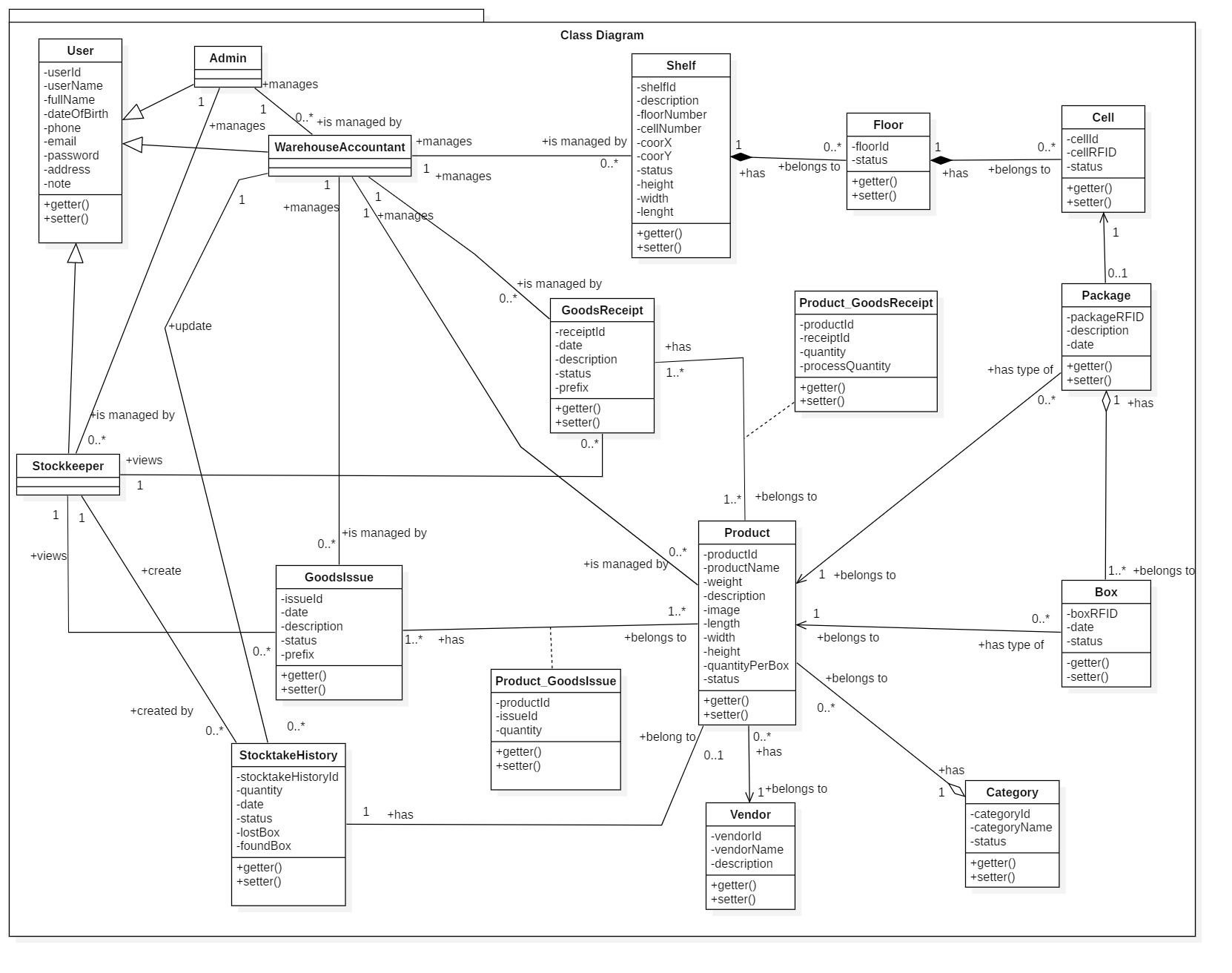


Figure 47 – Class Diagram

|  |  |  |
| --- | --- | --- |
| **CLASS DICTIONARY: DESCRIBE CLASS** | | |
| ***Class Name*** | ***Mapping column with Conceptual diagram*** | ***Description*** |
| **User** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the user information in general |
| **Admin** | Admin | Contains the admin information |
| **Stock-keeper** | Stock-keeper | Contains the stock-keeper information |
| **Warehouse Accountant** | Warehouse Accountant | Contains the warehouse accountant information |
| **Shelf** | Shelf | Contains the shelf information |
| **Floor** | Floor | Contains the floor information |
| **Cell** | Cell | Contains the cell information |
| **Package** | Package | Contains the package information |
| **Product** | Product | Contains the product information |
| **Category** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the category information |
| **Box** | Box | Contains the box information |
| **Vendor** | Vendor | Contains the vendor information |
| **Goods Receipt** | Goods Receipt | Contains the goods receipt information |
| **Goods Issue** | Goods Issue | Contains the goods issue information |
| **Product \_ Goods Receipt** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the attribute to define many to many relationship |
| **Product \_ Goods Issue** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the attribute to define many to many relationship |
| **Stocktake History** | StocktakeHistory | Contains the stocktake history information |

Table 16 - Class Dictionary

### Class Diagram Explanation

#### User

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | string | private | Unique identifier of user |
| userName | string | private | Username of user |
| password | string | private | Password of user |
| fullName | string | private | Full name of user |
| dateOfBirth | date | private | Date of birth of user |
| phone | string | private | Phone number of user |
| email | string | private | Email of user |
| address | string | private | Address of user |
| note | string | private | Notes about user |

Table 17 - <Class Diagram> User Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get user attributes |
| setter() | void | public | Set value of user attributes |

Table 18 - <Class Diagram> User Methods

#### Admin

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | string | private | Unique identifier of admin |
| userName | string | private | Username of admin |
| password | string | private | Password of admin |
| fullName | string | private | Full name of admin |
| dateOfBirth | date | private | Date of birth of admin |
| phone | string | private | Phone number of admin |
| email | string | private | Email of admin |
| address | string | private | Address of admin |
| note | string | private | Notes about admin |

Table 19 - <Class Diagram> Admin Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get admin attributes |
| setter() | void | public | Set value of admin attributes |

Table 20 - <Class Diagram> User Methods

#### Stock-keeper

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | string | private | Unique identifier of stock-keeper |
| userName | string | private | Username of stock-keeper |
| password | string | private | Password of stock-keeper |
| fullName | string | private | Full name of stock-keeper |
| dateOfBirth | date | private | Date of birth of stock-keeper |
| phone | string | private | Phone number of stock-keeper |
| email | string | private | Email of stock-keeper |
| address | string | private | Address of stock-keeper |
| note | string | private | Notes about stock-keeper |

Table 21- <Class Diagram> Stock-keeper Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get stock-keeper attributes |
| setter() | void | public | Set value of stock-keeper attributes |

Table 22- <Class Diagram> Stock-keeper Methods

#### Warehouse Accountant

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| userId | string | private | Unique identifier of warehouse staff |
| userName | string | private | Username of warehouse staff |
| password | string | private | Password of warehouse staff |
| fullName | string | private | Full name of warehouse staff |
| dateOfBirth | date | private | Date of birth of warehouse staff |
| phone | string | private | Phone number of warehouse staff |
| email | string | private | Email of warehouse staff |
| address | string | private | Address of warehouse staff |
| note | string | private | Notes about warehouse staff |

Table 23- <Class Diagram> Warehouse Accountant Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get warehouse staff attributes |
| setter() | void | public | Set value of warehouse staff attributes |

Table 24- <Class Diagram> Stock-keeper Methods

#### Shelf

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| shelfId | string | private | Unique identifier of shelf |
| description | string | private | Description of shelf |
| floorNumber | integer | private | Number of floor of each shelf |
| cellNumber | integer | private | Number of cell per floor of each shelf |
| coorX | integer | private | Coordinate X of shelf, determine by position in horizontal |
| coorY | integer | private | Coordinate Y of shelf, determine by position in vertical |
| status | boolean | private | Status of shelf |
| height | float | private | Height of shelf |
| width | float | private | Width of shelf |
| lenght | float | private | Length of shelf |

Table 25- <Class Diagram> Shelf Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get shelf attributes |
| setter() | void | public | Set value of shelf attributes |

Table 26- <Class Diagram> Shelf Methods

#### Floor

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| floorId | string | private | Unique identifier of floor |
| status | boolean | private | Status of floor |

Table 27- <Class Diagram> Floor Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get floor attributes |
| setter() | void | public | Set value of floor attributes |

Table 28- <Class Diagram>Floor Methods

#### Cell

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| cellId | string | private | Unique identifier of cell |
| cellRFID | string | private | RFID bind to cell |
| status | boolean | private | Status of cell |

Table 29- <Class Diagram>Cell Attribute

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get cell attributes |
| setter() | void | public | Set value of cell attributes |

Table 30- <Class Diagram>Cell Methods

#### Package

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| packageRFID | string | private | Unique RFID of package |
| date | date | private | Package creation date |
| description | string | private | Description of package |

Table 31- <Class Diagram>Package Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get package attributes |
| setter() | void | public | Set value of package attributes |

Table 32- <Class Diagram>Package Methods

#### Product

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| productId | string | private | Unique identifier of product |
| productName | string | private | Name of product |
| weight | float | private | Weight of product |
| description | string | private | Description of product |
| image | string | private | Image of product |
| length | float | private | Product box length |
| width | float | private | Product box width |
| height | float | private | Product box height |
| quantityPerBox | integer | private | Number of item per product box |
| status | boolean | private | Product status |

Table 33- <Class Diagram> Product Attribute

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get product attributes |
| setter() | void | public | Set value of product attributes |

Table 34- <Class Diagram> Product Methods

#### Category

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| categoryId | int | private | Unique identifier of category |
| categoryName | string | private | Name of category |
| status | boolean | private | Status of category |

Table 35- <Class Diagram>Category Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get category attributes |
| setter() | void | public | Set value of category attributes |

Table 36- <Class Diagram> Category Methods

#### Box

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| boxRFID | string | private | Unique RFID of box |
| date | date | private | Box stock in date |
| status | boolean | private | Status of box |

Table 37- <Class Diagram>Box Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get box attributes |
| setter() | void | public | Set value of box attributes |

Table 38- <Class Diagram> Box Method

#### Vendor

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| vendorId | int | private | Unique identifier of vendor |
| vendorName | string | private | Name of vendor |
| description | string | private | Description of vendor |

Table 39- <Class Diagram> Vendor Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get vendor attributes |
| setter() | void | public | Set value of vendor attributes |

Table 40- <Class Diagram> Vendor Methods

#### Goods Receipt

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| receiptId | string | private | Unique identifier of goods receipt |
| date | dateTime | private | Goods receipt creation date |
| description | string | private | Description of goods receipt |
| status | int | private | Status of goods receipt |
| prefix | string | private | Prefix of goods receipt |

Table 41- <Class Diagram>Goods Receipt Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get goods receipt attributes |
| setter() | void | public | Set value of goods receipt attributes |

Table 42- <Class Diagram>Goods Receipt Methods

#### Goods Issue

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| issueId | string | private | Unique identifier of goods issue |
| date | dateTime | private | Date of creating goods issue |
| description | string | private | Description of goods issue |
| status | int | private | Status of goods issue |
| prefix | string | private | Prefix of goods issue |

Table 43- <Class Diagram> Goods Issue Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get goods issue attributes |
| setter() | void | public | Set value of goods issue attributes |

Table 44- <Class Diagram> Goods Issue Methods

#### Product\_Goods Receipt

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| receiptId | string | private | Unique identifier of goods receipt |
| productId | string | private | Unique identifier of product |
| quantity | integer | private | Planning quantity of stock in |
| processQuantity | integer | private | Current scanned quantity of stock in |

Table 45- <Class Diagram>Product\_Goods Receipt Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get product\_receipt attributes |
| setter() | void | public | Set value of product\_receipt attributes |

Table 46- <Class Diagram>Product\_Goods Receipt Methods

#### Product\_Goods Issue

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| issueId | string | private | Unique identifier of goods issue |
| productId | string | private | Unique identifier of issue |
| quantity | integer | private | Planning quantity of stock out |

Table 47- <Class Diagram>Product\_Goods Issue Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get product\_issue attributes |
| setter() | void | public | Set value of product\_issue attributes |

Table 48- <Class Diagram>Product\_Goods Issue Methods

#### Stocktake History

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| stocktakeHistoryId | int | private | Unique identifier of stocktake history |
| quantity | string | private | Stocktaked quantity |
| date | date | private | Date of Stocktake History |
| status | boolean | private | Status of Stocktake |
| lostBox | string | private | Boxes that didn’t appear in stocktake process |
| foundBox | string | private | Lost boxes that was found during stocktake process |

Table 49- <Class Diagram>Stocktake History Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get stocktake attributes |
| setter() | void | public | Set value of stocktake attributes |

Table 50- <Class Diagram>Stocktake History Methods

### Sequence Diagram

Figure 48- <Sequence Diagram> Create Category



Figure 49- <Sequence Diagram> Create Invoice

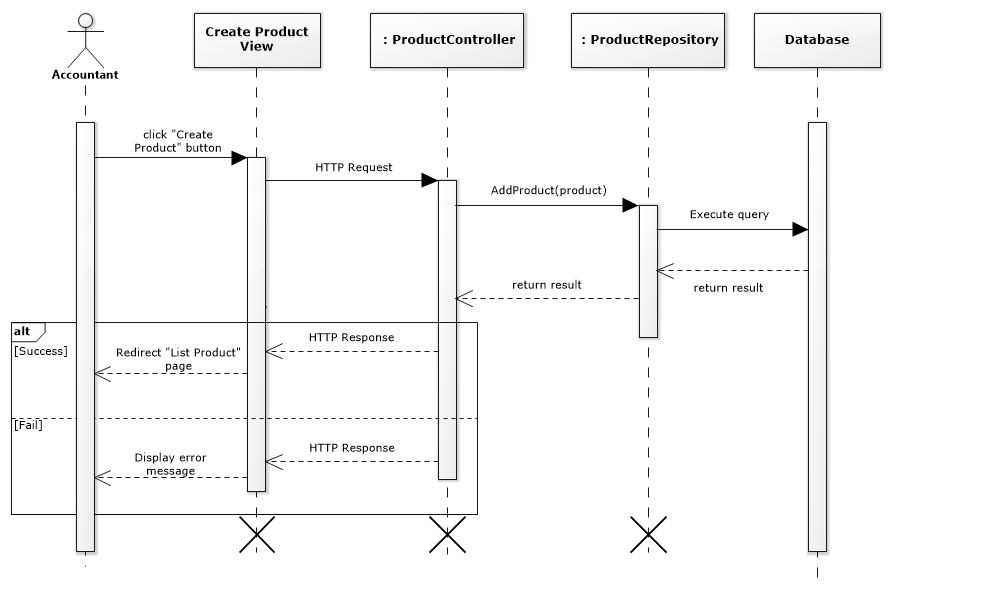


Figure 50- <Sequence Diagram> Create Product

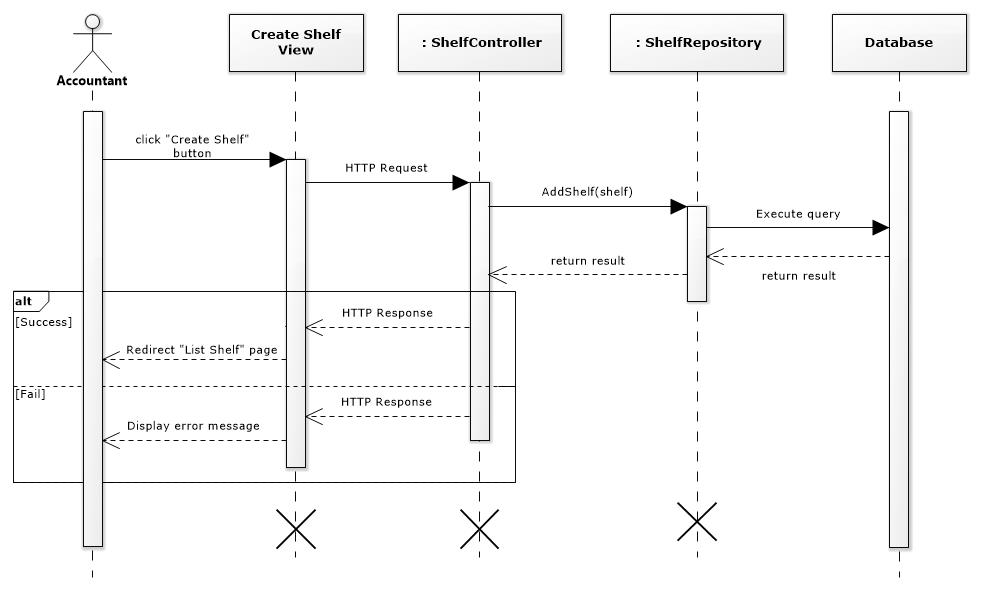


Figure 51- <Sequence Diagram> Create Shelf

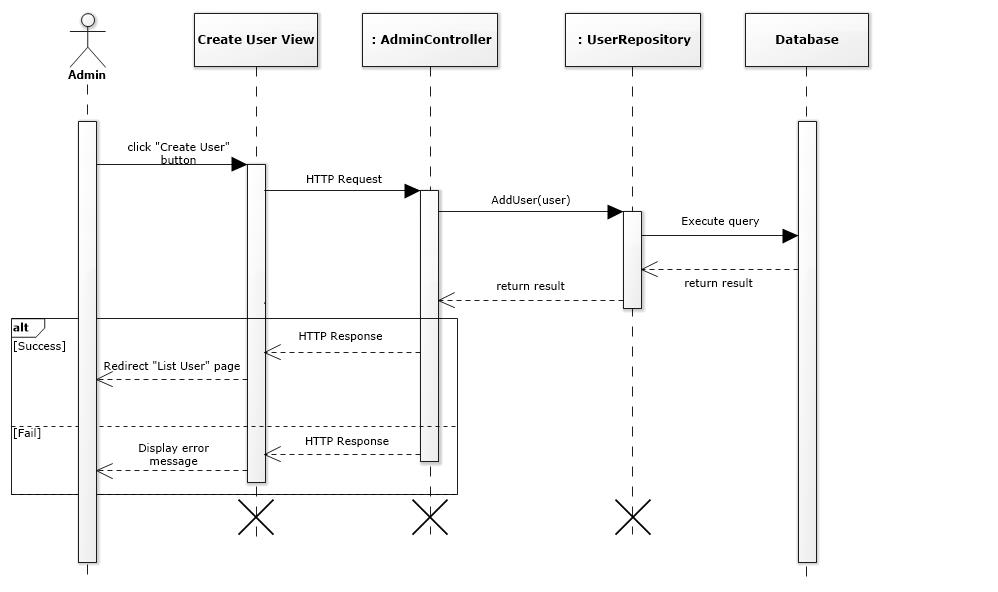


Figure 52- <Sequence Diagram> Create User

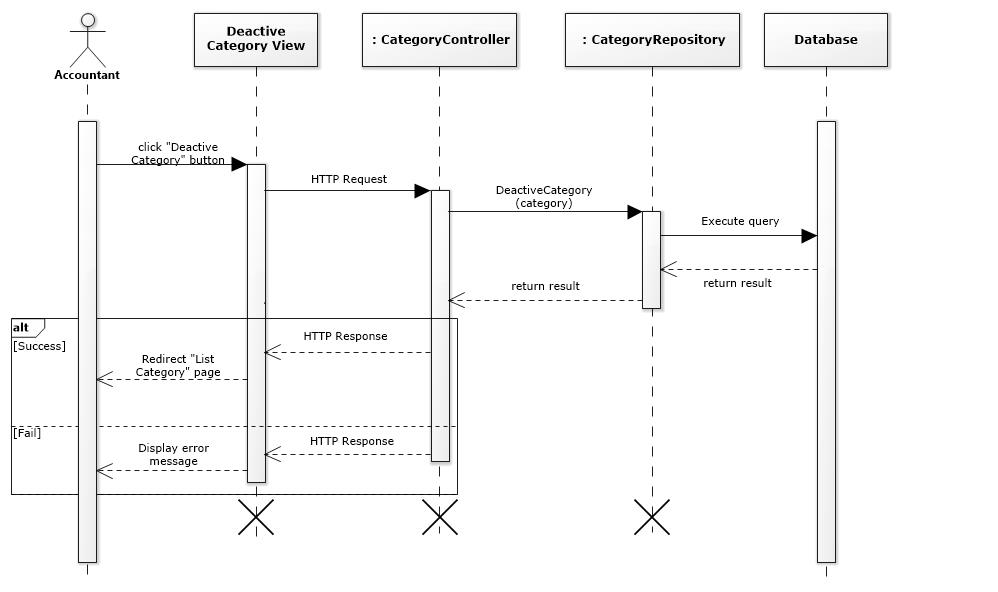


Figure 53- <Sequence Diagram> Deactivate Category

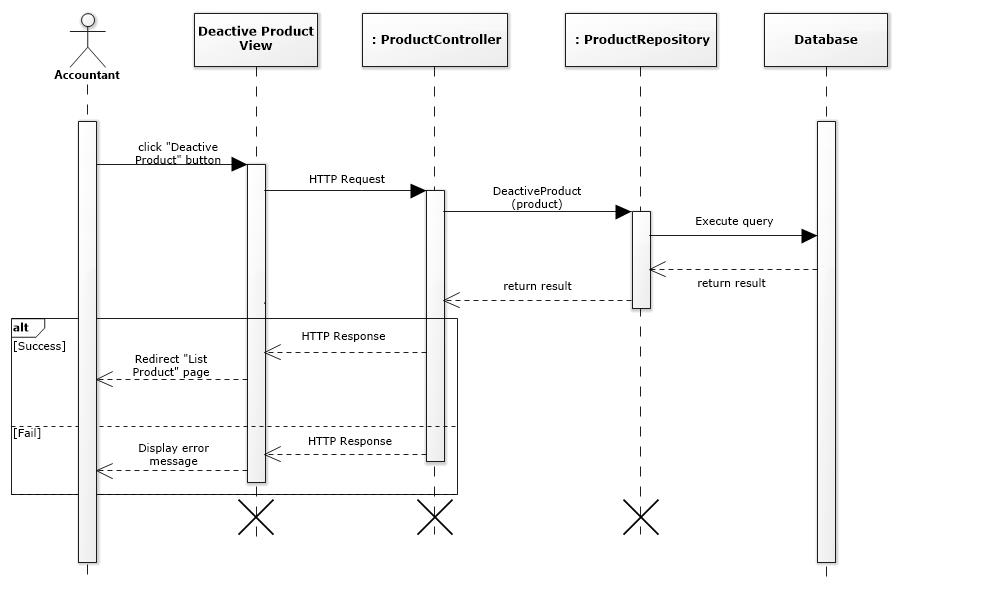


Figure 54- <Sequence Diagram> Deactivate Product

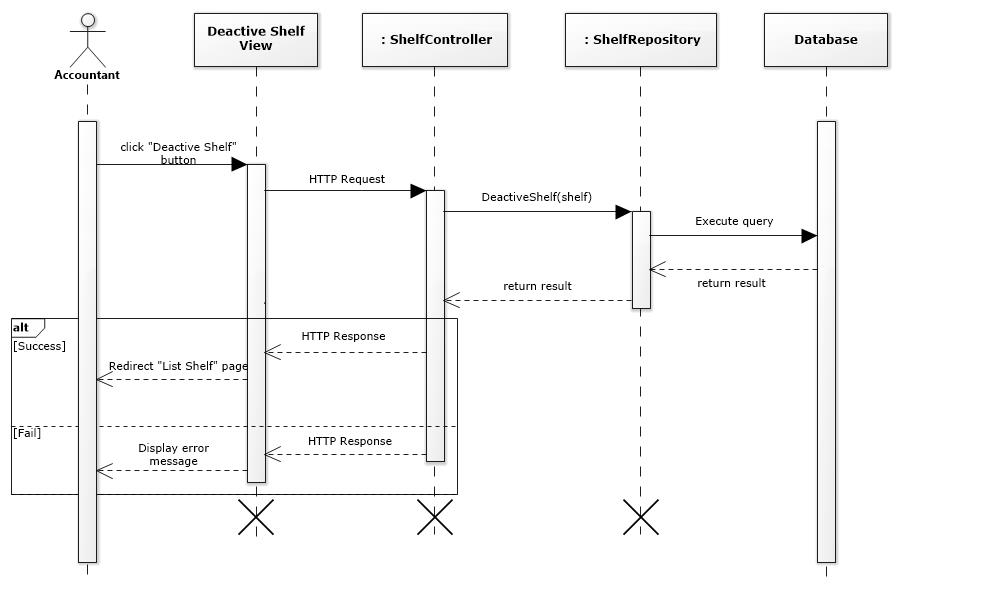


Figure 55- <Sequence Diagram> Deactivate Shelf

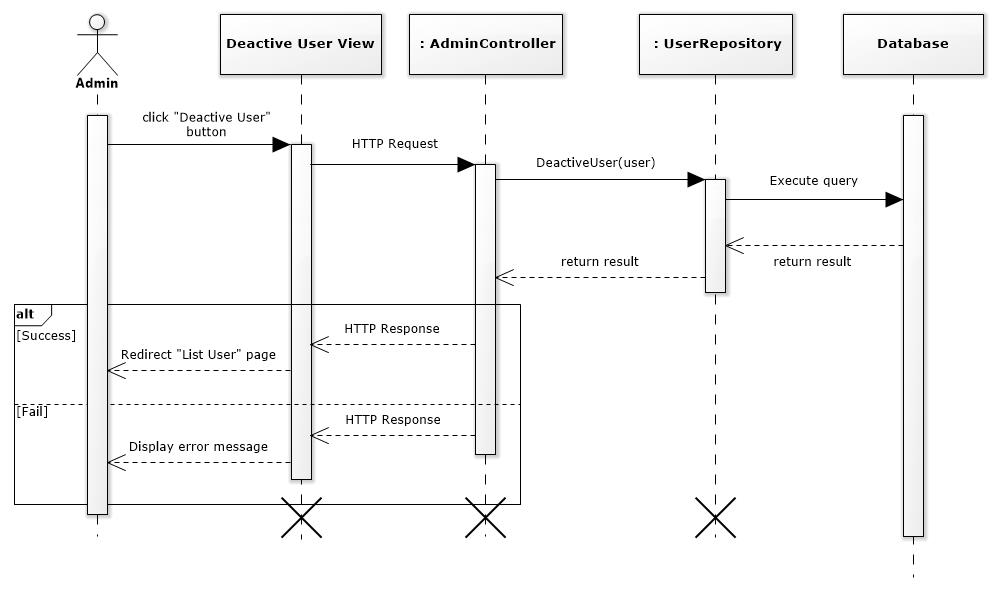


Figure 56- <Sequence Diagram> Deactivate User

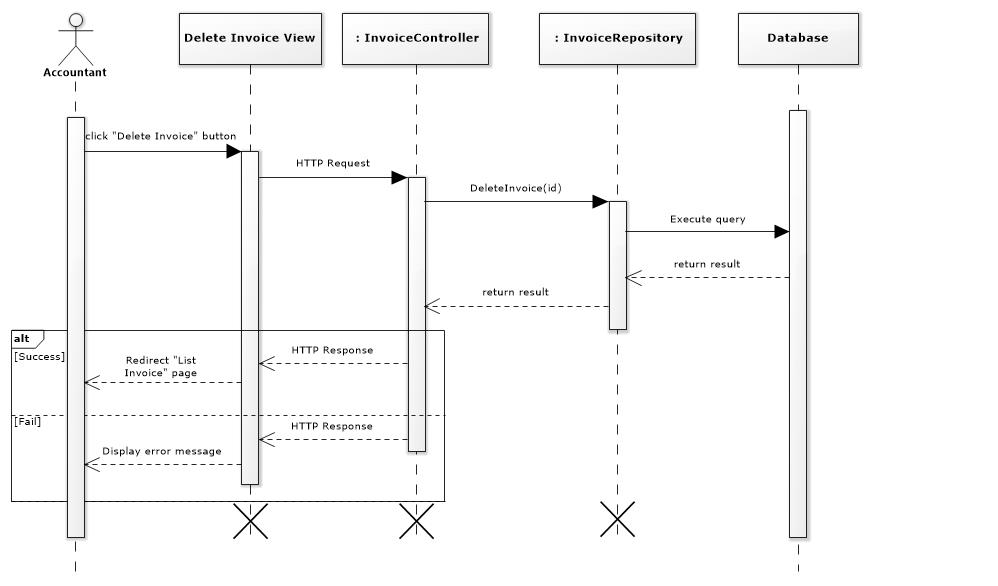


Figure 57- <Sequence Diagram> Delete Invoice

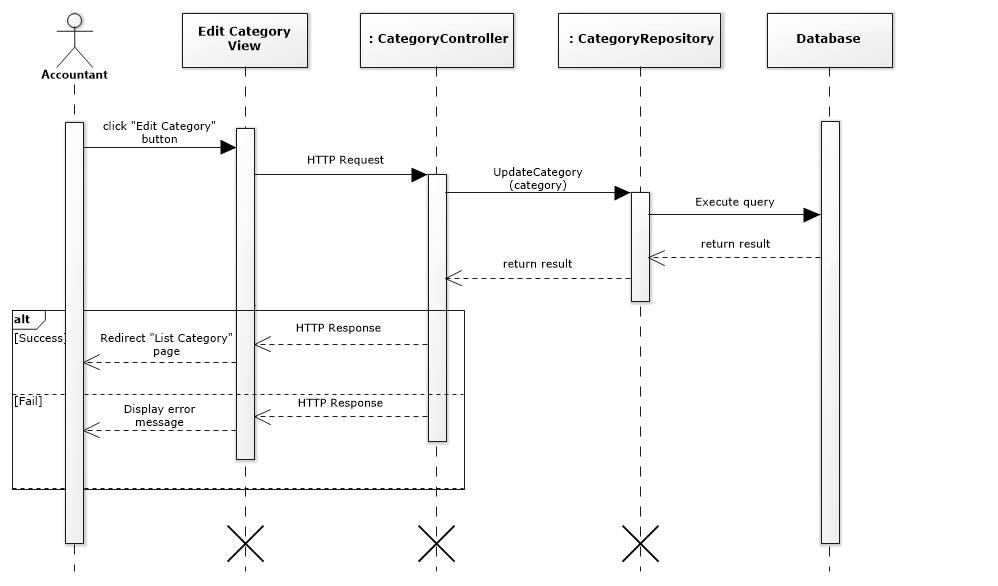


Figure 58- <Sequence Diagram> Update Category

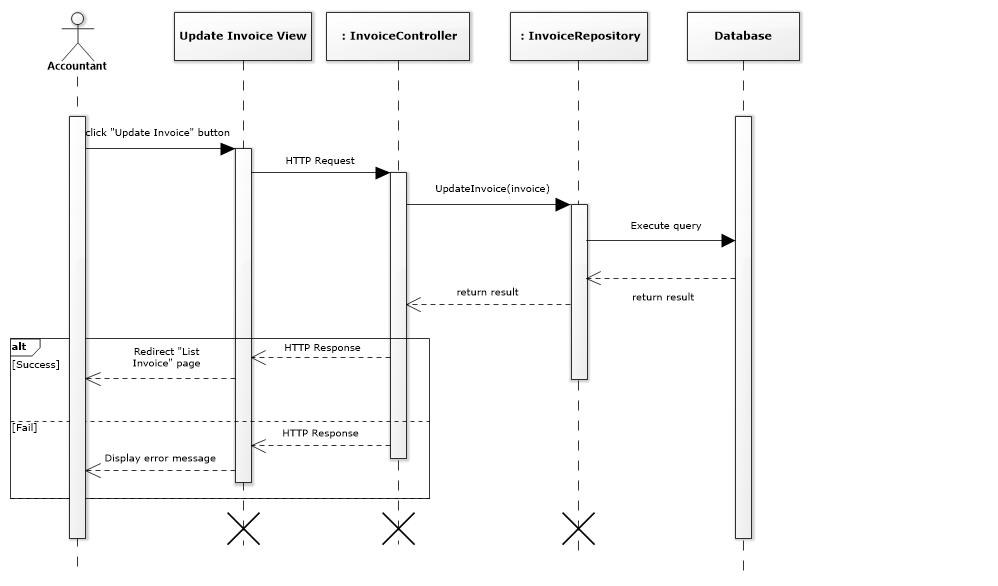


Figure 59- <Sequence Diagram> Update Invoice

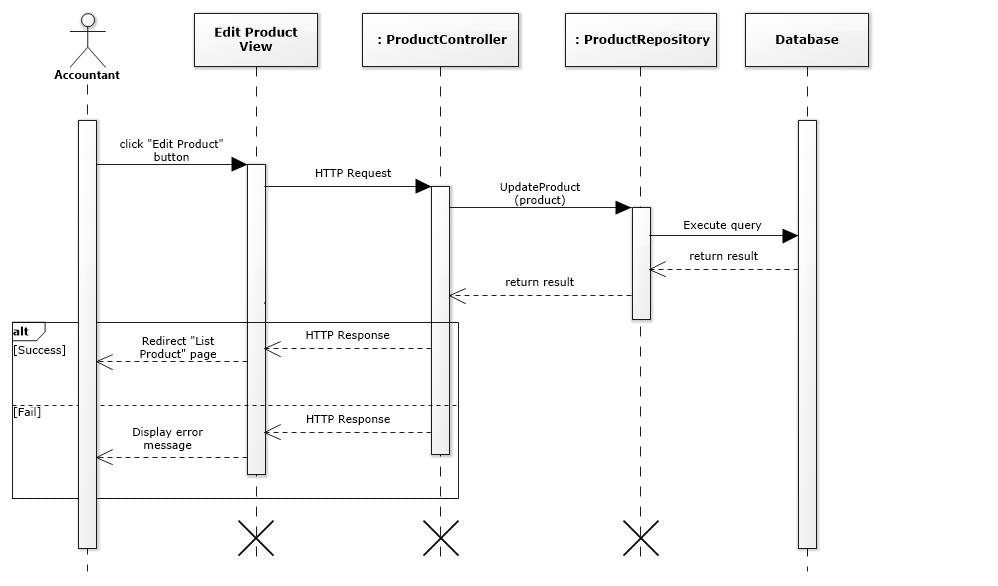


Figure 60- <Sequence Diagram> Update Product

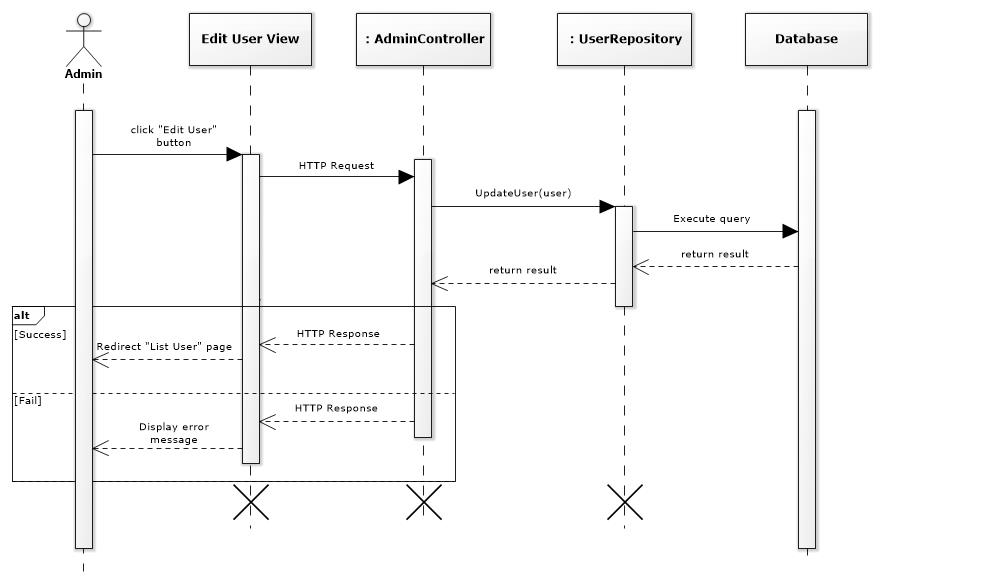


Figure 61- <Sequence Diagram> Update User

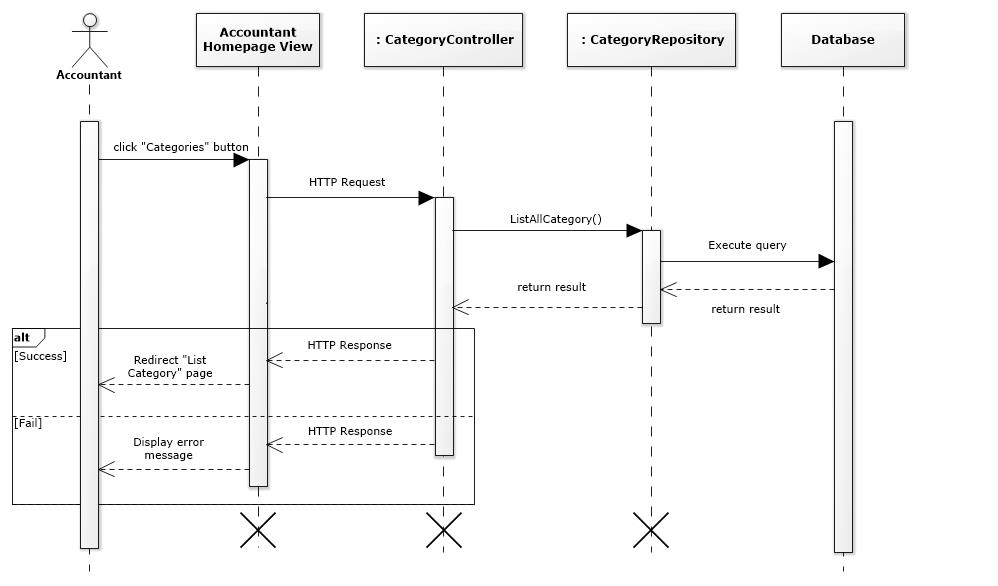


Figure 62- <Sequence Diagram> View Category

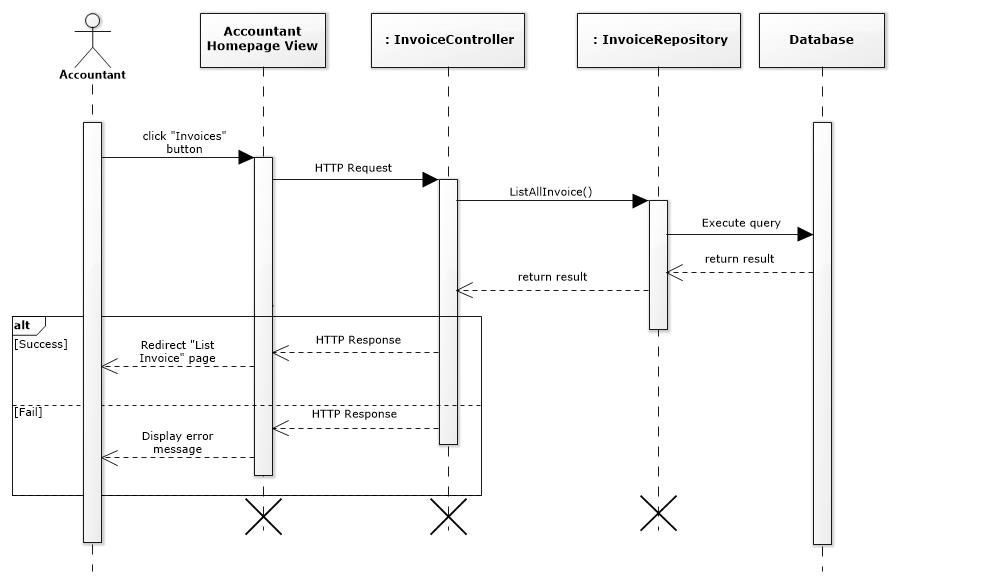


Figure 63- <Sequence Diagram> View Invoice

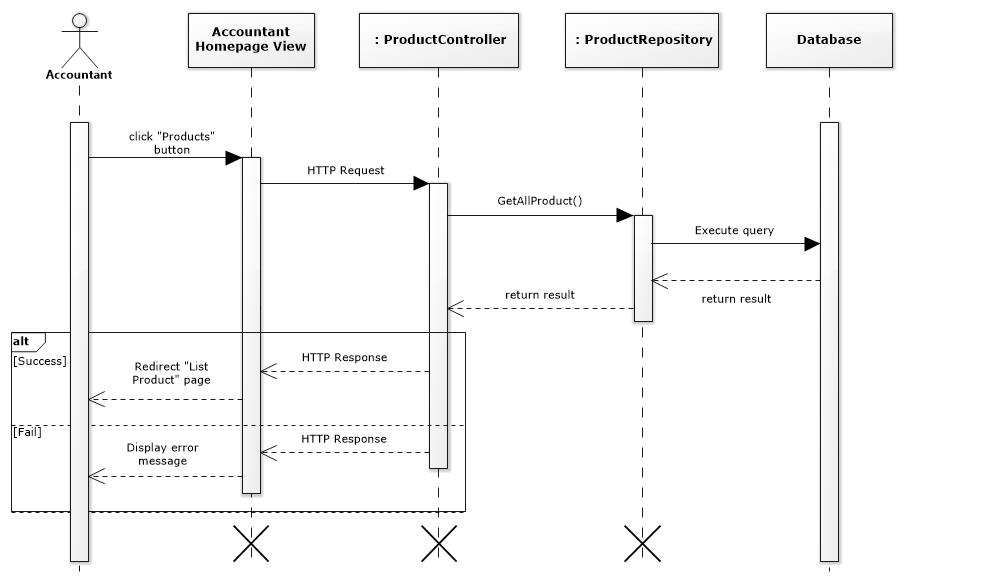


Figure 64- <Sequence Diagram> View Product

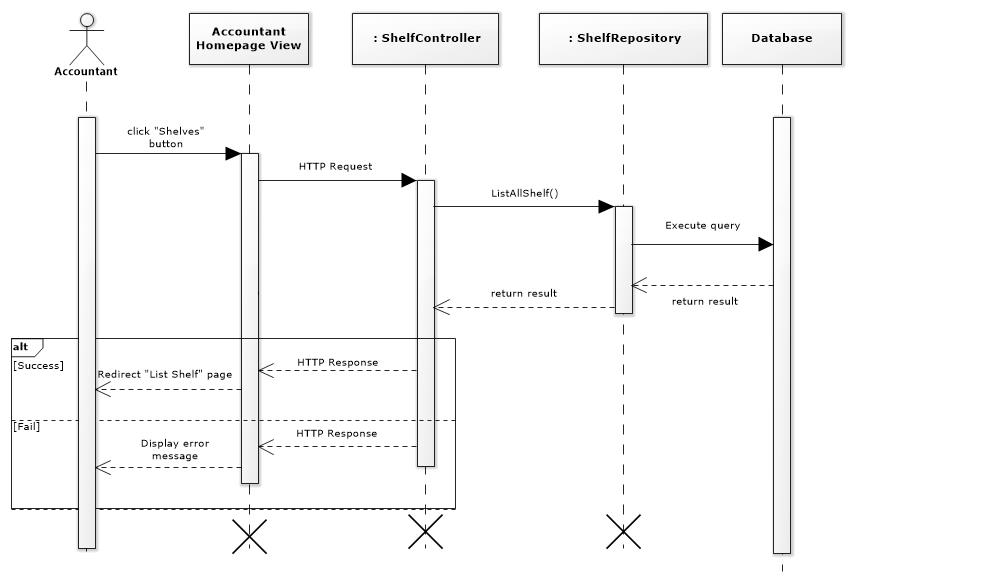


Figure 65- <Sequence Diagram> View Shelf

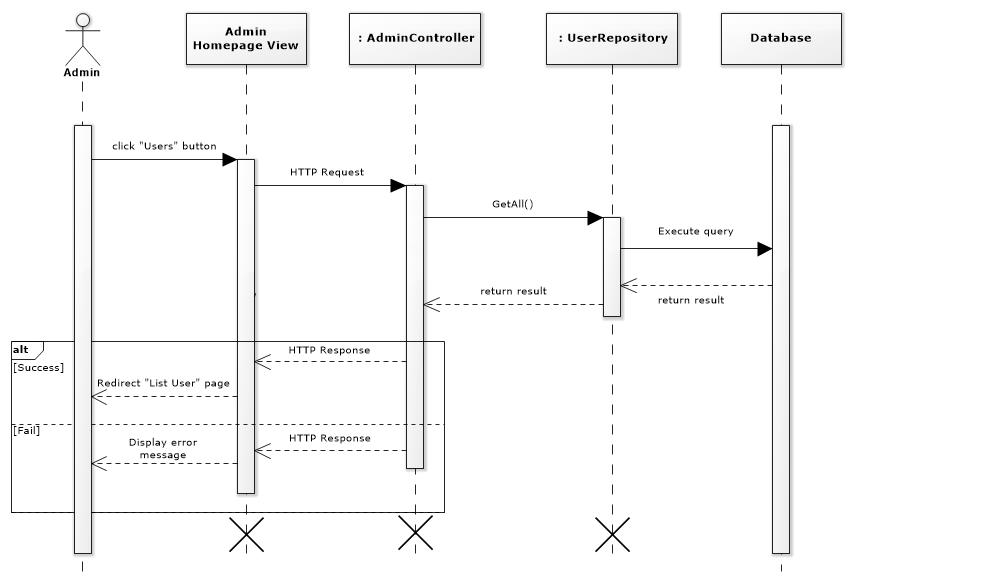


Figure 66- <Sequence Diagram> View User

### Activity Diagram

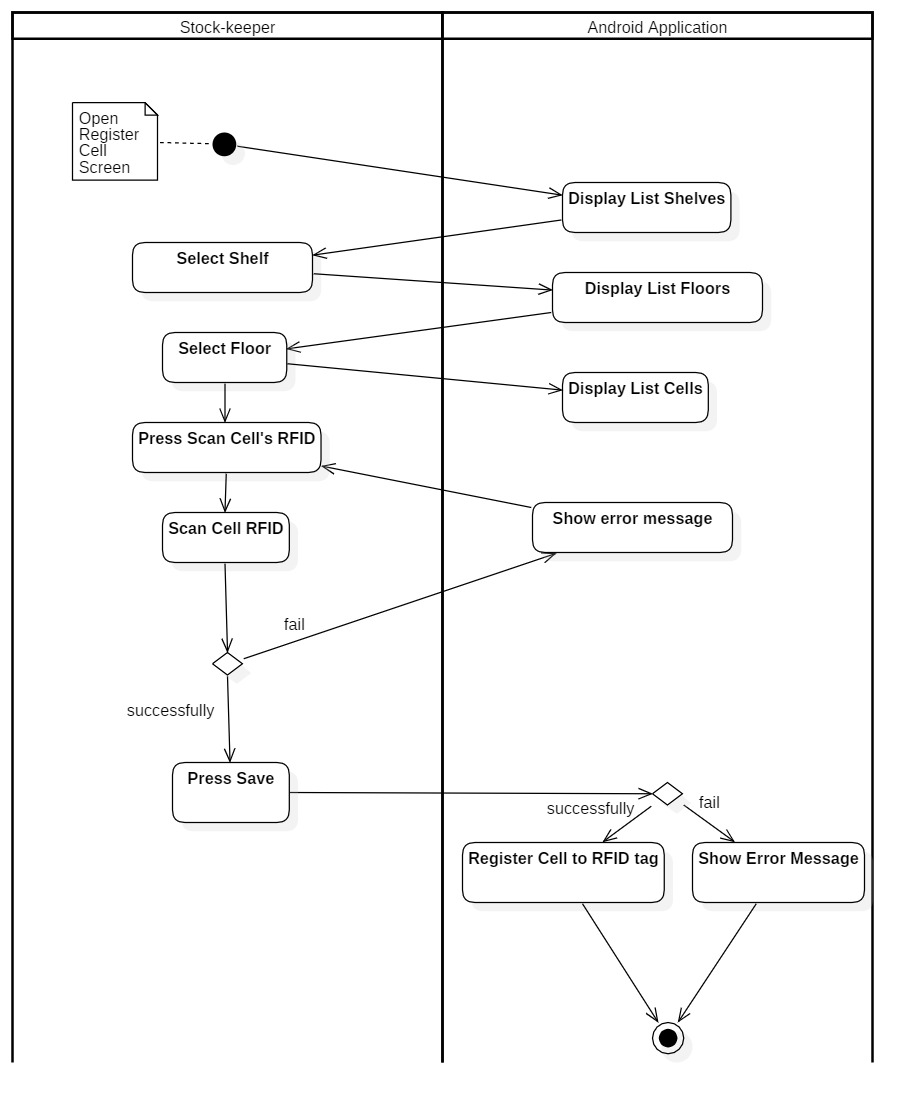


Figure 67- <Activity Diagram> Register Cell

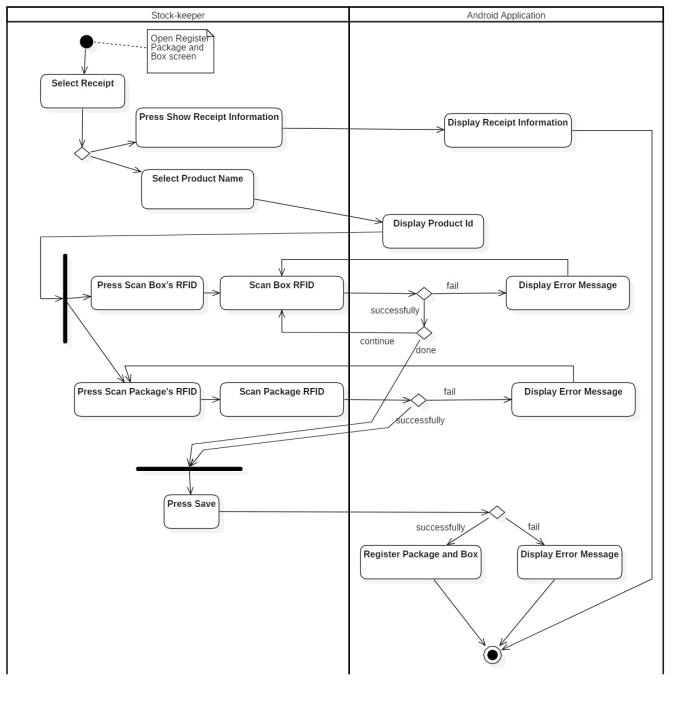


Figure 68- <Activity Diagram> Register Box and Package

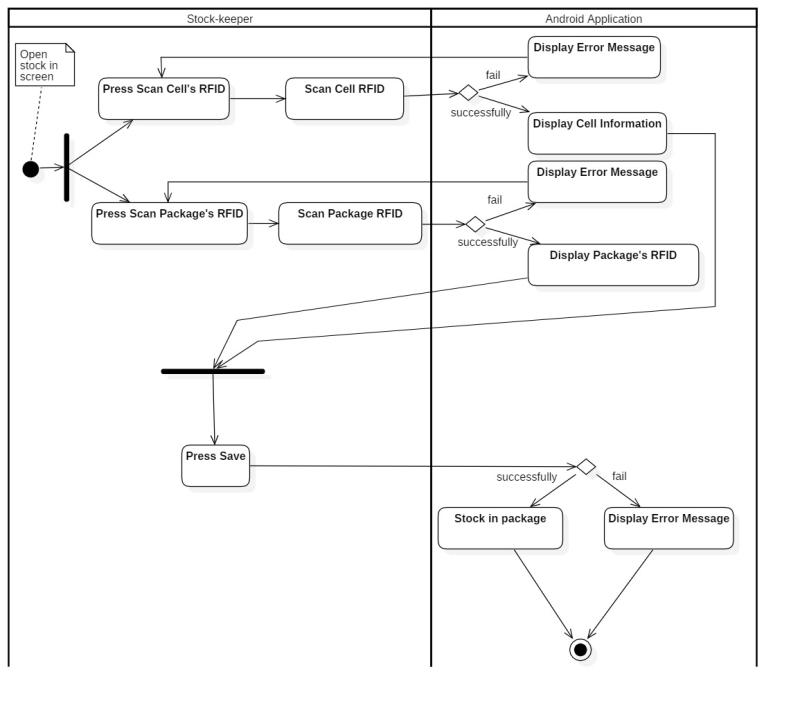


Figure 69- <Activity Diagram> Stock in

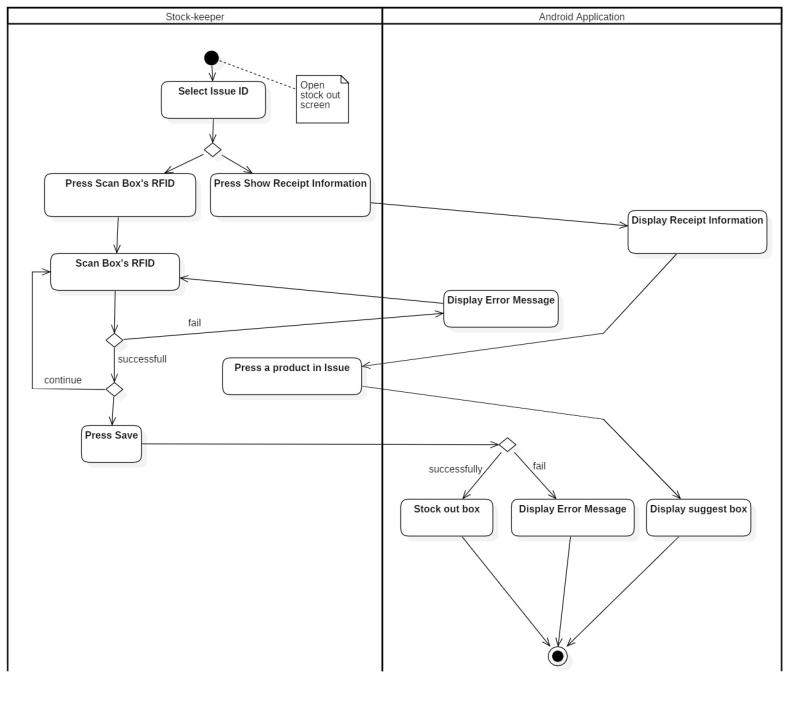


Figure 70- <Activity Diagram> Stock Out

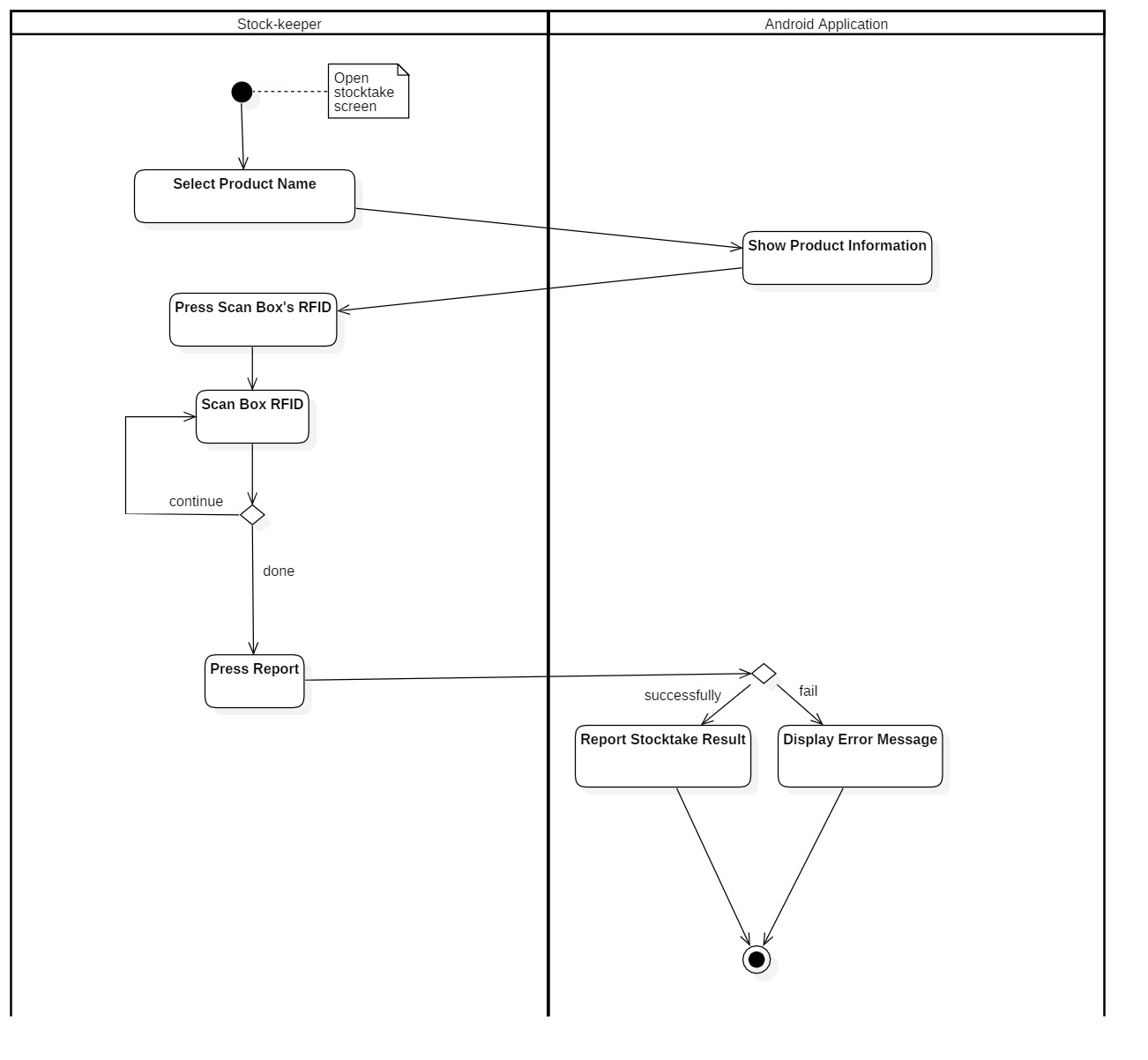


Figure 71- <Activity Diagram> Stocktake Inventory

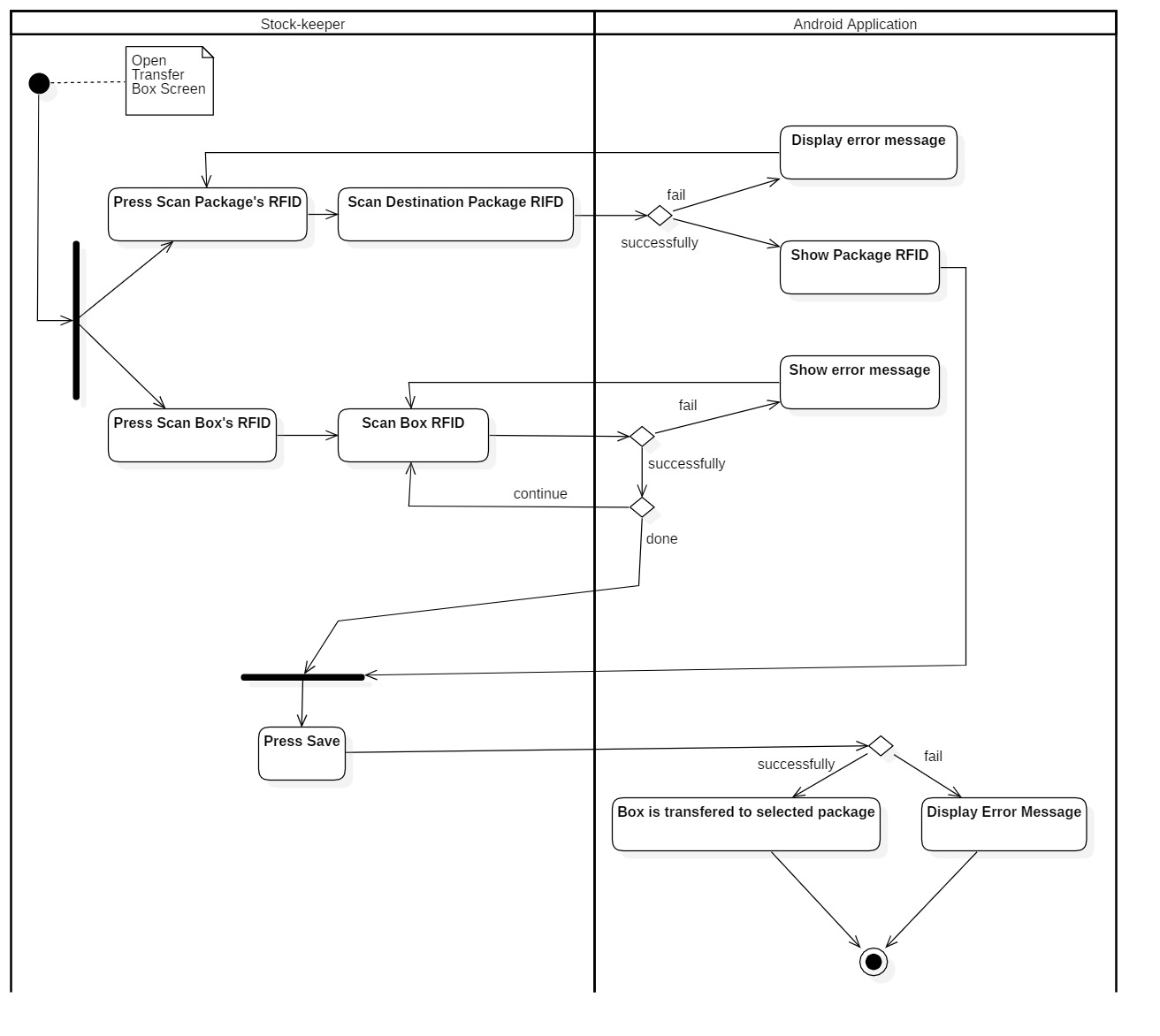


Figure 72- <Activity Diagram> Transfer Box

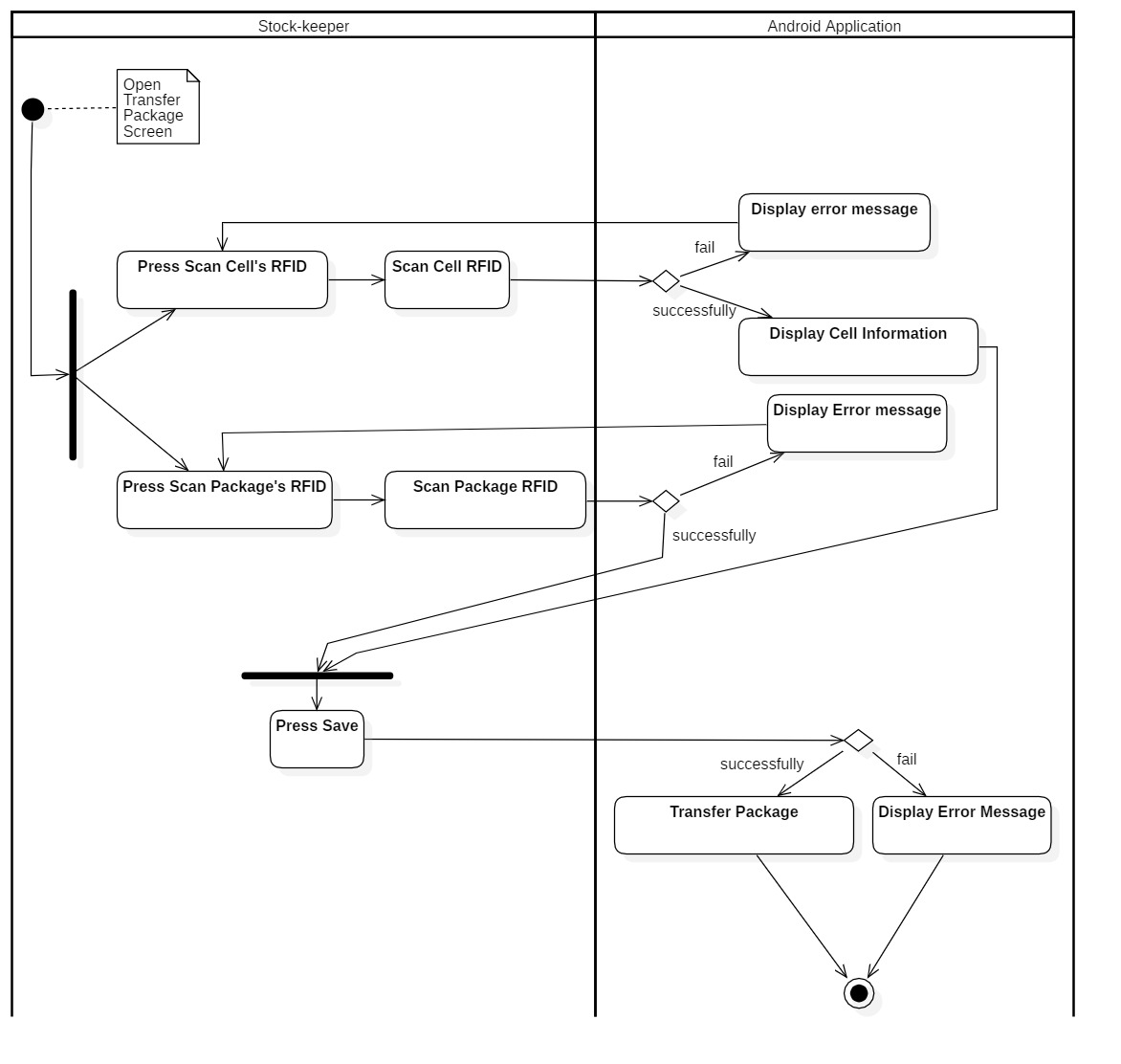


Figure 73- <Activity Diagram> Transfer Package

## User Interface Design

### Component Interface

Component Interface are Web Service API which will be used by Mobile Application.

| **Signature** | **Description** | **Input** | **Output** | **Output Format** |
| --- | --- | --- | --- | --- |
| getBoxRfidsByProductId(id) | Get box's rfid by using product’s id | id: String | Json list box’s rfid | String |
| getCellByFloorId(id) | Get cell by floor’s id | id: String | Json list cell’s information | String |
| registerCell(cell) | Register cell with rfid tag | cell: CellDTO | Http status code, String | String |
| getCellByCellRfid(id) | Get cell by cell’s rfid | id: String | Json cell’s information | String |
| checkCellIsEmpty(id) | Check cell is empty or not | id: String | Http status code, String | String |
| getFloorByShelfId(id) | Get floors by shelf’s id | id: String | Json list floor’s information | String |
| getFloorByCellId(id) | get floor by cell’s id | id: String | Json floor’s information | String |
| getAllActiveIssues() | Get all active issue (pending and processing) | N/A | Json list active (pending, processing) goods issue information | String |
| getAllActiveReceipt() | Get all active receipt (pending) | N/A | Json list active (peding) goods receipt’s information | String |
| suggestBox(request) | Suggest box using dijkstra and weighted average number | request: InvoiceInfoItem | Json suggesting box’s information | String |
| registerPackage(request) | Register package and box into system | request: RequestPackageInfo | Http status code, String | String |
| getPackageByPackageRfid(id) | Get package by package’s rfid | id: String | Json package’s information | String |
| stockIn(pac) | Stock in Package by mapping package with cell’s id | pac: PackageDTO | Http status code, String | String |
| stockOut(request) | Stock out box by remove box | request: RequestStockoutInfo | Http status code, String | String |
| transferPackages(pac) | Transfer package by update cell’s rfid | pac: PackageDTO | Http status code, String | String |
| transferBoxes(request) | Transfer box between packaged by update package’s rfid | request: RequestPackageInfo | Http status code, String | String |
| getAllProduct() | Get all products | N/A | Json list product’s information | String |
| getProudctByBoxId(id) | Get product by box’s id | id: String | Json product’s information | String |
| getAllShelves() | Get all shelves | N/A | Json list shelves’s information | String |
| getShelfByCellId(id) | Get shelves by floor’s id | id: String | Json shelf’s information | String |
| addStocktakeHistory(request) | Create stocktake history | request: StocktakeHistoryDTO | Http status code, String | String |
| login(request) | Login api | request: UserDTO | Json user’s information | String |

Table 51 - Component Interface

### User Interface Design

#### Mobile Application

##### Login

Figure 74- <UI Design> Login

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Username | Fill username | No | Yes | TextView | String |
| 2 | Password | Fill password | No | Yes | TextView | String |

Table 52- <UI Design> Login Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Login | Login into the application | Required | Transfer to the Main Menu |
| 4 | Cancel | Exit the application | N/A | Exit to the mobile screen |

Table 53- <UI Design> Login Buttons

##### Main Menu

Figure 75- <UI Design> Main Menu

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Register Package & Box | Register Package & Box into system | N/A | Transfer to Register Package & Box Screen |
| 2 | Stock In | Stock in Box into system | N/A | Transfer to Stock In Screen |
| 3 | Stock Out | Stock out box system | N/A | Transfer to Stock Out Screen |
| 4 | Transfer Product | Transfer boxes between packages and package between cells | N/A | Transfer to Transfer Product Screen |
| 5 | Stocktake Inventory | Check the current quantity of product’s box in warehouse | N/A | Transfer to Stocktake Inventory Screen |
| 6 | Register Cell (Shelf) | Register Cell into system | N/A | Transfer to Register Cell Screen |

Table 54- <UI Design>Main Menu Button

##### Register Shelf

Figure 76- <UI Design> Register Shelf

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Shelf | Shelf to register | Yes | Yes | Spinner | String |
| 2 | Floor | Floor to register | Yes | Yes | Spinner | String |
| 3 | Cell | Cell to register | Yes | Yes | Spinner | String |
| 4 | Cell RFID | Cell RFID to register with Cell | Yes | Yes | TextView | String |

Table 55- <UI Design> Register Shelf Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Save | Register Cell into system | N/A | Message from system |
| 6 | Clear | Clear all filled information in screen | N/A | Cell RFID will be blank |
| 7 | Cancel | Exit Register Cell screen | N/A | Transfer to Main Menu |
| 8 | Scan Cell RFID | Scan cell RFID and fill into Cell RFID field | N/A | Cell RFID String |

Table 56- <UI Design>Register Shelf Buttons

##### Register Package and Box

Figure 77- <UI Design>Register Packages and Box

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Goods Receipt | Receipt Id to register | Yes | Yes | Spinner | String |
| 2 | Product Name | Product Name to register | Yes | Yes | Spinner | String |
| 3 | Product Id | Product Id to register | Yes | Yes | TextView | String |
| 4 | Package RFID | Package RFID to register | Yes | Yes | TextView | String |
| 5 | Box RFID | Boxes RFID to register | Yes | Yes | TextView | String |

Table 57- <UI Design> Register Package and Box Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Receipt Information | View Receipt Information | N/A | Transfer to Goods Receipt Screen |
| 7 | Scan Package RFID | Scan Package RFID and fill into Package RFID field | N/A | Package RFID String |
| 8 | Scan Box RFID | Scan Box RFID and fill into Box RFID field | N/A | Box RFID String |
| 9 | Save | Register package and boxes into system | N/A | Message from system |
| 10 | Clear | Clear all filled information in screen | N/A | Clear all RFID field and all spinner will select the first item |
| 11 | Cancel | Exit Register Package & Box screen | N/A | Transfer to to Main Menu |

Table 58- <UI Design> Register Package and Box Button

##### Goods Receipt

Figure 78- <UI Design>Goods Receipt

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Goods Receipt Information | Show all goods receipt info | Yes | Yes | RecyclerView | String |

Table 59- <UI Design> Goods Receipt Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Exit | Exit Goods Receipt Screen | N/A | Transfer to Register Package & Box screen |

Table 60- <UI Design>Goods Receipt Button

##### Stock In

Figure 79- <UI Design> Stock In

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Cell RFID | Cell RFID for stock in | No | Yes | TextView | String |
| 2 | Cell | Cell ID for stock in | No | Yes | TextView | String |
| 3 | Floor | Floor ID for stock in | No | Yes | TextView | String |
| 4 | Shelf | Shelf ID for stock in | No | Yes | TextView | String |
| 5 | Package RFID | Package RFID for stock in | No | Yes | TextView | String |

Table 61- <UI Design> Stock In Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Scan Cell RFID | Scan Cell RFID and fill into Cell RFID field | N/A | Cell RFID String |
| 7 | Scan Package RFID | Scan Package RFID and fill into Package RFID field | N/A | Package RFID String |
| 8 | Save | Stock in package into warehouse | N/A | Message from system |
| 9 | Clear | Clear all filled information in screen | N/A | Clear all RFID field and all spinner will select the first item |
| 10 | Cancel | Exit Stock in screen | N/A | Transfer to to Main Menu |

Table 62- <UI Design>Stock In Buttons

##### Stock out

Figure 80- <UI Design> Stock Out

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Goods Issue | Issue Id to stock out | No | Yes | TextView | String |
| 2 | Box Scanned Information | Information of box and quantity of scanned boxes | No | Yes | RecyclerView | String |

Table 63- <UI Design> Stock Out Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Scan Box RFID | Scan Box RFID and fill product information of this box into product field | N/A | Box RFID String and product information |
| 4 | Goods Issue Information | View Goods Issue Information | N/A | Transfer to Issue Invoice Screen |
| 5 | Save | Stock out package from warehouse | N/A | Message from system |
| 6 | Clear | Clear all filled information in screen | N/A | Clear all RFID field |
| 7 | Cancel | Exit Stock Out screen | N/A | Transfer to Main Menu |

Table 64- <UI Design> Stock Out Buttons

##### Goods Issue

Figure 81- <UI Design> Goods Issue

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Goods Issue Information | Show all goods issue info | Yes | Yes | RecyclerView | String |

Table 65- <UI Design>Goods Issue Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Exit | Exit Goods Issue Screen | N/A | Transfer to Stock Out screen |

Table 66- <UI Design> Goods Issue Button

##### Suggest Box

Figure 82- <UI Design> Suggest Box

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Suggest Information | Show information about suggest box | Yes | Yes | ExpandableListView | String |

Table 67- <UI Design> Suggest Box Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Exit | Exit Suggest Box Screen | N/A | Transfer to Goods Issue screen |

Table 68- <UI Design> Suggest Box Button

##### Transfer Box

Figure 83- <UI Design> Transfer Box

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Package RFID | Package RFID which will be transfer | No | Yes | TextView | String |
| 2 | Scanned box Information | Information of box and quantity | No | Yes | RecyclerView | String |

Table 69- <UI Design> Transfer Box Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Scan Package RFID | Scan Package RFID and fill into Package RFID field | N/A | Package RFID String |
| 4 | Scan Box RFID | Scan Box RFID and fill product information of this box into product field | N/A | Box RFID String and product information |
| 5 | Save | Transfer box between package | N/A | Message from system |
| 6 | Clear | Clear all filled information in screen | N/A | Clear all RFID field and all spinner will select the first item |
| 7 | Cancel | Exit Transfer Box screen | N/A | Transfer to Main Menu |
| 8 | Transfer Box | Switch to Transfer Box Tab | N/A | Transfer to Transfer Box Tab |
| 9 | Transfer Package | Switch to Transfer Package Tab | N/A | Transfer to Transfer Package Tab |

Table 70- <UI Design>Transfer Box Button

##### Transfer Package

Figure 84- <UI Design> Transfer Package

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Cell RFID | Cell RFID of target cell | No | Yes | TextView | String |
| 2 | Cell | Cell Id of target cell | No | Yes | TextView | String |
| 3 | Floor | Floor Id contain target cell | No | Yes | TextView | String |
| 4 | Shelf | Shelf Id contain target cell | No | Yes | TextView | String |
| 5 | Package RFID | Package RFID to transfer | No | Yes | TextView | String |

Table 71- <UI Design> Transfer Package Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Scan Cell RFID | Scan Cell RFID and fill into Cell RFID field | N/A | Cell RFID String |
| 7 | Scan Package RFID | Scan Package RFID and fill into Package RFID field | N/A | Package RFID String |
| 8 | Save | Transfer package between cells | N/A | Message from system |
| 9 | Clear | Clear all filled information in screen | N/A | Clear all RFID field |
| 10 | Cancel | Exit Transfer Package screen | N/A | Transfer to Main Menu |
| 11 | Transfer Box | Switch to Transfer Box Tab | N/A | Transfer to Transfer Box Tab |
| 12 | Transfer Package | Switch to Transfer Package Tab | N/A | Transfer to Transfer Package Tab |

Table 72- <UI Design> Transfer Package Buttons

##### Stocktake inventory

Figure 85- <UI Design> Stocktake Inventory

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Product Name | Product name to stocktake | No | Yes | TextView | String |
| 2 | Product Id | Product Id to stocktake | No | Yes | TextView | String |
| 3 | Available Quantity | Quantity archived in system | No | Yes | TextView | String |
| 4 | Scanned Quantity | Real quantity in warehouse | No | Yes | TextView | String |

Table 73- <UI Design> Stocktake Inventory Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Scan Box RFID | Scan Box RFID and fill into Box RFID field | N/A | Scanned quantity increase 1 |
| 6 | Report | Report stocktake result to the system | N/A | Message from system |
| 7 | Clear | Clear all filled information in screen and scanned quantity | N/A | Scanned quantity return zero and all spinner will select the first item |
| 8 | Cancel | Exit Stocktake Inventory screen | N/A | Transfer to to Main Menu |

Table 74- <UI Design> Stocktake Inventory Buttons

#### Web System

##### Login

Figure 86- <UI Design> Login

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Username | Fill username | No | Yes | TextView | String |
| 2 | Password | Fill password | No | Yes | TextView | String |

Table 75- <UI Design> Login Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Login | Login into the application | Required | Transfer to Home page |

Table 76- <UI Design> Login Button

##### Logout

Figure 87- <UI Design> Logout

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Logout | Logout from the application | Required | Transfer to the Login page |

Table 77- <UI Design>Logout Button

##### Admin Homepage

Figure 88- <UI Design> Admin Homepage

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Total Users | Total number of users | Yes | No | Text | String |
| 2 | Active Users | Total number of active users | Yes | No | Text | String |
| 3 | Total Accountant | Total number of active accountant | Yes | No | Text | String |
| 4 | Total Stock-keeper | Total number of active stock-keeper | Yes | No | Text | String |

Table 78- <UI Design>Admin Homepage Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Show All User | Show all user | N/A | Transfer to “All User” page |
| 6 | Show All Accountant | Show only user with accountant role | N/A | Transfer to “All Accountant” page |
| 7 | Show All Stock-keeper | Show only user with stock-keeper role | N/A | Transfer to “All Stock-keeper” page |

Table 79- <UI Design>Admin Homapage Buttons

##### View All User

Figure 89- <UI Design> View All User

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Fullname | Fullname of user | Yes | Yes | Text | String |
| 2 | Username | Username of user | Yes | Yes | Text | String |
| 3 | Password | Password of user | Yes | Yes | Text | String |
| 4 | Email | Email of user | Yes | Yes | Text | String |
| 5 | Phone | Phone number of user | Yes | No | Text | String |
| 6 | Role | Role of user | Yes | Yes | Text | String |
| 7 | Status | Status of user | Yes | Yes | Text | String |

Table 80- <UI Design> View All User Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Create User | Create new user | N/A | Transfer to create user’s screen |
| 9 | Edit User | Edit user | N/A | Transfer to edit user’s screen |
| 10 | Deactive User | Deactivate user’s status | N/A | Show deactivate user dialog |
| 11 | Active User | Active user’s status | N/A | Show activate user dialog |
| 12 | Search | Search users by keyword | N/A | Show list of users is filtered by keyword |
| 13 | Show pagination | Show pagination | N/A | Show pagination |
| 14 | Show data colums | Show columns are checked | N/A | Display all columns are checked |

Table 81- <UI Design>View All Users Buttons

##### View All Accountant

Figure 90- <UI Design> View All Accountant

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Fullname | Fullname of accountant | Yes | Yes | Text | String |
| 2 | Username | Username of accountant | Yes | Yes | Text | String |
| 3 | Password | Password of accountant | Yes | Yes | Text | String |
| 4 | Email | Email of accountant | Yes | Yes | Text | String |
| 5 | Phone | Phone number of accountant | Yes | No | Text | String |
| 6 | Role | Role of accountant | Yes | Yes | Text | String |
| 7 | Status | Status of accountant | Yes | Yes | Text | String |

Table 82- <UI Design> View All Accountant Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Edit User | Edit user | N/A | Transfer to edit user’s screen |
| 9 | Deactive User | Deactivate user’s status | N/A | Show deactivate user dialog |
| 10 | Active User | Active user’s status | N/A | Show activate user dialog |
| 11 | Search | Search users by keyword | N/A | Show list of users is filtered by keyword |
| 12 | Show pagination | Show pagination | N/A | Show pagination |
| 13 | Show data colums | Show columns are checked | N/A | Display all columns are checked |

Table 83- <UI Design>View All Accountant Buttons

##### View All Stock-keeper

Figure 91- <UI Design> View All Stock-keeper

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Fullname | Fullname of stockkeeper | Yes | Yes | Text | String |
| 2 | Username | Username of stockkeeper | Yes | Yes | Text | String |
| 3 | Password | Password of stockkeeper | Yes | Yes | Text | String |
| 4 | Email | Email of stockkeeper | Yes | Yes | Text | String |
| 5 | Phone | Phone number of stockkeeper | Yes | No | Text | String |
| 6 | Role | Role of stockkeeper | Yes | Yes | Text | String |
| 7 | Status | Status of stockkeeper | Yes | Yes | Text | String |

Table 84- <UI Design>View All Stock-keeper Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Edit User | Edit user | N/A | Transfer to edit user’s screen |
| 9 | Deactive User | Deactivate user’s status | N/A | Show deactivate user dialog |
| 10 | Active User | Active user’s status | N/A | Show activate user dialog |
| 11 | Search | Search users by keyword | N/A | Show list of users is filtered by keyword |
| 12 | Show pagination | Show pagination | N/A | Show pagination |
| 13 | Show data colums | Show columns are checked | N/A | Display all columns are checked |

Table 85- <UI Design>View All Stock-keeper Buttons

##### Create User

Figure 92- <UI Design>Create User

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Fullname | Fullname of user | No | Yes | Text | String |
| 2 | Username | Username of user | No | Yes | Text | String |
| 3 | Password | Password of user | No | Yes | Text | String |
| 4 | Email | Email of user | No | Yes | Text | String |
| 5 | Phone | Phone of user | No | No | Text | String |
| 6 | Role | Role of user | No | Yes | Select | String |
| 7 | Address | Address of user | No | No | Text | String |
| 8 | Date of birth | Date of birth of user | No | Yes | Text | Date |
| 9 | Note | Note of user | No | No | Text | String |

Table 86- <UI Design>Create User Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 10 | Cancel | Cancel create user process | N/A | Transfer to list all user’s screen |
| 11 | Confirm | Create new user successfully | Required | Transfer to list all user’s screen with new user that is created |

Table 87- <UI Design>Create User Buttons

##### Edit User

Figure 93- <UI Design>Edit User

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Fullname | Fullname of user | No | Yes | Text | String |
| 2 | Username | Username of user | No | Yes | Text | String |
| 3 | Password | Password of user | No | Yes | Text | String |
| 4 | Email | Email of user | No | Yes | Text | String |
| 5 | Phone | Phone of user | No | No | Text | String |
| 6 | Role | Role of user | No | Yes | Select | String |
| 7 | Address | Address of user | No | No | Text | String |
| 8 | Date of birth | Date of birth of user | No | Yes | Text | Date |
| 9 | Note | Note of user | No | No | Text | String |

Table 88- <UI Design>Edit User Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 10 | Cancel | Cancel edit user process | N/A | Transfer to list all user’s screen |
| 11 | Confirm | Edit user successfully | Required | Transfer to list all user’s screen with user that is updated |

Table 89- <UI Design>Edit User Buttons

##### Deactivate User

Figure 94- <UI Design> Deactivate User

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Deactivate User | Deactivate user’s status | N/A | Transfer to list all user’s screen with user’s status is deactivated |
| 2 | Cancel | Cancel deactivate user process | N/A | Transfer to list all user’s screen |

Table 90- <UI Design>Deactivate User Buttons

##### Activate User

Figure 95- <UI Design>Activate User

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Activate User | Activate user’s status | N/A | Transfer to list all user’s screen with user’s status is activated |
| 2 | Cancel | Cancel activate user process | N/A | Transfer to list all user’s screen |

Table 91- <UI Design>Activate User Button

##### Accountant Homepage

Figure 96- <UI Design>Accountant Homepage

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read-only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Total Categories | Total number of categories | Yes | No | Text | String |
| 2 | Total products | Total number of products | Yes | No | Text | String |
| 3 | Total Shelfs | Total number of shelf | Yes | No | Text | String |
| 4 | Total Vendors | Total number of vendor | Yes | No | Text | String |
| 5 | Pending Goods Receipt List | Total Goods Receipt with pending status | Yes | No | Text | String |
| 6 | Pending Goods Issue List | Total Goods Issue with pending status | Yes | No | Text | String |

Table 92- <UI Design>Accountant Homepage Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 7 | Show All Goods Receipt and Issue | Show all goods receipt and issue | N/A | Transfer to “Receipts/Issues” page |
| 8 | Show Categories | Show all categories | N/A | Transfer to “Categories” page |
| 9 | Show All Products | Show all products | N/A | Transfer to “Products” page |
| 10 | Show All Shelves | Show all shelves | N/A | Transfer to “Shelves” page |
| 11 | Show All Stocktake History | Show all stocktake history | N/A | Transfer to “Stocktake Histories” page |
| 12 | Show Report Type | Show all report type | N/A | Transfer to “Reports” page |

Table 93- <UI Design>Accountant Homepage Button

##### View Category

Figure 97- <UI Design>View Category

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Category Name | Category name of category | Yes | Yes | Text | String |
| 2 | Description | Description of category | Yes | No | Text | String |
| 3 | Status | Status of category | Yes | Yes | Text | String |

Table 94- <UI Design>View Category Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | Create Category | Create new category | N/A | Transfer to create category’s screen |
| 5 | Edit Category | Edit category | N/A | Transfer to edit category’s screen |
| 6 | Deactivate Category | Deactivate category’s status | N/A | Show deactivate category dialog |
| 7 | Active Category | Active category’s status | N/A | Show activate category dialog |
| 8 | Search | Search category by keyword | N/A | Show list of categories is filtered by keyword |
| 9 | Show pagination | Show pagination | N/A | Show pagination |
| 10 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 95- <UI Design>View Category Button

##### Create Category

Figure 98- <UI Design>Create Category

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Category Name | Category name of category | No | Yes | Text | String |
| 2 | Description | Description of category | No | No | Text | String |

Table 96- <UI Design> Create Category Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Cancel | Cancel create category process | N/A | Transfer to list all category’s screen |
| 4 | Confirm | Create new category successfully | N/A | Transfer to list all category’s screen with new category that is created |

Table 97- <UI Design>Create Category Button

##### Edit Category

Figure 99- <UI Design> Edit Category

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Category Name | Category name of category | No | Yes | Text | String |
| 2 | Description | Description of category | No | No | Text | String |

Table 98- <UI Design>Edit Category Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Cancel | Cancel edit category process | N/A | Transfer to list all category’s screen |
| 2 | Confirm | Edit category successfully | N/A | Transfer to list all category’s screen with new category that is updated |

Table 99- <UI Design>Edit Category Buttons

##### Deactivate Category

Figure 100- <UI Design> Deactivate Category

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Deactivate Category | Deactivate category’s status | N/A | Transfer to list all category’s screen with category’s status is deactivated |
| 2 | Cancel | Cancel deactivate category process | N/A | Transfer to list all category’s screen |

Table 100- <UI Design> Deactivate Category Buttons

##### Activate Category

Figure 101- <UI Design> Activate Category

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Activate Category | Activate category’s status | N/A | Transfer to list all category’s screen with category’s status is activated |
| 2 | Cancel | Cancel activate category process | N/A | Transfer to list all category’s screen |

Table 101- <UI Design>Activate Category Buttons

##### View All Product

Figure 102- <UI Design>View All Product

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Product Id | Id of product | Yes | Yes | Text | String |
| 2 | Product Name | Name of product | Yes | Yes | Text | String |
| 3 | Inage | Image of product | Yes | No | Text | String |
| 4 | Category | Category of product | Yes | Yes | Text | String |
| 5 | Quantity | Box’s quantity of product | Yes | Yes | Text | Integer |
| 6 | Quantity per box | Quantity product item per product’s box | Yes | Yes | Text | Integer |
| 7 | Status | Status of product | Yes | Yes | Text | String |

Table 102- <UI Design>View All Product Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Create Product | Create new user | N/A | Transfer to create product’s screen |
| 9 | Import Product | Import product’s list excel file | N/A | Transfer to import product’s screen |
| 10 | Edit Product | Edit product | N/A | Transfer to edit user’s product |
| 11 | Deactivate Product | Deactivate product’s status | N/A | Show deactivate product dialog |
| 12 | Active Product | Active product’s status | N/A | Show activate product dialog |
| 13 | Search | Search users by keyword | N/A | Show list of products are filtered by keyword |
| 14 | Show pagination | Show pagination | N/A | Show pagination |
| 15 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 103- <UI Design>View All Product Buttons

##### Create Product

Figure 103- <UI Design> Create Product

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Product Id | Id of product | No | Yes | Text | String |
| 2 | Product Name | Name of product | No | Yes | Text | String |
| 3 | Image | Image of product | No | No | File | String |
| 4 | Description | Description of product | No | No | Text | String |
| 5 | Weight | Weight of product | No | No | Text | Float |
| 6 | Height | Height of product | No | No | Text | Float |
| 7 | Width | Width of product | No | No | Text | Float |
| 8 | Length | Length of product | No | No | Text | Float |
| 9 | Quantity per box | Quantity product item per product’s box | No | Yes | Text | Integer |
| 10 | Category | Category of product | No | Yes | Select | String |
| 11 | Vendor | Vendor of product | No | Yes | Select | String |

Table 104- <UI Design>Create Product Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 12 | Cancel | Cancel create product process | N/A | Transfer to list all product’s screen |
| 13 | Confirm | Create new product successfully | N/A | Transfer to list all product’s screen with new product which was created |

Table 105- <UI Design>Create Product Buttons

##### Edit Product

Figure 104- <UI Design>Edit Product

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Product Id | Id of product | Yes | Yes | Text | String |
| 2 | Product Name | Name of product | No | Yes | Text | String |
| 3 | Image | Image of product | No | No | File | String |
| 4 | Description | Description of product | No | No | Text | String |
| 5 | Weight | Weight of product | No | No | Text | Float |
| 6 | Height | Height of product | No | No | Text | Float |
| 7 | Width | Width of product | No | No | Text | Float |
| 8 | Length | Length of product | No | No | Text | Float |
| 9 | Quantity per box | Quantity product item per product’s box | No | Yes | Text | Integer |
| 10 | Category | Category of product | No | Yes | Select | String |
| 11 | Vendor | Vendor of product | No | Yes | Select | String |

Table 106- <UI Design>Edit Product Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 12 | Cancel | Cancel edit product process | N/A | Transfer to list all product’s screen |
| 13 | Confirm | Edit product successfully | N/A | Transfer to list all product’s screen with product that was edited |

Table 107- <UI Design>Edit Product Buttons

##### Deactivate Product

Figure 105- <UI Design> Deactivate Product

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Deactivate Product | Deactivate product’s status | N/A | Transfer to list all product’s screen with product’s status is deactivated |
| 2 | Cancel | Cancel deactivate product process | N/A | Transfer to list all product’s screen |

Table 108- <UI Design>Deactivate Product Buttons

##### Activate Product

Figure 106- <UI Design> Activate Product

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Activate Product | Activate product’s status | N/A | Transfer to list all product’s screen with product’s status is activated |
| 2 | Cancel | Cancel activate product process | N/A | Transfer to list all product’s screen |

Table 109- <UI Design>Activate Product Buttons

##### Import Product

Figure 107- <UI Design> Import Product

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Upload File | File list of products | No | Yes | File | String |

Table 110- <UI Design>Import Product Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Sample File | Download sample file | N/A | A sample file will be downloaded |
| 3 | Confirm | Import product successfully | N/A | Transfer to list all product’s screen with products that are created |
| 4 | Cancel | Cancel import product process | N/A | Transfer to list all product’s screen |

Table 111- <UI Design>

##### Detail Product

Figure 108- <UI Design>Detail Product

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Create Product | Create new user | N/A | Transfer to create user’s screen |
| 2 | Edit Product | Edit product | N/A | Transfer to edit user’s product |
| 3 | Search | Search users by keyword | N/A | Show list of products are filtered by keyword |

Table 112- <UI Design>Detail Product Buttons

##### View Shelf

Figure 109- <UI Design>View Shelf

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Shelf Id | Shelf id of shelf | Yes | Yes | Text | String |
| 2 | Description | Description of shelf | Yes | No | Text | String |
| 3 | Floor Number | Number of floor of shelf | Yes | Yes | Text | Integer |
| 4 | Cell Number | Number of cell per floor | Yes | Yes | Text | String |
| 5 | Height | Height of cell | Yes | No | Text | Integer |
| 6 | Width | Width of cell | Yes | No | Text | Integer |
| 7 | Length | Length of cell | Yes | No | Text | Integer |
| 8 | Status | Status of Shelf | Yes | Yes | Text | String |
| 9 | Number of empty cell | Number empty cell in shelf | Yes | Yes | Text | Integer |

Table 113- <UI Design>View Shelf Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 10 | Create Shelf | Create new shelf | N/A | Transfer to create shelf’s screen |
| 11 | Edit Standard Shelf Size | Edit standard floor number and cell number | N/A | Transfer to edit shelf’s size screen |
| 12 | Edit Shelf | Edit shelf | N/A | Transfer to edit shelf’s screen |
| 13 | Deactivate Shelf | Deactivate shelf’s status | N/A | Show deactivate shelf dialog |
| 14 | Activate Shelf | Activate shelf’s status | N/A | Show activate shelf dialog |
| 15 | Search | Search shelves by keyword | N/A | Show list of shelves are filtered by keyword |
| 16 | Show pagination | Show pagination | N/A | Show pagination |
| 17 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 114- <UI Design>View Shelf Button

##### Create Shelf

Figure 110- <UI Design>Create Shelf

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Shelf Id | Shelf id of shelf | No | Yes | Text | String |
| 2 | Description | Description of product | No | No | Text | String |
| 3 | Floor Number | Number floor of shelf | Yes | Yes | Text | Integer |
| 4 | Cell Number | Description of shelf | Yes | Yes | Text | String |
| 5 | Coordinate X | Position of shelf in horizontal | No | Yes | Text | Integer |
| 6 | Coordinate Y | Position of shelf in vertical | No | Yes | Text | Integer |
| 7 | Height | Height of shelf | No | No | Text | Integer |
| 8 | Width | Width of shelf | No | No | Text | Integer |
| 9 | Length | Length of shelf | No | No | Text | Integer |

Table 115- <UI Design>Create Shelf Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 10 | Cancel | Cancel create shelf process | N/A | Transfer to list all shelf’s screen |
| 11 | Confirm | Create new shelf successfully | N/A | Transfer to list all shelf’s screen with new shelf that is created |

Table 116- <UI Design>Create Shelf Buttons

##### Edit Shelf

Figure 111- <UI Design> Edit Shelf

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Shelf Id | Shelf id of shelf | Yes | Yes | Text | String |
| 2 | Description | Description of product | No | No | Text | String |
| 3 | Floor Number | Number floor of shelf | Yes | Yes | Text | Integer |
| 4 | Cell Number | Description of shelf | Yes | No | Text | String |
| 5 | Coordinate X | Position of shelf in horizontal | No | Yes | Text | Integer |
| 6 | Coordinate Y | Position of shelf in vertical | No | Yes | Text | Integer |
| 7 | Height | Height of shelf | No | No | Text | Integer |
| 8 | Width | Width of shelf | No | No | Text | Integer |
| 9 | Length | Length of shelf | No | No | Text | Integer |

Table 117- <UI Design>Edit Shelf Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 10 | Cancel | Cancel edit shelf process | N/A | Transfer to list all shelf’s screen |
| 11 | Confirm | Edit shelf successfully | N/A | Transfer to list all shelf’s screen with new shelf that is edited |

Table 118- <UI Design> Edit Shelf Buttons

##### Deactivate Shelf

Figure 112- <UI Design> Deactivate Shelf

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Deactivate Shelf | Deactivate shelf’s status | N/A | Transfer to list all shelf’s screen with shelf’s status is deactivated |
| 2 | Cancel | Cancel deactivate shelf process | N/A | Transfer to list all shelf screen |

Table 119- <UI Design>Deactivate Shelf Buttons

##### Activate Shelf

Figure 113- <UI Design> Activate Shelf

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Activate Shelf | Activate shelf’s status | N/A | Transfer to list all shelf’s screen with shelf’s status is activated |
| 2 | Cancel | Cancel activate shelf process | N/A | Transfer to list all shelf screen |

Table 120- <UI Design> Activate Shelf Button

##### Detail Shelf

Figure 114- <UI Design>Detail Shelf

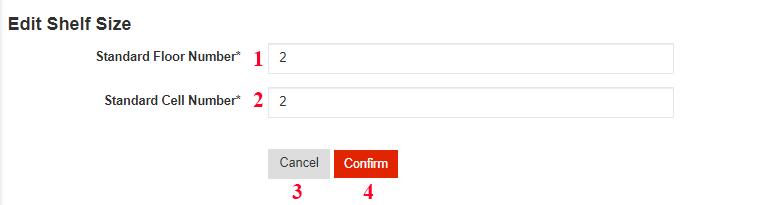
* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Floor Id | Floor Id of shelf | Yes | Yes | Text | String |
| 2 | Cell Id | Cell Id of shelf | Yes | No | Text | String |
| 3 | Cell RFID | RFID tags in cell | Yes | Yes | Text | String |
| 4 | Status | Status of cell | Yes | No | Text | String |

Table 121- <UI Design>Detail Shelf Fields

##### Edit Standard Shelf Size

Figure 115- <UI Design>Edit Standard Shelf Size



* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Standard Floor Number | Floor number of shelf | No | Yes | Text | Integer |
| 2 | Standard Cell Number | Cell number per Floor | No | Yes | Text | Integer |

Table 122- <UI Design> Edit Standard Shelf Size Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Cancel | Cancel edit shelf size process | N/A | Transfer to list all shelf’s screen |
| 4 | Confirm | Edit shelf size successfully | N/A | Transfer to list all shelf’s screen with new shelf size that is edited |

Table 123- <UI Design>Edit Standard Shelf Size Buttons

##### View Goods Receipt

Figure 116- <UI Design> View Goods Receipt

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Receipt Id | Id of receipt | Yes | Yes | Text | String |
| 2 | Date | Creation date of receipt | Yes | Yes | Text | Date |
| 3 | Status | Status of receipt | Yes | Yes | Text | String |
| 4 | Created by | Creator of receipt | Yes | Yes | Text | String |

Table 124- <UI Design>View Goods Receipt Fields

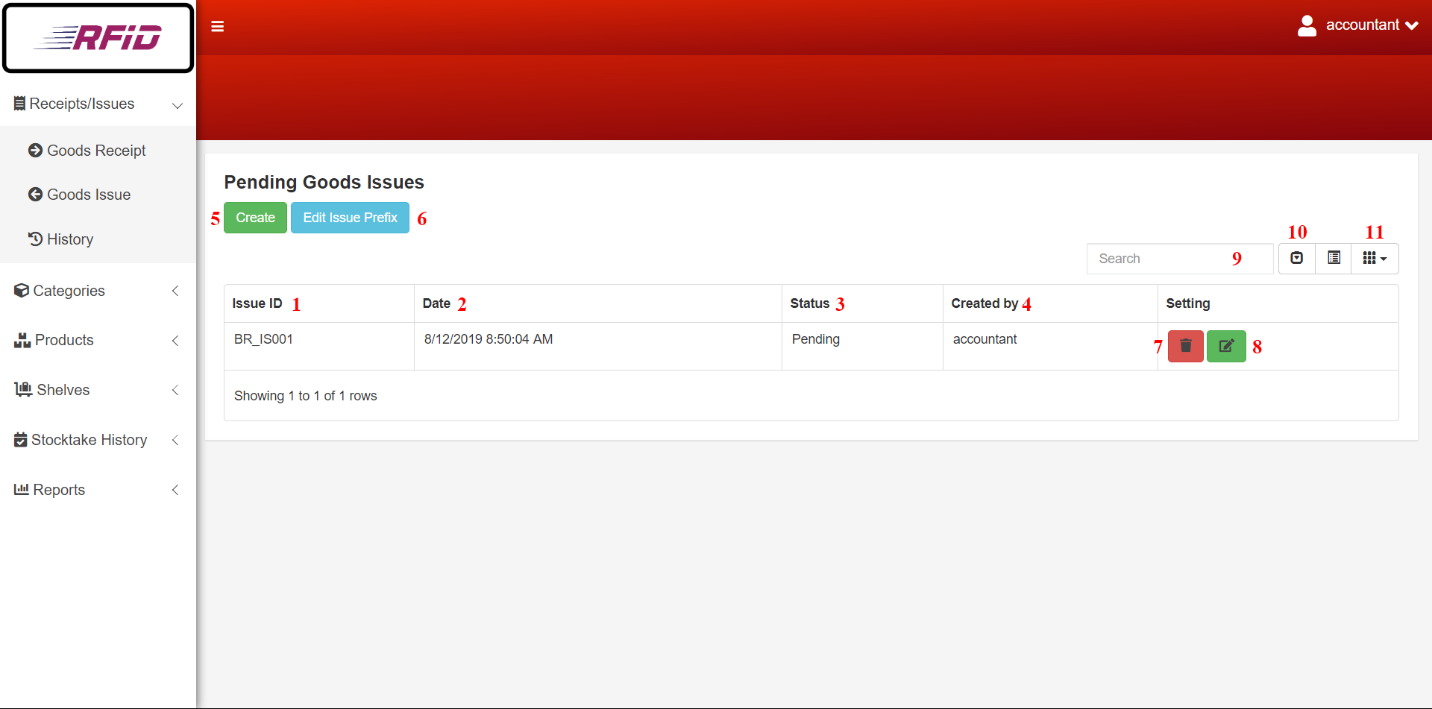
* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Create Receipt | Create new receipt | N/A | Transfer to create receipt screen |
| 6 | Edit Prefix | Edit prefix of receipt | N/A | Transfer to edit prefix screen |
| 7 | Delete receipt | Delete receipt | N/A | Show delete receipt dialog |
| 8 | Edit Receipt | Edit receipt | N/A | Transfer to edit receipt screen |
| 9 | Search | Search receipt by keyword | N/A | Show list of receipt are filtered by keyword |
| 10 | Show pagination | Show pagination | N/A | Show pagination |
| 11 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 125- <UI Design>View Goods Receipt Buttons

##### View Goods Issue

Figure 116- <UI Design> View Goods Issue



* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Issue Id | Id of issue | Yes | Yes | Text | String |
| 2 | Date | Creation date of issue | Yes | Yes | Text | Date |
| 3 | Status | Status of issue | Yes | Yes | Text | String |
| 4 | Created by | Creator of issue | Yes | Yes | Text | String |

Table 126- <UI Design>View Goods Issue Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Create Issue | Create new issue | N/A | Transfer to create issue screen |
| 6 | Edit Prefix | Edit prefix of issue | N/A | Transfer to edit prefix screen |
| 7 | Delete Issue | Delete issue | N/A | Show delete issue dialog |
| 8 | Edit Issue | Edit issue | N/A | Transfer to edit issue screen |
| 9 | Search | Search issue by keyword | N/A | Show list of issues are filtered by keyword |
| 10 | Show pagination | Show pagination | N/A | Show pagination |
| 11 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 127- <UI Design>View Goods Issue Buttons

##### View Invoice History

Figure 116- <UI Design> View Invoice History

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Invoice Id | Invoice id of invoice | Yes | Yes | Text | String |
| 2 | Date | Creation date of invoice | Yes | Yes | Text | Date |
| 3 | Status | Status of invoice | Yes | Yes | Text | String |
| 4 | Created by | Creator of invoice | Yes | Yes | Text | String |

Table 128- <UI Design>View Invoice Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Search | Search invoice by keyword | N/A | Show list of invoices are filtered by keyword |
| 6 | Show pagination | Show pagination | N/A | Show pagination |
| 7 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 129- <UI Design>View Invoice Buttons

##### Edit Prefix

Figure 117- <UI Design>Edit Prefix

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Prefix | Prefix string of invoice Id | No | Yes | Text | String |

Table 130- <UI Design> Edit Prefix Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Cancel | Cancel edit prefix process | N/A | Transfer to view goods receipt or to view goods issue screen |
| 3 | Confirm | Edit Prefix of Invoice | N/A | Transfer to view goods receipt or to view goods issue screen |

Table 131- <UI Design> Edit Prefix Buttons

##### Create Invoice

Figure 118- <UI Design> Create Invoice [2]

Figure 119- <UI Design> Create Invoice [1]

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Product Id | Product id of product | Yes | Yes | Text | String |
| 2 | Product Name | Product name of product | Yes | Yes | Text | String |
| 3 | Category | Category of product | Yes | Yes | Text | String |
| 4 | Vendor | Vendor of product | Yes | Yes | Text | String |
| 5 | Quantity | Quantity of product | No | Yes | Text | Integer |

Table 132- <UI Design> Create Invoice Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Choose Product | Choose products to add into invoice | N/A | Transfer to choose product screen |
| 7 | Cancel | Cancel create invoice process | N/A | Transfer to list all invoice screen |
| 8 | Finish | Create new invoice successfully | N/A | Transfer to list all invoices’ screen with new invoice that is created |

Table 133- <UI Design> Create Invoice Buttons

##### Choose Product

Figure 120- <UI Design>Choose Product

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Checkbox | Checkboxes for choose products | No | Yes | Checkbox | Boolean |
| 2 | Available Quantity | Available Quantity in Warehouse. Only appear in Stock out | Yes | Yes | Text | String |

Table 134- <UI Design>Choose Product Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Confirm | Add chosen products to invoice | N/A | Transfer to add product to invoice screen |
| 4 | Search | Filter Product list with key | N/A | Filtered product list |
| 5 | Cancel | Cancel choosing products | N/A | Transfer to add product to invoice screen |

Table 135- <UI Design>Choose Product Button

##### Edit Invoice

Figure 121- <UI Design>Edit Invoice

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Quantity | Quantity of product | No | Yes | Text | Integer |

Table 136- <UI Design>Edit Invoice Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Choose Product | Choose products for edit invoice | N/A | Transfer to choose product screen |
| 3 | Cancel | Cancel edit invoice process | N/A | Transfer to list all invoice screen |
| 4 | Finish | Edit invoice successfully | N/A | Transfer to list all invoices’ screen with edited invoice |

Table 137- <UI Design>Edit Invoice Button

##### Delete Invoice

Figure 122- <UI Design>Delete Invoice

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Delete | Delete invoice | N/A | Transfer to list all invoice screen and remove this invoice |
| 2 | Cancel | Cancel delete invoice process | N/A | Transfer to list all invoice screen |

Table 138- <UI Design>Delete Invoice Button

##### Detail Invoice

Figure 123- <UI Design>Detail Invoice

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Date | Date of invoice | Yes | Yes | Text | Date |
| 2 | Type | Type of invoice | Yes | Yes | Text | String |
| 3 | Status | Status of invoice | Yes | Yes | Text | String |
| 4 | Description | Description of invoice | Yes | Yes | Text | String |
| 5 | User create | User that create invoice | Yes | Yes | Text | String |
| 6 | Product Id | Product id of product | Yes | Yes | Text | String |
| 7 | Product Name | Product name of product | Yes | Yes | Text | String |
| 8 | Category | Category of product | Yes | Yes | Text | String |
| 9 | Vendor | Vendor of product | Yes | Yes | Text | String |
| 10 | Quantity | Quantity of product | Yes | Yes | Text | Integer |

Table 139- <UI Design>Detail Invoice Fields

##### Create Report

Figure 124- <UI Design>Create Report

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Good Receive/Issue | Good receive/issue report | N/A | Transfer to good receive/issue pdf layout |
| 2 | Product | Product report | N/A | Transfer to good product pdf layout |
| 3 | Category | Category report | N/A | Transfer to category pdf layout |
| 4 | Vendor | Vendor report | N/A | Transfer to vendor pdf layout |

Table 140- <UI Design>Create Report Buttons

##### Awaiting Confirm Stocktake

Figure 125- <UI Design>Awaiting Confirm Stocktake

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Date | Creation date of stocktake | Yes | Yes | Text | Date |
| 2 | Product Id | Product Id of stocktake | Yes | Yes | Text | String |
| 3 | Product Name | Product Name of stocktake | Yes | Yes | Text | String |
| 4 | Username | Stocktake creator | Yes | Yes | Text | String |
| 5 | Status | Status of invoice | Yes | Yes | Text | String |
| 6 | Stocktake Quantity | Stocktaked quantity of stocktake | Yes | Yes | Test | Integer |
| 7 | Available Quantity | Available quantity of product | Yes | Yes | Text | Integer |

Table 141- <UI Design> Awaiting Confirm Stocktake Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Confirm stocktake | Confirm stocktake status | N/A | Transfer to “Update Stocktake Status”screen |
| 9 | Search | Search stocktake by keyword | N/A | Show list of stocktake are filtered by keyword |
| 10 | Show pagination | Show pagination | N/A | Show pagination |
| 11 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 142- <UI Design> Awaiting Confirm Stocktake Buttons

##### History of Stocktake

Figure 125- <UI Design>History of Stocktake

* **Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 2 | Product Id | Product Id of stocktake | Yes | Yes | Text | String |
| 3 | Product Name | Product Name of stocktake | Yes | Yes | Text | String |
| 4 | Username | Stocktake creator | Yes | Yes | Text | String |
| 5 | Status | Status of invoice | Yes | Yes | Text | String |
| 6 | Stocktake Quantity | Stocktaked quantity of stocktake | Yes | Yes | Test | Integer |
| 7 | Available Quantity | Available quantity of product | Yes | Yes | Text | Integer |

Table 143- <UI Design> History of Stocktake Fields

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Search | Search stocktake by keyword | N/A | Show list of stocktake are filtered by keyword |
| 9 | Show pagination | Show pagination | N/A | Show pagination |
| 10 | Show data columns | Show columns are checked | N/A | Display all columns are checked |

Table 144- <UI Design> History of Stocktake Buttons

##### Update Stocktake Status

Figure 126- <UI Design>Update Stocktake Status

* **Button**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Update | Update stocktake status | N/A | Transfer to list all stocktake screen and update this stocktake status |
| 2 | Cancel | Cancel update stocktake status process | N/A | Transfer to list all stocktake screen |

Table 145- <UI Design>Update Stocktake Status Buttons

## Database Design

### Entity Relationship Diagram (ERD)

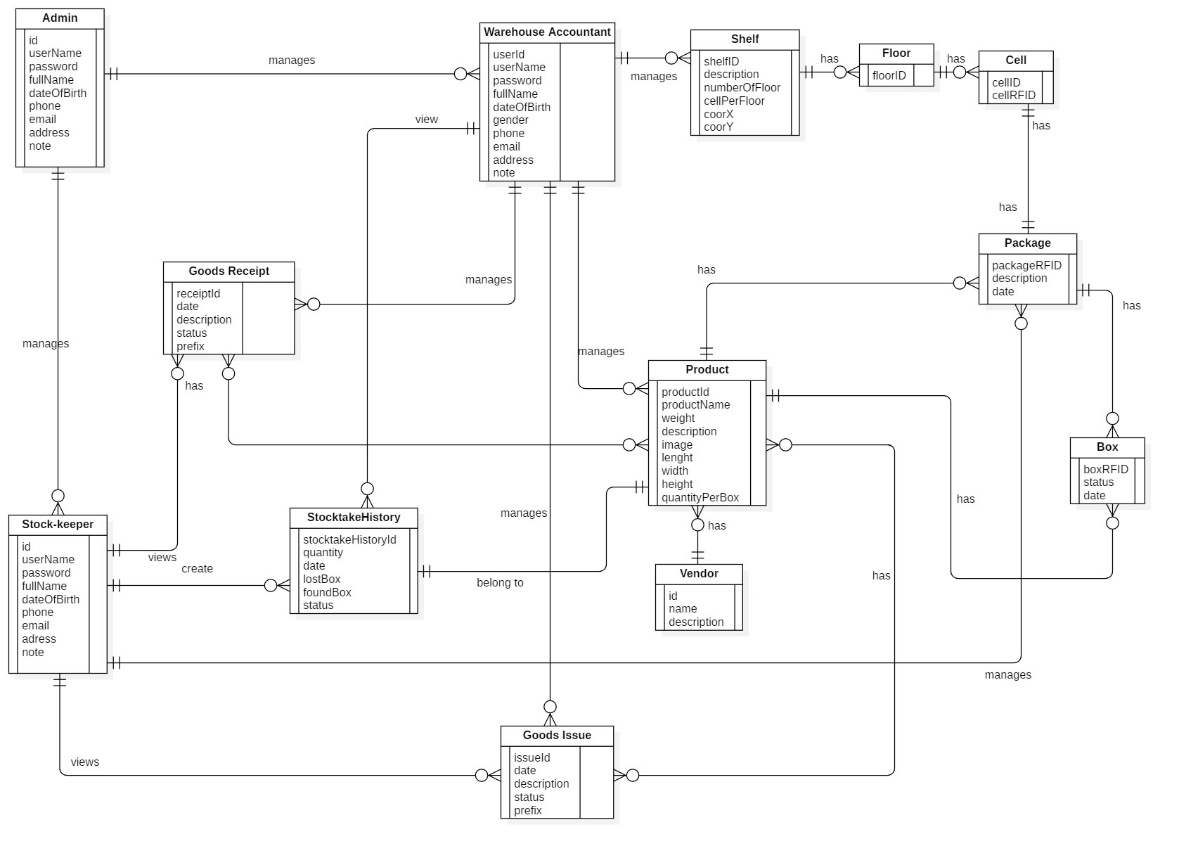


Figure 127 - Entity Relationship Diagram

### Entity Dictionary

| **Entity Data Dictionary: Describe content of all entities** | | |
| --- | --- | --- |
| **Entity Name** | **Mapping entity with conceptual diagram** | **Description** |
| Admin | Admin | Contains information of admin. |
| Stock-keeper | Stock-keeper | Contains information of stock-keeper. |
| Warehouse Accountant | Warehouse Accountant | Contains information of warehouse accountant. |
| GoodsReceipt | GoodsReceipt | Contains information of goods receipt. |
| GoodsIssue | GoodsIssue | Contains information of goods issue. |
| Product | Product | Contains information of product. |
| StocktakeHistory | Stocktake History | Contains information of stocktake history. |
| Vendor | Vendor | Contains information of vendor. |
| Shelf | Shelf | Contains information of shelf. |
| Floor | Floor | Contains information of floor. |
| Cell | Cell | Contains information of cell. |
| Package | Package | Contains information of package. |
| Box | Box | Contains information of box. |

Table 146 - Entity Data Dictionary

## Algorithms

### Find box base on the longest stock in date and closest warehouse’s gate

#### Definition

This algorithm intended to suggest stock-keeper the suitable box when they need to find box to stock out.

#### Problem Definition

When stock-keeper get an issue invoice, the stock-keeper do not know where to pick the box to stock out. In addition, a box have longest stock in date need to be stock out first but stock-keeper also do not know about this. This algorithm, provide information that help stock-keeper can choose the closest box and have longest date in warehouse.

#### Solution

To suggest box information for stock-keeper, we use weighted mean formula to solve this problem:­

**With:**

**: the weighted mean**

**x: the value of criteria**

**w: the weighted number**

In current problem, we have two criteria need to consider there are:

- The first criteria is the difference days between stock in date and now.

- The second criteria is the distance from source shelf to target shelf. (Because the farther away the distance, the lower priority so that the distance value need to be multiply with -1)

The priority of shelves is based on weighted mean values. The shelf that have larger weighted mean is better.

#### Example

**We have list shelves:**

|  |  |  |
| --- | --- | --- |
| **Criteria**  **Shelf** | **Days (70%)** | **Distance (30%)** |
| A | 20 | 10 |
| B | 25 | 8 |
| Shelf C | 20 | 8 |

**Calculate weighted mean values:**

**Order weighted mean values descending:**

|  |  |
| --- | --- |
| **Shelf** | **Weighted Average Number** |
| **B** | **15.1** |
| **C** | **11.6** |
| **A** | **11** |

#### Complexity

In total, the complexity of this algorithm is O(n).

#### Flowchart

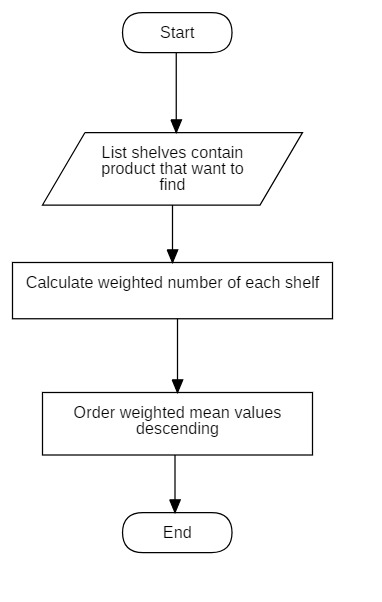
****

Figure 128<Algorithm> Optimizing Stock Out Product Flowchart

### Convert shelves’ coordinates to matrix algorithm

#### Definition

This algorithm converts shelves’ coordinates into matrix.

#### Definition Problem

Each shelf in the warehouse has its coordinates and we need to convert them into matrix for Dijkstra algorithm to analyze.

#### Solution

To convert all shelves’ coordinates into matrix, we follow these steps:

* Assume the coordinates of two shelves are (XA, YA), (XB, YB), distance between two column is x and distance between two row is y
* Compare the coordinates:
* If XA = XB and |YA – YB| = 1: two shelves are adjacent with distance between two shelves is x
* If XA = XB and |YA – YB| > 1: two shelves are not adjacent
* If XA = XB and |YA – YB| = 0: two shelves are one
* If YA = YB and |XA – XB| = 1: two shelves are adjacent with distance between two shelves is y
* If YA = YB and |XA – XB| > 1: two shelves are not adjacent
* If YA = YB and |XA – XB| = 0: two shelves are one

**Example:**

Assume thatdistance between two columns is x = 2 and distance between two rows is y = 1



Figure 129- <Algorithm> Shelf Coordinate Map

After calculating:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A1** | **A2** | **A3** | **B1** | **B2** | **B3** | **C1** | **C2** | **C3** |
| **A1** | **0** | **2** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **A2** | **2** | **0** | **2** | **0** | **1** | **0** | **0** | **0** | **0** |
| **A3** | **0** | **2** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **B1** | **1** | **0** | **0** | **0** | **2** | **0** | **1** | **0** | **0** |
| **B2** | **0** | **1** | **0** | **2** | **0** | **2** | **0** | **1** | **0** |
| **B3** | **0** | **0** | **1** | **0** | **2** | **0** | **0** | **0** | **1** |
| **C1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **2** | **0** |
| **C2** | **0** | **0** | **0** | **0** | **1** | **0** | **2** | **0** | **2** |
| **C3** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **2** | **0** |

Table 147- <Algorithm> Shelf Coordinate Mapping Table

#### Complexity

In total, the complexity of this algorithm is O(n2).

#### Flowchart

Figure 130- <Algorithm> Shelf Coordinate Mapping Flowchart

### Dijkstra algorithms

#### Definition

This algorithm is used to find the shortest path.

#### Problem Definition

Our use of this algorithm is to calculate the shortest distance from the source shelf (1; 1) to the other shelves.

#### Solution

In order to calculate the shortest distance, we use Dijkstra algorithm.

Create a list ListVertices which contains all vertices that already had been calculated the shortest distance.

Set the source vertex 0 and distance equals 0. It will be picked first and added to ListVertices;

While ListVertices does not contain all vertexes:

* Choose a vertice with minimum distance value.
* Add it to ListVertices.
* Find its adjacent vertices and update distance value of its adjacent vertice. (If adjacent vertice has the distance value already, compare the new distance with the old distance. If the new distance is smaller, update the new distance, if the new distance is smaller, keep the old distance)

#### Example

We have the following example:

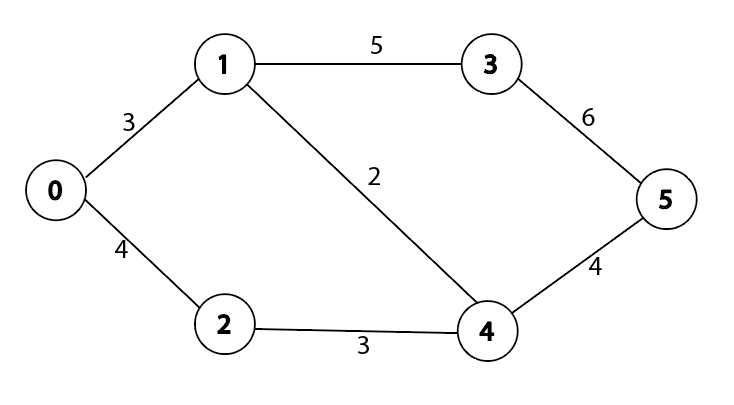


Figure 131- <Algorithm> Dijkstra Example

First we choose 0 as the source vertex, add it into ListVertices. Its adjacent is 1 and 2 and the distance value are updated as 3 and 4.

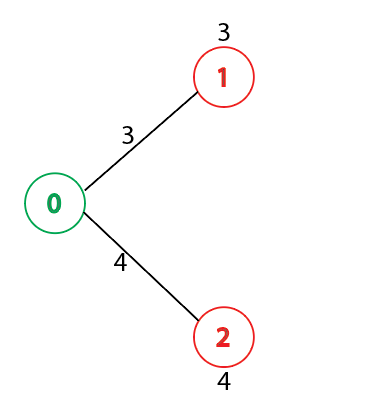


Figure 132- <Algorithm> Dijkstra Example

Pick the vertex with minimum distance value and add it to ListVertices. In the current situation vertex with minimum distance is 1 and ListVertices become {0, 1}. Update the distance value of vertex 1’s adjacent vertices are vertex 3 and vertex 4 as 8 and 5.

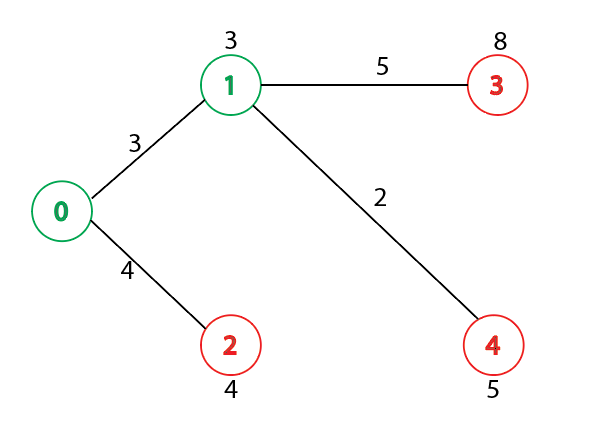


Figure 133- <Algorithm> Dijkstra Example

Pick the next vertex with minimum distance value and add it to ListVertices and the ListVertices becomes {0, 1, 2}. We pick vertex 2 with distance value is 4. The adjacent of vertex 2 is vertex 4 but the distance from vertex 2 to vertex 4 (4 + 3 = 7) is larger than old distance (5) so we keep old distance.

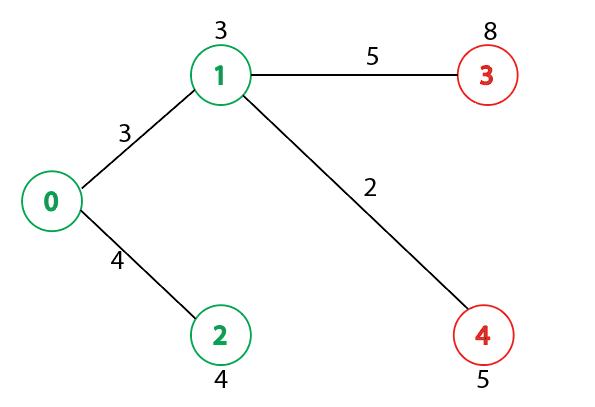


Figure 134- <Algorithm> Dijkstra Example

Iterative it until all ListVertices contain all of vertices we have the shortest distance value from source vertex to other vertices.

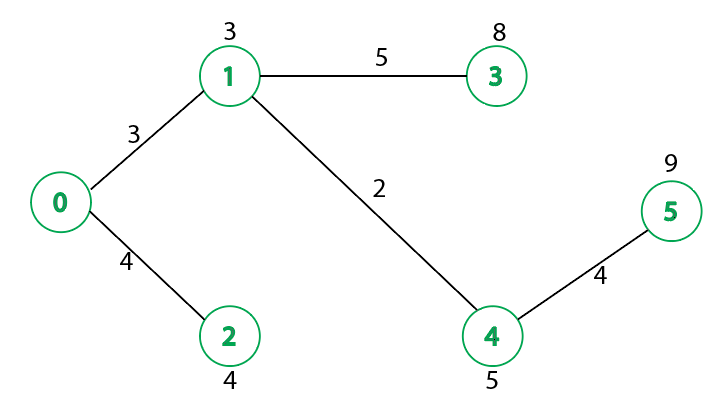


Figure 135- <Algorithm> Dijkstra Example

#### Complexity

In total, the comlexity of this algorithm is O(n2)

#### C:\Users\Ryan\Desktop\Chart\Dijkstra.jpgFlowchart

Figure 136- <Algorithm> Dijkstra Flowchart