Users:

Super admin

Subscribers/User/Client

Staff

Pages:

**Application Roles Management**

**Description**: As a super user I want to manage a roles for the application that I can create, update, view, delete and can see the list of roles of the application in order for me to assign a specific role of the application to the client/user.

**Database table fields:** application\_roles = (fields(id, application\_role\_name), created\_at, updated\_at))

**Features:**

Create

Update

View

Delete

Application roles list

**Acceptance criteria:**

* Create a validation in the front-end and back-end
* Application role name(required, will not accept numbers, max length 20, will not accept special characters)

**Applications Management**

**Description**: As an super user I want to manage an application for the client/user

**Database table fields:** applications = (fields(id, application\_name), created\_at, updated\_at))

**Features:**

Create

Update

View

Delete

Application list

**Acceptance criteria:**

* + - Create a validation in the front-end and back-end
  + Application name(required, will not accept numbers, max length 20, will not accept special characters)

**Features Management**

**Description**: As an super user I want to manage a features for a specific application

**Database table fields:** features = (fields(id, application\_id, feature\_name, created\_at, updated\_at))

**Features:**

Create

Update

View

Delete

Featureslist

**Acceptance criteria:**

* + Create a validation in the front-end and back-end
  + Create a dropdown in form with a choices of Applications(From DB)
  + application\_id must be from the dropdown
  + Feature name(required, will not accept numbers, max length 20, will not accept special characters)

**Users Management**

**Description**: As a super user I want to manage a users/clients where I can add, update, and delete an information and assign an application they can access based on the application management. Also I want to monitor their subscriptions and can also add and update

**Database table fields:**

users = (fields(id, first\_name, middle\_name, last\_name, address, contact\_number, application\_role\_id, username, password, created\_at,updated\_at)

subscriptions (fields(id, application\_id, user\_id, month\_duration, amount, created\_at, updated\_at))

**Features:**

Create (User Information, Subscriptions)

Update (User Information, Subscriptions)

View - (User Information, Subscriptions and can view history list)

Delete (User Information, Subscriptions)

Users list

**Acceptance criteria:**

* User
  + Create a validation in the front-end and back-end
  + Create a dropdown in form with a choices of Applications(From DB)
  + application\_id must be from the dropdown
  + first\_name, middle\_name, last\_name (required, will not accept numbers, max length 20, will not accept special characters)
  + address(required, max length 50, will not accept special characters)
  + mobile(required, max length15, will not accept special characters)
  + Application\_role\_id must be from the dropdown based on application role management
  + Username(required, must be unique)
  + Password(required, Should have at least 1 special character, uppercase and a number)
  + email(required, must be in the email format)
* Subscription
  + View subscribed application based on table **user**\_**applications**
  + Settle subscription and store it to **subscriptions** table.
  + View subscription history

**Staffs Management**

**Description**: as a user I want to manage my staffs record

**Database table fields:**

staffs = (fields(id, first\_name, middle\_name, last\_name, address, email, mobile, role\_id, branch\_id, username, password,created\_by(user\_id) ,created\_at,updated\_at))

**Features:**

Create

Update

View

Delete

Staffs list

**Acceptance criteria:**

* Staffs
  + Create a validation in the front-end and back-end
  + Create a dropdown in form with a choices of roles(From DB)
  + Create a dropdown in form with a choices of branch(From DB)
  + created\_by(must be from who added the stock based on the logged in user)
  + first\_name, middle\_name, last\_name (required, will not accept numbers, max length 20, will not accept special characters)
  + address(required, max length 50, will not accept special characters)
  + mobile(required, max length15, will not accept special characters)
  + username(required, must be unique)
  + password(required, Should have at least 1 special character, uppercase and a number)
  + email(required, must be in the email format)

**Stocks management**

**Description**: as a user I want to manage my stocks records

**Database table fields:**

stocks = (fields(id, owned\_by(user\_id), stock\_photo, stock\_name, created\_at, updated\_at ))

stocks\_photos = (fields(id, stock\_id, stock\_photo, created at, updated\_at))

**Features:**

Create

Update

View

Delete

Stocks list

**Acceptance criteria:**

* Stocks
  + Create a validation in the front-end and back-end
  + owned\_by(must be from who added the stock based on the logged in user)
  + stock\_name (required, max length 20, will not accept special characters)
  + Add the stock photo to the other table(stocks\_photos) referenced by stock\_id and can add multiple, but it’s optional and can be to be followed

**Inventory management:**

**Description**: as a user I want to manage my inventory records with different variants

**Database table fields:**

inventory = (fields(id, stock\_id, remarks, sku, stock\_qty, weight\_kg, price,unit, height\_mm, width\_mm, length\_mm, thickness\_mm, size, color, selling\_price,production\_cost,created\_by(staff\_id),created\_at, updated\_at))

inventory\_photos = (fields(id, inventory\_id, inventory\_photo, created at, updated\_at))

**Features:**

Create

Update

View

Delete

Inventory list

**Acceptance criteria:**

* Inventory
  + Create a validation in the front-end and back-end
  + created\_by(must be from who added the stock based on the logged in staff)
  + stock\_id(must be referenced from stocks)
  + remarks, price,unit, height\_mm, width\_mm, length\_mm, thickness\_mm, size, color (are nullable it will depend on the clients business if which field are they going to use. we will add a display choices from the user settings for this in the future)
  + selling\_price,production\_cost must be from **Purchase Order page+**
  + **``**
  + Add the inventory photo to the other table(inventory\_photos) referenced by inventory\_id and can add multiple, but it’s optional and can be to be followed

Purchase Order:

Has access: staff

Features:

Create, Update, Delete, View, Complete, list for inventory

Each Item from approved Purchase Order will be reflected to the inventory specifically in that branch

Transaction

Has access: staff

Features:

Create, update, delete, view list for transactions