**Design - User to Machine Mapping**

# **Background**

After successful login by the Admin person, user can onboard the new users along with their bio-metric to the machine, which is tagged to a particular Registration center. Through admin portal all the users, machines and devices are configured. But mapping of the users and their Bio-metric information to a particular machine would happen through this screen. These bio-metric detail would be used later to perform the de-duplication and authentication validation.

The **target users** are

* Super Admin
* Registration Supervisor
* Registration officer

The key **requirements** are

User Mapping:

On login successful of the user, we show off “New Registration” screen. There exists the “Menu”, which lists the options/ features. On clicking the “User Mapping” link, it should land up in “User Mapping Screen”. This consist of list of users available at that registration center. A table with User name, User ID, User Role is shown off. On clicking the particular user, his/her details should appear below.

Capture the selected user’s 10 finger print along with the IRIS detail and validate against the server. If valid then store the respective detail into the local system.

* Super admin – can map or unmap himself or other RO / RS.
* RS – can map or unmap other RS/RO.
* One Station ID can be mapped to multiple users

The key **non-functional requirements** are

* Security:
  + Should not store any sensitive information as plain text information.
  + The data which resides in the data-base should be in encrypted format.
* Network:
  + Should able to communicate to the configured REST URL with proper authentication.
  + The http read timeout parameter to be explicitly set, if client unable to connect to the REST service.
  + Connectivity should happen through SSL mode. The respective key to be loaded during the call.

# **Solution**

The key solution considerations are –

**Service**

* On clicking the “User Mapping” list in menu tab on Registration screen, it should hit the service [**UserMappingService**] to fetch the user details like name, ID and role from local DB.
* Create **UserMappingService** and create DTO for the same.
  + When this service is triggered for fetch() functionality - gets the user details.
  + It hits DAO and repository to fetch **list<users>.**
  + When a particular user is selected and made active or inactive then on saving it hits. save() functionality – maps / Un-maps the user from a Station ID.
  + It hits DAO and repository to map / um-map the user to a Station ID [which is machine specific].
  + If user provided the fingerprint and iris in the UI and submit the page, the same will be validated against the data available in the server by calling the respective REST service.
    - One request per finger and iris. Totally there are 12 requests to be triggered. **TODO**: Need to plan for bunch of image validation.
  + If most of the fingers and iris are matching [based on ISO template] then activate the user mapping to the station id.
  + Send an Alert message (say) “User mapped successfully” or an error message.
* Handle exceptions in using custom Exception handler and send correct response to client.

**UI:**

* Design UI using FXML and map the UI individual components in UserMappingController class.
* UserMappingController – it should communicate between UI screen and Service ‘UserMappingService’ class to render the data to screen and capture the data from screen.
* Based on validation across the POJO class from the UserMappingService, build the UI screen
* Create the proper alert success/error to intimate the user.

**Classes**:

**UI**: UserMappingScreen

**Controller**: UserMappingController

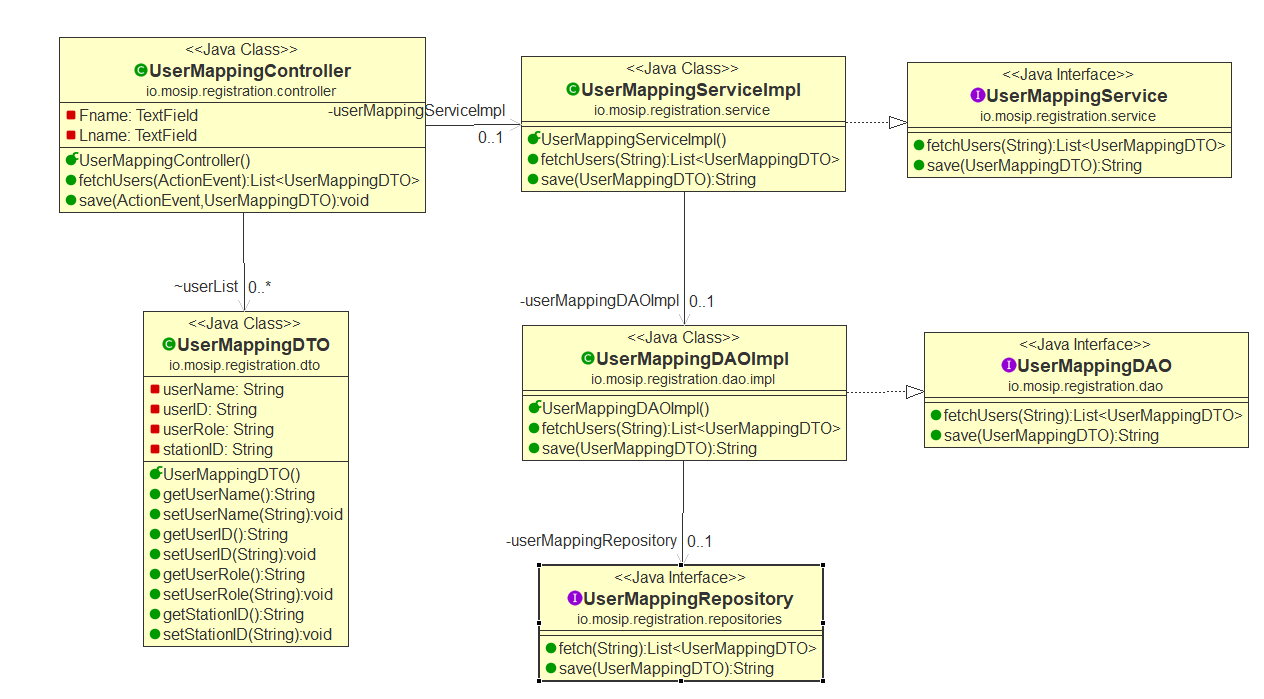
**Service**: UserMappingService

**DAO**: UserMappingDAO

**Repository:** UserMappingRepository

**DTO**: UserMappingDTO

**Class Diagram:**



Sequence Diagram: