
Design Document for **Fiesta Fetch**

Group **MK1_4**

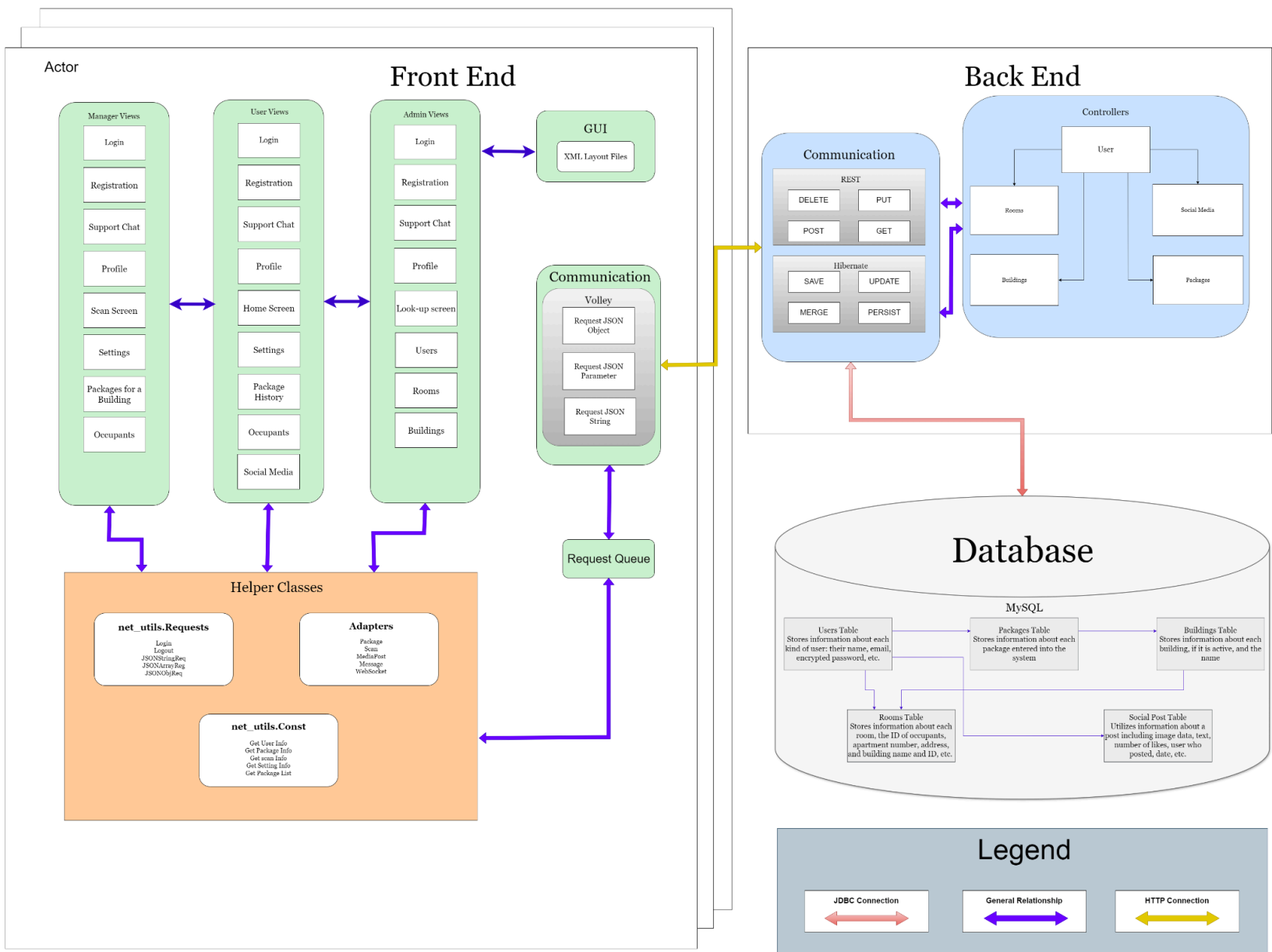
Ben Steenhoek: 25 % contribution

Harsh Modi: 25% contribution

Ben Bartels: 25% contribution

Lucas Schakel: 25% contribution

BLOCK DIAGRAM PICTURE



Frontend

Create Account and Login

- Create account screen generates a page with the following elements:
 - EditText: Password
 - EditText RePassword
 - EditText: Email
 - Button: CreateAccount
 - Button: ResetPassword

Upon clicking the button 'CreateAccount' the values of the Password and Email are sent as a POST request to the server if Password and RePassword match. When clicking the button 'ResetPassword', the user can change their password after verification.

- Login screen generates a page with the following elements:
 - EditText: UserID
 - EditText: Password
 - Button: Login

Login Screen takes user input of their username and password, and if that username and password is valid, it will take the user to the main menu, where they are logged in as that user.

Package Entry and Scanner

- Managers are the only type of users that use the scan function, so the manager dashboard generates a page with the following elements:
 - EditText: PackageTrackingNumber
 - EditText: PackageAddress
 - EditText: PackageRecipient
 - Button: AddPackage

Manager dashboard will take the user input of package information and send that information as a POST request to the server. When using the package scanner feature, these fields will auto-populate. This information will notify the user has a package to pickup and show up on their home screen.

Admin Dashboard

- Admin dashboard generates a page with the following elements:
 - Table: Users
 - Table: Buildings
 - Table: Rooms
 - Button: AddEntry
 - Button: UpdateEntry
 - Button: DeleteEntry

This screen should load one of the 3 tables: users, buildings or rooms. The selected table is populated into a recycler view with all of the entries in the table through a GET request. Administrators should be able to create, update, and delete entries in the tab.

Backend

Communication

The backend uses mappings to update the database based on information sent to the predefined URL mappings. These include:

- Post: send information on an item that'll be created and added to the database.
- Get: request information, often with an identifier for the specific item requested from the database
- Put: send information along with an identifier to update a specific item in the database
- Delete: send an identifier to delete a specific item from the database, this will also nullify any ties to the deleted item

Controllers

The controllers contain the mappings for communication between frontend and the database.

These include:

- User: Contains the above mappings to create users, which contains a one-to-many relationships with the package table and a many-to-one relationship with the rooms table.
- Packages: This contains a many-to-one relationship with the users table. A user can view their packages on the frontend, a manager and admin can create and update packages. They can also delete them but only really needed if they scanned a package twice.
- Buildings: this contains a one-to-many relationship with the rooms table, this is used at the beginning of the search when a package is scanned.
- Rooms: contains a one-to-many relationships with the users table and a many-to-one relationship with the Buildings table. This is used by the Admins to see who is in what room, as well as the second table used when searching for a scanned package.
- Social Media: Websocket that allows users to take pictures and make posts about their packages.

Table Relationships

