**Design Document for Rate My Dishes**

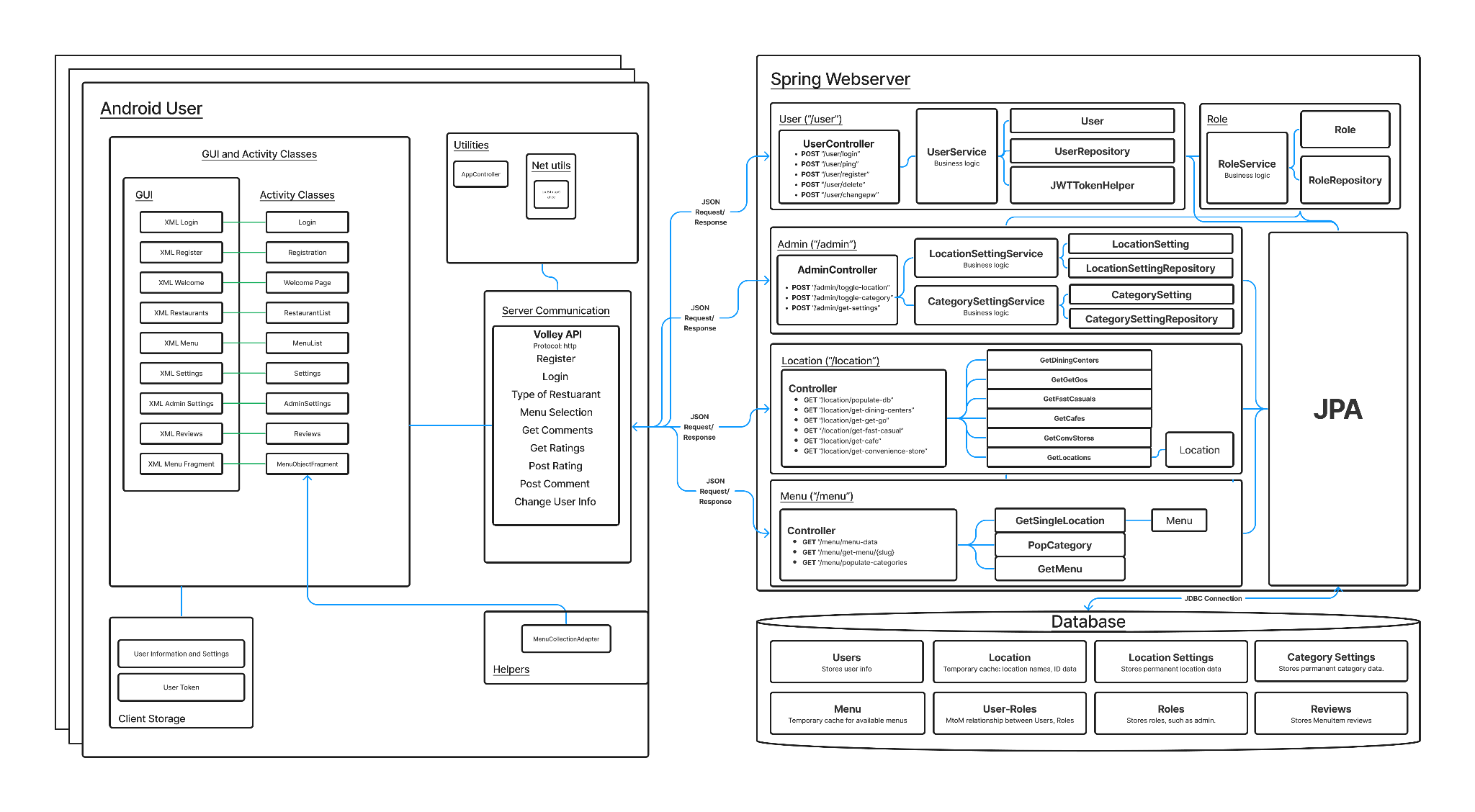
Group 1\_as\_6

Jacob Duba: 25% contribution

Karthik Kasarabada: 25% contribution

Michael Gohr: 25% contribution

Brandon Zaragoza: 25% contribution

Block Diagram

Use this third page to describe complex parts of your design.

**Frontend: Account Creations-** Upon clicking the button ‘CreateAccount’ the values of the UserID, and Password are sent as a request to the server. If the passwords do not match or are too short the user will be prompted with an error message.

**Frontend: Login Screen-** Login Screen takes user input of their username and password and sends a request to the server, 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. If the username or password is invalid the user will be prompted with an error message. User login information is saved across sessions so they do not have to login every time they launch the application.

**Frontend: Settings-** Any user can load this page but only admin users can view the admin settings button. By clicking the button ‘Logout’ the user’s saved credentials will be removed and they will be taken back to the login screen. By clicking the button ‘change password’ the user will be asked to confirm their current password. If they confirm correctly a request will then be sent to the server to change the password to whatever the user wants their new password to be. By clicking the ‘delete account’ the user will be prompted to confirm their password in order to delete their account. If they confirm their password and confirm their delete their credentials will be removed and the account will no longer be usable. The admin settings button will redirect the admin user to the admin settings screen. The back button takes the user back to the welcome page.

**Frontend: Restaurant List-** Restaurant list generates a page with varying elements but the back button will always be available. Upon clicking the back button the user will be taken back to the Welcome page. This screen is also populated with a different number of buttons depending on how many restaurants are open or types of restaurants exist for that selection. Each of these buttons will take the user to the menu for that specified restaurant

**Backend: Roles-**In order to have a many to many relationship, users can have multiple roles. Although we only have one role “admin” at the moment, it is a JPA Entity being stored in SQL. When a user tries to access admin settings, we use a function in the RoleService class to see if the user has a role titled “admin”.

**Backend: User-** We use JWTs to handle authentication! This explains the JWTHelper class inside our user object, which turns User objects into token that the controllers can send back to the Android user.

**Backend: Locations and Menus-** For Locations, we delete, update, and add columns into the Location Repository based on the payload received from a GET Request sent to ISU Dining . We use them as a “cache.” This task is automated, running every 10 minutes in the production environment. For Menus, we actually generate JSON data and store it in the database so we do not have to generate this data on the fly when the Android User requests a menu. This allows the server to react much faster. This process is automated, running every 10 minutes in the production environment.

**Backend: Reviews-** For Reviews, we store Review Objects into the Reviews Repository. Each object is mapped to a unique Menu Item and stored text values for comments and an int value for the rating of an item. Additionally, each Review object points to a unique user.

Table Relationships Diagram