**Design Document for Closetics**

Group 2\_muzakr\_3

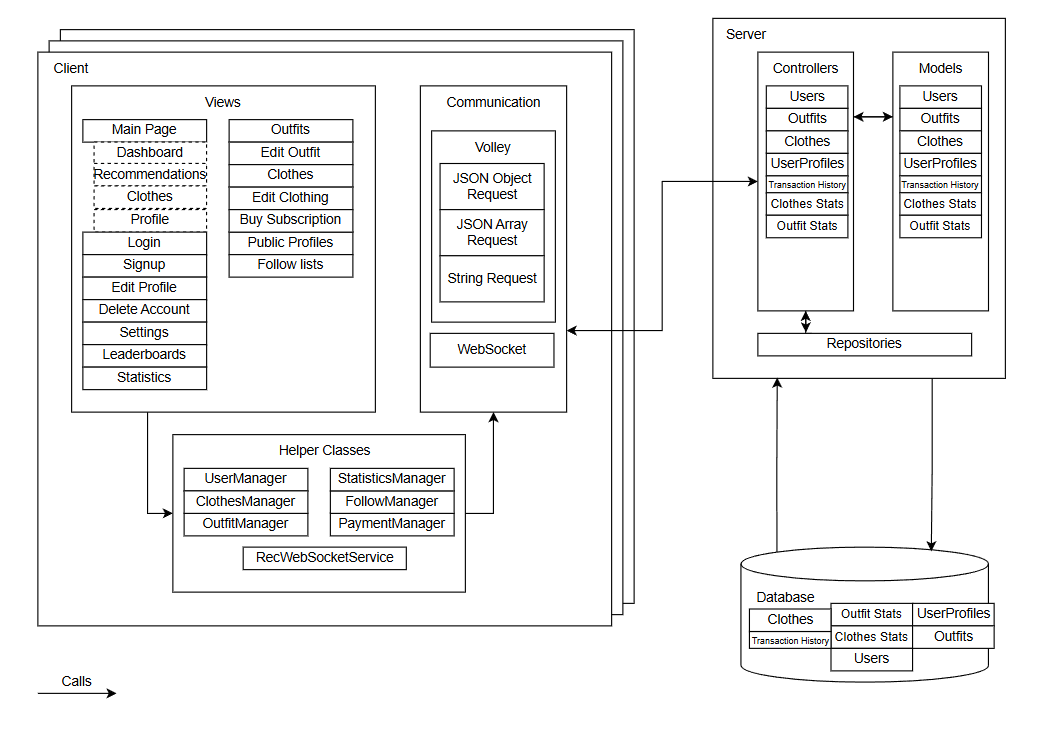
Ashten Finders: 25% contribution

Brayden Hayworth: 25% contribution

Niall Sharma: 25% contribution

Stas Bronevskiy: 25% contribution

Block Diagram



Block Diagram Description

**Frontend**

Views – the UI of the application, they are the main way a user inputs and receives information. Our project uses Android views, consisting of Java classes and XML layouts. Our views can be divided into 5 main categories:

* Managing User Account
* Managing Personal Clothes and Outfits
* Interacting with other Users
* Leaderboards and Personal Statistics
* Other (Settings, Payments, …)

Helper classes – used as a middle man between Views and Communication with the Server. Usually are Java classes with template request methods.

Communication – internal part of the application that handles all interactions with the Server. Consists of Volley HTTP Request library and WebSocket library for Android.

**Backend**

Database:

Our database contains numerous tables storing all the information required by our application. The database utilized is MySQL. We use table relationships to further show the complex association between our different data points (User profiles and Users, Outfits and Clothes, …). Our Java backend utilizes Springboot and Hibernate to communicate with the database.

Controllers:

The controller classes are what allow the backend database to communicate with the front end. They contain the CRUD mapping that allows the front-end to retrieve data from the backend as well as add data to the backend. They contain specific logic to handle different scenarios and provide consistent, correct, and expected responses. This includes error handling and input validation.

Repositories:

We have a repository for each controller, the controller uses it, and this allows us to query the database for the data we need to send when a user makes a request to a certain endpoint within the controller. Most of the queries are built-in with JPA repositories built-in queries, but we made use of custom queries in some cases to retrieve more complex sets of data.

Database Schema

