Design Document for Total Knockout Chess

Group **1-AN-5**

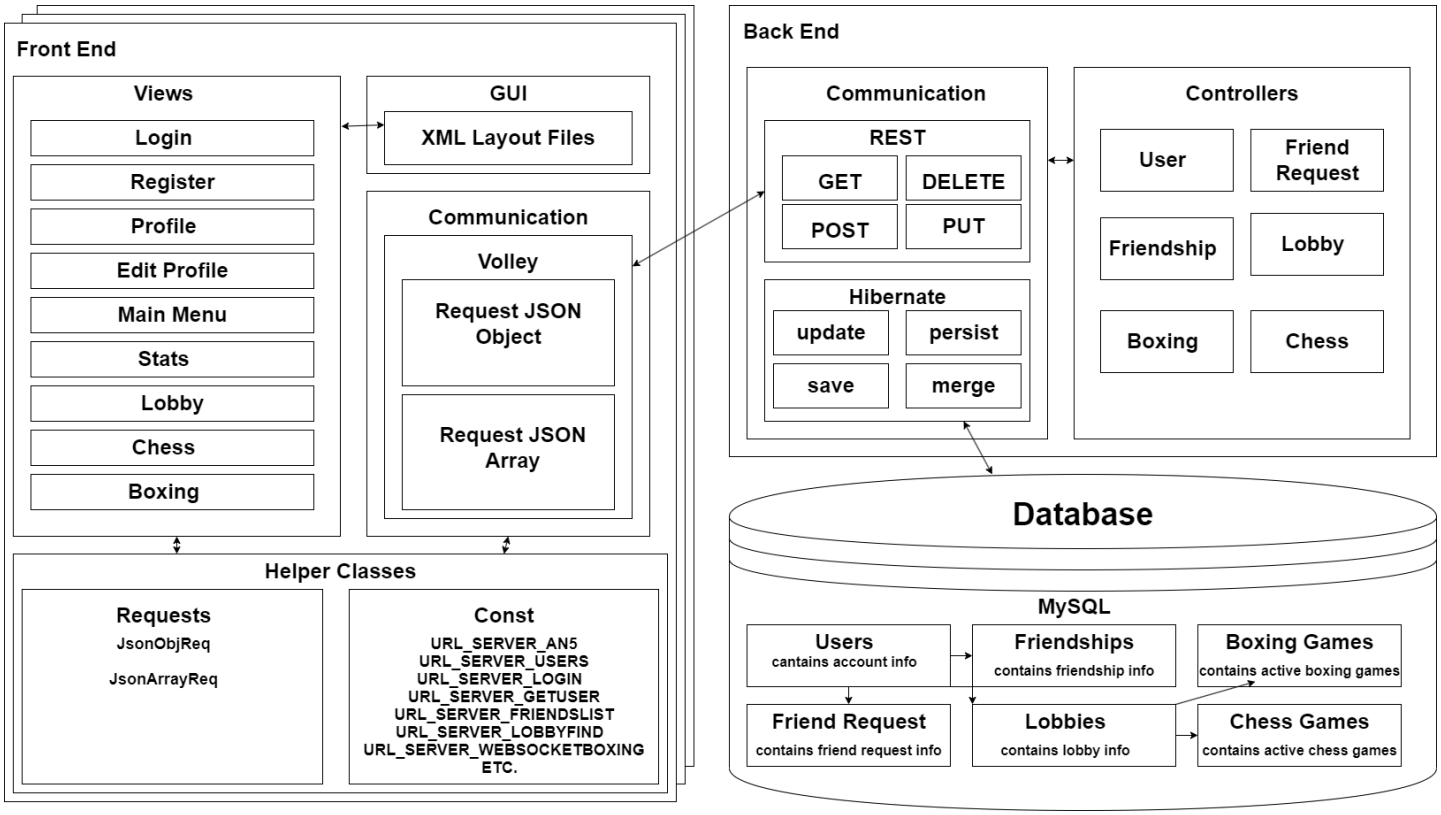
Zachary Scurlock 25% contribution

Connor Hand 25% contribution

Lex Somers 25% contribution

Brian McCreary 25% contribution

Block Diagram



Block Diagram Description

Frontend:

The front end of our app consists of activity java classes called “Views”; XML files referred to as a “GUI”; and a singular java class, “Const.java”, which holds constant strings used for URL mappings. Each GUI’s XML code creates different objects. Those objects include buttons, text boxes, layouts that can hold other objects, etc. In each View, code exists that links different elements from the GUIs to various declared Java objects. When a user interacts with a GUI element, the views then use either web sockets or a tool called Volley to send and receive requests. There are different types of requests that can be sent which send different pieces of data in the form of JSONObjects, JSONArrays, or strings. After a request is sent, the front end then waits for a response from the backend containing some form of data and then does something with that data within the Views.

Backend:

Using Spring Boot as our foundation, the backend enables communication between the front end and the database. Separated into two main sections, the REST portion enables the frontend requests to be mapped to the appropriate controllers, who then are able to communicate back to the hibernate module before syncing up with the database. An example of a common interaction (to add a friend) from the frontend would look like the following: a put request would be routed to the friend request controller, the controller would then forward the request to the hibernate module, which in turn would tell the friend request database to save a new request.

Database:

Our database contains tables for all of our stored information. This information includes data such as users, friends, friend requests, active boxing games, active chess games, and lobbies. Our database serves as the heart of our application through the utilization of MySQL, a relational database used to support client-server architecture. These tables allow us to have excellent connections between our data points.

Database Schematic