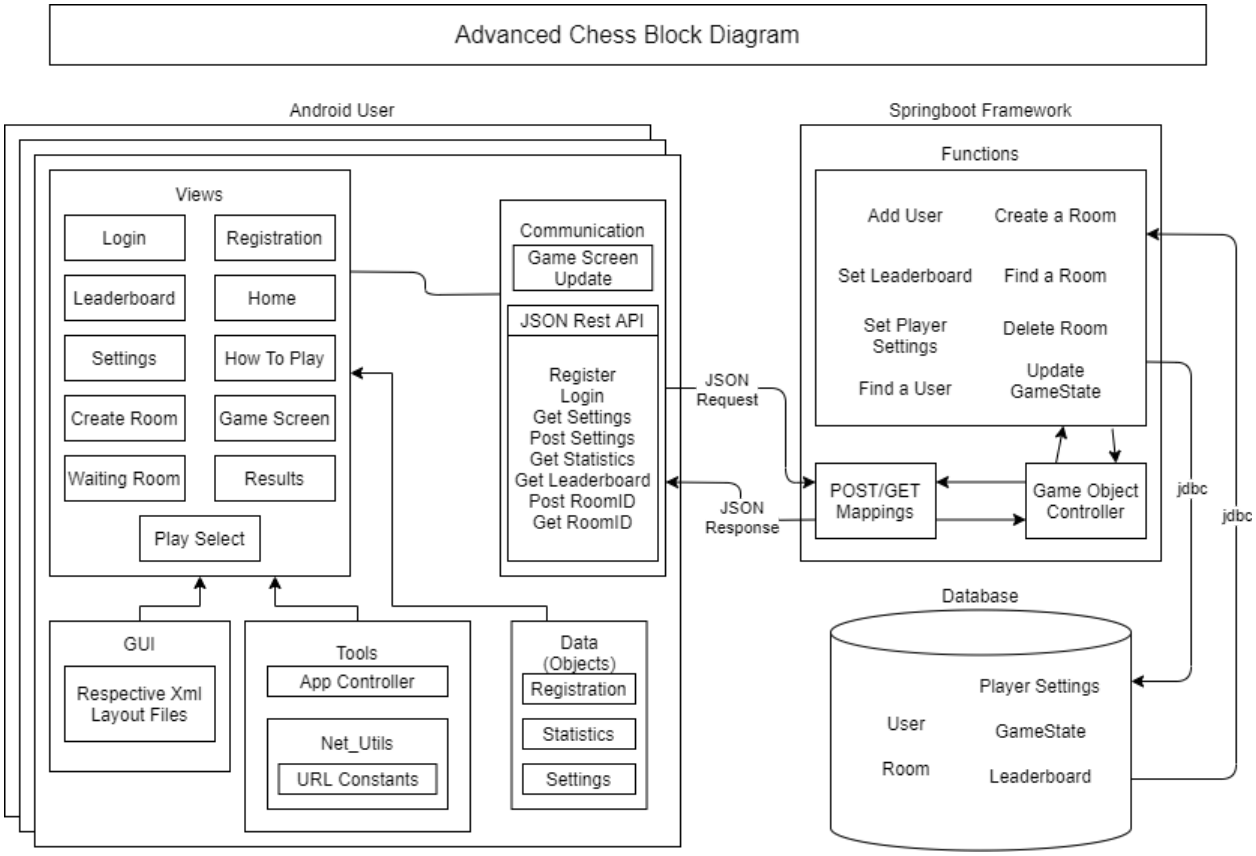
Design Document for Advanced Chess

Team TZ_3

Carver Bartz: 25% contribution
Jake Garnett: 25% contribution
Theron Gale: 25% contribution
Ian Swaner: 25% contribution

Block Diagram



Description

Android User GUI

Our Android application will have a comprehensive GUI. Each screen of the application will be fully functional with a corresponding xml file. We will have a visualization of the chessboard for all players that is updated in real-time as each player takes their turn.

Android Tools/Helpers

Our Android app thus far is planned to have an AppController class and a URL Const class. The AppController class is utilized to streamline requests and allocate them into a queue to reduce recycled code, and the URL Const class is used to contain the base for all URLs used for all JSON requests.

Communication

We will communicate with the server using GET and POST requests which have been mapped in the Backend code. These requests will update the user's chess game accordingly, for example, if a player decides to move a pawn the new game-state will be processed through a GET request and everyone else's game-state will be updated through a POST request. The Backend processes and stores the game state and allows the game to be updated for each client.

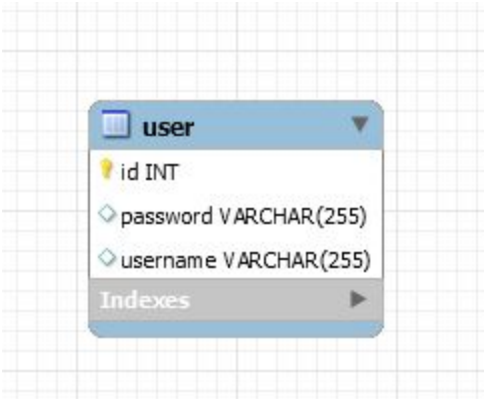
SpringBoot Framework/Server

The SpringBoot Framework/Server will contain the advancedchess-1.0.0.jar file. It will serve as a link between the Frontend GUI and the SQL database that contains all of the data. The Android User will be able to send POST and GET requests via their application to the server. The request will then utilize a certain function to interact with the database. The final step is the server will return the result to the Android User's client.

Database

The database will be an SQL database and hold the information of the user (ID, user name, and password). The database will also hold LeaderBoard statistics and the user's preferred settings, these will all have a reference to the user's unique ID. A room will also be stored in the database once it's created and will have an ID for other players to use to find that room. The room will hold a game-state that will update as the game is in progress.

Table Relationship Diagram



Right now our relationship diagram is incomplete, we are still working on the game logic to allow players to play together through the server. We do know that we will have a leaderboard object that will hold the wins and losses along with other statistics and will be referenced by the user, and we will have a similar system with the user's preferences.