Design Document for

Group 1_UG_1

Member 1
Caden: 25% contribution

Member 2
Manny: 25% contribution

Member 3
Jack: 25% contribution

Member 4
Ben: 25% contribution

Complex Design Explanation

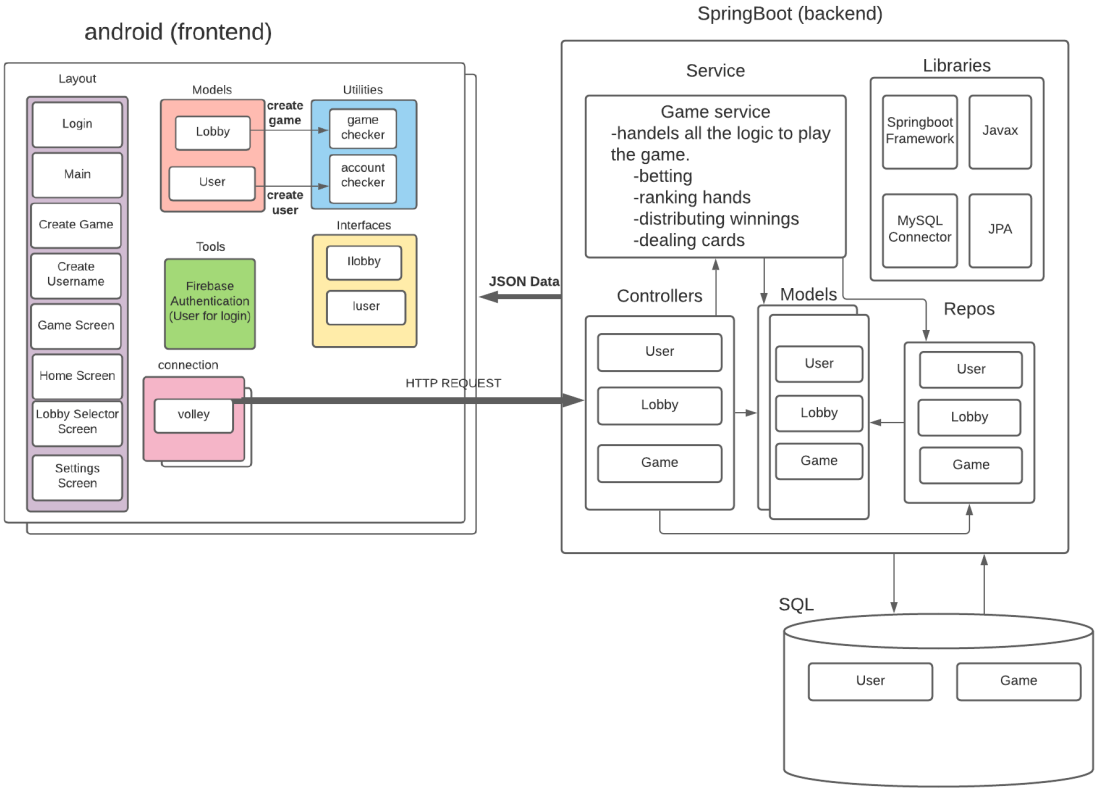
In our frontend we used a library called Firebase Authentication to help us manage our users. We take in a username and password and pass it to Firebase to handle our account creation where they store all of the information needed and we can ask Firebase for the user's ID which we store in our database for our purposes.

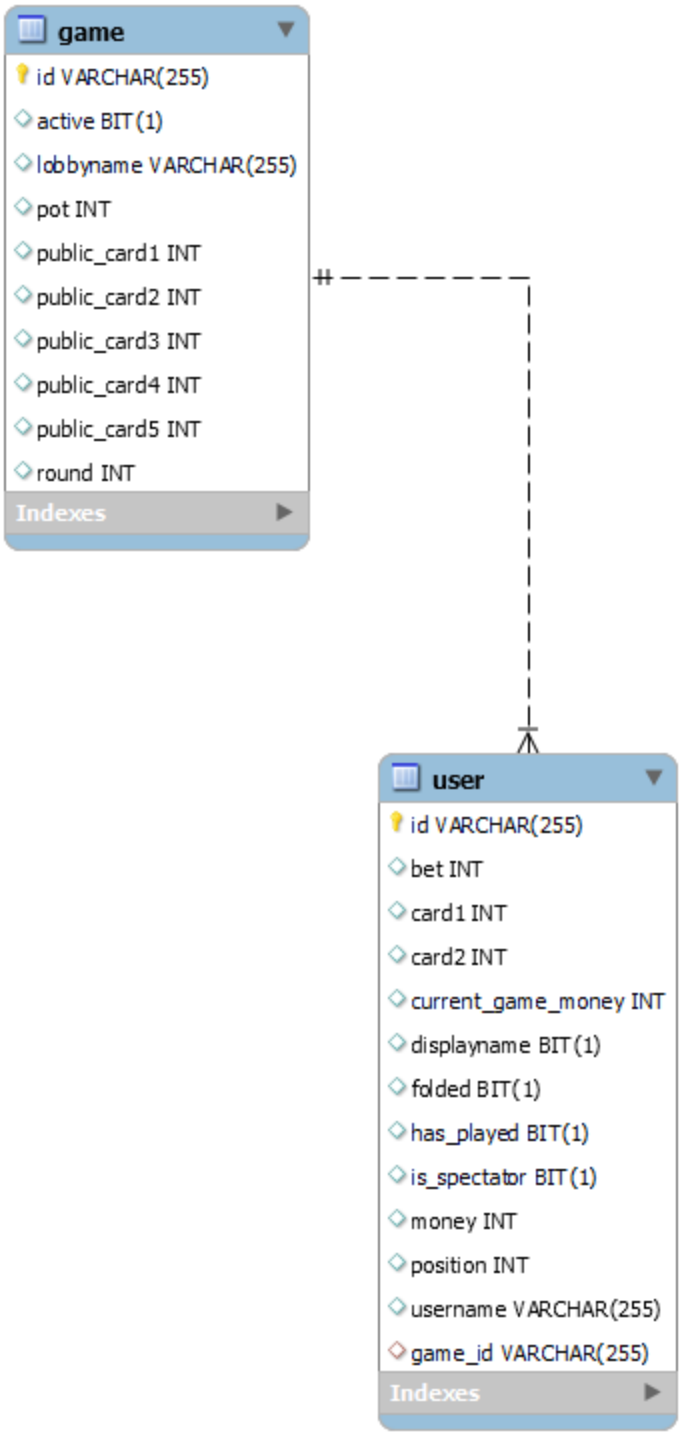
We also have a few different packages to separate some of our code. We used a Models package to put all of our models in. We have User and Lobby models to match with the tables that we use and in those models we handle our converting to and from JSON for our http requests. We also have an interfaces package. We use this to help use make our http calls easier and it also allows us to test our code easier.

We use XML for designing each screen and user Java classes to control how each XML file flow with each other and the user and application interact.

In our backend, we implemented a service called game service. This is used to set up the game and for the game functionality. It has functions like gameInit, which initializes the games, and play, which lets the user bet, fold or raise during that round.

We also made a private class called Deck to make a deck of cards with the functions of deal and deal public cards. This allows for the game to simulate a real deck so that the players are dealt real cards without repeats from a 52 card deck.


android (frontend)
SpringBoot (backend)
Layout
Login
Main
Create Game
Create Username
Game Screen
Home Screen
Lobby Selector Screen
Settings Screen
Models
Lobby
User
create game
create user
Utilities
game checker
account checker
Tools
Firebase Authentication (User for login)
Interfaces
Ilobby
Iuser
connection
volley
HTTP REQUEST
JSON Data
Service
Game service
-handels all the logic to play the game.
-betting
-ranking hands
-distributing winnings
-dealing cards
Libraries
Springboot Framework
Javax
MySQL Connector
JPA
Controllers
User
Lobby
Game
Models
User
Lobby
Game
Repos
User
Lobby
Game
SQL
User
Game


game
id VARCHAR(255)
active BIT(1)
lobbyname VARCHAR(255)
pot INT
public_card1 INT
public_card2 INT
public_card3 INT
public_card4 INT
public_card5 INT
round INT
Indexes
user
id VARCHAR(255)
bet INT
card1 INT
card2 INT
current_game_money INT
displayname BIT(1)
folded BIT(1)
has_played BIT(1)
is_spectator BIT(1)
money INT
position INT
username VARCHAR(255)
game_id VARCHAR(255)
Indexes