Design Document for Coup Mobile

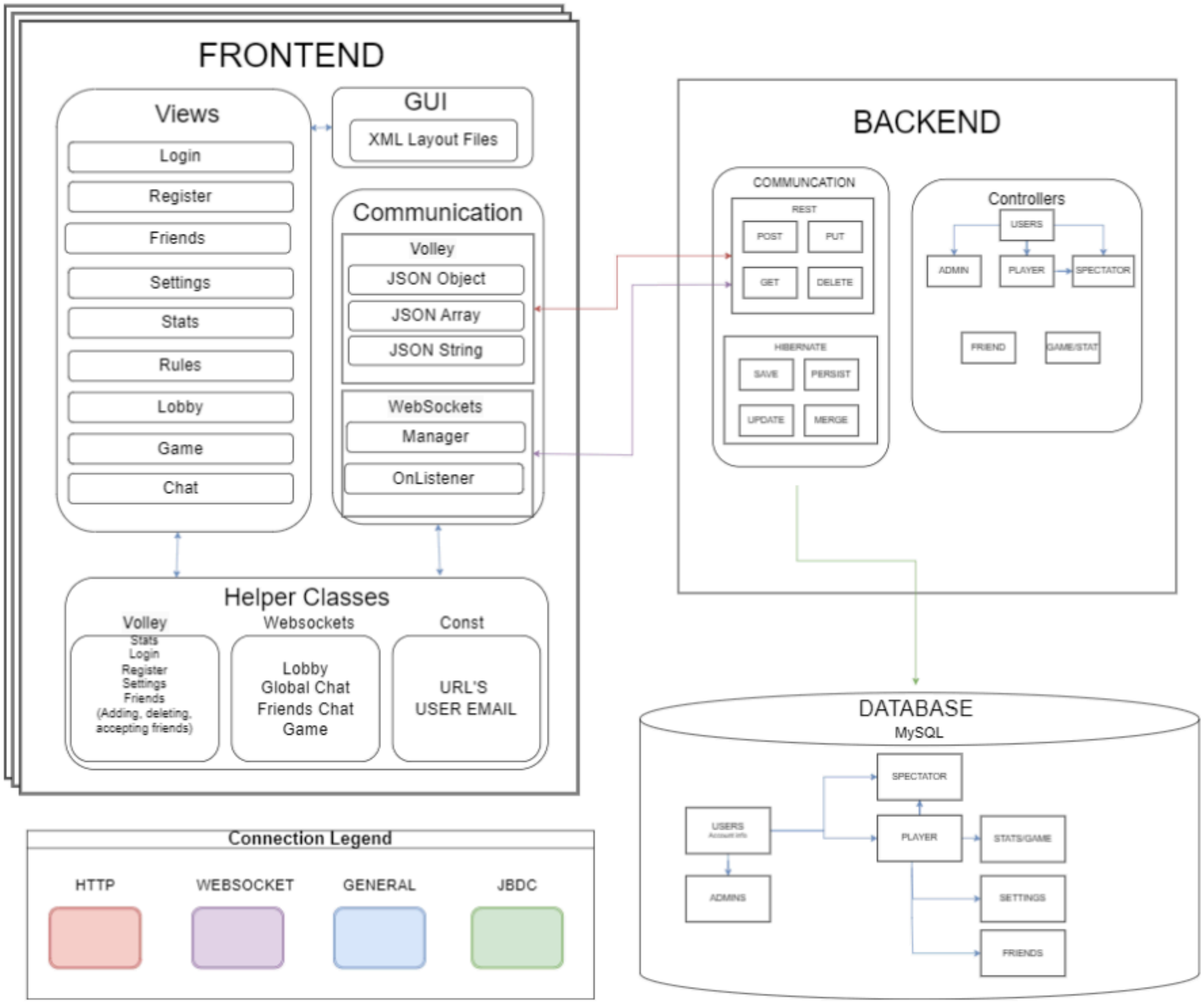
Group Mk1_8

Member1 Name: Charles Arroyo 25 Percent

Member2 Name: Ponciano Ramirez 25 percent

Member3 Billy Dang 25 Percent

Member4 Name: Bo Oo 25 Percent


FRONTEND
Views
Login
Register
Friends
Settings
Stats
Rules
Lobby
Game
Chat
GUI
XML Layout Files
Communication
Volley
JSON Object
JSON Array
JSON String
WebSockets
Manager
OnListener
Helper Classes
Volley
Stats
Login
Register
Settings
Friends
(Adding, deleting, accepting friends)
Websockets
Lobby
Global Chat
Friends Chat
Game
Const
URL'S
USER EMAIL
BACKEND
COMMUNCATION
REST
POST
PUT
GET
DELETE
HIBERNATE
SAVE
PERSIST
UPDATE
MERGE
Controllers
USERS
ADMIN
PLAYER
SPECTATOR
FRIEND
GAME/STAT
DATABASE
MySQL
USERS
Account info
ADMINS
SPECTATOR
PLAYER
STATS/GAME
SETTINGS
FRIENDS
Connection Legend
HTTP
WEBSOCKET
GENERAL
JBDC

Frontend

Game Actions :

A key component of our gameplay experience are the Action buttons, each corresponding to a move within the game: 'Income', 'Foreign Aid', 'Taxes', 'Coup', 'Bluff', 'Assassinate', 'Steal', 'Block', and 'Exchange'. These buttons are designed for quick access, enabling players to make decisions quickly in the context of the game. Pressing an action button triggers a move that affects the game state which is also seen by all players allowing them to make a move to respond back.

Social Interaction:

The Friends section is the social aspect of our game, allowing players to 'Add friend', 'Delete Friend', or 'Invite' others to a game session. The 'Exit' button is used for leaving the friend list. A 'Search Friend' EditText provides a way to connect with other players, and a dynamic ListView ensures players have a real-time update of their friends list.

Game Board View and Player Interaction:

The game action button is in the center of the game, with Image Views displaying each player's cards(hidden), ensuring that personal and opponent's gameplay elements are visually different and constantly in view for all players in the game. The Image Views for Player 1 Card 1 through Player 4 Card 2 are placed to resemble the feel of a physical environment, allowing for a more easier player adaptation from tabletop to the screen. Chat interactions are seen through a Chat Box with a Send Chat button, allowing for constant banter and bluffing that is a key component of the game.

Backend

Communication

The backend uses mapping and HTTP requests to update and send data to our MySQL database. These include:

- POST: Creates information into the database
- GET: Get’s requested information from the database, and sends verification to the front end.
- PUT: Updates existing information in the database.
- DELETE: Deletes a table/user/column from database.

Controllers

The backend implements multiple controllers, with most of them being connected to a User.

Accounts:

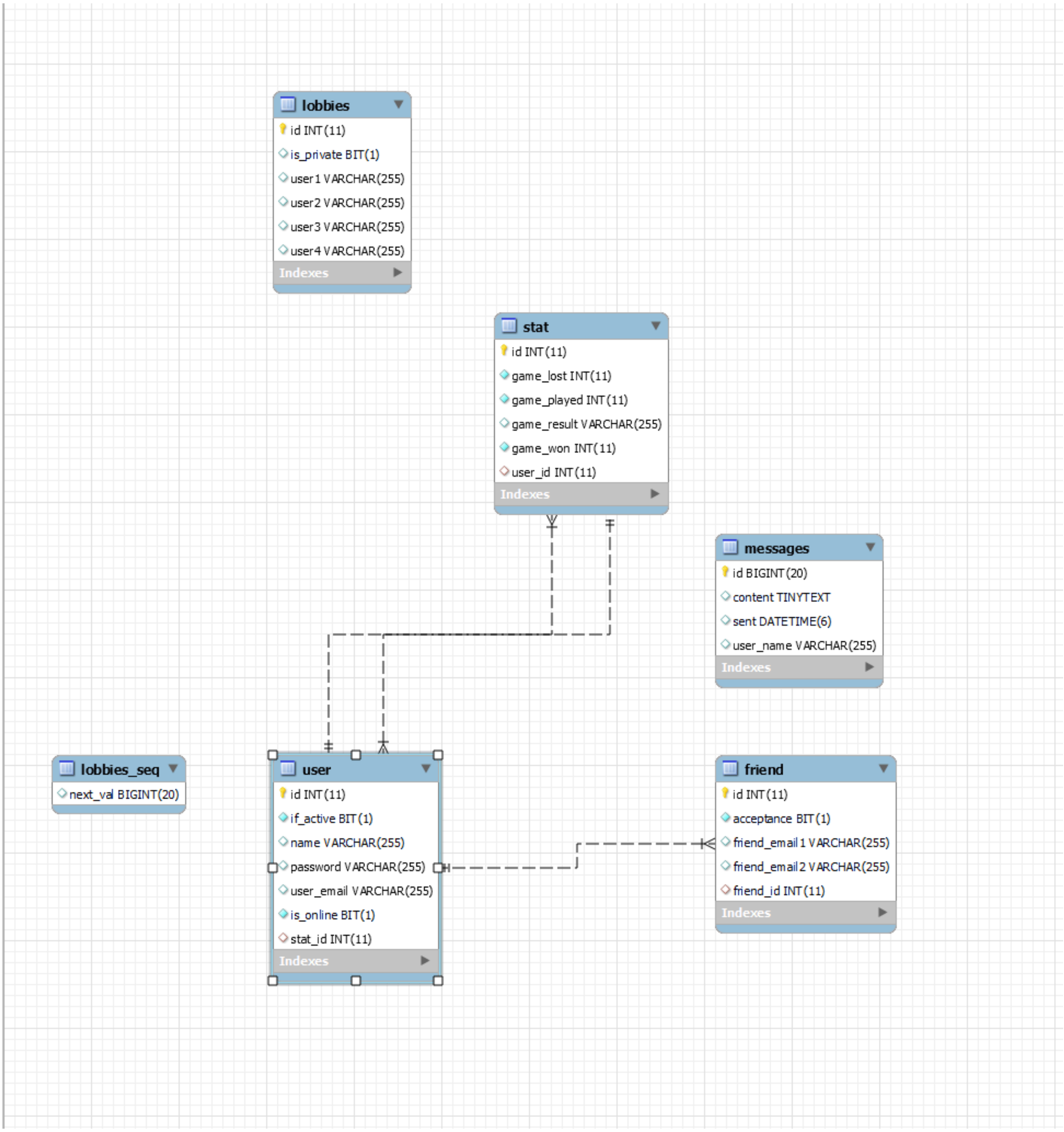
- **User/Player**: Contains above communication request to create Users, with basic attributes (email, password) and the ability to play games.
- **Admin** (unimplemented): Contains above communication requests
  - Delete Users
  - See overall game stats: How many times games are being played a week, top players.
  - Monitor chats between all users.
- **Spectator**: (unimplemented) Any user can be a spectator, but they are not allowed to play in games, but can watch current games occurring.

Relationships: (Example: Reviews: Restaurant and Customers) HTTP

- **Friends**: This is a One to Many relationship between **User/Player** and **User/Player**. Once a User/Player has friends, they can direct messages.
- **Stats**: One to Many relationship between **User/Player** and **User/Player**. Once a user plays a game, they can see their game stats, and others.
- **Admin Stats**: One to Many relationship between **User/Player** and **Admin**. Admins have the ability to see all stats between all players, which will be consolidated into stats. Admins need this information to see how their game is doing.

Relationships: Websockets

- **Lobby:** One to One relationship between **User/Player** and **User/Player** and One to Many **User/Player** and **Admin.** Users/Players can see all the users in the lobby. Admins can see all the lobbies existing.
- **Message**: One to Many relationship between **User/Player** and **Admin**. Admins have the ability to see all the chats of players, to monitor cursing/explicit language.


lobbies
id INT(11)
is_private BIT(1)
user1 VARCHAR(255)
user2 VARCHAR(255)
user3 VARCHAR(255)
user4 VARCHAR(255)
Indexes
stat
id INT(11)
game_lost INT(11)
game_played INT(11)
game_result VARCHAR(255)
game_won INT(11)
user_id INT(11)
Indexes
messages
id BIGINT(20)
content TINYTEXT
sent DATETIME(6)
user_name VARCHAR(255)
Indexes
lobbies_seq
next_val BIGINT(20)
user
id INT(11)
if_active BIT(1)
name VARCHAR(255)
password VARCHAR(255)
user_email VARCHAR(255)
is_online BIT(1)
stat_id INT(11)
Indexes
friend
id INT(11)
acceptance BIT(1)
friend_email1 VARCHAR(255)
friend_email2 VARCHAR(255)
friend_id INT(11)
Indexes