Design Document for Cycino

Group <3_swarna_5>

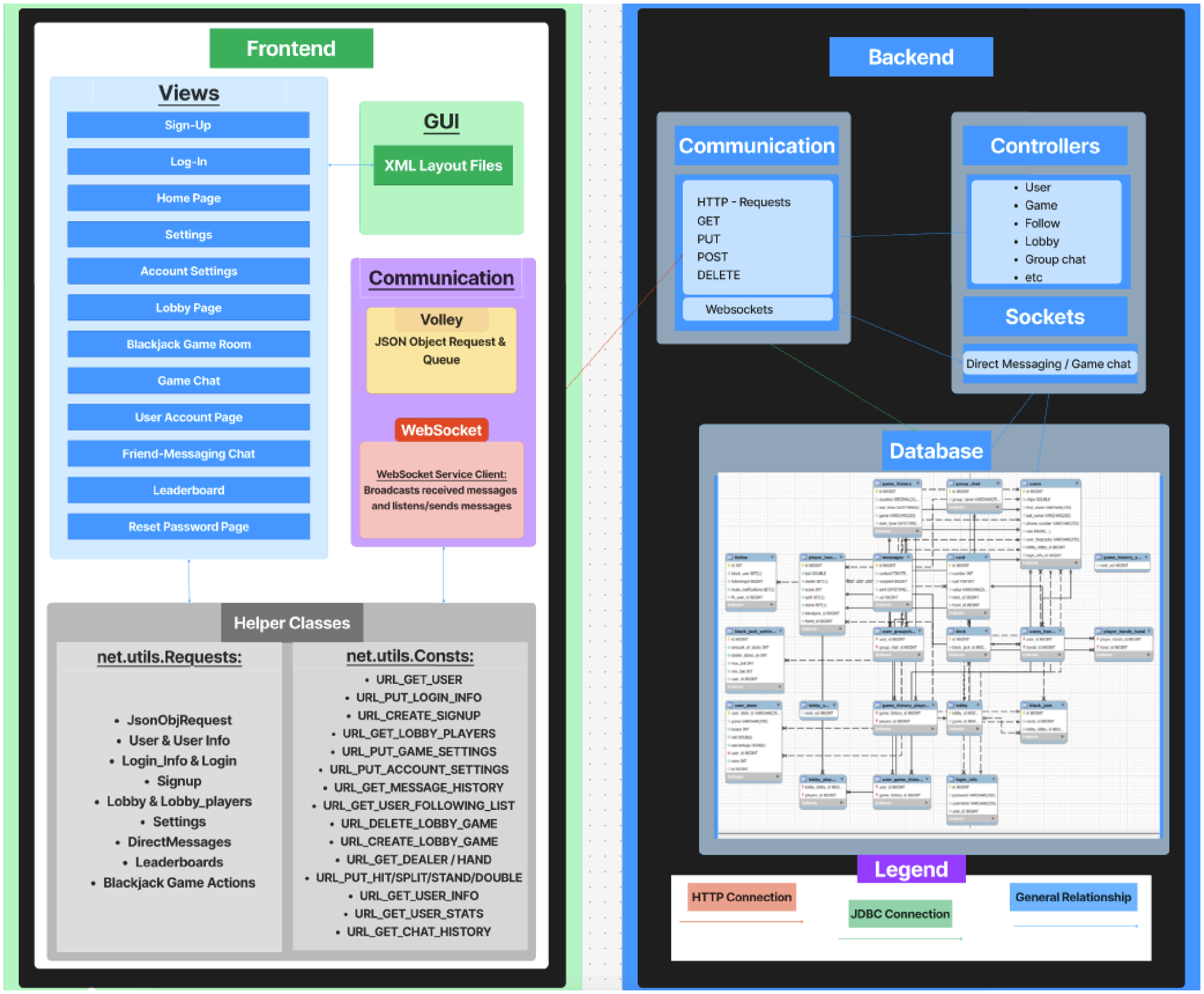
Evan Litzer: 25%

Filip Forsberg: 25%

Ben Eschman : 25%

Sam Craftman: 25%

Block Diagram



Design description

Frontend: *currently implemented*

***Signup*(Username, Password)**

- ❖ Signup user generates user login information, creates a user account, and then prompts user to their newly created account page after logging in.
    - ➢ EditText: Username
    - ➢ EditText: Password
    - ➢ Button: SignUpUser
- ❖ When the SignUpUser button is clicked, Username and Password values are sent as POST request to server.
    - ➢ Users can then use their username and password to log into the app (check GET information).

Deal()

- ❖ Commences a round of blackjack through first dealing the player(s) two face-up cards and the dealer one each face-up and face-down. Achieved by calling startGame(), handling the actions and setting random cards.
- ❖ Users are then sequentially prompted with their next decision with buttons and their/dealer's presented cards.
    - ➢ Button: Deal (Once pressed, then disappears) | Button: Hit | Button: Stand
    - ➢ ImageView: dCard1, dCard2, p1Card1, p1Card2
    - ➢ TextView: dScore, pScore
    - ➢ ImageView: pCard3, dCard3, pCard4, dCard4 (if needed)
- ❖ Goal is to be as close to 21 without going over → Hit deals them another card, Stand stops dealing more cards.
    - ➢ Dealer has preset dealerStandOnValue integer to stop it from hitting after reaching number
- ❖ Cards are generated with GET requests and set to corresponding drawables.
- ❖ User decisions send PUT requests to the server that then either deal another card or finalize user hands.

***updateGameSettings*(dealerStandOn, maxBet, minBet, numberOfDecks, userID)**

- ❖ updateGameSettings changes the settings of the current user's blackjack game.
    - ➢ EditText: dealerStandsOnEdit, minBetEdit, maxBetEdit, numberOfDecksEdit
    - ➢ Button: updateDealerStandsOnButton, updateMinBetButton, updateMaxBetButton, updateNumberOfDecksButton
    - ➢ TextView: dealerStandOnTV, minBetTV, maxBetTV, deckNumberTV
- ❖ GameSettings page takes input in the editTexts for each of these settings and updates them when a button is clicked.
    - ➢ When the button is pressed, a PUT request is sent to the server changing the afflicted value.
    - ➢ GET request then displays the settings in the TextViews.

Backend:

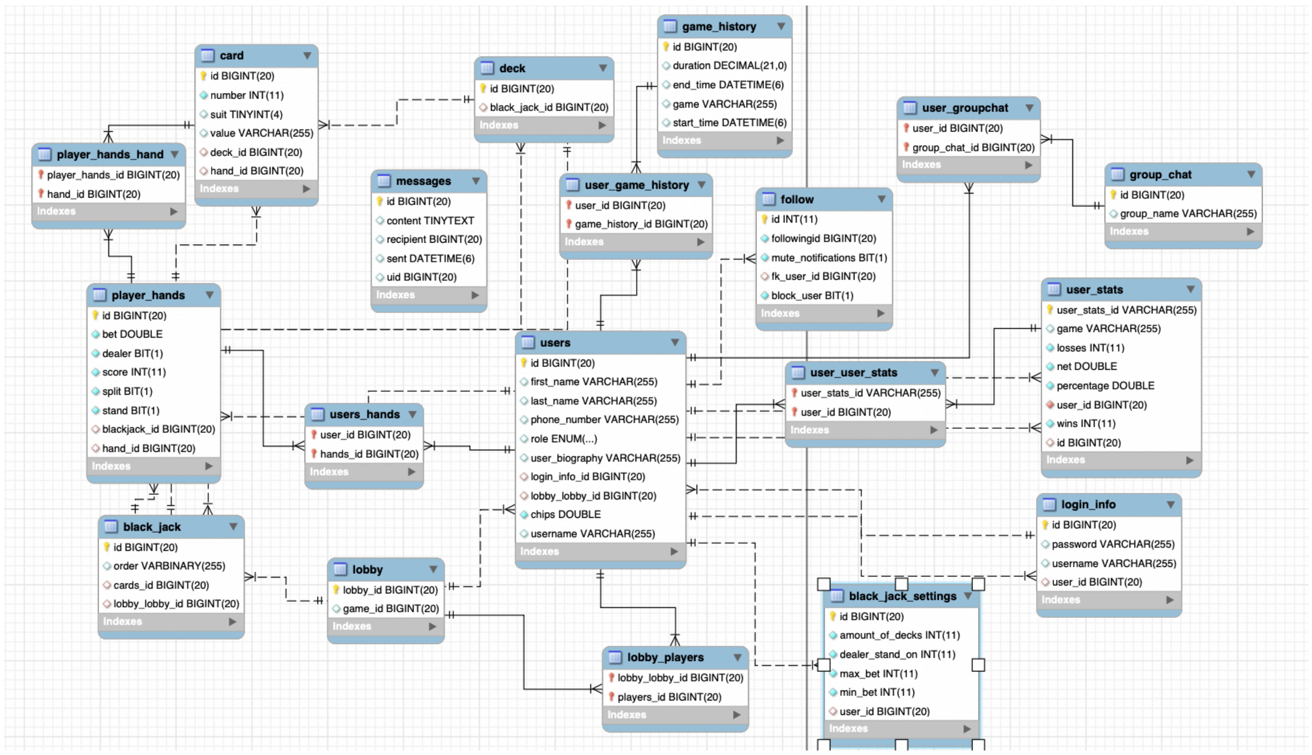
Swagger Link: http://coms-3090-052.class.las.iastate.edu:8080/swagger-ui/index.html#/

General Structure

Our application is based around users playing games. Naturally, to play games, save statistics, chat with others, the application is centered around the user entity. Following the user entity, we can further divide our application into 2 big categories: user-to-user interactions & user-to-game interactions.

The user entity in itself holds all the necessary data for user profiles, but it also carries a lot of data to allow for different types of connections with the rest of the application. The connections being:

- One-to-One
    - logininfo-user
        - We keep profile information separate from the login information
    -
- Many-to-One
    - lobby-user
        - Since it is an online game, many users can share a [game]lobby together
    - userStats-user
        - users have a userStats for each game.
- One-to-Many
    - hands
    - followList
        - User connections are, logically speaking, one-way in our application. One can think of Instagram for inspiration. Following someone =/= that they follow you back.
- Many-to-Many
    - gameHistories
        - users each have their own game history of multiple games
    - groupChats


card
id BIGINT(20)
number INT(11)
suit TINYINT(4)
value VARCHAR(255)
deck_id BIGINT(20)
hand_id BIGINT(20)
Indexes
deck
id BIGINT(20)
black_jack_id BIGINT(20)
Indexes
game_history
id BIGINT(20)
duration DECIMAL(21,0)
end_time DATETIME(6)
game VARCHAR(255)
start_time DATETIME(6)
Indexes
user_groupchat
user_id BIGINT(20)
group_chat_id BIGINT(20)
Indexes
group_chat
id BIGINT(20)
group_name VARCHAR(255)
Indexes
player_hands_hand
player_hands_id BIGINT(20)
hand_id BIGINT(20)
Indexes
messages
id BIGINT(20)
content TINYTEXT
recipient BIGINT(20)
sent DATETIME(6)
uid BIGINT(20)
Indexes
user_game_history
user_id BIGINT(20)
game_history_id BIGINT(20)
Indexes
follow
id INT(11)
followingid BIGINT(20)
mute_notifications BIT(1)
fk_user_id BIGINT(20)
block_user BIT(1)
Indexes
player_hands
id BIGINT(20)
bet DOUBLE
dealer BIT(1)
score INT(11)
split BIT(1)
stand BIT(1)
blackjack_id BIGINT(20)
hand_id BIGINT(20)
Indexes
users
id BIGINT(20)
first_name VARCHAR(255)
last_name VARCHAR(255)
phone_number VARCHAR(255)
role ENUM(...)
user_biography VARCHAR(255)
login_info_id BIGINT(20)
lobby_lobby_id BIGINT(20)
chips DOUBLE
username VARCHAR(255)
Indexes
user_stats
user_stats_id VARCHAR(255)
game VARCHAR(255)
losses INT(11)
net DOUBLE
percentage DOUBLE
user_id BIGINT(20)
wins INT(11)
id BIGINT(20)
Indexes
users_hands
user_id BIGINT(20)
hands_id BIGINT(20)
Indexes
user_user_stats
user_stats_id VARCHAR(255)
user_id BIGINT(20)
Indexes
black_jack
id BIGINT(20)
order VARBINARY(255)
cards_id BIGINT(20)
lobby_lobby_id BIGINT(20)
Indexes
lobby
lobby_id BIGINT(20)
game_id BIGINT(20)
Indexes
login_info
id BIGINT(20)
password VARCHAR(255)
username VARCHAR(255)
user_id BIGINT(20)
Indexes
black_jack_settings
id BIGINT(20)
amount_of_decks INT(11)
dealer_stand_on INT(11)
max_bet INT(11)
min_bet INT(11)
user_id BIGINT(20)
Indexes
lobby_players
lobby_lobby_id BIGINT(20)
players_id BIGINT(20)
Indexes