
Design Document for Cards With Friends

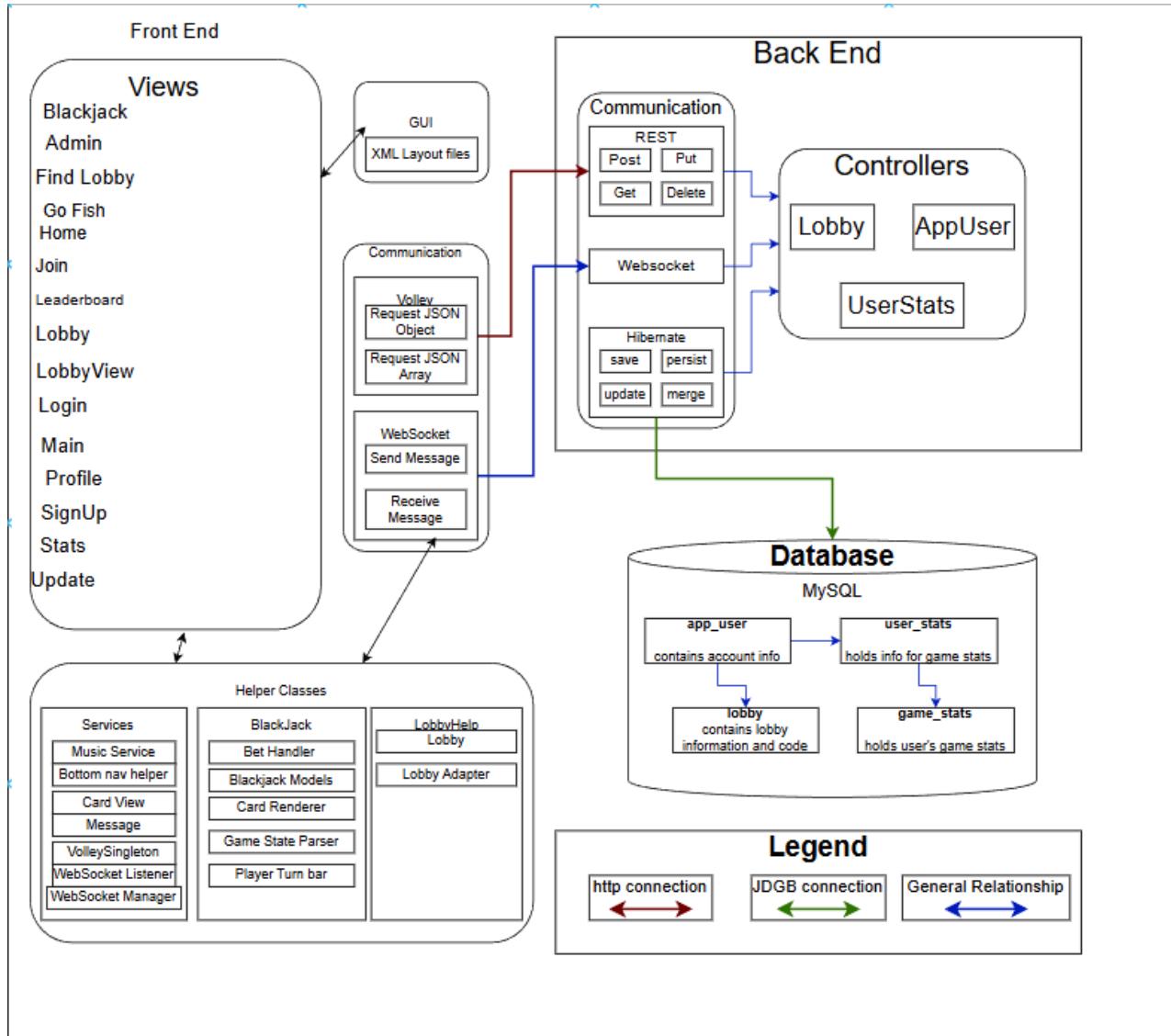
Group 1_Srijita_5

Jake Breyfogle: 25% contribution

Colten Stevens: 26% contribution 😊

Alex Behm: 24.5% contribution

Elijah Herring: 24.5% contribution



Frontend:

- BlackjackActivity
 - Generates and handles the blackjack gamestate sent by the websocket and displays it to the user
 - This activity uses multiple helper classes:
 - BlackjackModels: Defines an object that represents the gamestate being sent over by the websocket
 - CardRenderer: Handles displaying the cards and hand values of both the player and the dealer with animations
 - GameStateParser: Parses the websocket JSON to create a blackjack game model
 - PlayerTurnBar: Displays a UI bar at the bottom of the screen with everyone in the game to indicate whose turn it is and who you are currently viewing
- Gofish
 - Place where users can play Gofish with the following elements
 - Popup menu asking which player you want to ask a question to
 - Follow up popup menu asking which card
 - Players ask other players for cards in an effort to make pairs, called books
 - Every players information is on display at the top of the screen
 - Information that is displayed: books collected,

Backend:

- BlackJack Game:
 - Blackjack card: extends My card class (my card holds both suit and value) to now have a isShown variable to determine if card is face up or face down on table.
 - BlackJack Dealer: Robot blackjack dealer that HITS or STANDS based on value of current cards in hand. (hand = a list of Blackjack Cards)
 - BlackJack Game: Game engine behind BlackJack. Uses all other blackjack classes to construct a game with a dealer and players. functions in blackjack game called by websocket and executed in a blackjack game. After the end of every round, game stats are updated.
 - BlackJack player: mimics a player in blackjack, each player has a hand and carries functions to determine what actions they can take in the game. canSplit or canDouble boolean.
 - Blackjack Websocket, broadcast game state as well as notifies players if it is their turn. Takes in player decision jsons and then calls corresponding functions in the BlackJack game.
- Go Fish game:
 - GoGish Deck: a deck of 52 cards that are pulled from into go fish player hands.
 - Go Fish Player: an object mimicking a person playing go fish. GoFish player contains a user name and a current hand (list of cards). A number of completed books and a user reference called to update game stats at the end of the game.
 - Go Fish game: game engine behind the game go fish. contains functions to deal out cards to players from constructed go fish deck, how a player can take a turn, and lastly whose turn it is in the game.
 - Go Fish Service: Sends game state through the web socket. Handles updating players game stats at end of go fish game. also handles people joining/leaving the game lobby.
- Lobby Folder:
 - Lobby: Contains join code, lobby ID, GameType, and list of app users in sql database.

- Lobby controller: Contains functions for http request to lobby sql table. Allows our front end to create, delete, edit users in lobbies etc.
 - LobbyDTO: Constructs json to send of LobbyList web socket for out browse lobby feature.
 - LobbyListWebSocket: Allows the front end to view all current lobbies in sql database and join from that page.
 - Lobby Service: keeps the web socket open while all lobbies are being parsed by DTO and sent over the web socket.
 - LobbyWebSocket: Basic web socket that acts as a chat room for players in the lobby before the host starts the game.
- User folder:
 - App User: SQL entity containing a username, password, email, firstname, lastname and age. Also contains boolean values for isModerator and isAdmin.
 - App User Controller: handles https request to, create , delete, and edit app users in SQL database.
- User stats:
 - User Stats: Links app user to a certain set of stats
 - Game Stats: extends user stats and creates basic stats for all games such as games played
 - Blackjack Stats: holds stats specifically for blackjack for a certain app user
 - Crazy8Stats: holds stats specifically for Crazy8 for a certain app user
 - EuchereStats: holds stats specifically for Euchre for a certain app user
 - GoFish stats: holds stats specifically for GoFish for a certain app user

