Design Document for Clever Quest

Group: 1_muzakr_6

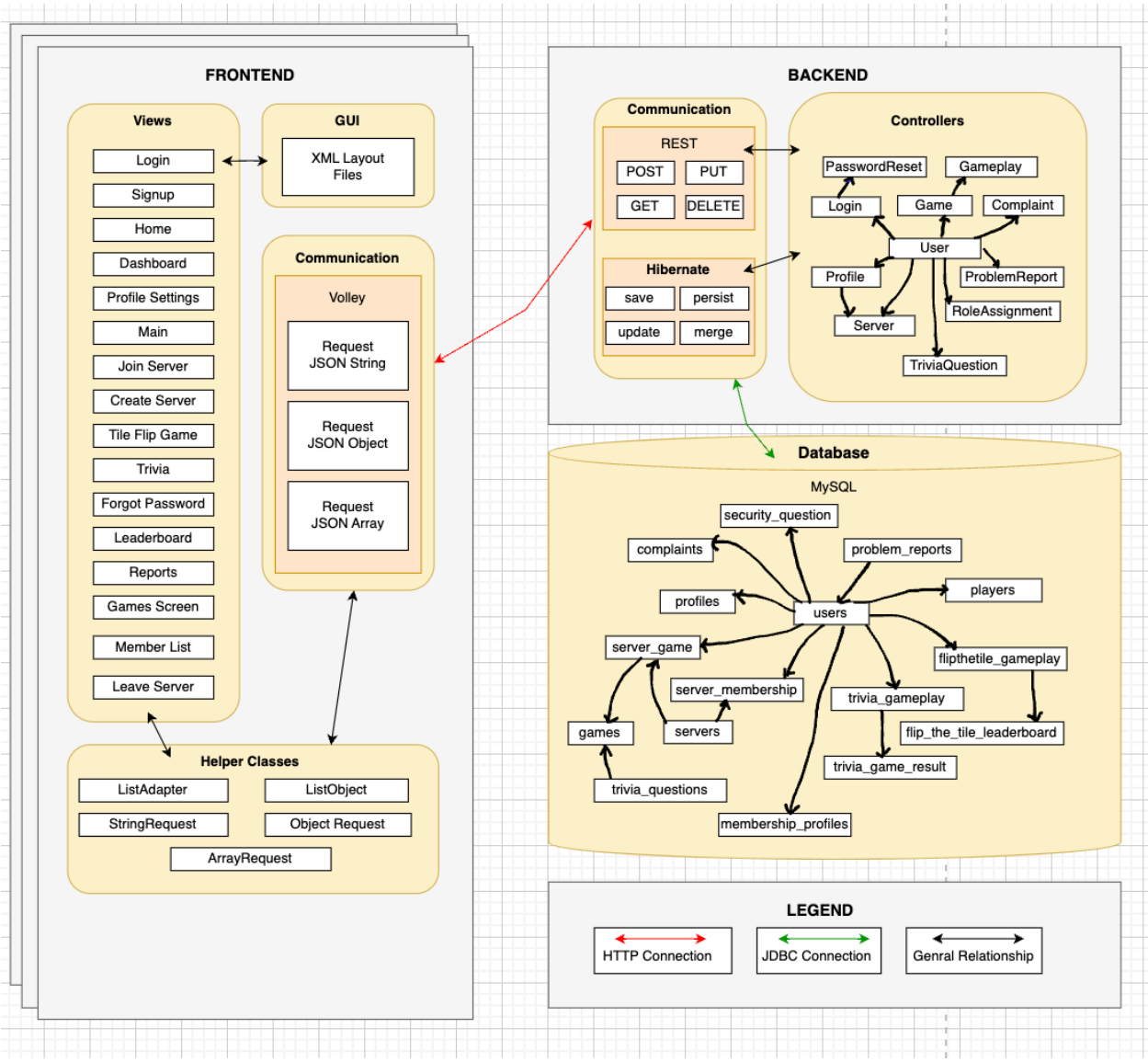
Maisha Rahman Chowdhury: 25% contribution

Siyona Gorre: 25% contribution

Clair Ammons: 25% contribution

Shiny Pokuri: 25% contribution

Block diagram



Frontend *(Currently Implemented)*

Activities

The user interacts with the app with various android views. When they log in, they will be presented with a list of servers they are a member of, allowing them to click on any server to enter it, or create or join a new server. Once in a server, the user is presented with a dashboard that allows them to do various things depending on their role within the server. Normal players can do things such as view the server's members, play games, or view the leaderboard. If they are a server admin they can assign roles or games to users. Moderators can view reports made by users and possibly punish users based on them. When the user joins a trivia game, they will be shown a question. Upon that user or any other answering correctly, the next question will be shown to all connected users. When the user joins a tile flip game, they will be presented with a grid and will be expected to flip the tiles over in order to match colors. At the end of each game, they will see their scores, which will then be displayed on the leaderboard.a

Backend *(Currently Implemented)*

Communication

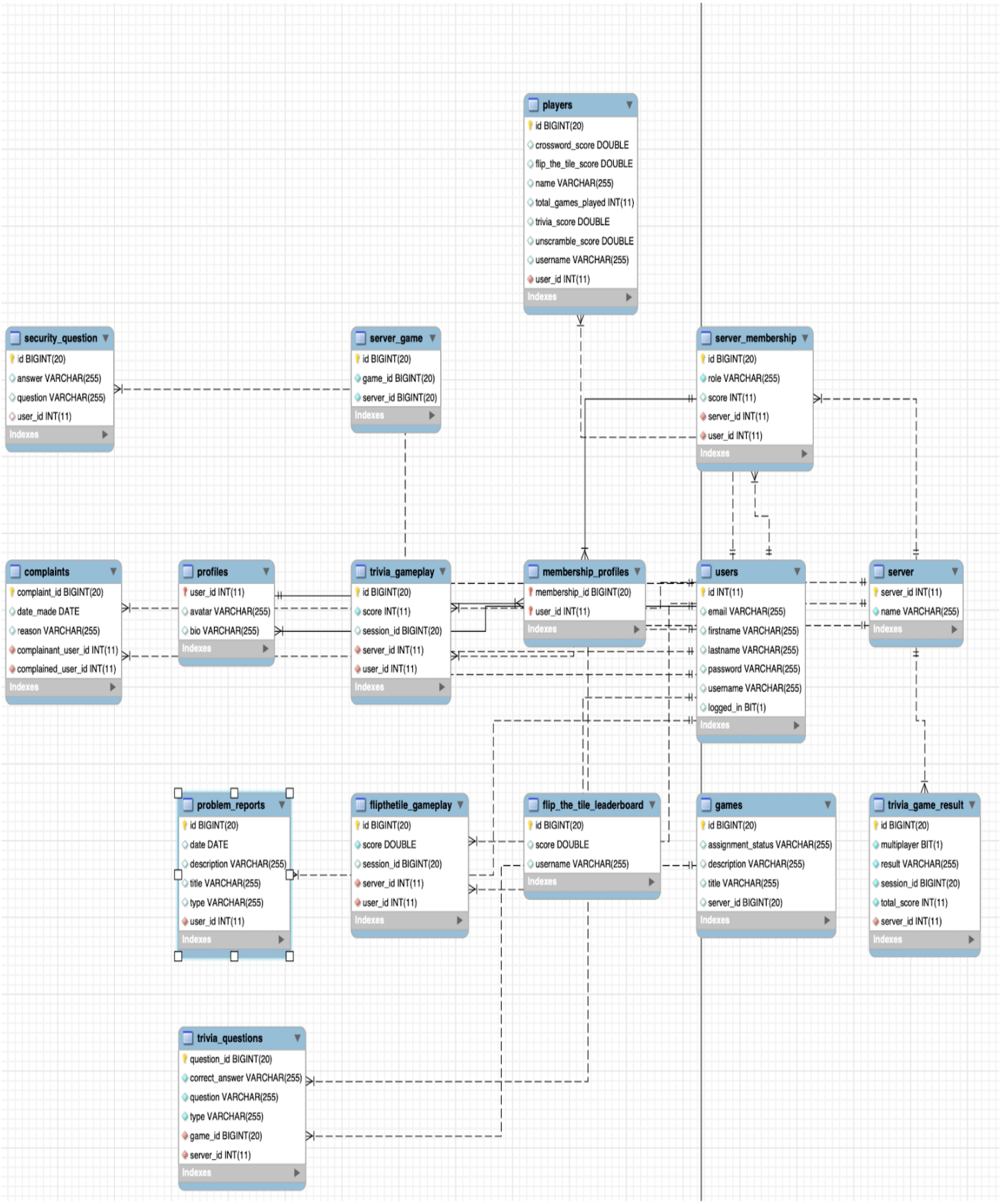
The backend uses mappings to update the database based on information sent to the given mappings' URLs. These include:

- **POST**: send information on an item to be added to the database.
- **GET**: request information, often with an identifier for the item requested from the database.
- **PUT**: send information to update a specific item in the database.
- **DELETE**: send an identifier to delete a specific item from the database.

Controllers

User: Contains the above mapping for creating a new user, gets the user by username, logs out, and updates a user's information in users table
Login: Checks the password with the database and logs a player in, sets logged in to true, invalid otherwise checks from users table.
PasswordReset: Resets password with the previously added security question set by the user, let's user choose a security question and answer, updates the security question in the database, and gets a userId by email checks from users table, security_question table has OneToOne relation users table.

Complaint: Gets all complaints, Creates a new complaint, Gets a complaint by ID, Deletes a complaint by ID, Updates a complaint, Gets complaints by complainant user ID checks from complaints table.OneToOne relation users table.
ProblemReport: Gets all reports, Creates a new report, Gets a specific report by ID, Updates an existing report by ID, Deletes a report by ID in problem_reports table.
Profile: Displays Profile Information, Creates Profile Endpoint, Edits Profile Information, Updates Password, Username, Avatar, Bio, Deletes Account, from users, profiles table.ManyToMany mapped by profiles with the server_membership table ,OneToOne relation users table.
Server: Creates Server, Joins a server by request parameters, Joins a server by request body, Leaves a server using request body. OneToMany with server_membership, uses the servers table for server information, membership related information in server_membership table.
TriviaQuestion: Fetches the Game with ID 2 from the database, Fetches the Server using serverId, Creates TriviaQuestion and sets the game and server, Saves trivia question ManyToOne with servers table, uses trivia_questions table.
Game: Gets all available games, Gets all games for a specific server, Assigns a game to a server, Assigns default games to a server when a player joins, Updates game details, Deletes a game assignment from a specific server in game table
Gameplay: Gets all information about a game and each session and scores, uses server_gameplay table.
RoleAssignment: The admin (who created the server) assigns a moderator, admin, uses servers table.


players
id BIGINT(20)
crossword_score DOUBLE
flip_the_tile_score DOUBLE
name VARCHAR(255)
total_games_played INT(11)
trivia_score DOUBLE
unscramble_score DOUBLE
username VARCHAR(255)
user_id INT(11)
Indexes
security_question
id BIGINT(20)
answer VARCHAR(255)
question VARCHAR(255)
user_id INT(11)
Indexes
server_game
id BIGINT(20)
game_id BIGINT(20)
server_id BIGINT(20)
Indexes
server_membership
id BIGINT(20)
role VARCHAR(255)
score INT(11)
server_id INT(11)
user_id INT(11)
Indexes
complaints
complaint_id BIGINT(20)
date_made DATE
reason VARCHAR(255)
complainant_user_id INT(11)
complained_user_id INT(11)
Indexes
profiles
user_id INT(11)
avatar VARCHAR(255)
bio VARCHAR(255)
Indexes
trivia_gameplay
id BIGINT(20)
score INT(11)
session_id BIGINT(20)
server_id INT(11)
user_id INT(11)
Indexes
membership_profiles
membership_id BIGINT(20)
user_id INT(11)
Indexes
users
id INT(11)
email VARCHAR(255)
firstname VARCHAR(255)
lastname VARCHAR(255)
password VARCHAR(255)
username VARCHAR(255)
logged_in BIT(1)
Indexes
server
server_id INT(11)
name VARCHAR(255)
Indexes
problem_reports
id BIGINT(20)
date DATE
description VARCHAR(255)
title VARCHAR(255)
type VARCHAR(255)
user_id INT(11)
Indexes
flipthetile_gameplay
id BIGINT(20)
score DOUBLE
session_id BIGINT(20)
server_id INT(11)
user_id INT(11)
Indexes
flip_the_tile_leaderboard
id BIGINT(20)
score DOUBLE
username VARCHAR(255)
Indexes
games
id BIGINT(20)
assignment_status VARCHAR(255)
description VARCHAR(255)
title VARCHAR(255)
server_id BIGINT(20)
Indexes
trivia_game_result
id BIGINT(20)
multiplayer BIT(1)
result VARCHAR(255)
session_id BIGINT(20)
total_score INT(11)
server_id INT(11)
Indexes
trivia_questions
question_id BIGINT(20)
correct_answer VARCHAR(255)
question VARCHAR(255)
type VARCHAR(255)
game_id BIGINT(20)
server_id INT(11)
Indexes