UNIVERSITÀ DI PISA

MSc in Computer Engineering

Distributed Systems and Middleware Technologies

The Winning Agreement

TEAM MEMBERS:

Giovanni Barbieri

Federico Cavedoni

Alessio Di Ricco

Academic Year 2023-2024

Content

1. Project Specifications

1.1 Introduction

The application we would like to propose is inspired by the game **"L' Intesa Vincente"** of Rai1. In which there is one minute to guess as many words as possible with the following method:

There are three players, two of them know the word while the third does not and has to guess it. The two players must make the third player understand the word by saying one word each. When the third person wants to try to guess, he writes the word and press the button to send it. If it is correct, they gain a point, otherwise they lose a point.

We would like to make a webapp with the following features.

1.2 Functional Requirements

In our application we have implemented 3 types of actors:

- **Unregistered user**
- **Registered user**
- **Admin**

The possible action that each actor can perform are:

1. An **unregistered user** can:
   - Create an account.
   - Log-in to an account.

2. A **registered user** can:
   - Login/logout
   - Add friends to play with
   - Invite friends to play.
   - Play a match in one of the two roles.
   - Leave the game.
   - Browse his past matches.

3. An **admin** can:
    - Login/logout
    - Remove user.
    - Browse all the matches.

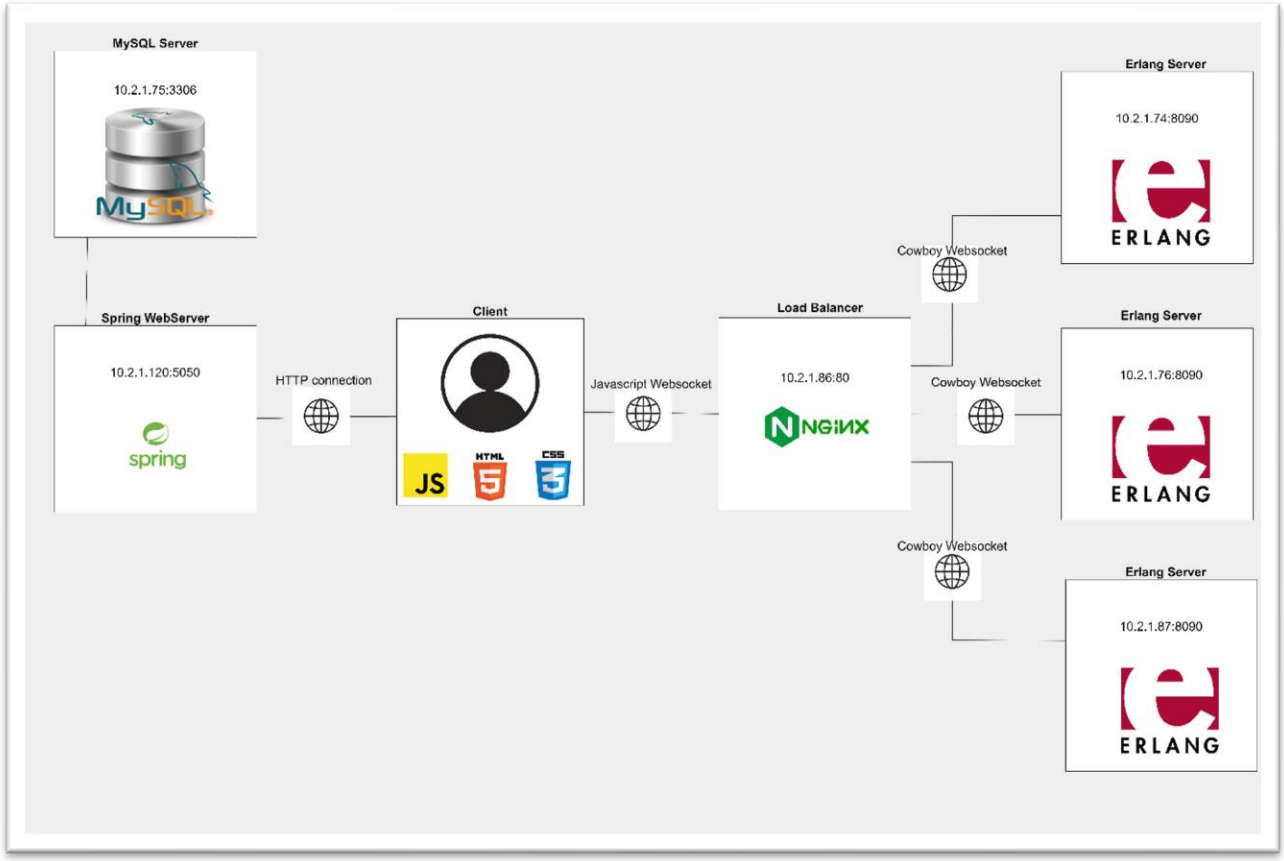
1.3 Non-Functional Requirements

- Concurrent service accesses management
- Strong consistency during the game phase
- High service availability
- Allow high horizontal scalability.

1.4 Synchronization and Communication issues

- Multiple clients exchange words with each other in games concurrently. The game interface of each client node must be synchronized.
- Timers of users must be synchronized.
- Notify the user if he has been invited by another user to play a game.
- The server can independently handle any number of matches, which can also run concurrently.

2. System Architecture



2.1 Architecture overview

The application architecture is divided into various distributed components to manage different functionalities. For all operations except gaming, the client interacts with the Spring Web Server, residing on node 10.2.1.120 and listening on port 5050.

The necessary data is stored within the MySQL server, hosted on node 10.2.1.75 on port 3306. As for gaming, it is handled by the Erlang Server, distributed across three nodes: 10.2.1.74, 10.2.1.76, 10.2.1.87, all operating on port 8090. When the client requests to play, it passes through node 10.2.1.86, where Nginx is running, responsible for load balancing requests among Erlang nodes, directing the request to the least loaded Erlang server at the moment.

Communication with Erlang occurs via WebSocket, utilizing the Cowboy framework.

2.2 Spring Web Server

The Spring Web Server serves for managing various functionalities within the application. It handles user authentication, data retrieval from the mysql db, and processing of requests from clients. Specifically, its responsibilities include:

User Authentication: Verifying user credentials and granting access to authenticated users.

Signup: Allows users to create new accounts within the application, providing necessary credentials and information for registration.

View, remove, and add friends: Enables users to manage their social connections by viewing, deleting, and adding friends to their network.

Invite friends to a game: Provides users with the ability to invite their friends to join them in a game.

Browse games: Permits users to explore their past games.

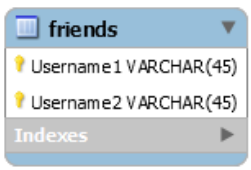
View and remove users (only for admin): Grants administrative users the privilege to view and remove other users from the platform.

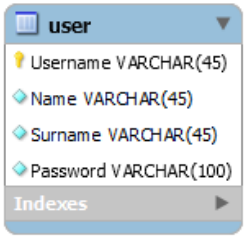
Player Waiting Management: handle the waiting process for players awaiting others to join the game session.

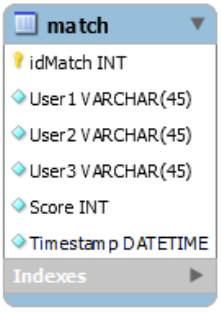
Unique ID Assignment: Assigning a unique identifier to each game session, ensuring distinct identification and management of concurrent games.

Once all players are ready, the Spring Web Server orchestrates the commencement of the game session. It responds to clients by providing the usernames of all participants in the game.

2.3 MySQL Server







2.4 Erlang Server

We chose to use Rebar3 for the development of the Erlang server with Cowboy in order to simplify the management of the project's dependencies and automate the build process, therefore installing it on the 3 containers that run the Erlang server.

The erlang server has two main modules:

the **server_handler** module is responsible for managing WebSocket connections and interactions between players during a game. Initially, the init function is called to initialize the WebSocket connection and set the initial state of the server. Next, the websocket_handle function handles messages sent by clients, decoding the requested action and coordinating responses based on the current state of the game.

the **player_handler** module manages the actions and interactions of individual players during a game. Functions within this module allow players to submit a word, guess a word, and wait for other players' actions.

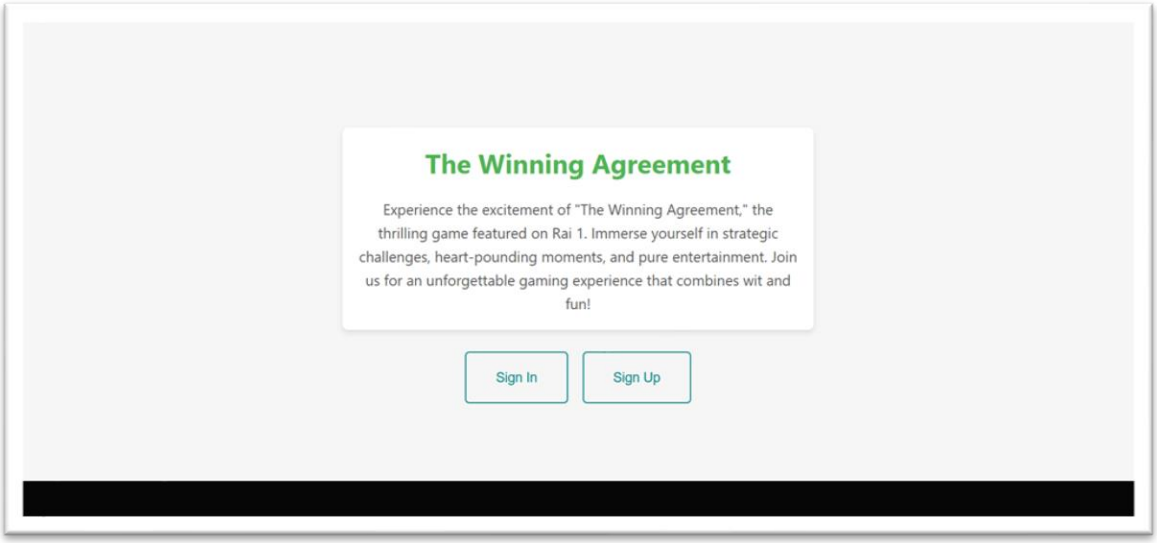
Each player corresponds to a relative process on the server, the process is registered with the username of the player to which it refers, so that the other players in the game can communicate with this process only by knowing the username of the player to whom it refers.

The usernames of the players in the game is information that each player has previously received from the spring web server

3. User Manual

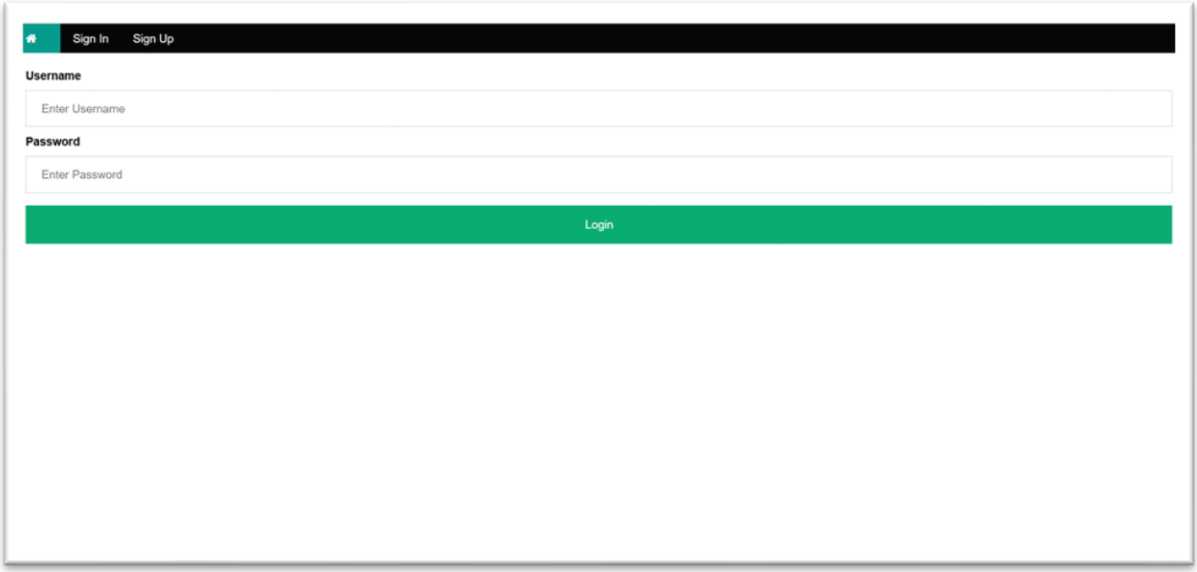
3.1 Home page

This is the homepage of the webapp, where the user can read a brief description of the game and then decide to log in or register by pressing the corresponding buttons.



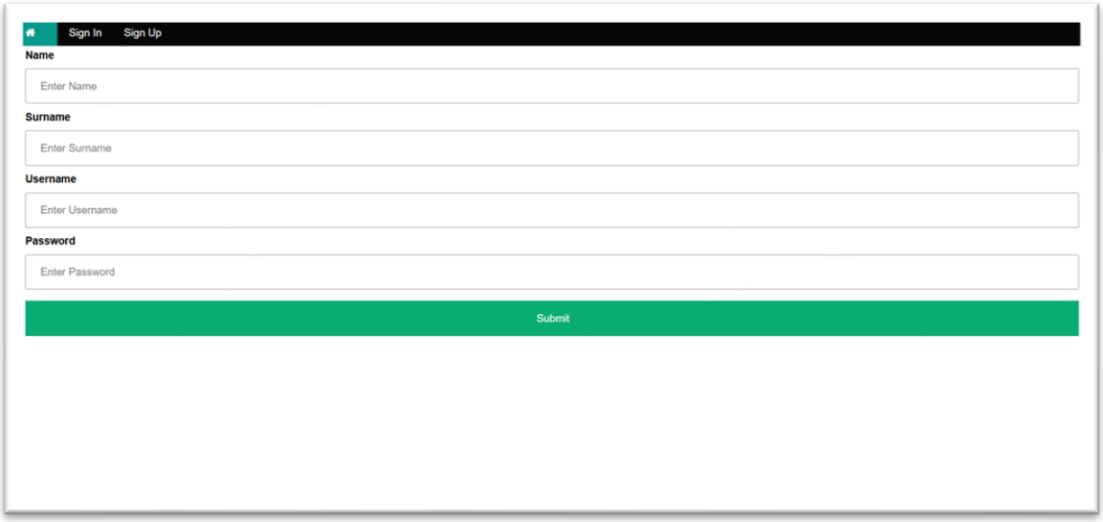
3.2 Login page

This is the login page where the user can log in using the 'username' and 'password' of his account. Alternatively, the user can use the buttons to return to the home page or to go to the registration page.



3.3 Signup page

On this page the user can register on the site by entering his personal data, such as name, surname, username and password. Once registered, the user will be able to log in to the site using the credentials entered for registration. The user can also navigate between the login page and the home page using the appropriate buttons.



3.4 User section

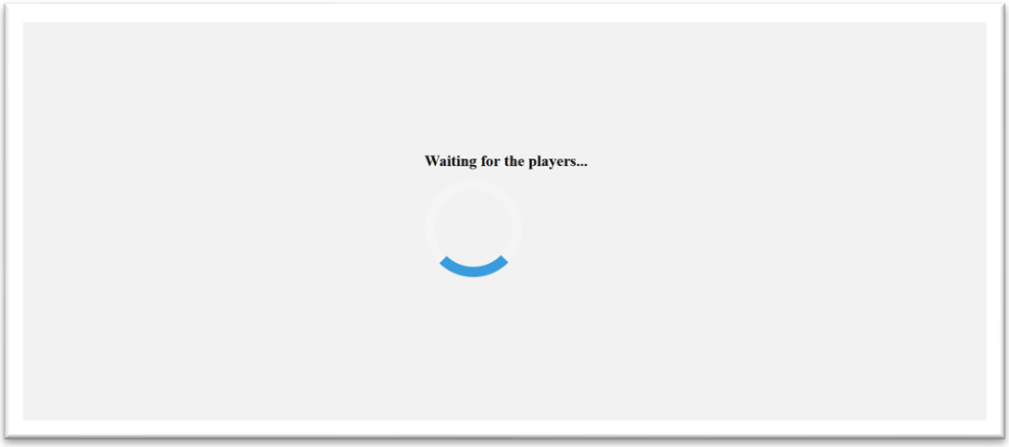
3.4.1 User HomePage

After logging in, the user accesses the homepage of his profile, from which he can start a game, check on his friends or see his past games via the buttons in the center. Finally, the user can decide to log out using the button at the top left.

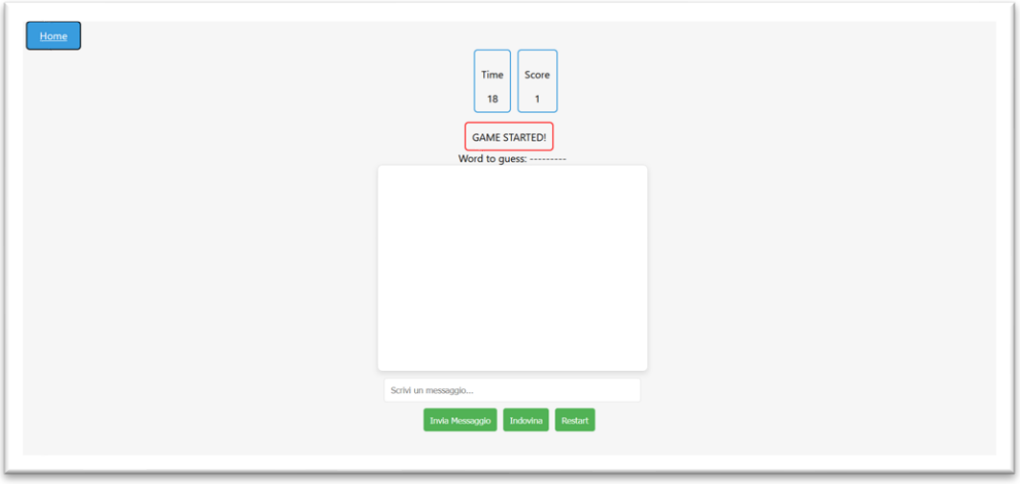
3.4.2 Game page

When a user presses the 'Play game' button he can decide whether to be a 'guesser', i.e. the one who has to guess, or a 'player', i.e. the one who has to make the guesser guess the word.

Initially there will be a loading page, in which the webapp waits for three players, a guesser and two players, to play the game.

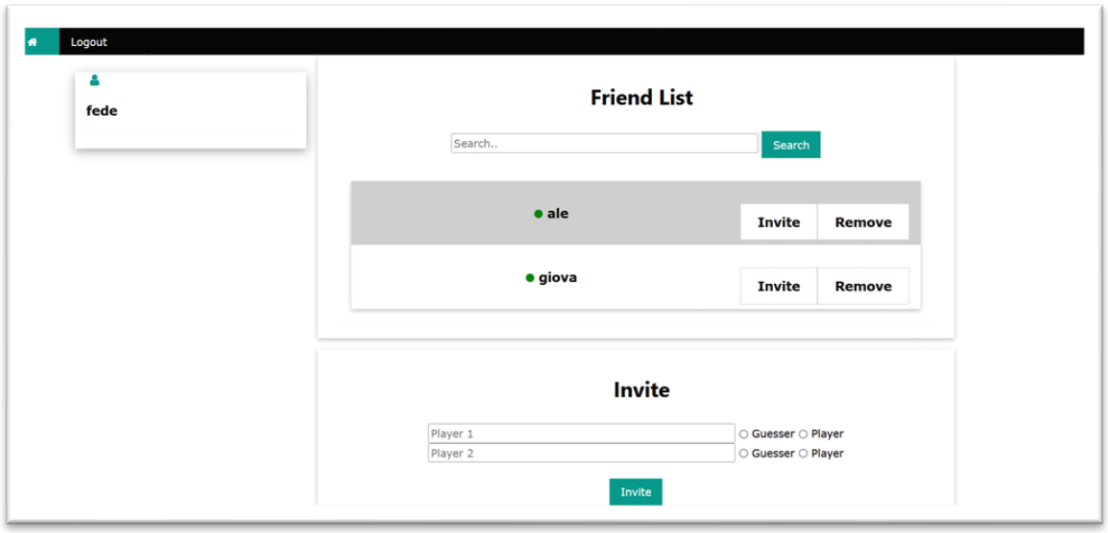


When the three players have connected, the game will begin, a word will be distributed to the two players who will have to write words to make the guesser guess by sending them via the appropriate button. The guesser can try to guess the word by pressing the appropriate button and, if the word is correct, he will receive a point. At the end of the timer the game will be saved in the game history. Players will be able to restart the game or return at home at any time using the appropriate buttons.



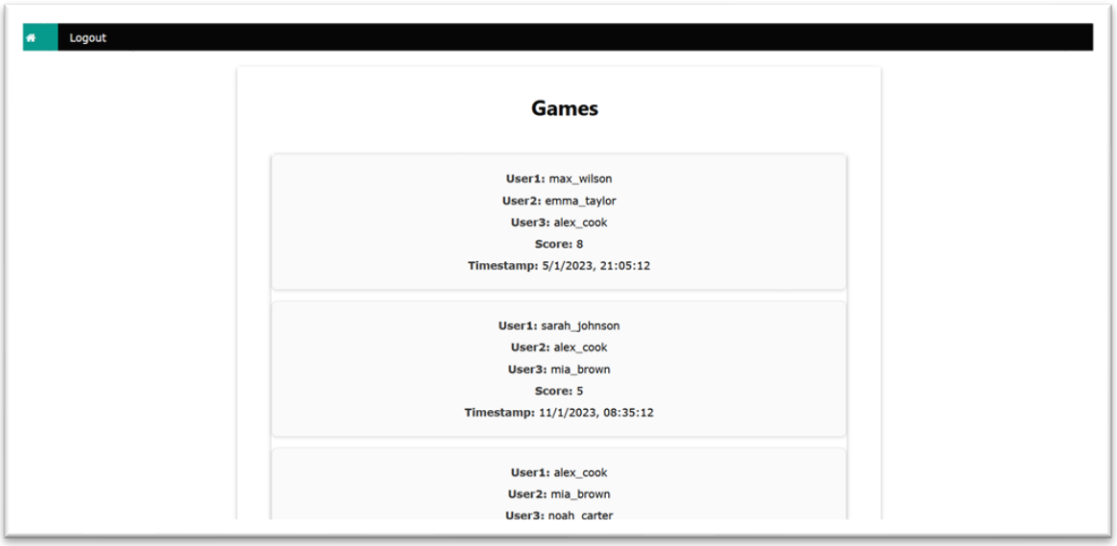
3.4.3 Friends page

On this page the user can check if his friends are online or offline using the color of the dot to the left of the friend's name in the friends list. Moreover, the user can invite a friend to play the game, can remove a friend or search and add a user to his friends via the search bar. Finally, the user can, using the buttons at the top left, return to the main page or log out.



3.4.4 Browse games page

On this page the user can see all his past games, the players, the score and the timestamp in which they were played. The user can also return to the home page or log out using the buttons at the top left.



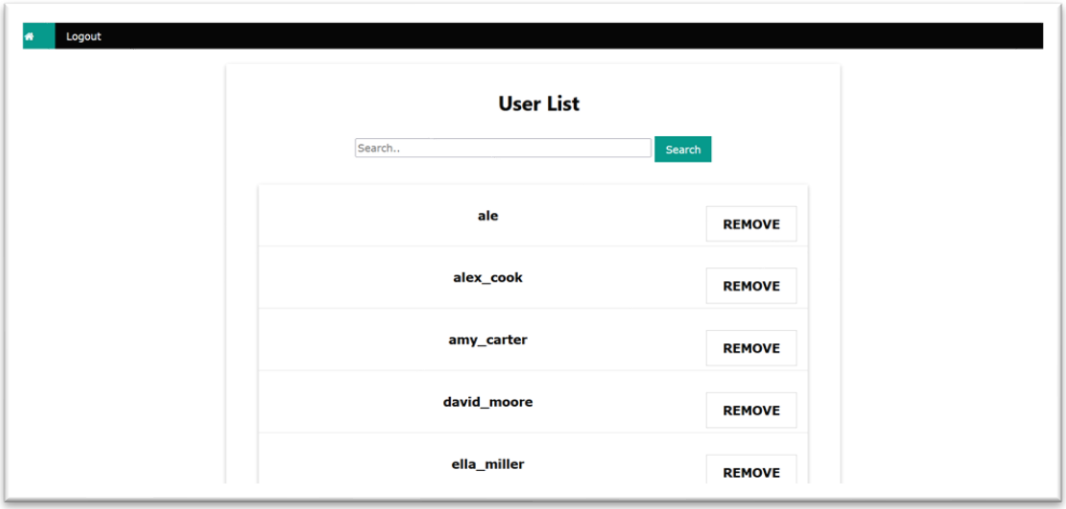
3.5 Admin section

3.5.1 Admin HomePage

If you log in to the admin profile you will be redirected on this page, which represents the homepage of the admin who can decide to open the user management page, open the global match search page or log out using the appropriate buttons.

3.5.2 Users Page

This is the user management page, where the admin can see all the users registered on the site, search for one in particular using the search bar and remove users from the site by deleting them from the database. Finally, the admin can return to the home page or log out using the buttons at the top left.



3.5.3 Browse Games page

This is the page where the admin can see all the games played on the webapp by all users and search for the games played by a specific user via the search bar. Finally, the admin can return to the home page or log out using the buttons at the top left.