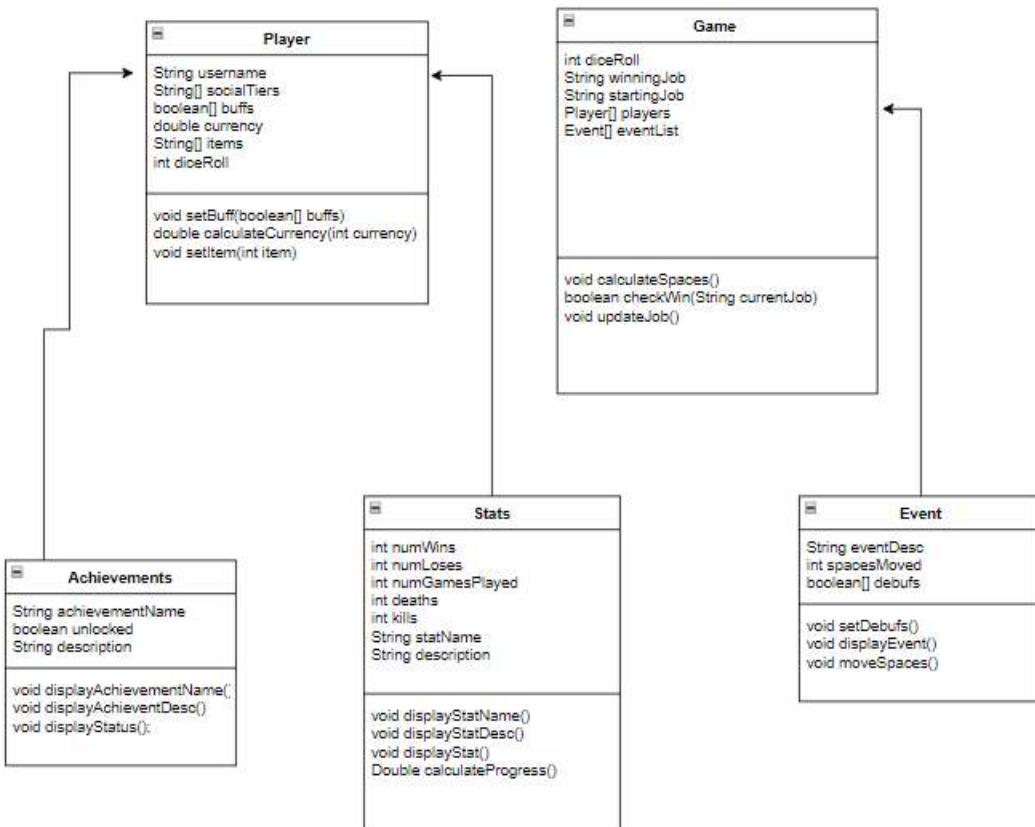


List of Frameworks/ Architecture

- **Socket.io** – We need to avoid using HTTP send/response for messages as this method would not be able to handle multiple players using the same server to play multiple games. Therefore we need a way to handle bidirectional and even-o-based communication between our servers and users through web sockets. Socket.io will provide the services needed to achieve this goal with a high amount of scalability. The socket.io library consists of two parts, which are: a client-side library that runs in the browser, and a server-side library for Node.js.
- **ReactJS** – This library will be used to build our interface for our application. ReactJS is a flexible framework that will allow us to build our own components that are managed in their own state so that we can design our application closer to the ideas that we have in mind. React will also efficiently update and render the right components according to how our data changes in the game. Therefore, React will be how we interface our backend to our frontend.
- **AWS Cognito** – Cognito will be used to track everything that is connected to the player's account. This includes the log in system and storing everything attached to the players account such as: images, player status, and any other objects. AWS Cognito will handle the authentication and authorization system. Therefore Cognito will also cover security concerns such as proper encoding and encryption methods to securely store usernames and password information.
- **AWS CloudFront** – This service will be the network that we use to distribute our application. We will use an EC2 instance for our Web Application server. An EC2 instance is a scalable virtual machine that will provide all the CPU, storage, and memory that is needed for our application. The server will run on linux.
- **Terraform** – This is an open source infrastructure as code language that we will be using to build our VPC to create and modify AWS resources as needed. Using Terraform to manage our AWS infrastructure means we can create small changes without worrying about breaking the network due to its ability to save backups of prior commits.

Class Diagram



Robust Diagram 1

Use Case: Create Account

BASIC COURSE:

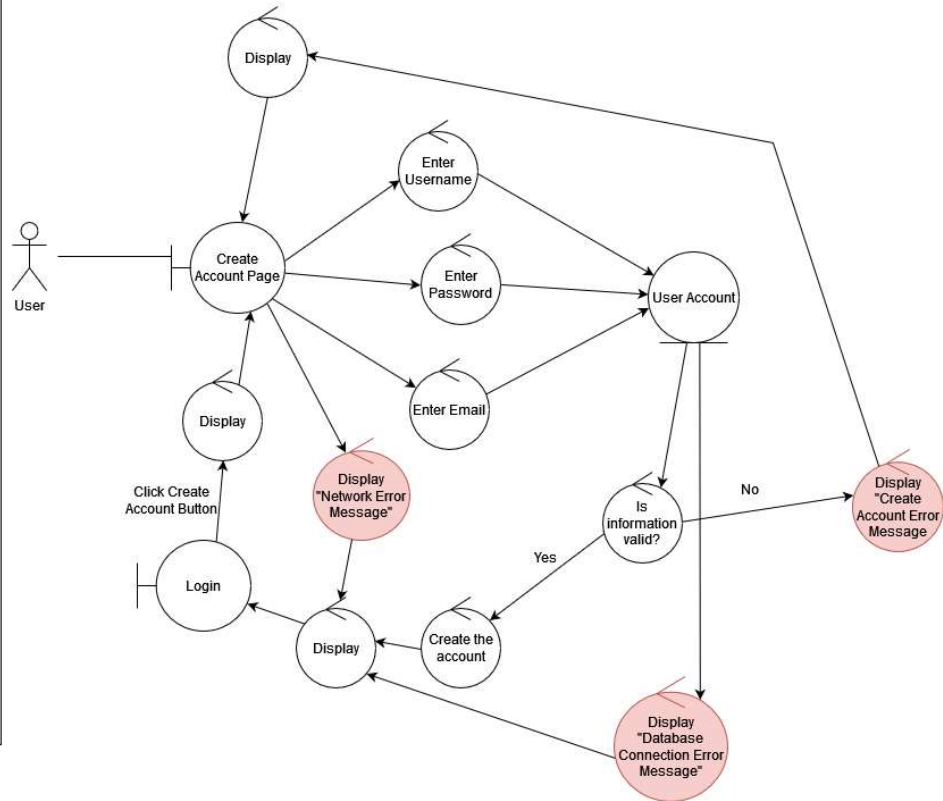
On the Login Page, the User clicks the Create Account Button and the system displays the Create Account Page. The user enters their username, password, email, and phone number into the fields and clicks the Create Account Button. The system checks this information in the database to see if it is valid. The system then displays the Home Page.

ALTERNATE COURSES:

Account existing error: The system shows a Create Account Error Message displaying the username, password, email, and/or phone number already exist for an account.

Network error: The system shows a Network Error Message and redirects the user to the Login Page.

Database connection error: The system shows a Database Connection Error Message and redirects the user to the Login Page.



Robust Diagram 2

Use Case: Create Private Lobby

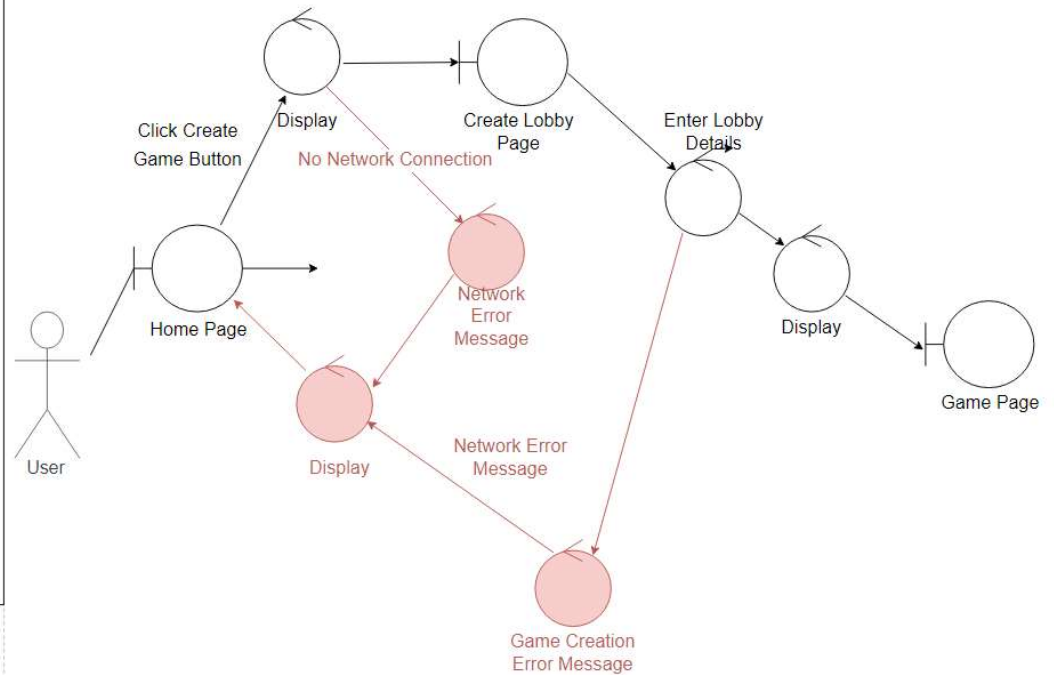
BASIC COURSE:

On the Home Page, the user clicks the Create Game Button, then the system displays the Create Lobby Page. The user enters the information for the game they are trying to create. The system then takes the user to the Game Page where the game is played.

ALTERNATE COURSES:

Network error: The system shows a Network Error Message and redirects the user to the Home Page.

Game Creation error: The system shows a Game Creation Error Message and redirects the user to the Home Page.



Robust Diagram 3

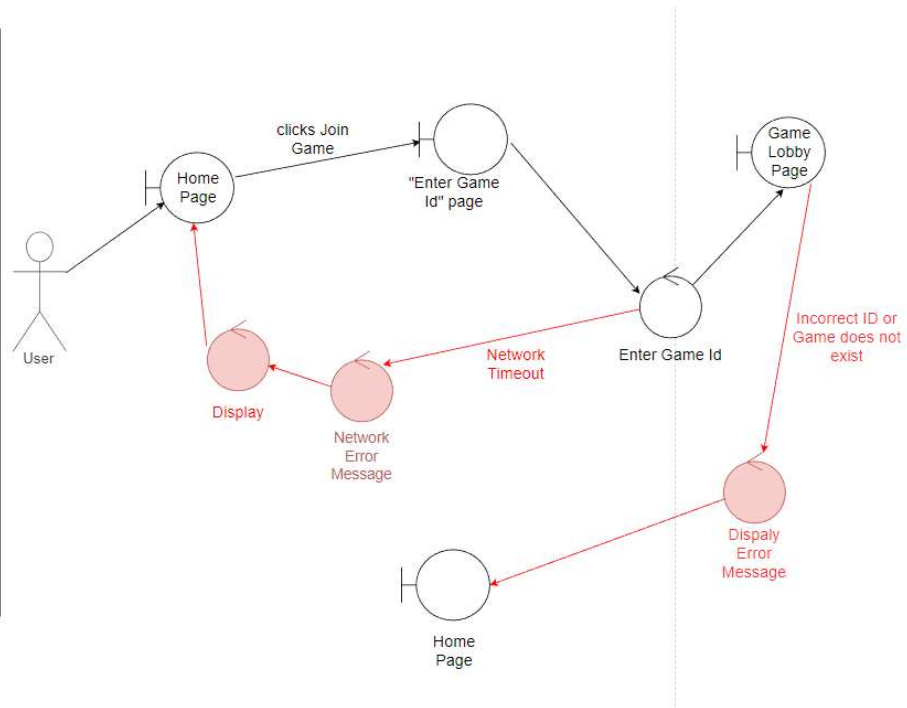
Joining a Game

Base Case: After this user has a successful authentication logging into their account, the user will have an option to join another player's existing game. Upon choosing to join a user's game, the user will need to provide a unique game ID. If the ID is correct, the system will then connect the user with all other players within the lobby enabling the user to chat and interact with other players.

Alternate Case:

Network timeout: If the user attempting to join a game has a server timeout, the user will be provided an error message saying there was an error attempting to connect to the users lobby. They will then be taken back to their homepage where they can try to attempt the process again.

Incorrect ID or Game does not exist: The game ID provided is incorrect or no longer exists. If the game ID is incorrect or does not exist, an error message will be provided stating that the ID is incorrect or does not exist and will be taken back to their home page.



Robust Diagram 4

Viewing Stats/ Achievements Page

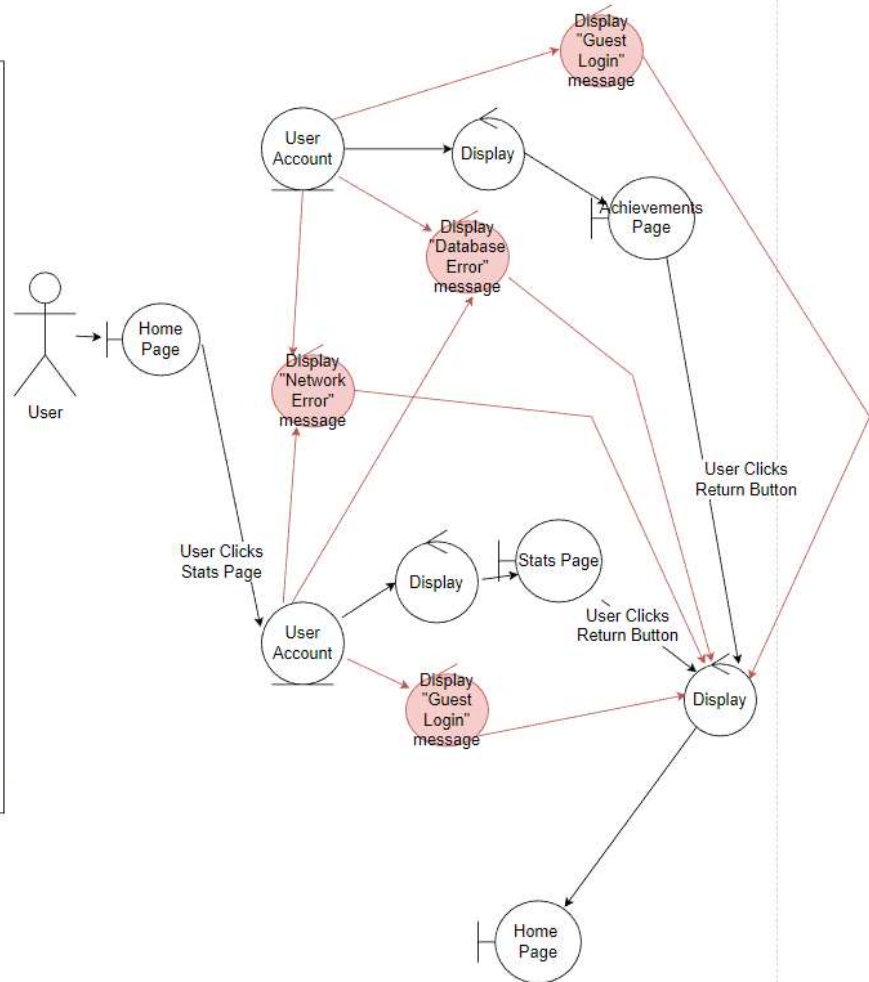
Base Case: Once the User has successfully logged in to their account they will be on the Home Page, which will have both the Achievements Button and the Stats Button. Once the User clicks the Achievements Button they are sent to the Achievements Page. If the User clicks the Stats Button they will be sent to the Stats Page. The system will gather the saved stats and achievements on the Users Account to display on the respective pages. Both pages will have Return Button. When the User clicks these buttons, they will be returned to the Home Page.

Alternate Case:

Guest Login Error: The system will display a Guest Login Message and return to the Home Page.

Network Error: The system shows a Network Error Message and redirects the user to the Home Page.

Database Connection Error: The system shows a Database Connection Error Message and redirects the user to the Home Page.



Sequence Diagram 1

Use Case: Create Account

BASIC COURSE:

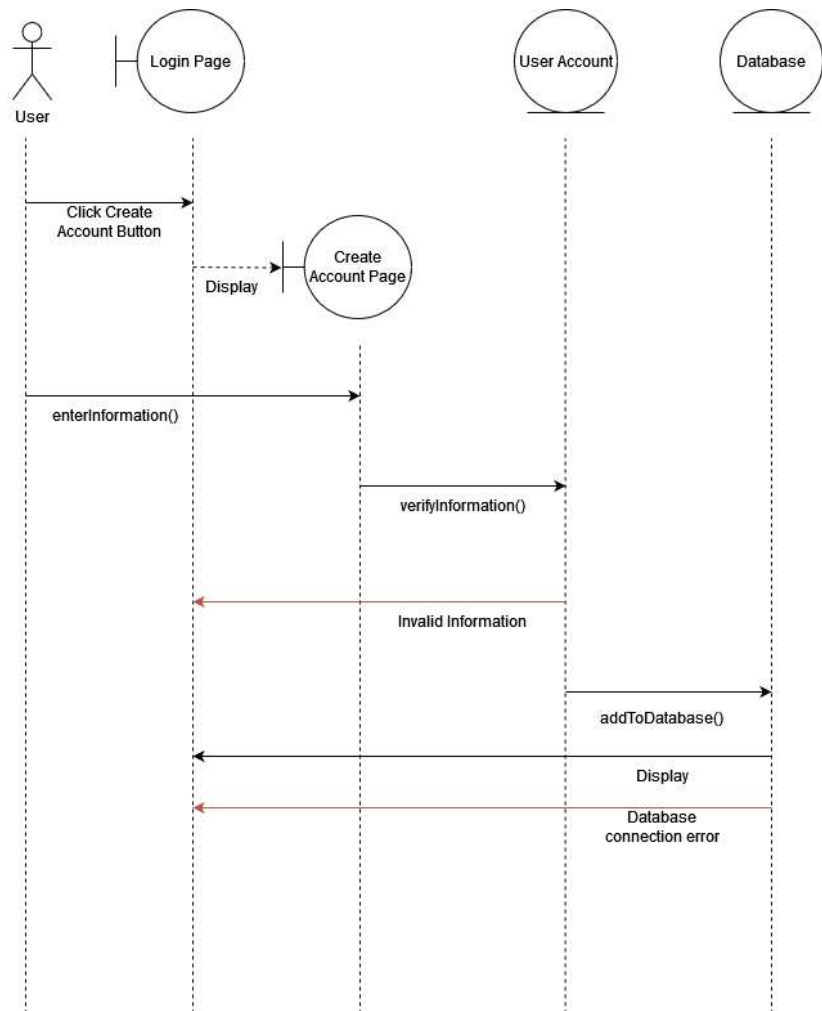
On the Login Page, the User clicks the Create Account Button and the system displays the Create Account Page. The user enters their username, password, email, and phone number into the fields and clicks the Create Account Button. The system checks this information in the database to see if it is valid. The system then displays the Home Page.

ALTERNATE COURSES:

Account existing error: The system shows a Create Account Error Message displaying the username, password, email, and/or phone number already exist for an account.

Network error: The system shows a Network Error Message and redirects the user to the Login Page. Home

Database connection error: The system shows a Database Connection Error Message and redirects the user to the Login Page.



Sequence Diagram 2

Use Case: Create Private Lobby

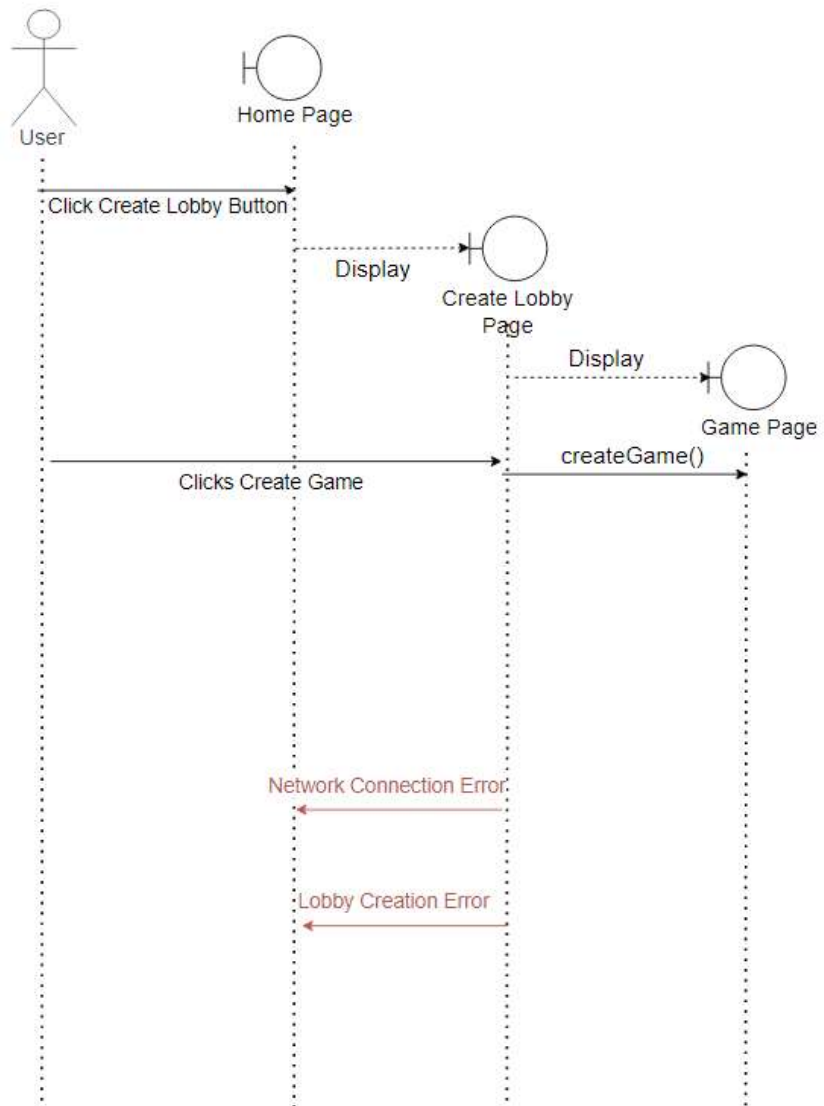
BASIC COURSE:

On the Home Page, the user clicks the Create Game Button, then the system displays the Create Lobby Page. The user enters the information for the game they are trying to create. The system then takes the user to the Game Page where the game is played.

ALTERNATE COURSES:

Network error: The system shows a Network Error Message and redirects the user to the Home Page.

Game Creation error: The system shows a Game Creation Error Message and redirects the user to the Home Page.



Sequence Diagram 3

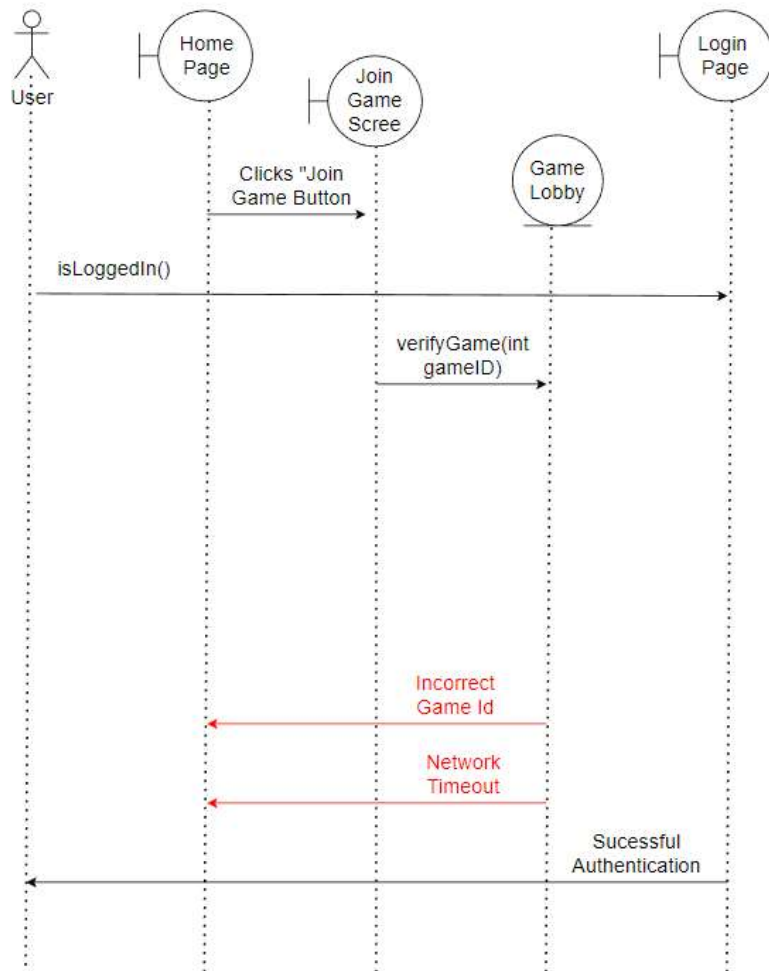
Joining a Game

Base Case: After this user has a successful authentication logging into their account, the user will have an option to join another player's existing game. Upon choosing to join a user's game, the user will need to provide a unique game ID. If the ID is correct, the system will then connect the user with all other players within the lobby enabling the user to chat and interact with other players.

Alternate Case:

Network timeout: If the user attempting to join a game has a server timeout, the user will be provided an error message saying there was an error attempting to connect to the users lobby. They will then be taken back to their homepage where they can try to attempt the process again.

Incorrect ID or Game does not exist: The game ID provided is incorrect or no longer exists. If the game ID is incorrect or does not exist, an error message will be provided stating that that the ID is incorrect or does not exist and will be taken back to their home page.



Sequence Diagram 4

Viewing Stats/ Achievements Page

Base Case: Once the User has successfully logged in to their account they will be on the Home Page, which will have both the Achievements Button and the Stats Button. Once the User clicks the Achievements Button they are sent to the Achievements Page. If the User clicks the Stats Button they will be sent to the Stats Page. The system will gather the saved stats and achievements on the Users Account to display on the respective pages. Both pages will have Return Button. When the User clicks these buttons, they will be returned to the Home Page.

Alternate Case:

Guest Login Error: The system will display a Guest Login Message and return to the Home Page.

Network Error: The system shows a Network Error Message and redirects the user to the Home Page.

Database Connection Error: The system shows a Database Connection Error Message and redirects the user to the Home Page.

