Polybius

Team 9 - Design Document

Lucas Hahn, Leo Liu, Christopher Sculley, Thomas Simons

Table of Contents

Purpose

For many reasons, the video game community tends to not interact with others face-to-face. While multiplayer games are a staple in the video game community, these games generally lack real world interaction. This lack of in-person communication means that social skills cannot be learned and improved through traditional, 2D video games. However, augmented reality is an upcoming technology that can be utilized to emulate virtual elements onto a real space.

Polybius aims to give players a unique gaming experience, by blending real-world interaction with imaginary gameplay in a way seldom seen before. Our purpose is to give players the opportunity to interact with each other and make real world connections by allowing players to play in collaborative 3D augmented reality games around a location in the real world. By displaying a map of local games and offering social features like profiles, statistics and messaging, our app promotes social interaction both virtually and in-person.

Functional Requirements

1. User Accounts

   As a user,

   a. I want to be able to create an account so that I can have a personalized experience
   b. I want to add other players as a friends so that I can build established connections
   c. I want to have a profile that others can see so that others can connect with me
   d. I want to see many statistics about the games I've played so that I can track my performance
   e. I want to send challenges to players to beat my high score so that I can challenge others
   f. I want to be able to search for other users so that I can connect with others
   g. I want a messaging system to message my friends so that we can continue to communicate outside of matches
   h. I want a easy-to-use interface that feels satisfying to use so that I will enjoy using the application

2. Game posting

   As a user,

a. I want to be able to find games around me so that I can easily join new games using a map
b. I want to be able to find specific minigames so that I can play the games I like the most
c. I want to host games to play with nearby players so that I can meet new people
d. I want to delete a game from the map when I'm done playing

3. Gaming Features

   As a gamer,

   a. I want bonuses for meeting up with new people so that I will be encouraged to meet others
   b. I want to keep track of my highscores so that I can see my progress and share my achievements
   c. I want to see the top players of any minigame so that I can strive to be the best
   d. I want to play this game multiple times so that I can meet new people
   e. I want to play tic tac toe using augmented reality against other players
   f. I want to play Connect 4 using augmented reality against other players
   g. I want to play paper toss using augmented reality against other players
   h. (If time allows)I want to play Pong using augmented reality against other players

4. Community Involvement

   As a member of the gaming community,

   a. I want to spectate a game that is already in progress so that I can participate in the experience
   b. I want to be able to create a tournament so that I can encourage more collaboration than normal
   c. I want to be able to join a community for the game so that I can feel included
   d. I want to be able to play in a tournament so that I can compete in larger and rarer events than normal
   e. I want to be able to give feedback on the game so that it will be heard and considered
   f. I want to be able to report other players if I feel they exhibit inappropriate behavior so that I can avoid interacting with them

Non-Functional Requirements

1. Performance

   As a developer,

   a. I would like Polybius to run smoothly on IOS and Android
   b. (If time allows) I would like the user interface to be fluid and pleasant to use
   c. (If time allows) I would like all web communication to be fast and seamless

2. Backend

   As a developer,

   a. I would like the server to handle multiple users playing concurrently
   b. I would like data to be easily stored and accessed
   c. I would like to search for user profiles
   d. (If time allows) I would like to see leaderboards for certain statistics or games

3. Augmented Reality / 3D Rendering

   As a developer,

   a. I would like Polybius to render 3D augmented reality games
   b. I would like Polybius to track 2D targets through the camera view
   c. I would like Polybius to support multiple AR game types
   d. (If time allows) I would like gameplay to be natural and fun to use

4. Networking

   As a developer,

   a. I would like current game locations to be displayed on a map
   b. I would like messages between players to take up little data

Design Outline

Components

1. Mobile Client (iOS/Android)

a. We will have a mobile app that allows players to login, search & create games, chat with other users, and look at player statistics.
b. The mobile client will connect to the web server and get user data via SQL queries to the SQL database.
c. The UI will be designed for easy and smooth usability. There will be separate menus for game creation, game searching, chatting, settings, and user profiles.
d. Users will be able to find games within a specified radius; this is to prevent searching for games that are too far away.

2. Web Server

a. The web server will be built to handle mobile client networking. It will contain the SQLite database for user profiles, authentication, and statistics.
b. Users will create an account, and that profile will be stored in the database.
c. User chats will be stored on the web server. When users send each other messages each message will be stored on the database. After users read messages sent to them, the messages will be deleted from the database.
d. Each user profile will contain statistics about each game that they've played, a hidden user ID for use with the minigames, an image-target for each minigame with their specific user-ID QR code attached to it, and a username and password for authentication as well as an email for user registration.
e. The web server will be built with Smartfox with extension for a database
f. User registration will automatically generate a QR code for the user

3. Minigames

a. Each minigame will be attached to an image target for Vuforia augmented reality tracking.
b. Each image target will contain a QR code for a specific user-ID, as well as a specific image for the specific minigame. The user-ID ensures that

players will only be able to play the game in the specific lobby that they join. Players won't be able to track a different image target other than the correct target with the correct ID.

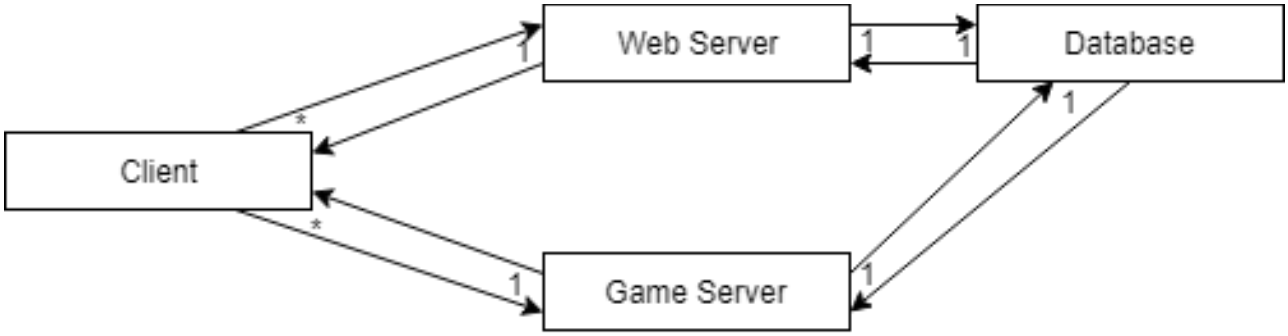
c. Minigame rooms and lobbies will be created through the game server. The mobile client will send requests to the SmartFox game server.
d. Players will be able to find games to join through the mobile client, and will connect to lobbies through the game server.

4. Game Server

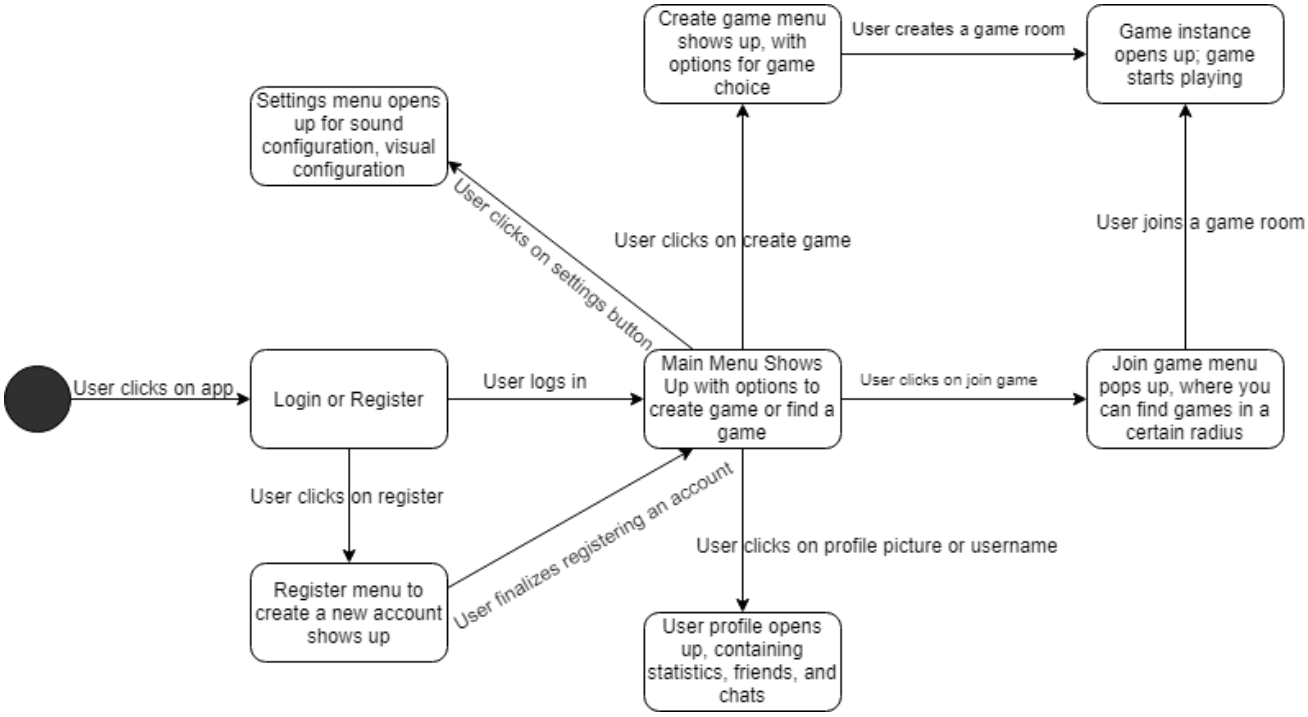
a. We will use SmartFox to create our game server, which can support 100 concurrent users at a time.
b. The game server will have lobbies for each specific game, and rooms for each specific game instance.
c. Players must connect to a lobby to find rooms for the game that they want to play.
d. Hosts, or players who create a new room, will have to print out their QR code ID image for the game that they want to play. The host will decide where to meet up to play the game.

High Level Overview

The mobile client will interface with the web server and game server, creating a one-to-many architecture. The web server receives data into the SQLite database. It will also store user info, excluding the statistics, into the database. The game server will also interface with the database, storing and receiving player statistics only. It will also receive user-ID's for augmented reality tracking. The game server will also allow the user to play each minigame.



State Diagram



Design Issues

Functional Issues:

1. How should users login to the app?
    - Option 1: Create an Account
    - Option 2: 3rd-party Integration
    - Option 3: No login needed

    Choice: Option 1

    For Polybius, having players create their own account was what allowed the most customization. We wanted players to have total customization of their profile and easy game connection. In addition, each user gets assigned a unique QR code that allows them to host games anywhere they choose, and using a 3rd-party app wouldn't allow for easy creation.

2. Should user info be exposed, and if so what info?
    - Option 1: Don't let users choose and only expose statistics and username
    - Option 2: Let users choose what info they want to expose, except for important info such as email or password. Info will not be exposed by default, except for username
    - Option 3: Expose everything, except for key info, automatically by default unless explicitly stated by the user.

    Choice: Option 2

    Allowing users to expose their own data provides a trusting and safe environment. By also locking certain info, such as email and password, we prevent data breaches by keeping that data locked in the database. Users also have flexibility and can choose what they want to share with the community.

3. How should chat messages be saved?
    - Option 1: Temporarily, and they will be deleted after the chat message is read and the instance is closed
    - Option 2: Kept forever

    Choice: Option 1

By deleting messages once they have been read and the user has closed out of the chat, we save storage space. Since the primary goal of Polybius is not to function as a chat messaging app, we expect users to send out chats to others mostly to find a game to play together.

4. How should users find games to play?
    - Option 1: Only by searching through game lobbies
    - Option 2: Allow players to search through lobbies as well as send and accept game invites

    Choice: Option 2

    Players will have more options to find a game if they can search through lobbies and also send and receive game invites. If they add a friend through the mobile client, the player will be able to send them an invite to their current game, or receive one. This versatility means that players will be able to find more games through different means, and will have a better playing experience.

5. How far should the search radius be?
    - Option 1: User selected value
    - Option 2: Fixed value within a 1.5 mile radius

    Choice: Option 2

    With a fixed radius value of 1.5 miles or less, the players will not have to travel a significant distance to reach a game. The goal of the game is to create a community of players, thus the small distance will allow players who are close to each other to meet and play together.

Non-Functional Issues:

1. What should be used for our web server and mobile client interface?
    - Option 1: Django (Python)
    - Option 2: Flask (Python)
    - Option 3: Smartfox

    Choice: Option 3

    We chose to work with SmartFox as our interfaces between the database and the client. Since we are already using it to run our game instances if we combine it

with our database it can streamline the process of sending and receiving information. It also allows us to consolidate our servers into one powerful server.

2. What game engine should we use?
    - Option 1: Unity3D
    - Option 2: Unreal Engine
    - Option 3: GameMaker Studio
    - Option 4: CryEngine

    Choice: Option 1

    Unity3D is by far the most versatile free game engine on the market. It offers complete functional flexibility, with anything being possible. It also has an “asset store”, which users can use to buy and download assets. Vuforia support is included with Unity3D, which makes implementing augmented reality smooth and easy.

3. What language should we use?
- Option 1: JavaScript
- Option 2: Boo Script
- Option 3: C#

    Choice: Option 3

    C# is an object-oriented language similar to Java. It is the most powerful of the three, it is easy to use, and it is a language the team is comfortable with. The documentation for C# is also extensive.

4. What framework should we use for GPS and map technologies?
- Option 1: Mapbox
- Option 2: Leaflet.js
- Option 3: Custom map viewer in Unity
- Option 4: Google Maps API

    Choice: Option 1

    Mapbox is the best solution for our project since it has native Unity support, is very versatile, and is free to use. Mapbox will be integrated in Polybius, and the

maps will allow users to see where the game is located. The Mapbox SDK will be used to point players to the right location for their game.

5. Which database solution should we use?
    - Option 1: MySQL
    - Option 2: SQLite
    - Option 3: mongoDB
    - Option 4: PostgreSQL

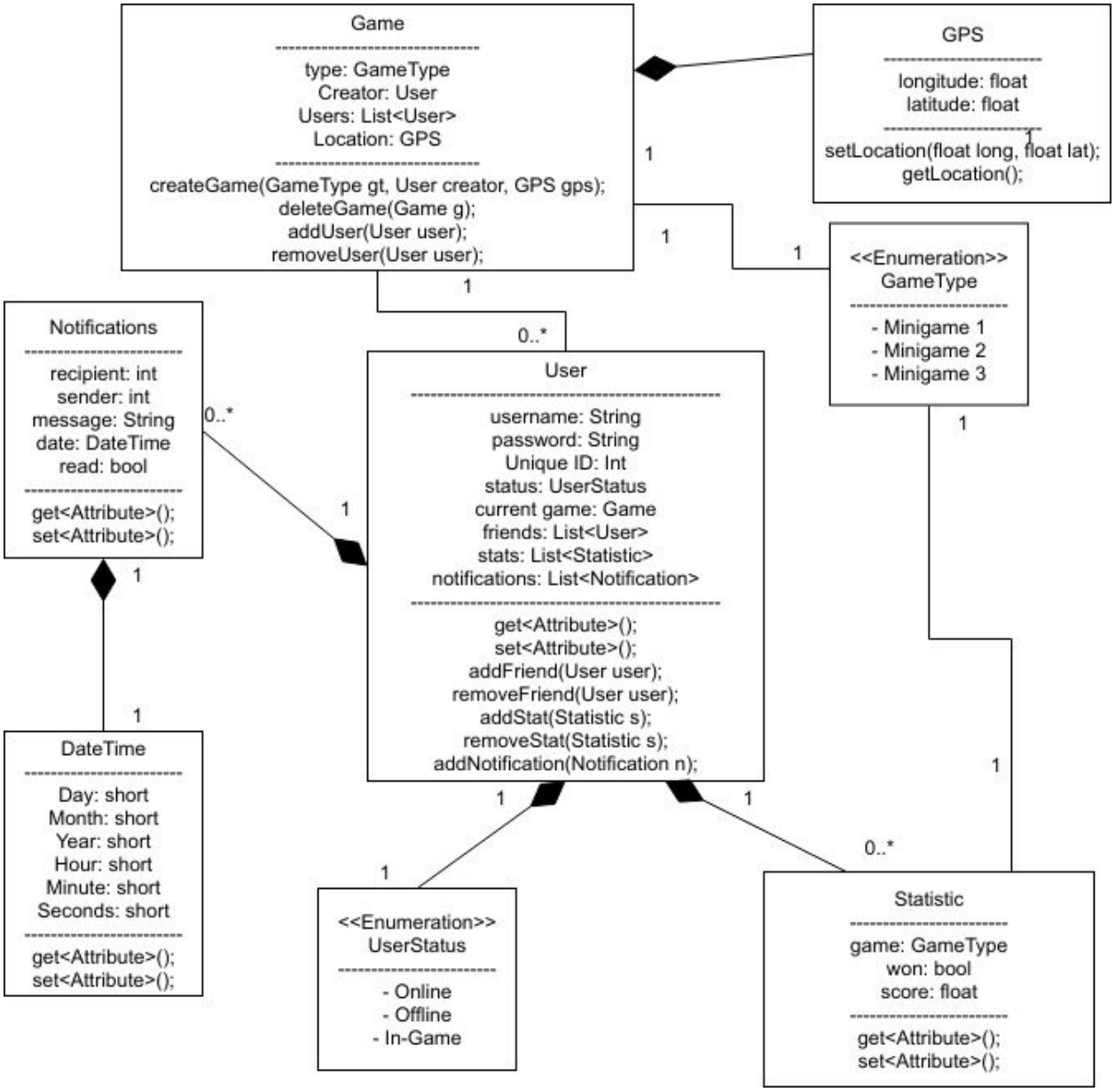
    Choice: Option 1

    MySQL is natively supported and included in most modern builds such as Smartfox, allowing for widespread support among many different distributions. In addition, MySQL's focus on ease-of-use and efficiency makes it an attractive solution for our needs.

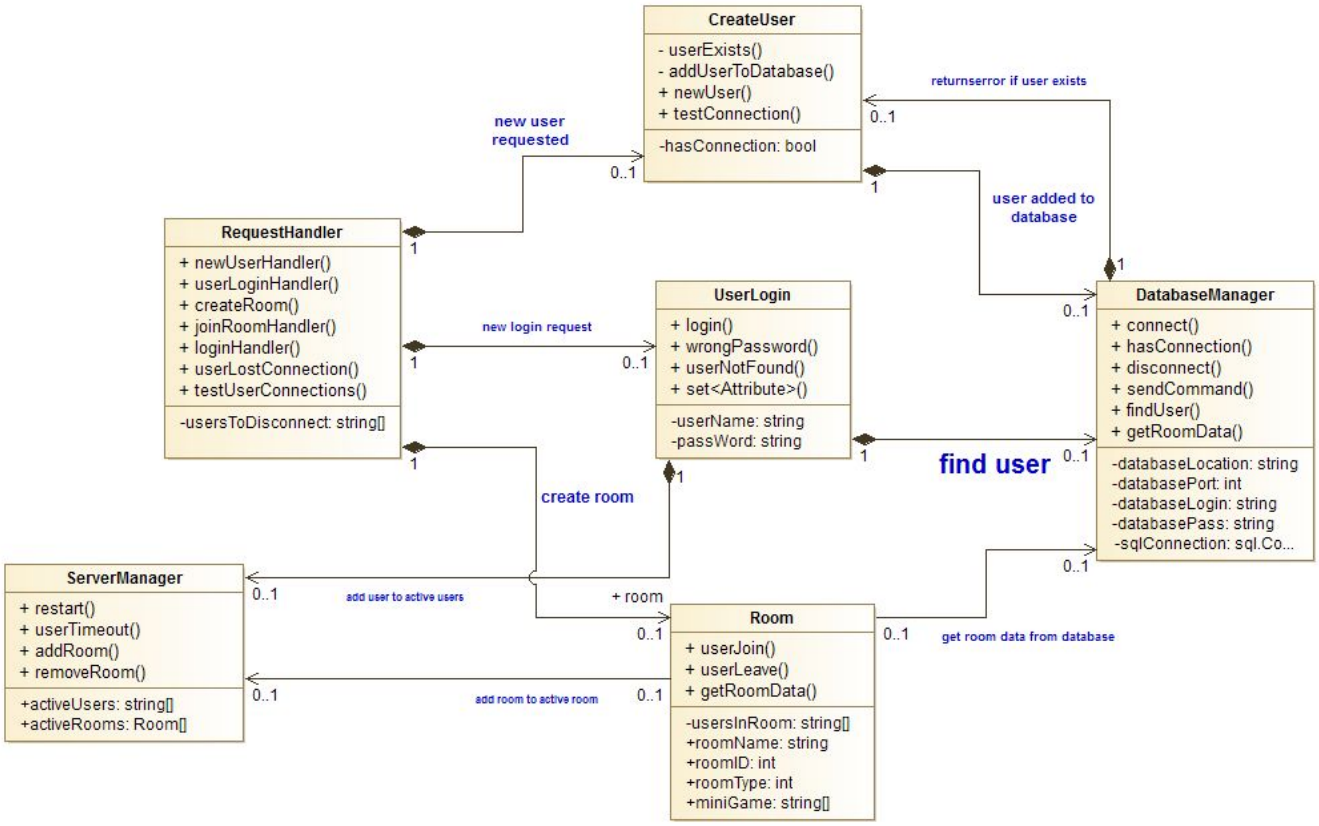
Design Details

Class Design

Client Class Diagram



Server Class Diagram:



Descriptions of Classes and Interactions

Client Classes & Interactions:

Game:

- Represents a game instance
- Lists the users in the game instance that are playing
- Stores the location of the host, aka where players need to meet up to play
- Game type selected is what game the players will be playing

GPS:

- Stores a location with a given longitude and latitude
- Locations will be where the host is; other players must meet up with the host in order to play
- The GPS location will be visible on a map in the mobile client

GameType:

- The type of minigame chosen, for example, Connect 4 or Pong
- Hosts will choose what game they want to play
- Server will create a room in the lobby for the respective game

User:

- Represents each user profile
- Contains information for each user, such as username and password
- User class is created on registration, and each user profile is stored on the database
- Friends list contains a list of other users that the current user is friends with
- Statistics are tracked by the game server and stored in the database
- Users will be able to chat with their friends

Notifications:

- Represents messages between users and global messages
- Contains information on the sender of the message
- Contains the unique user ID of the recipient
- Contains a string of the message
- Notifications are sent to the user when another user messages them

DateTime:

- Represents the time a message has been sent
- Contains information specific to the time of day
- Connects to sent messages with a sent time

UserStatus:
- Represents the online activity of a user
- Enum that holds whether or not the user is Online, Offline, or Busy
- Set by the user themself
- Status indicator made for the chat messaging system

Statistic:
- Represents an individual statistic for a specific gametype
- Each statistic will have a score, which shows the count of that statistic
- Statistics will be stored in the database corresponding to the specific user they belong to.

Server Classes & Interactions:

RequestHandler:
- Represents the inbound and outbound connections between the server and the clients
- Contains an array of client names that need to be disconnected
- Methods are for various requests to and from the server

UserLogin:
- Manages the login requests from users returning errors if the login failed
- Contains the username and an encrypted password string used to match with the database
- Login fails are sent back to the client

CreateUser:
- Manages new user requests from clients
- Checks for existing user and returns error
- Adds the new user to the database

ServerManager:
- Manages all the backend server functions
- Contains arrays of active rooms and users logged in

- Checks if users have timed out

Room:

- Represents the game room that is created by the user
- Manages what users are in the room and what minigame is loaded in
- Starts minigames

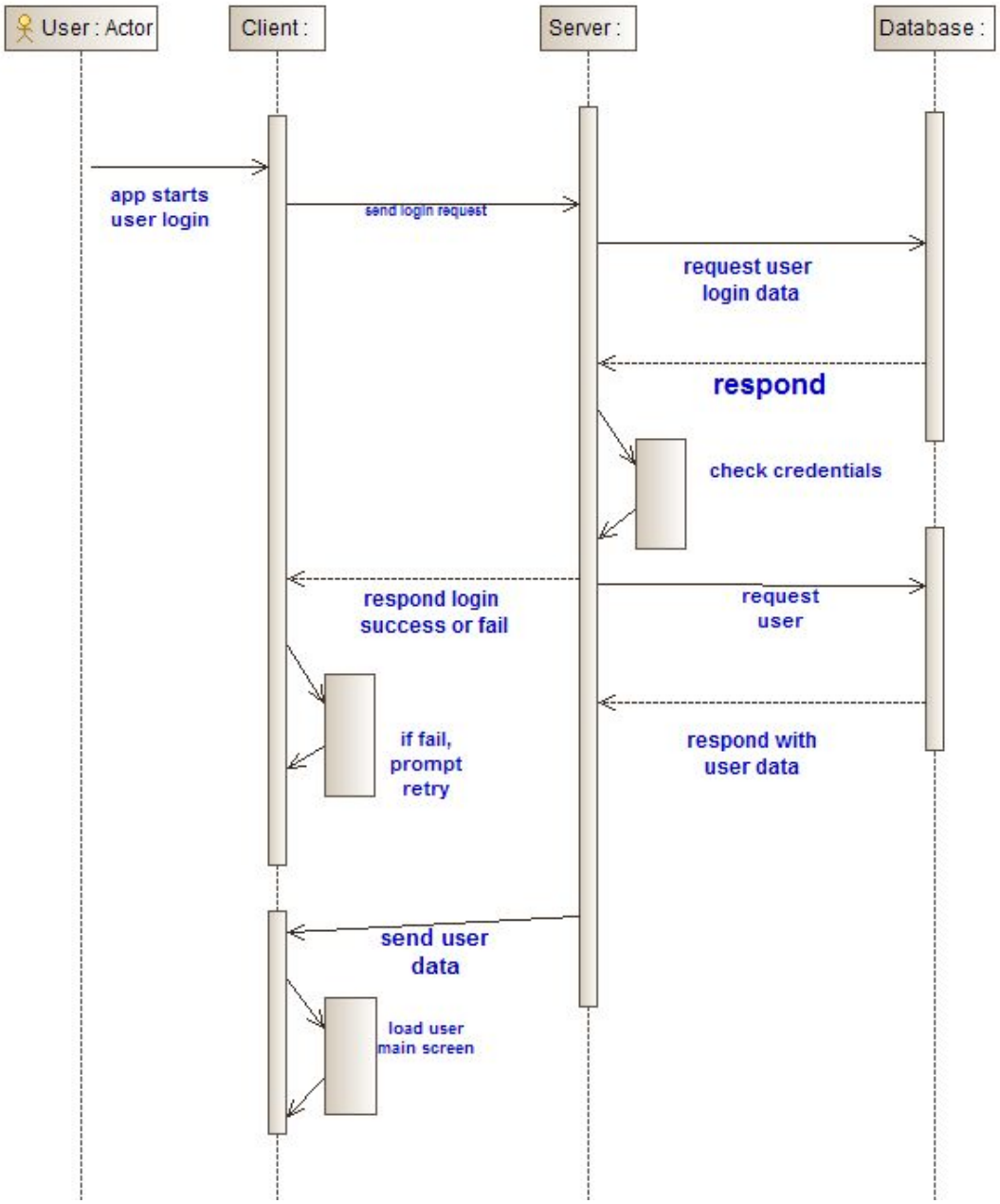
DatabaseManager:

- Represents the connections between the server and the database
- Contains methods to communicate with the database
- Contains information needed to connect to the database

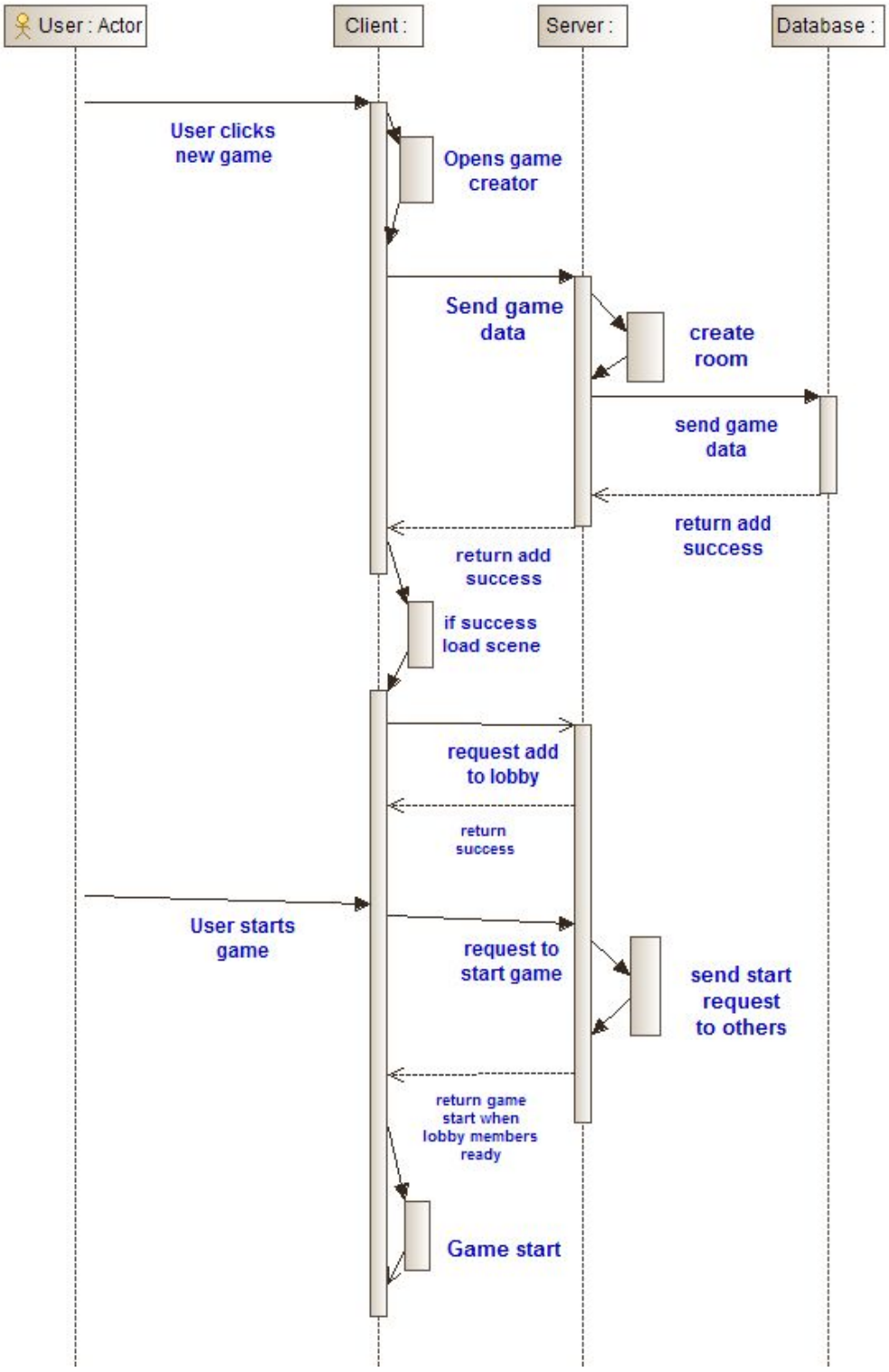
Sequence Diagrams

These sequence diagrams show how the client will interact with the server, which then interacts with the database, illustrating the flow of data between the main components. Depending on the response from the server, the client will update the UI with the appropriate information accordingly.

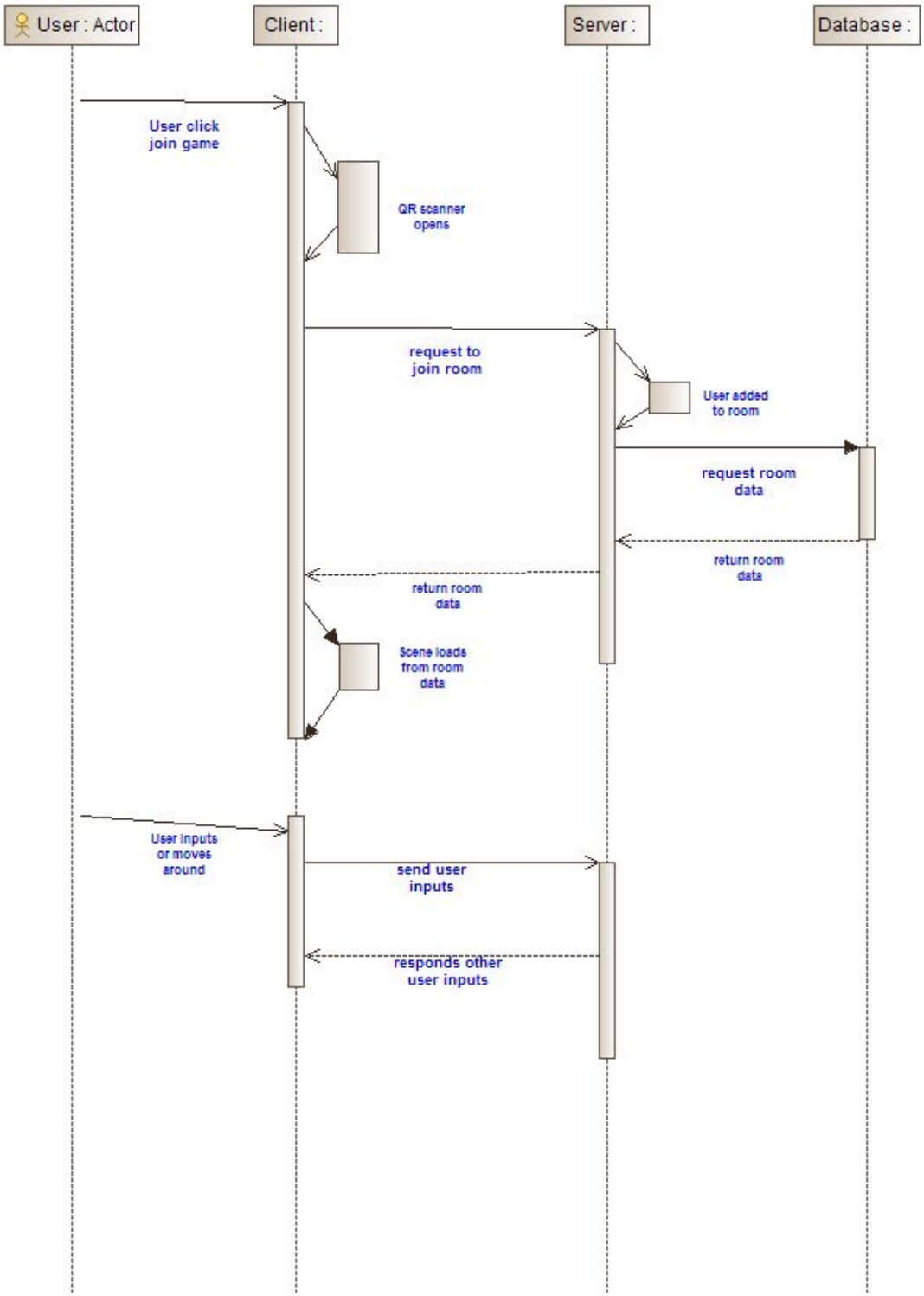
1. User logs into the app



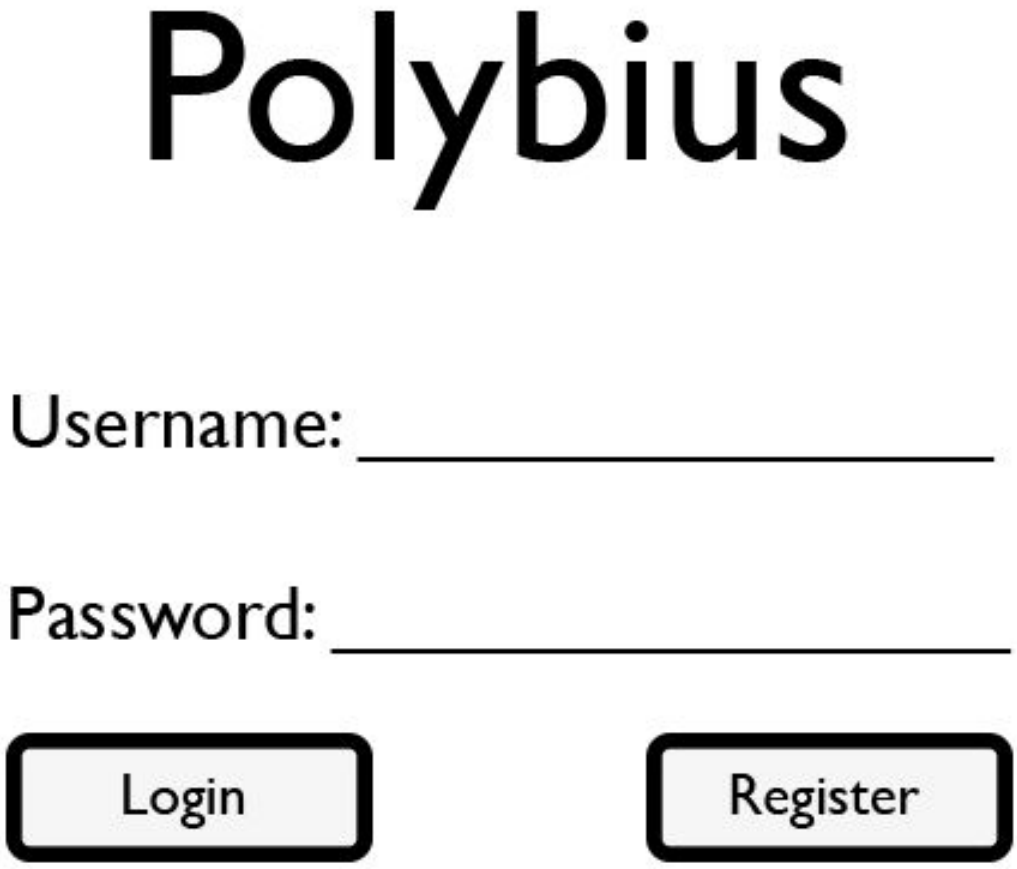
2. Player hosts a minigame



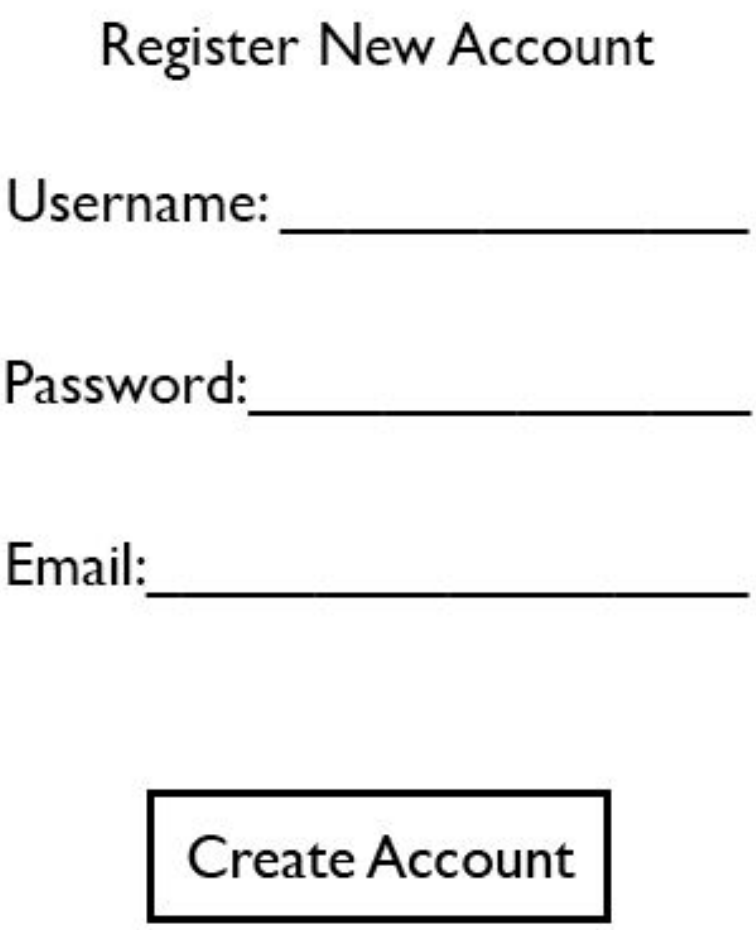
3. Player joins a minigame



UI Mockups

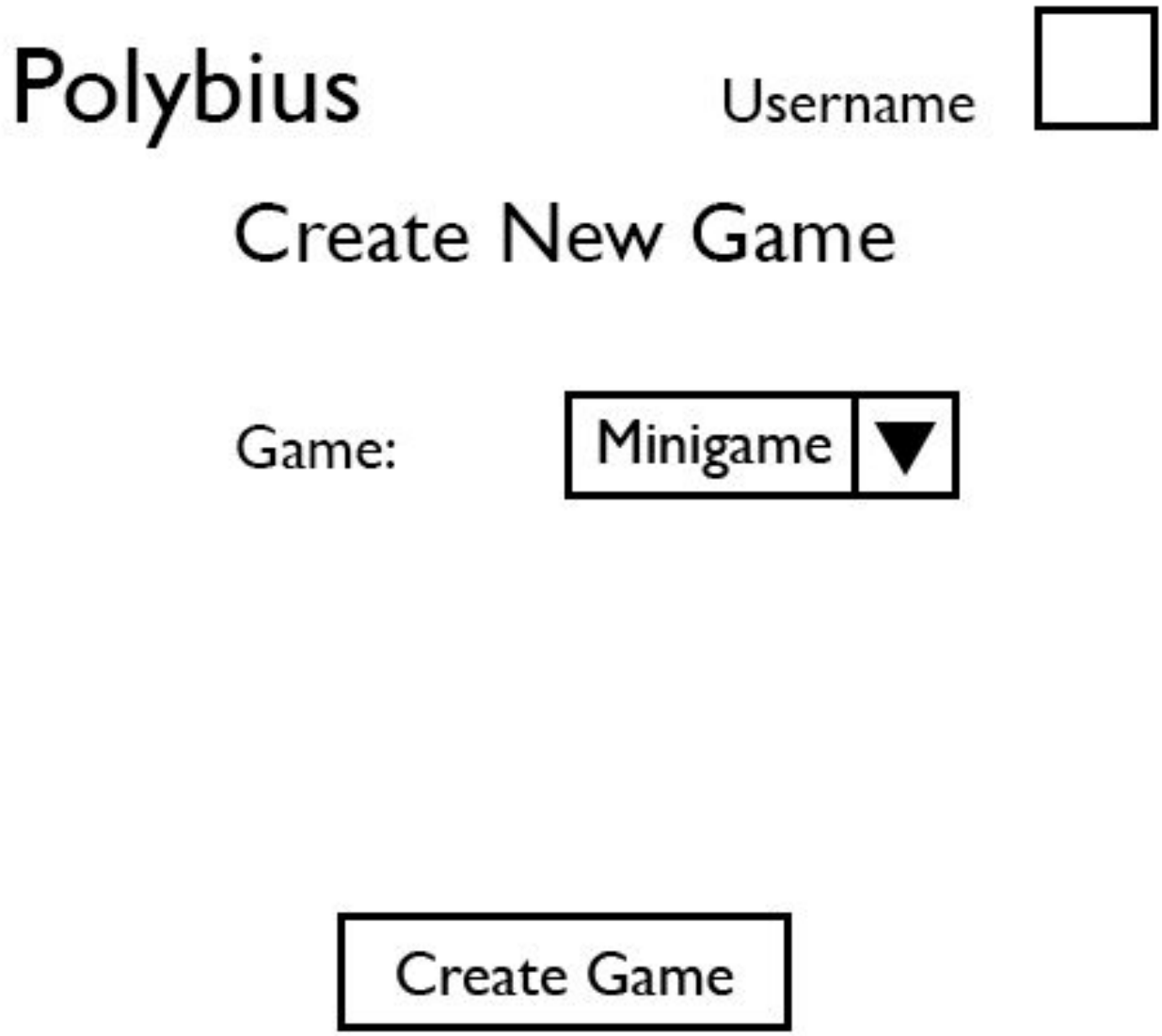


This is the page that the user sees when opening the app for the first time. After logging in for the first time they will be logged in automatically every time they reopen the app. They can choose to register an account on this menu.

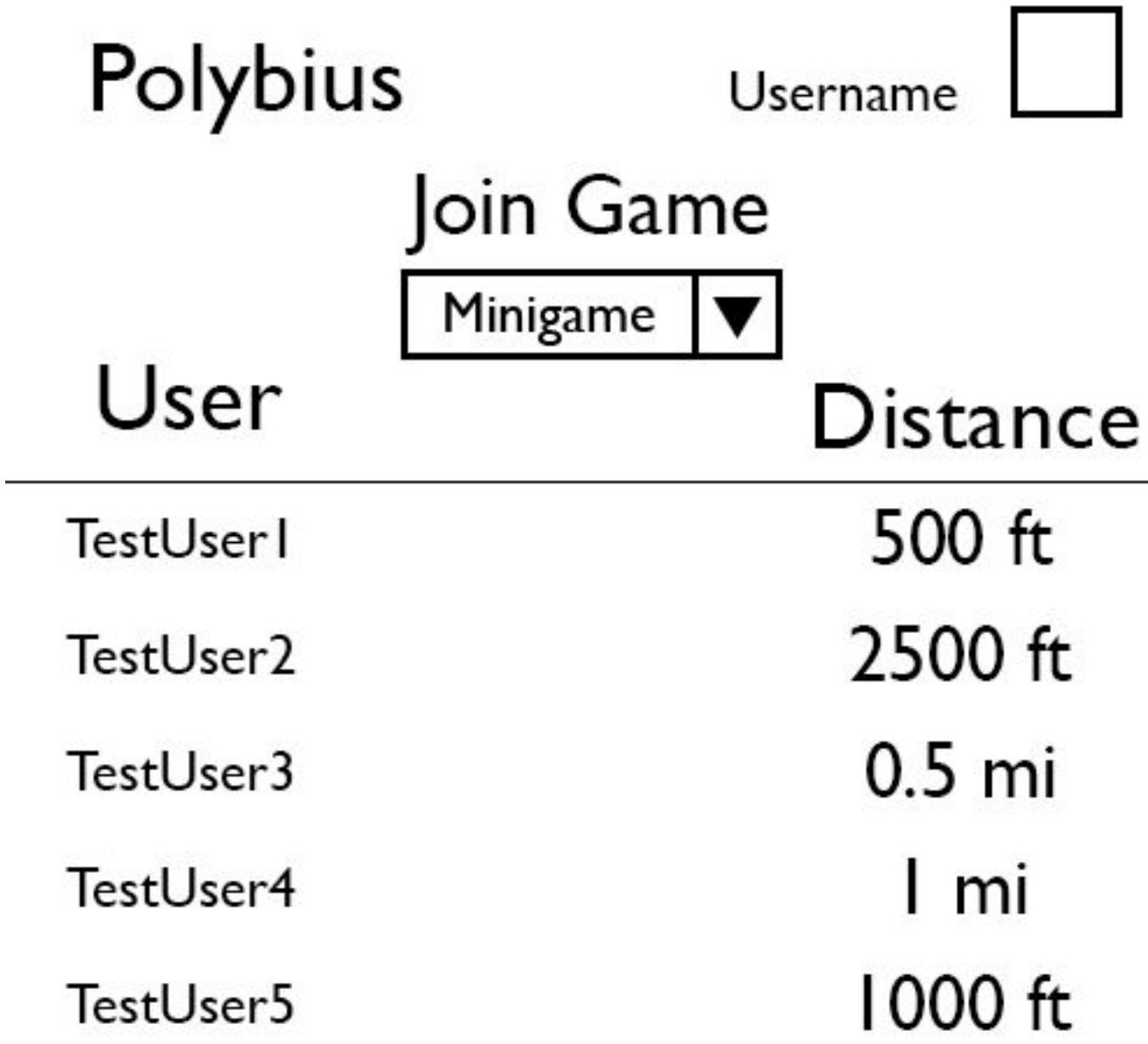


This is the register new account menu. A user will need to input a username, a password, and a valid email. They will receive a verification email in order to verify their account, which will finalize their account. Their account information will be stored on the database.

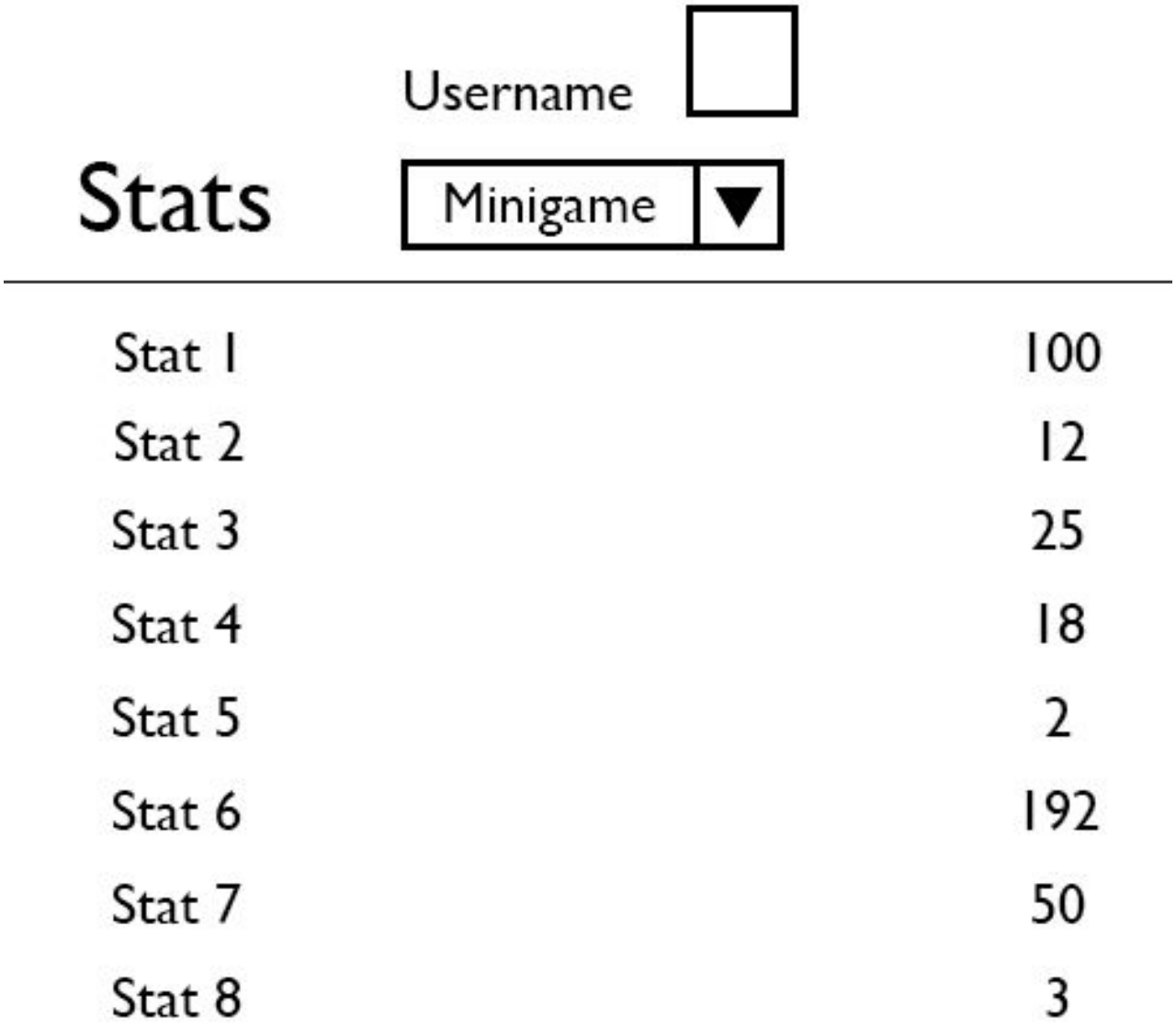
This is the screen that players will see after logging in. The header contains the game's logo, as well as the player's username and profile picture (the box). They will be able to create a game, join a game, or tweak the settings.



This is the create new game menu. Players will be able to select which game they want to play with a dropdown selection, and then by clicking on the “Create Game” button the game server will drop the player into a new game room and wait for other players to join.



This is the menu for when the player wants to join a game. When a player is joining a game, there will be a list of different games that other players are hosting. The distance is shown as well, so players have an indication of where the game is located relative to their position. After selecting a game, a map will pop up and show the player where exactly the game host is so they can travel to that location.



This is the menu that will pop up when a player clicks on their username or profile picture. It will bring them to this menu that shows their respective statistics for each game. Players will be able to choose which game to show statistics for by selecting the game from the dropdown.