

DuelGames



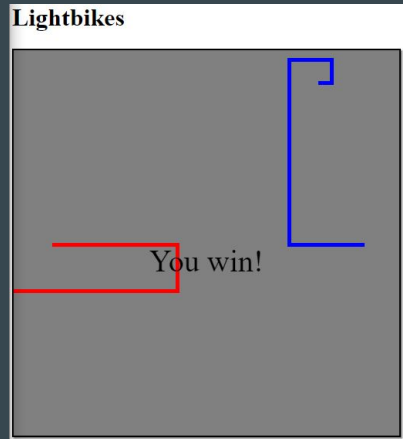
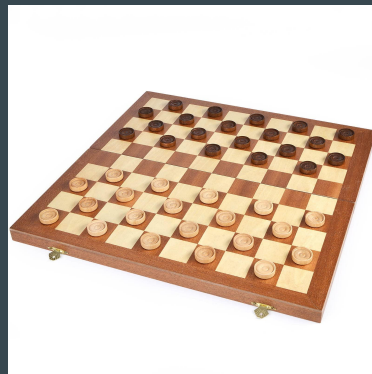
Mason Vick

Introduction

...

Overview

- Website for users to play games against each other
 - Online through casual or ranked
 - With friends through code generation
- Account Creation
 - Create an account to keep track of your stats
- Leaderboard System
 - See your stats against other players



Background

Good Experience

SSMS/mssql

javascript

Low Experience

NodeJS

Websockets

HTML

No Experience

Socket.IO

Related Work



Devs can upload games and earn from ads
Estimated \$5 million annual revenue



800k+ visits Nov 2024



4.5M+ visits Nov 2024

similarweb.com & growjo.com

Requirements

...

Use Cases

Account Creation

- User login/signup
- Google login
- Save account data

Play Games

- Lightbikes
- Tic Tac Toe
- Ranked/Casual
- Play via Code

View Leaderboards

- See your stats
- See other players' stats

Phases

MVP:

- Working Game
- Account Creation
- Collect Game Data

V1.0:

- Second Game
- Leaderboard
- Google Accounts

V2.0:

- Better Leaderboard
- Ranked Play
- Play Via Code

Tic Tac Toe

Create New Game

Enter Game Code

Join Game

Quick Match

Ranked Match

Breaks when clicked (old reset button)

Login

Username:

Email:

Password:

Register

Login with Google

Design/Implementation

...

Backend built on Node.js

- Starts HTTP server & attaches express app
- Hosts Socket.IO server
- Mssql connection to database
- Listens for client connections

Express App:

Handles HTTP Requests

Serves static HTML files

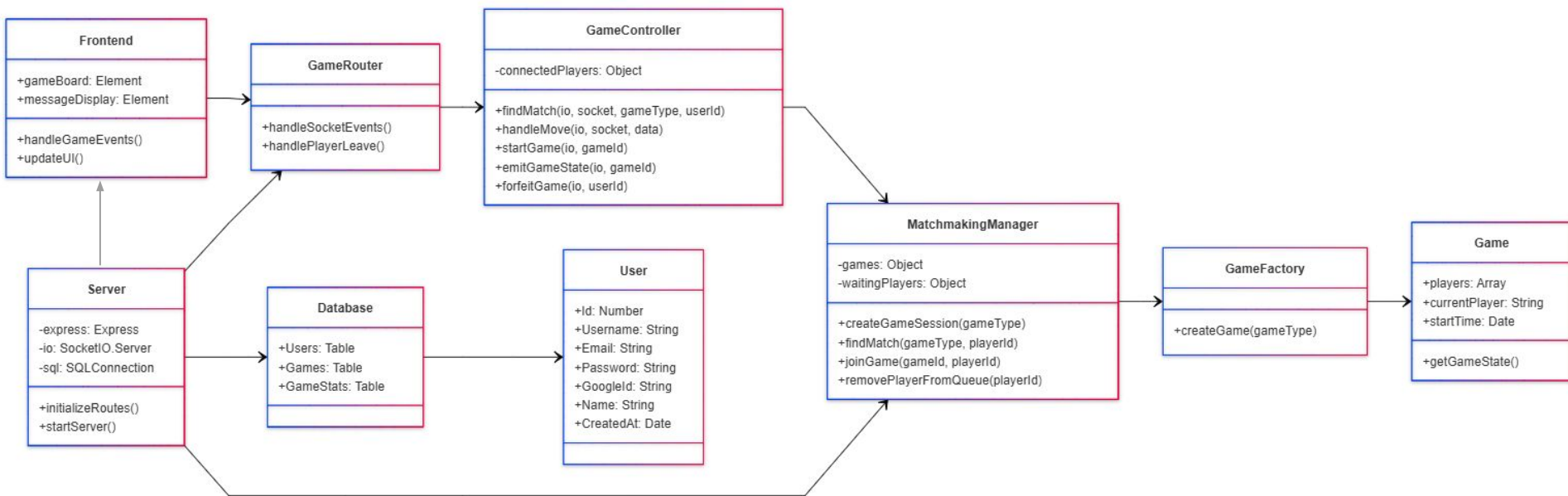
Initialize game routes



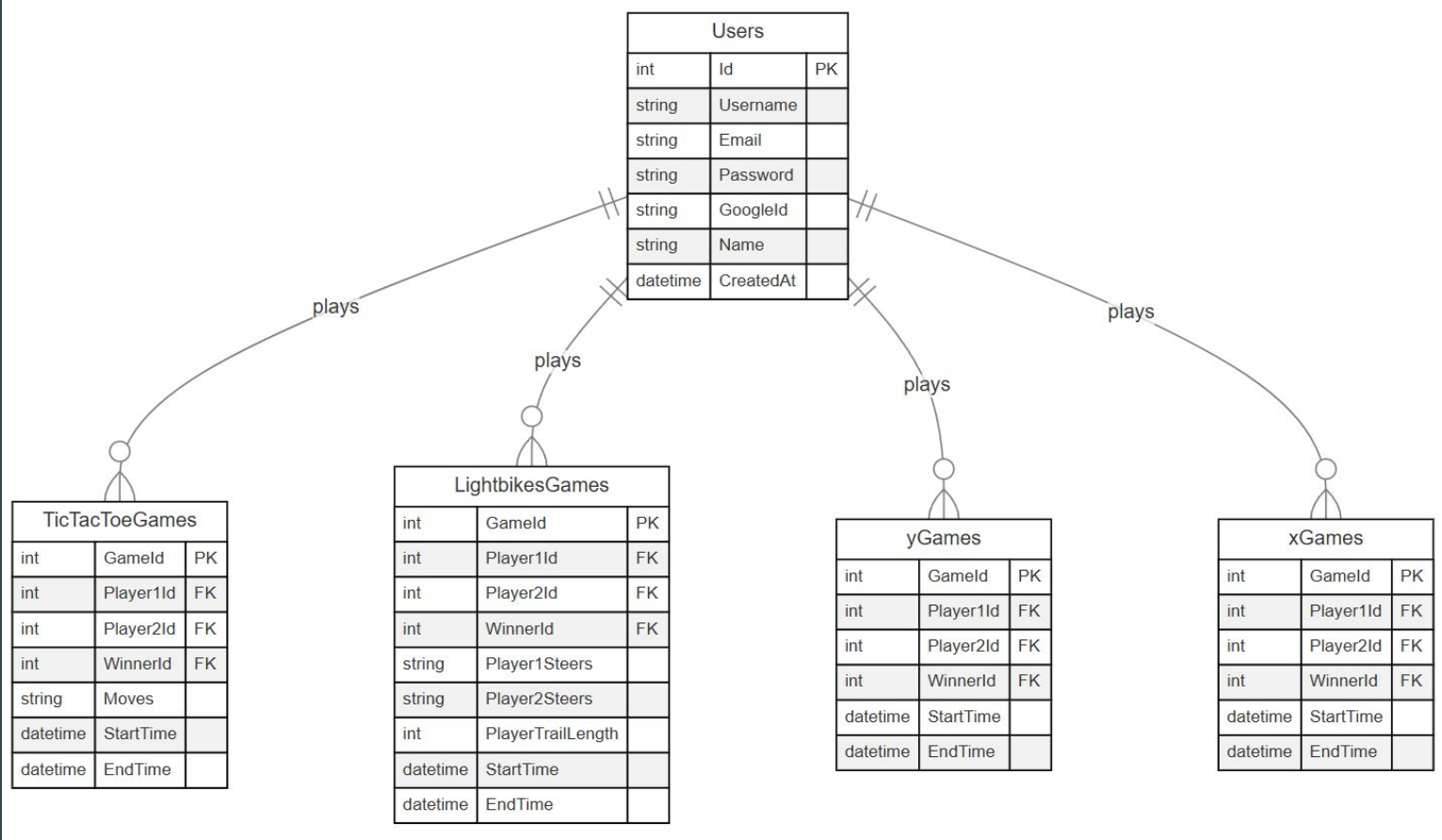
Languages/Frameworks/Tools

- Languages: Javascript, HTML, CSS, SQL
- Frameworks/libraries: Express.js, passport.js (MW), bcrypt
- Backend Tools: socket namespaces/sessions/rooms, Google OAuth 2.0 Protocol

UML Diagram



DB Diagram



Login & Register

LoginRegister

Login

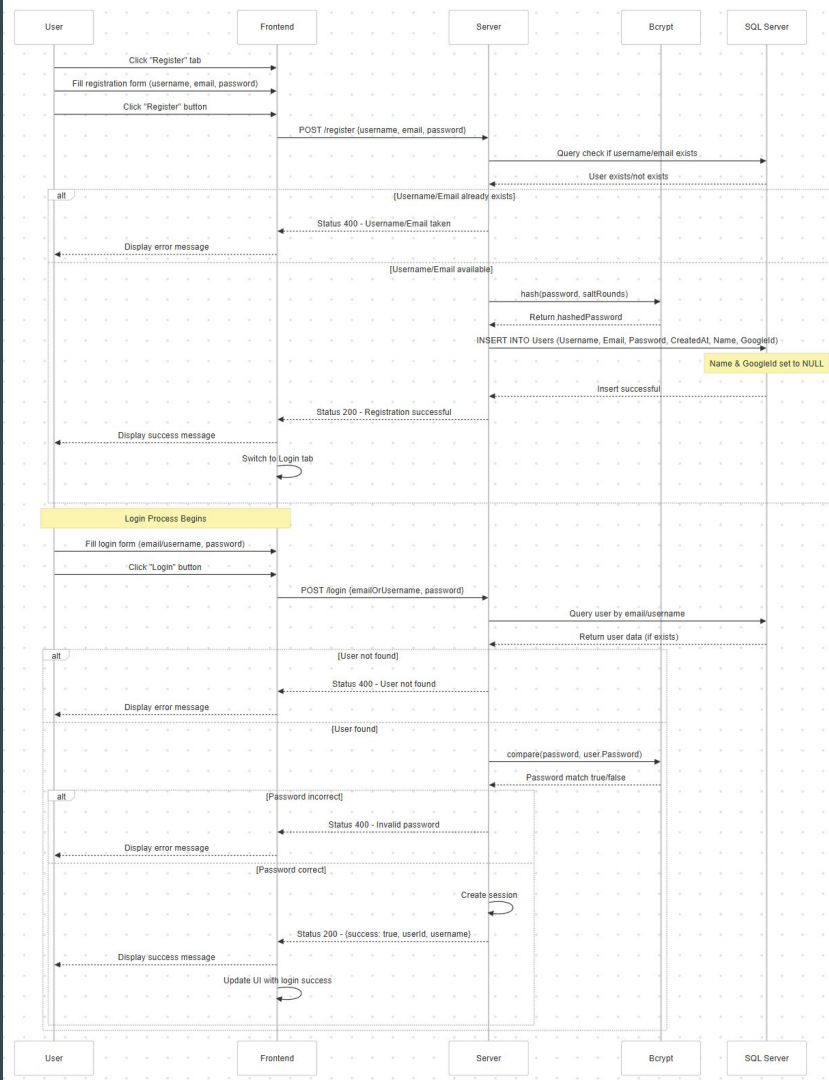
Email or Username:

Password:

Login

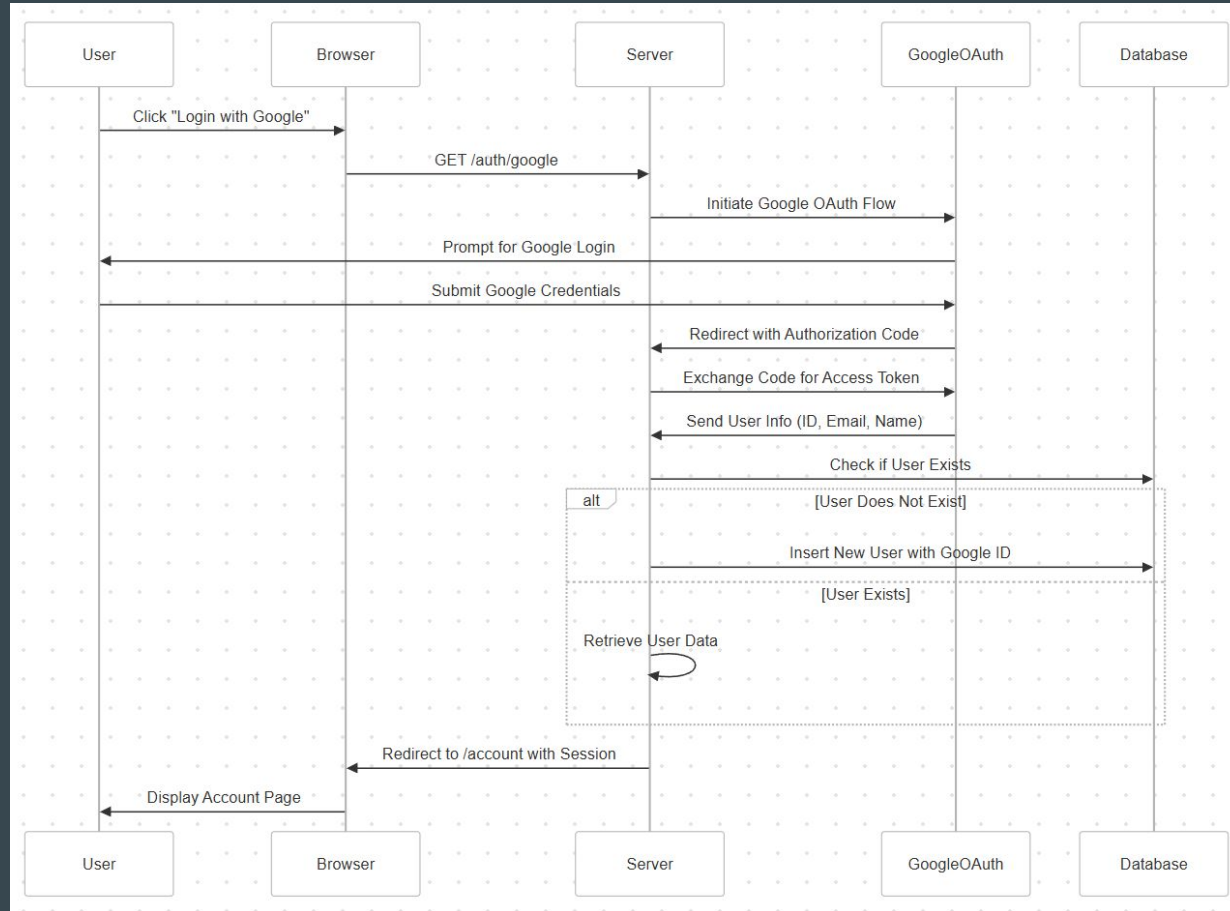
Login with Google

- Accounts via traditional & Google
- userId to track users across sessions
- Hashing via bcrypt

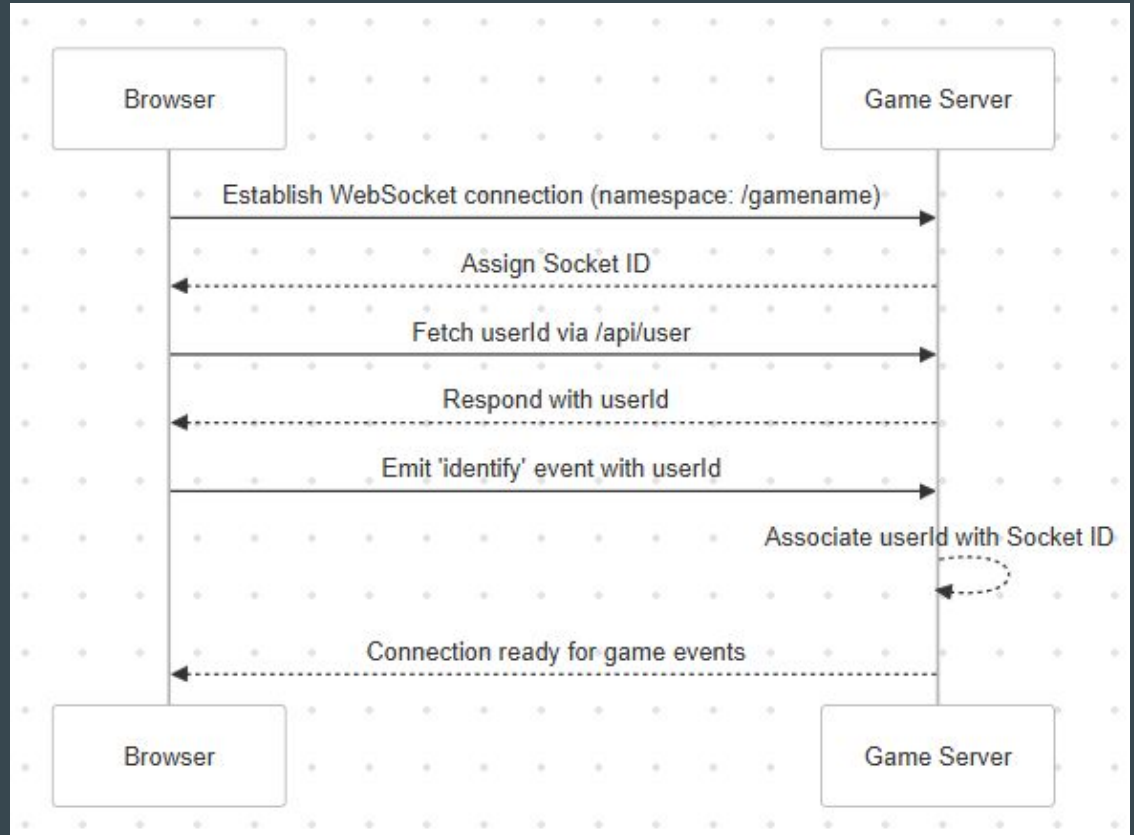
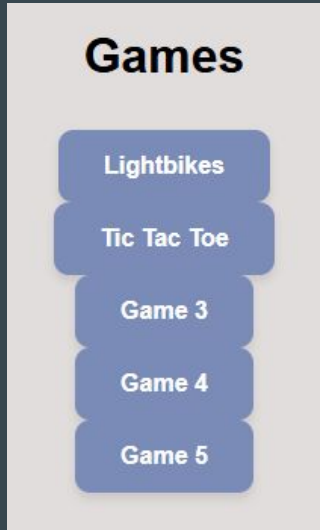


Google OAuth2.0 & passport.js

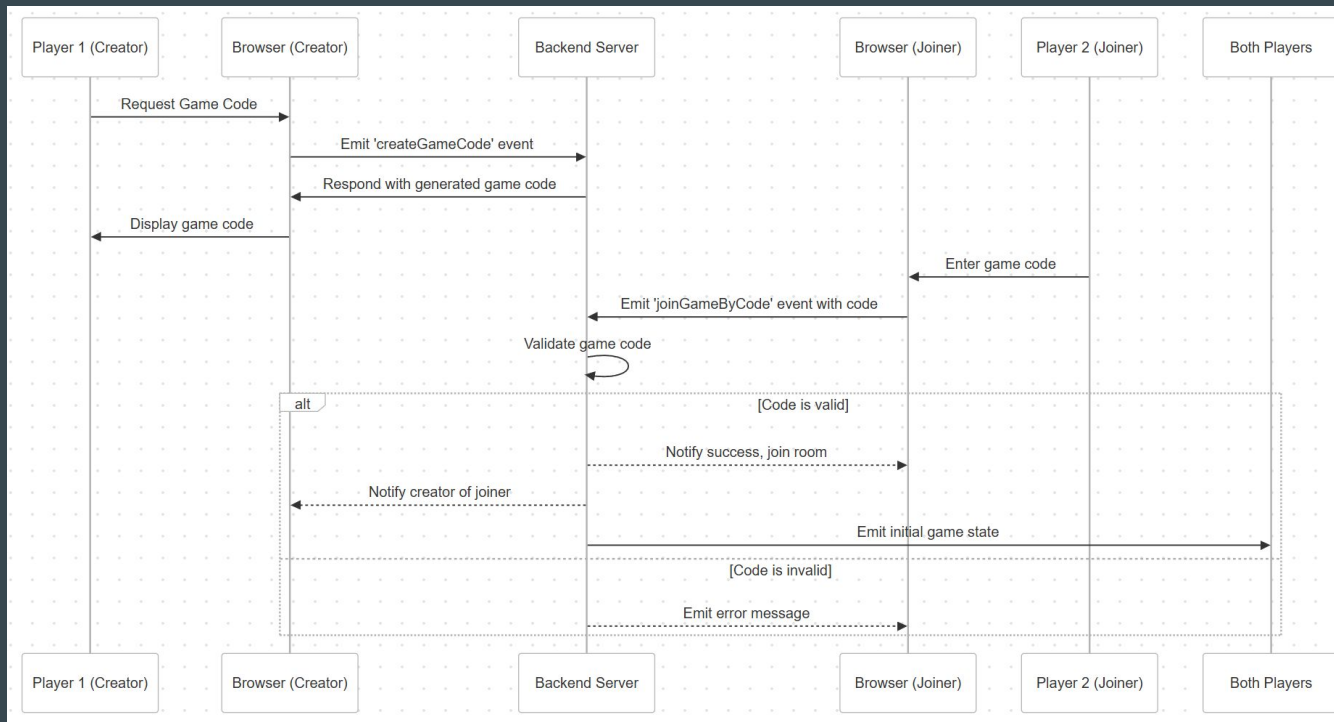
- GET redirects to OAuth page
- User logs in via Google
- OAuth sends authorization code
- Server handles data
- Add to DB/sign in



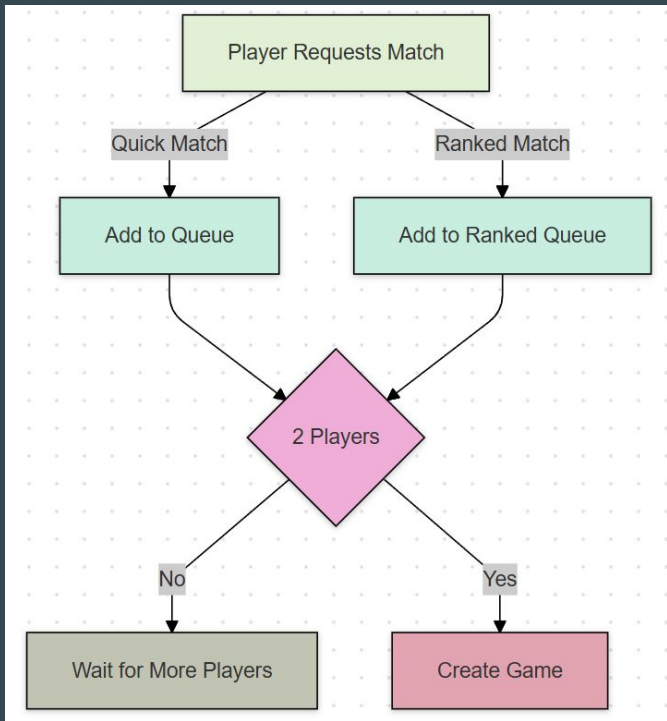
Entering a WebSocket Namespace (Game Page)



Join Game By Code



Quick Match/Ranked Match



```
document.getElementById('quickMatchButton').addEventListener('click', () => {
  console.log("Requesting quick match with userId:", userId);
  if (userId) {
    document.getElementById('searchingPopup').style.display = 'block';
    socket.emit('quickMatch', { gameType: 'lightbikes', userId });
  } else {
    alert('User ID not found, please login.');
```

game.html

quick/ranked & light bikes/tic tac toe

```
socket.on('rankedMatch', async ({ gameType, userId }) => {
  console.log(`Ranked match requested by ${userId} for ${gameType}`);
  const skillRating = await getPlayerSkillRating(userId, gameType);
  lightbikesController.findRankedMatch(io, socket, gameType, userId, skillRating);
});
```

game.js (router)

Matchmaking Manager

- Keeps track of queues and active games
- Used by a controller
- Uses Game Factory

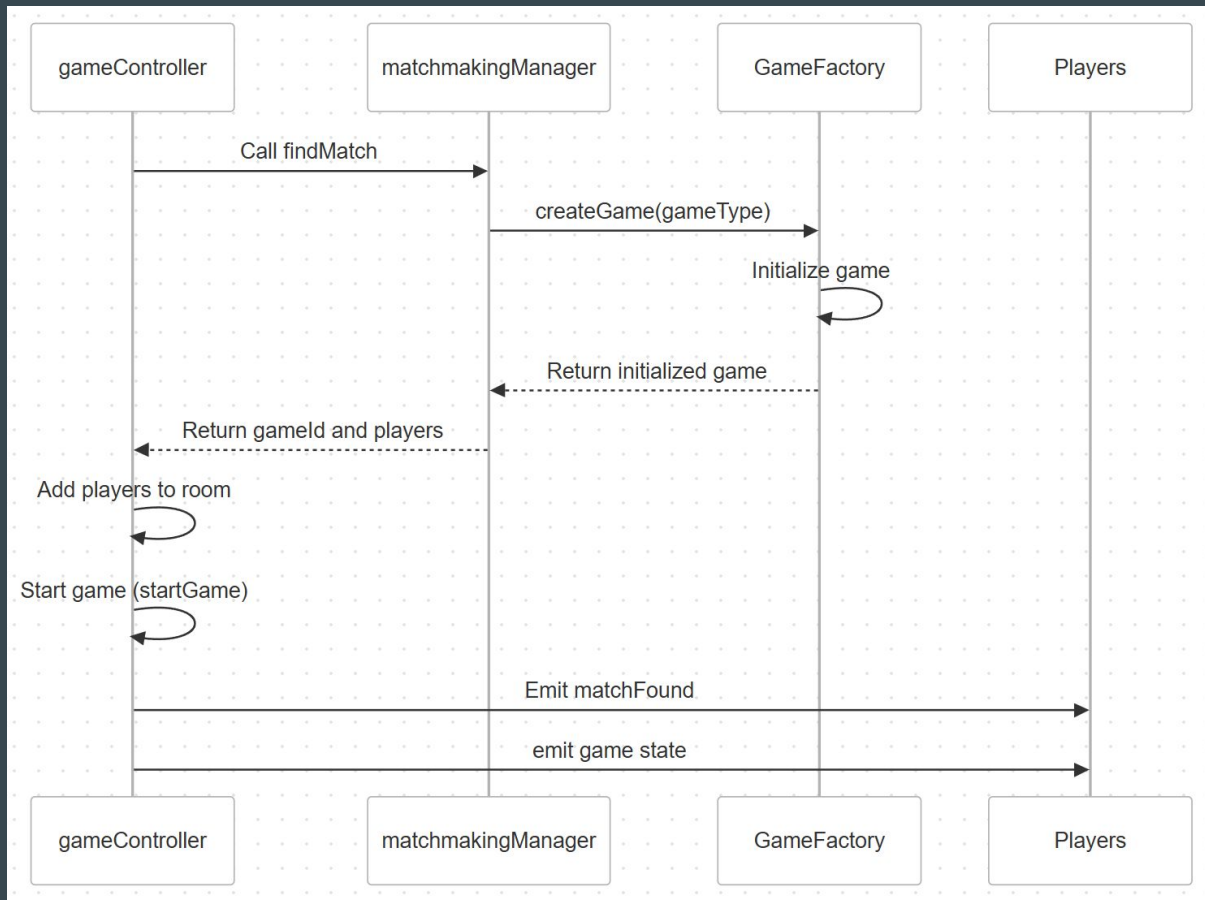
```
module.exports = {  
  createGameSession,  
  findLightbikesMatch,  
  joinLightbikesGame,  
  findTicTacToeMatch,  
  joinTicTacToeGame,  
  findRankedMatch,  
  games  
};
```

GameFactory

- Holds Classes for each game
- Initializes Games

```
class TicTacToeGame {  
  constructor() {  
    this.board = Array(9).fill(null); // 3x3 board  
    this.currentPlayer = 'X';          // 'X' goes first  
    this.players = [];                 // player IDs  
    this.playerSymbols = {};           // Maps players to X or O  
    this.winner = null;                 // 'X' or 'O'  
    this.startTime = new Date();        // time when game starts  
  }  
}
```

Playing a Game



Turn Based Game Flow

Tic Tac Toe

Create New Game

Enter Game Code

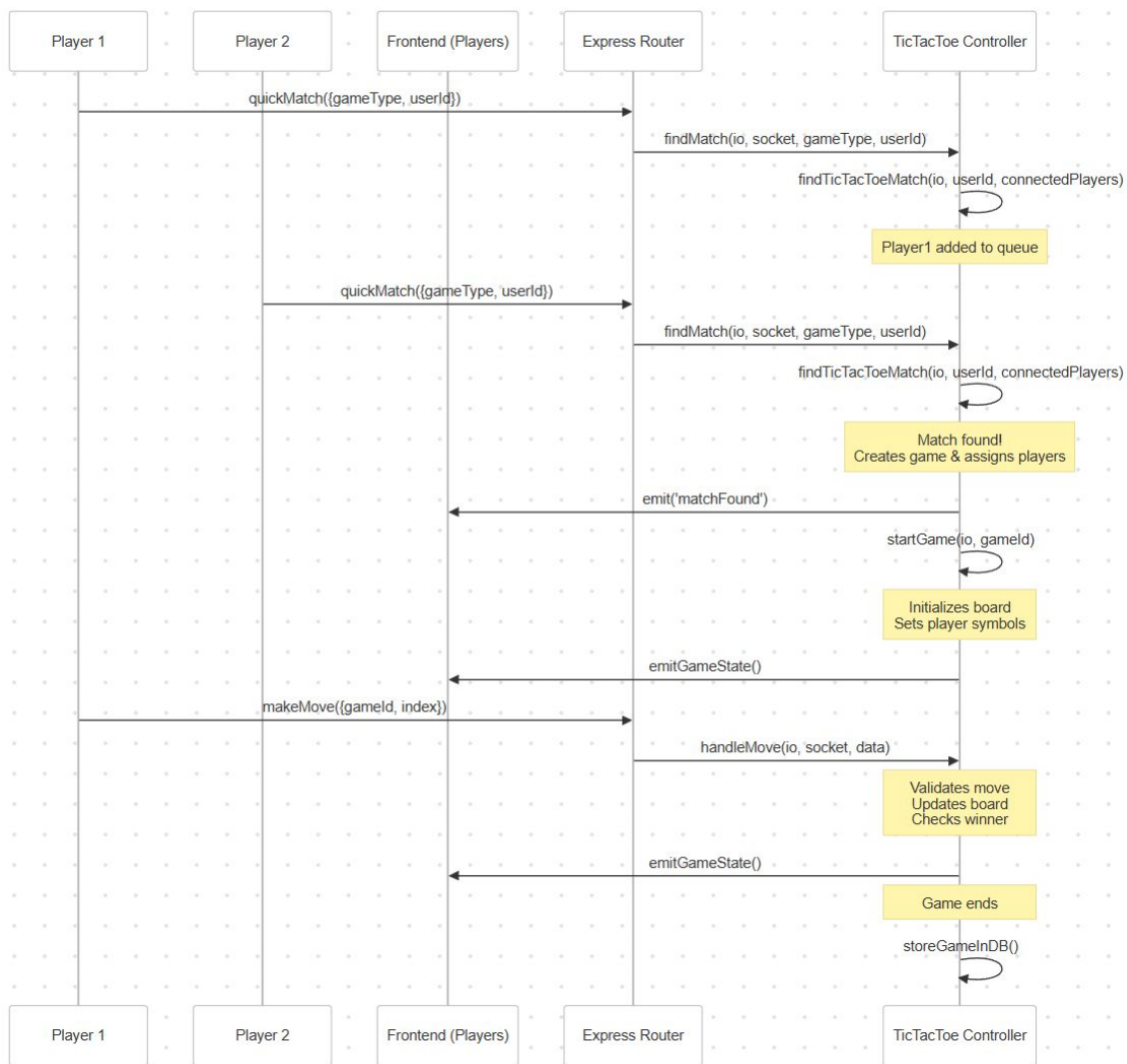
Join Game

Quick Match

Ranked Match

O	X	O
	X	
	O	X

Opponent's turn (X)



Real-Time Game Flow

Lightbikes

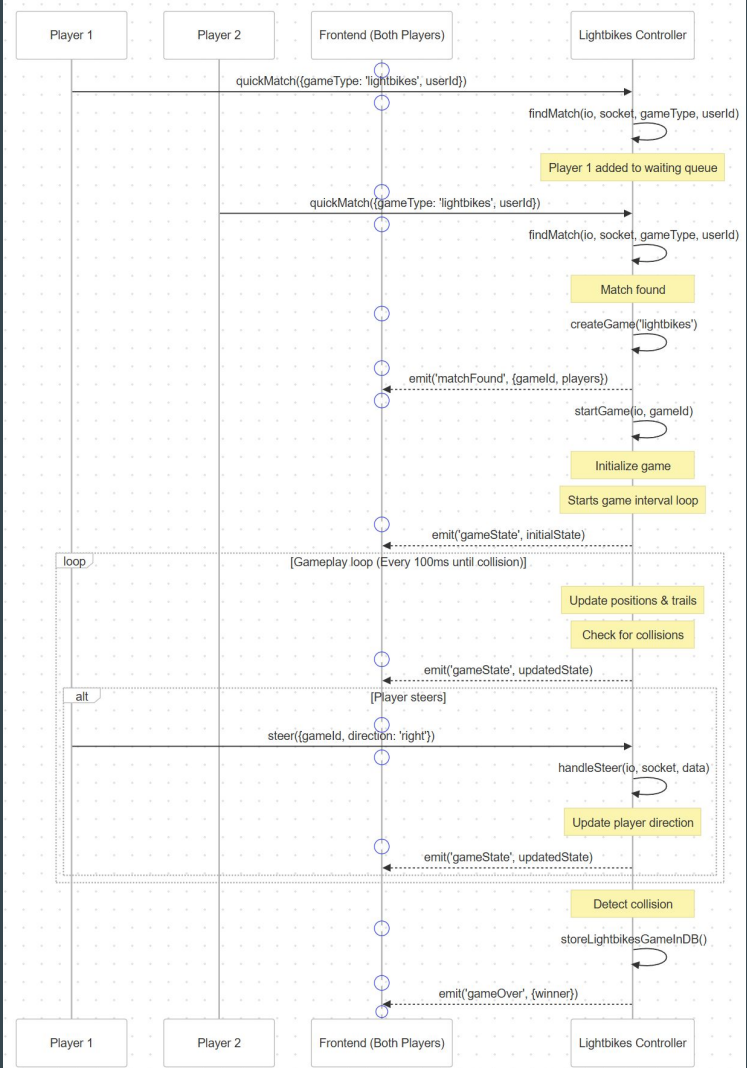
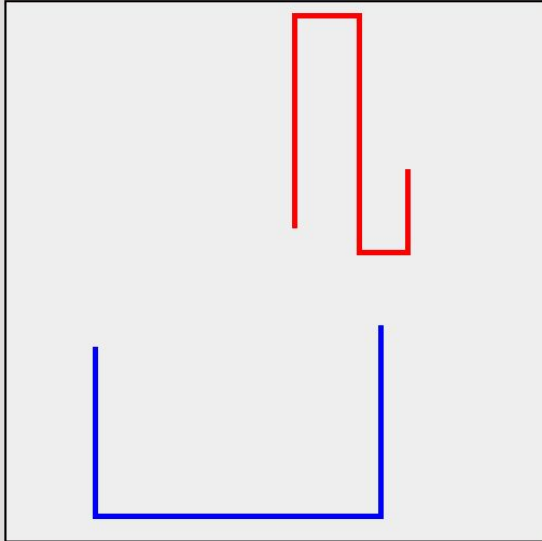
Create New Game

Enter Game Code

Join Game

Quick Match

Ranked Match



Individual Game Data Structures

```
▼ Object i  
    gameOver: false  
    ▶ grid: (10000) [null, null, null, null, null, null, null, null, null, null, nu...]  
▼ playerPositions:  
    ▼ 15:  
        color: "blue"  
        direction: "up"  
        ▶ steers: (5) ['up', 'left', 'up', 'right', 'up']  
        ▶ trail: (130) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]  
            traillength: 129  
            x: 13  
            y: 40  
        ▶ [[Prototype]]: Object  
        ▶ 16: {x: 35, y: 61, direction: 'up', trail: Array(130), traillength: 129, ...}  
        ▶ [[Prototype]]: Object  
▼ players: Array(2)  
    0: 16  
    1: 15  
    length: 2  
    ▶ [[Prototype]]: Array(0)  
winner: null
```

```
▼ {board: Array(9), currentPlayer: '0', winner: null, players: Array(2),
  ▶ board: (9) ['X', '0', '0', null, 'X', 'X', '0', 'X', null]
    currentPlayer: "0"
  ▼ playerSymbols:
    15: "X"
    16: "0"
    ▶ [[Prototype]]: Object
  ▶ players: (2) [15, 16]
    winner: null
```

Demo

Live demo :)

Testing

Account Creation

Form Validation

Database Queries

Google OAuth

Play Games

Game Logic

UI interaction

Socket Communication

View Leaderboards

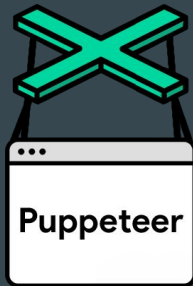
Querying Data Accurately

Calculations with Data



Mocha/Chai for unit tests

Puppeteer for UI tests



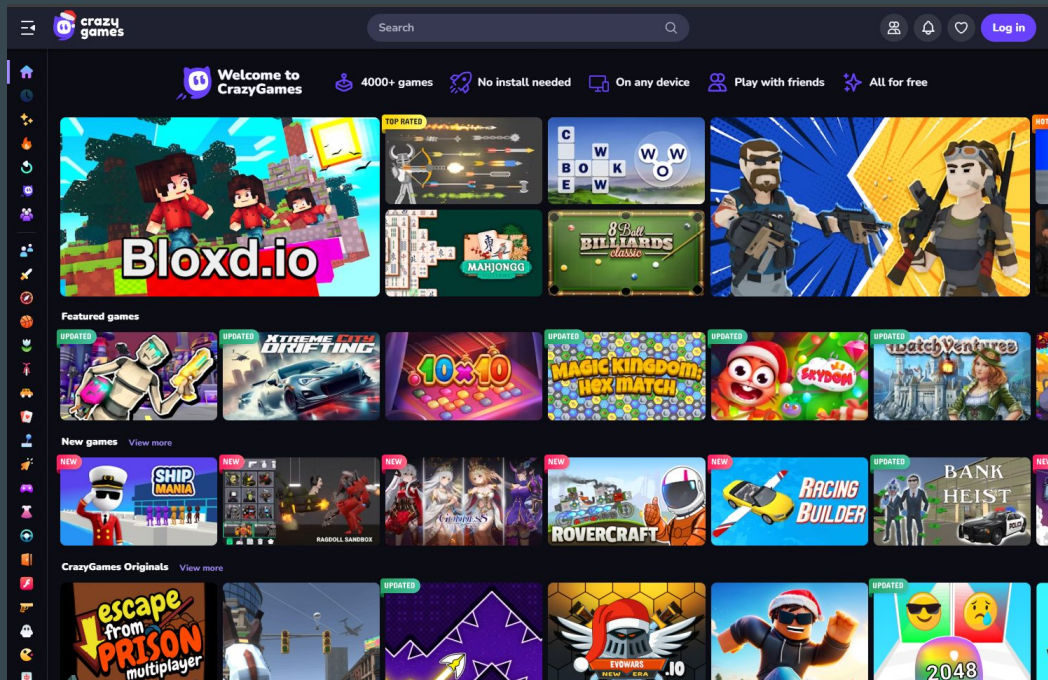
Sinon for mocking dependencies

Deployment

- Optimize code (queries, structure, security)
- More error handling (edge cases, cheaters, bots)
- Backend: PM2 - Process manager for node.js apps hosted on a droplet
- Frontend: Hosted separately or on the same droplet
- CI/CD Pipeline to automate builds/deployments
- Add HTTPS

Future Work

- More Games
- Refined Ranking System
 - Actual ranking score/tiers
 - Separate ranked vs. casual game data
- Friends list (leaderboard too)
- Better UI
- Convert html scripts to js files
- More Testing
- MMM function simplification
- Controllers only talk to routers



Thank you!