Aiden Inglis

NMIT  322 Hardy St Nelson 7010

DAT602 – Assignment 1

# Milestone 1

## Game Description

### Base Idea

My idea for a multi-player game is an adventure, turn based game where the players will move around the map and fight a monster to win the game while picking up equipment and items along the way to make them stronger. The game will be max 15 minutes and the players that survive(if they survive) by the end of the game after the monster dies wins (+1 point). If the players die, they lose 1 point from their score.

### Logging In

When the game is first started the player will be faced with a log in screen where the player will have to enter their username and password to enter into the lobby which will be checked against the database. If they don’t have an account they can click the “don’t have an account” link below and then make an account to log them in with(which the account will be created then checked against the database that it exists. If the user fails their password 5 times in a row then the player will be locked out and will need to contact an administrator to have them unlock it. When registering a new account the user will be required to make a username, password, and enter their email.

### Game Lobby

Once players have registered and logged into their account, they will be directed to the lobby. In the lobby the user will be able to see the online players and active games listed. In the lobby users will be able to create games, log out, join an existing game. For administrators, if a user has administrative rights, then they have some extra buttons in addition to normal plays as they can shutdown games, create user, update user, delete user. In the lobby, if the user wants to create a game, they will have the choice of picking between two maps in which the boss and the items are different. The player will have a button that will allow the player to delete their own account. (admins will not be able to use this feature).

### Gameplay

When users start a game, on the left panel they will see what equipment they have and in the middle top of the screen they will be able to see whose turn it is. Each turn the player will be able to move one tile.

#### Players Movement and Items

When it comes to the gameplay the players will be restricted to a 10 by 10 grid in which they can move around in. The players will start in the same tile and each game will have a max player count of four at one time. Players may move around the board picking up items and equipment before fighting the boss/monster. If a player’s turn takes longer than 3 minutes then their turn is skipped.

#### Monster Combat

a player can encounter a monster on a tile. If they do then they will engage in combat with the user. This will be turn based with the player and monster, so when the player moves to the tile, they will then have to attack the creature, or do nothing, the monster will then do one thing which is attack and deal damage to the player/s in its square. The player deals base damage and then additional damage with effects/equipment. Players may choose to use items that are in their inventory at any time.

#### Objectives and scoring

**Objective:**

The main objective of the game is to go into combat with the monster and then kill it to win. You may divert from this goal by getting equipment and items along the way.

**Scoring:**

There will be a scoring system based off of how many games the user has won and lost.

It is going to work like this:

If player dies: -1 point

If player leaves the game before its finished then no change.

If the player is in a game when the monster dies: +1 point

These points will be shown in the lobby as a sort of leaderboard for active players(meaning only the active players’ scores will be shown.

If the game takes longer than 15 minutes the game ends and no points will be changed.

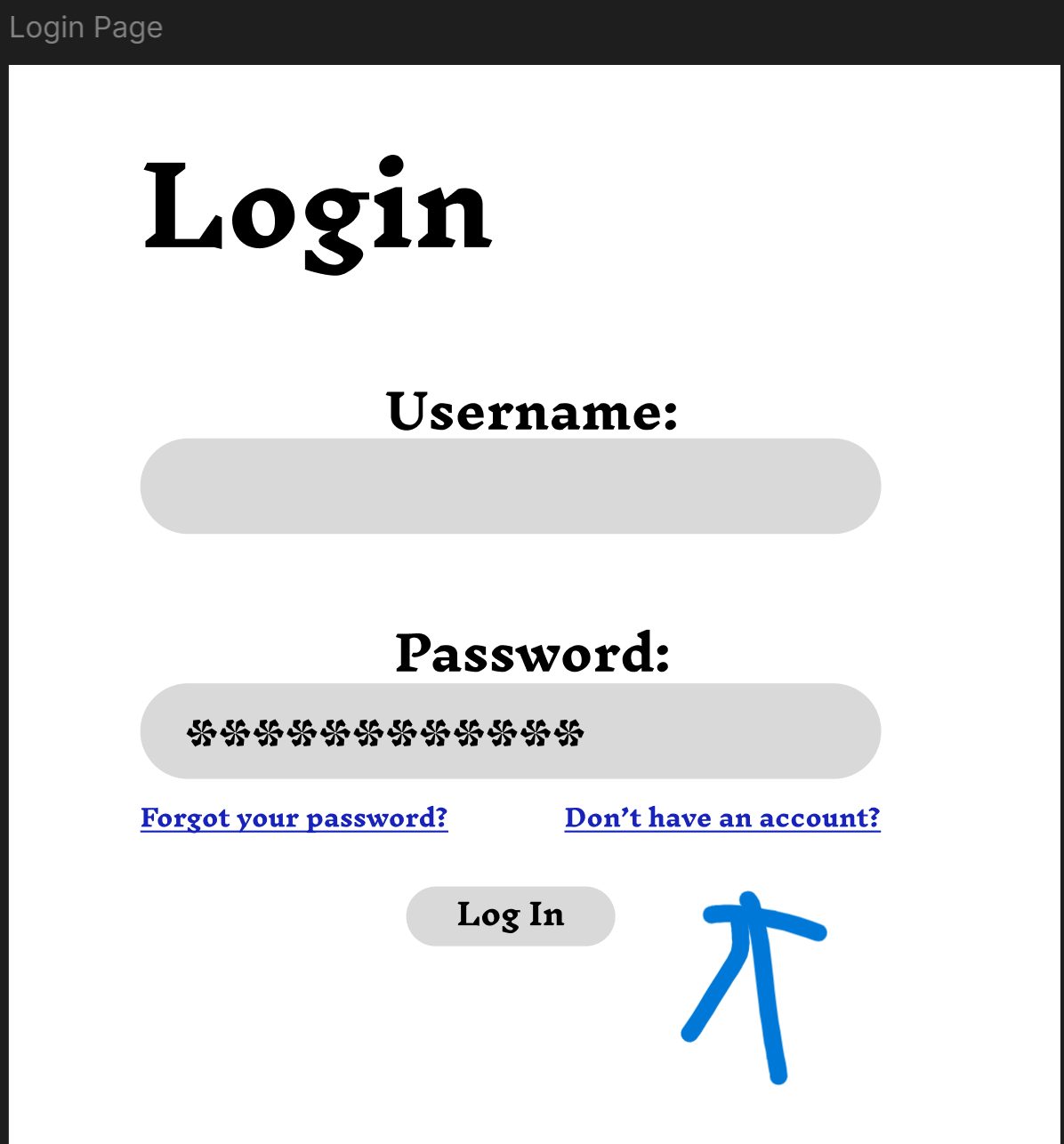
### Administrators

Administrators in this game will higher privileges than normal users as they will be able to do things like end active games, modify/delete users registered users, and lock/unlock registered accounts. An administrator’s job is to monitor the lobby and reply to requests and act on those requests if necessary (actions will be made upon their own volition).

## Storyboards

### Player registration.

The player will launch the application and then be faced with the login screen, if they do not have a login yet they will have to make one.



They will be redirected to the registration page in which they will enter their email, and make a username and password.

A screenshot of a register page

Description automatically generated

They will then press the registration button after the users details have all been entered correctly.

The player will then be logged in and sent to the lobby.

### Player login, including lock out, Failed.

So this user is trying to login. He will first add his username and password into the inputs below and press the “login” button.

If the password or user name is wrong then they will get this page.

A screen shot of a login screen

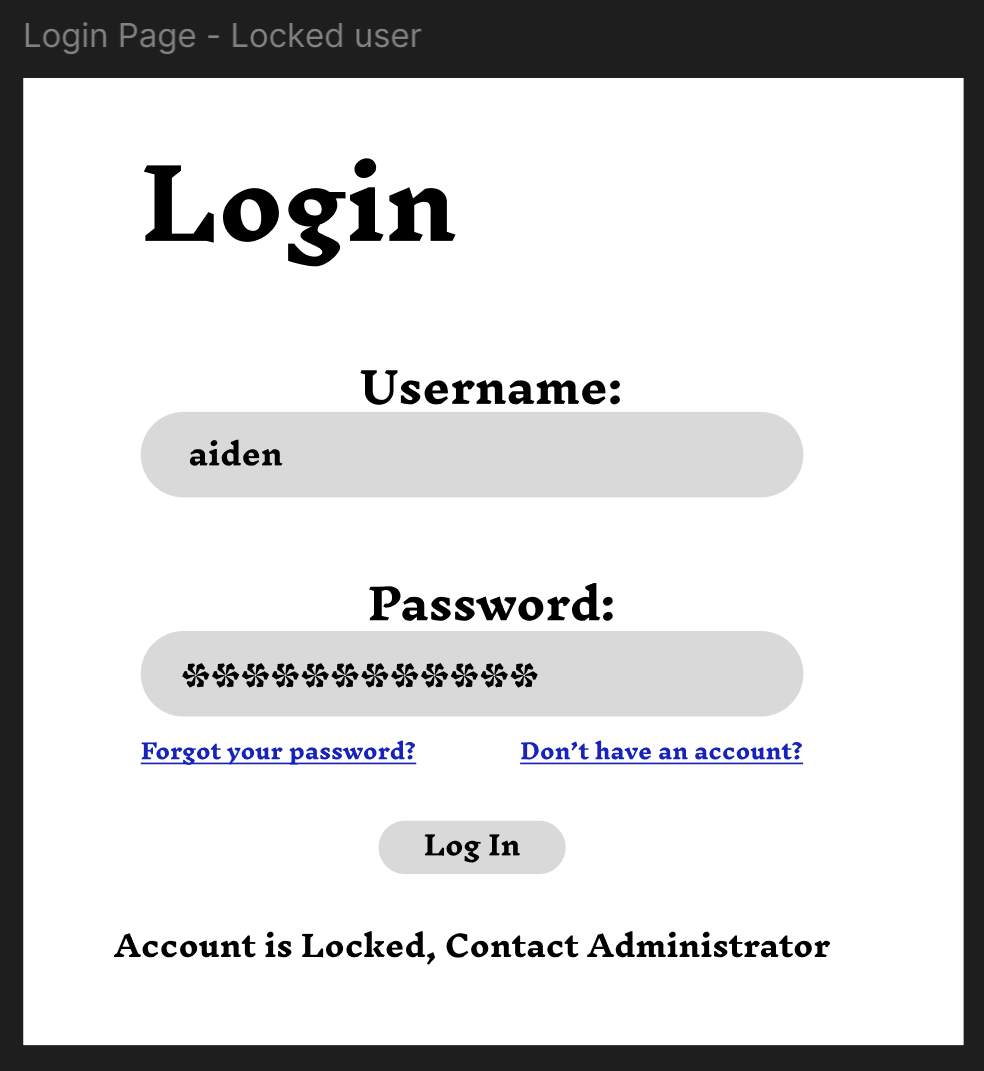
Description automatically generated

Or User not found if the username isn’t found in the database.

A screenshot of a login page

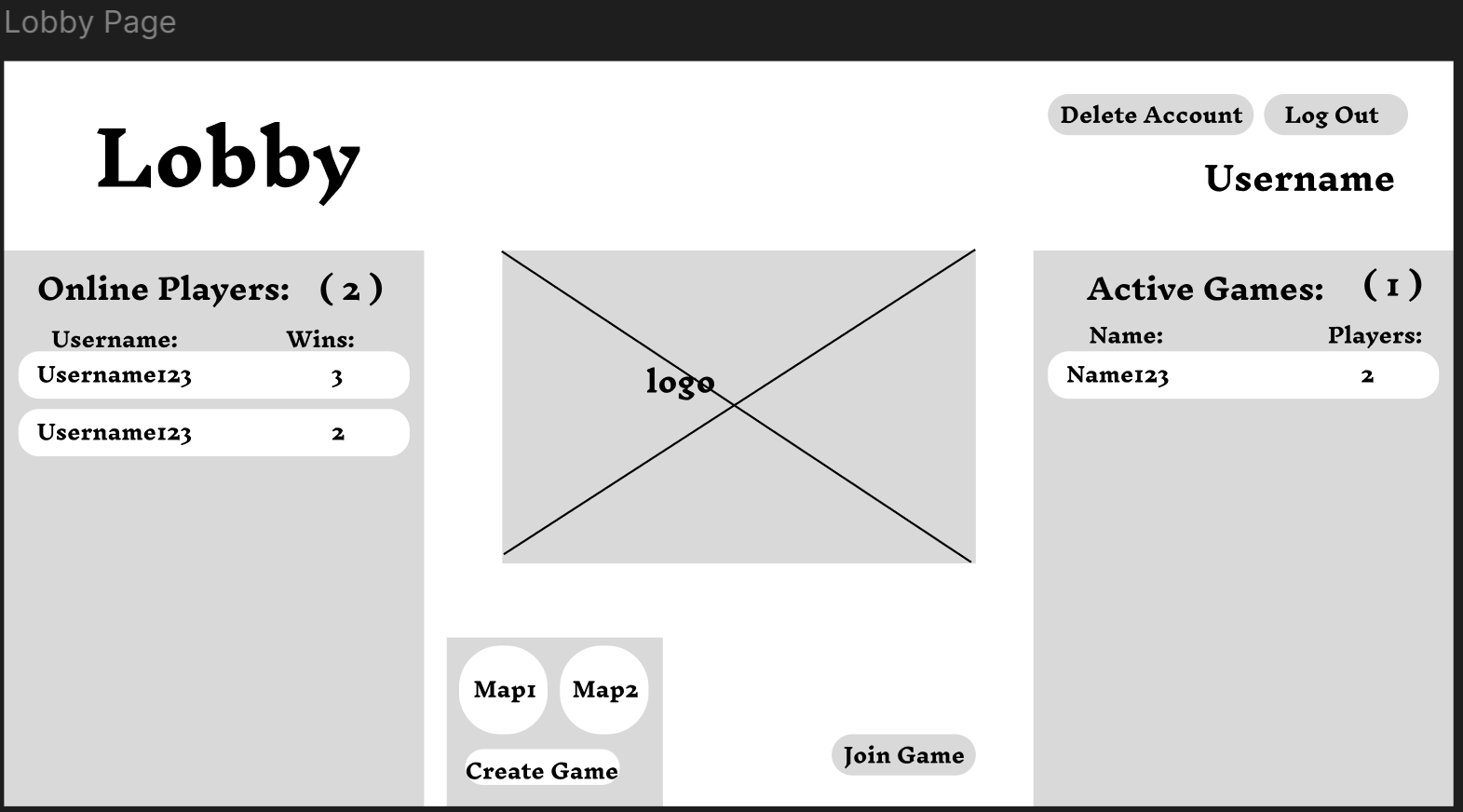
Description automatically generated

If you have failed a users password 5 times then the user account will be locked and user will be shown this screen below.



### Player deletes own account.

So A player will log in with their account and start in the lobby as demonstrated before in the player registration.



They will then click the button at the top of the screen second to the right which says ‘delete account’.

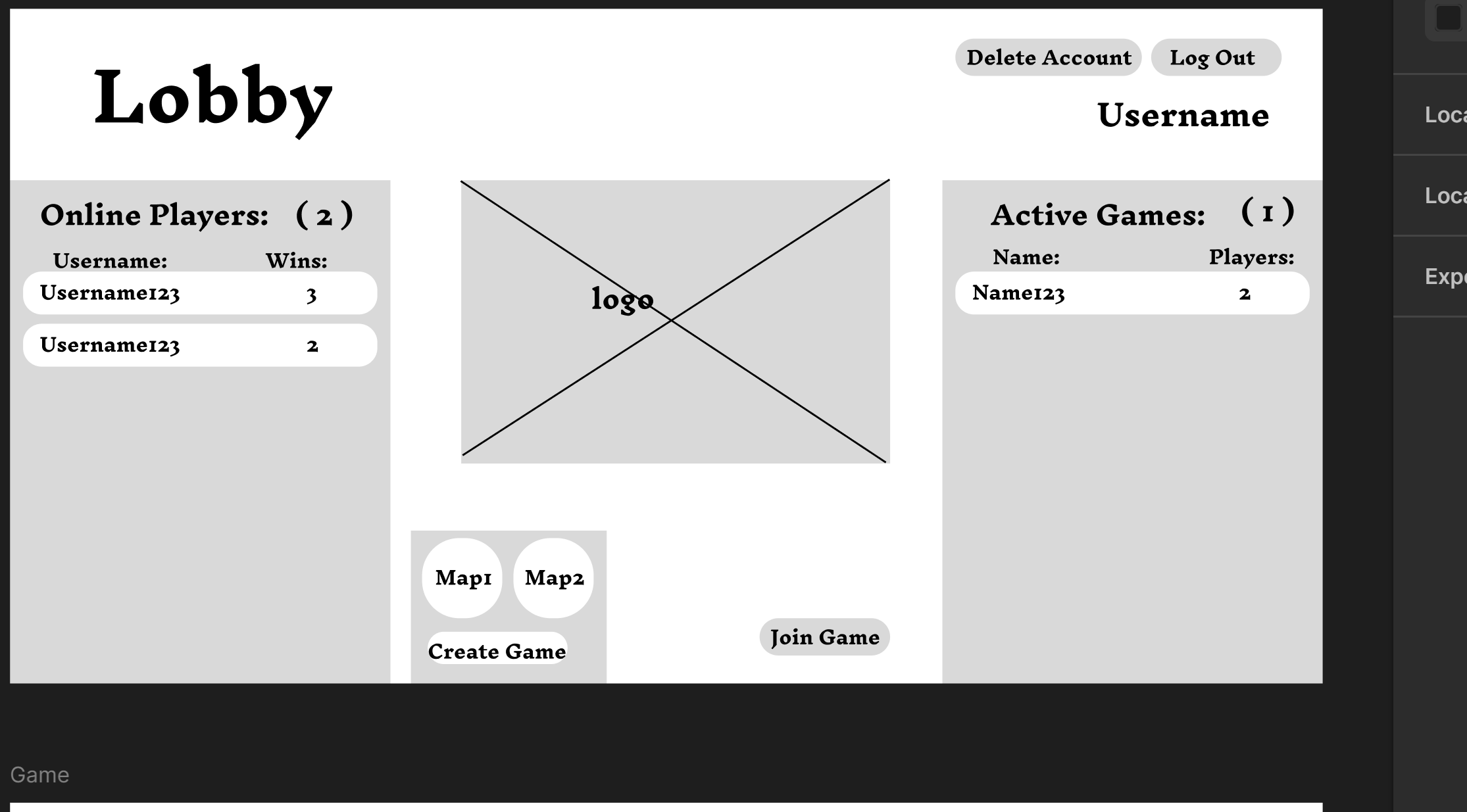
A screenshot of a computer

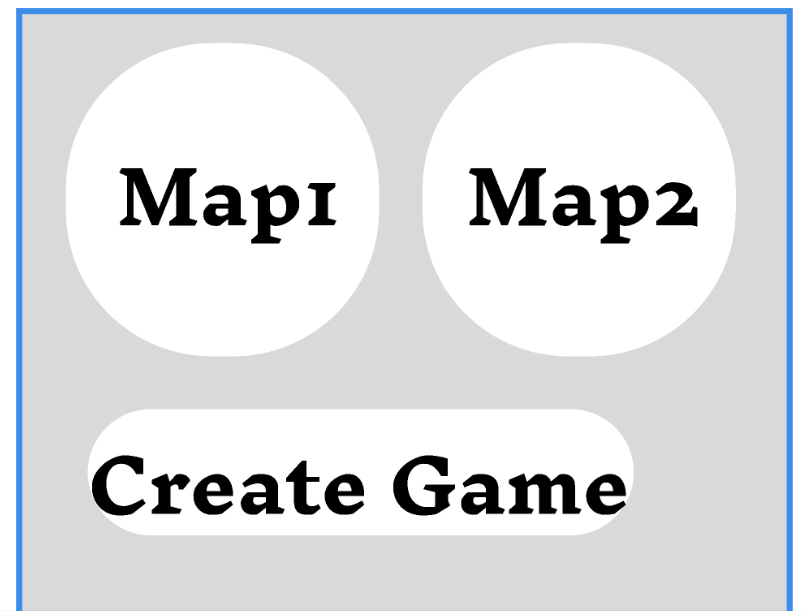
Description automatically generated

The player will then get a confirmation window and if they select yse then their account will be erased from the database.

### How to join a game.

To join a game a player must either click the join matches from the lobby which are located on the right side. Another method that the player can use is creating a game in which they can choose between two maps and then must click create.

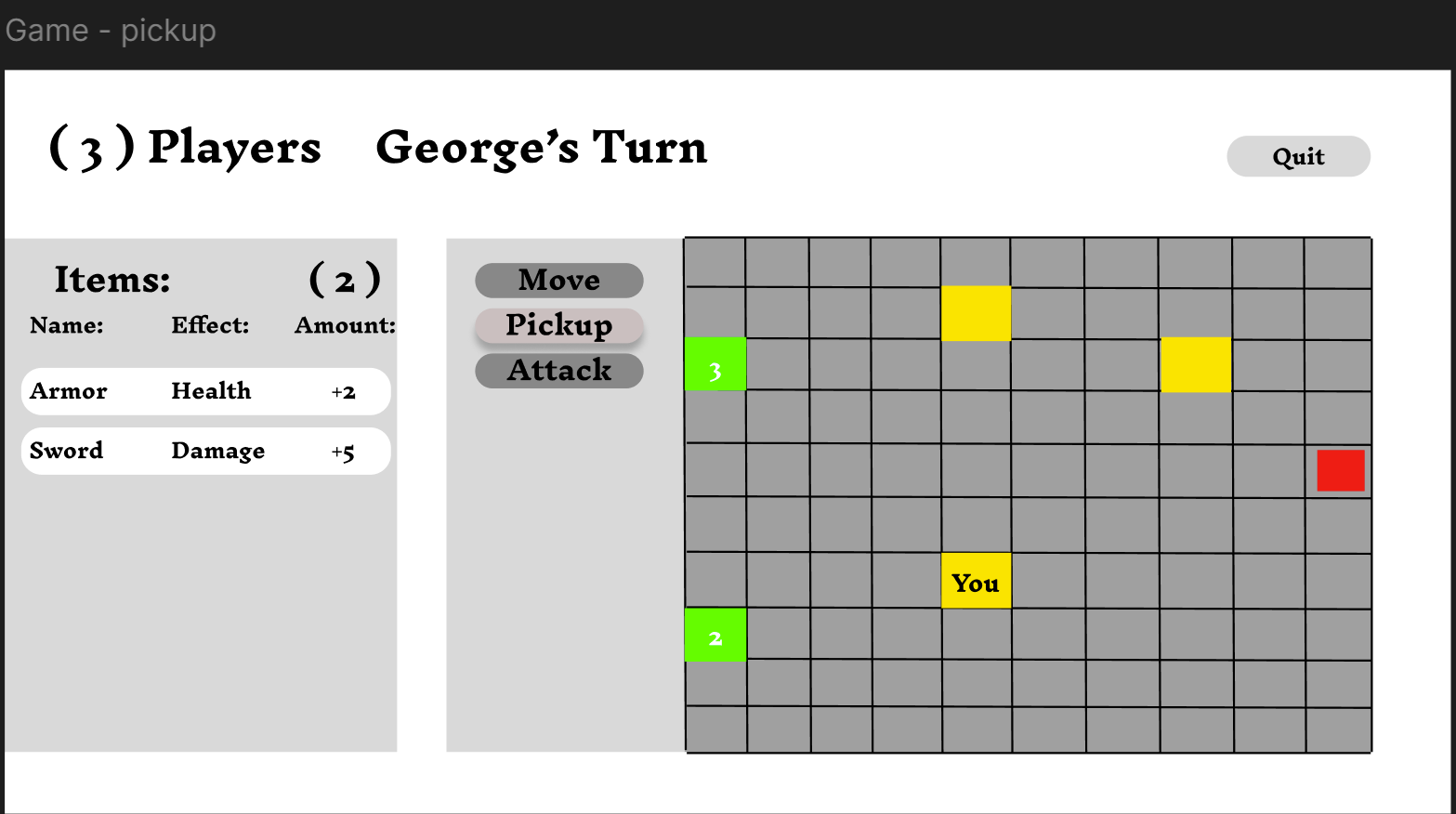


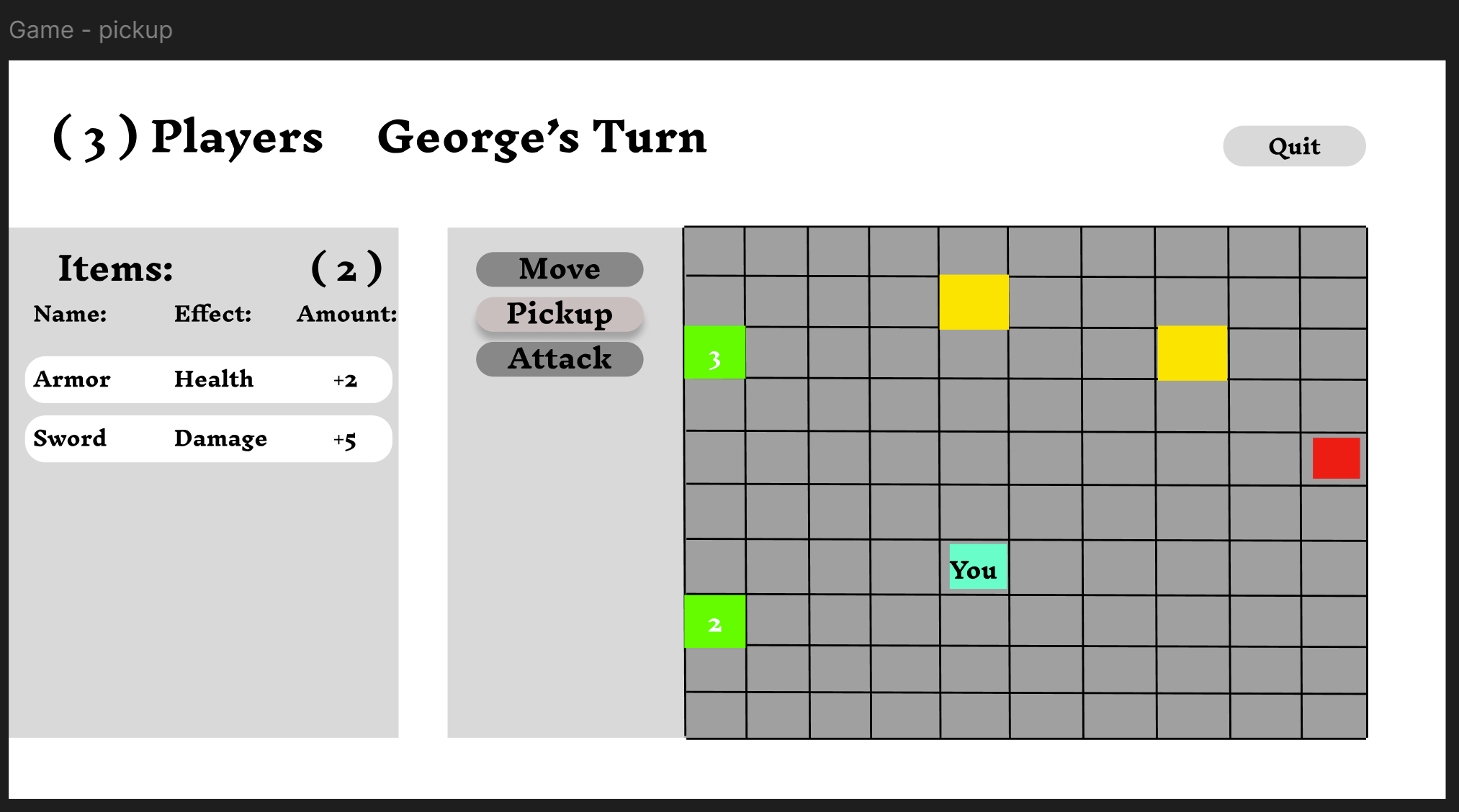


After the player has pressed create, they will then be taken to a new game with their specified map.

### Picking up an item on a tile.

When the player has moved to a spot with a yellow background the pickup option becomes available and then the player can click on that button and then a item will be picked at random from a pool of items and will then be available in the players inventory.





Once an item is picked up you would see a item/s on the left.

### Player - game play movement

When it is your turn you can select the move option and then click any tile adjacent to you(these tiles will be shaded differently) and once you click on a tile you will be moved there. In this example I will head towards the boss.

A screenshot of a game

Description automatically generated

A screenshot of a game

Description automatically generated

### Admin – Delete player account

First the Admin must click on a user account found in the left pane.

Then the Admin must click on the ‘delete user’ button and then a popup will ask for confirmation.

A screenshot of a computer

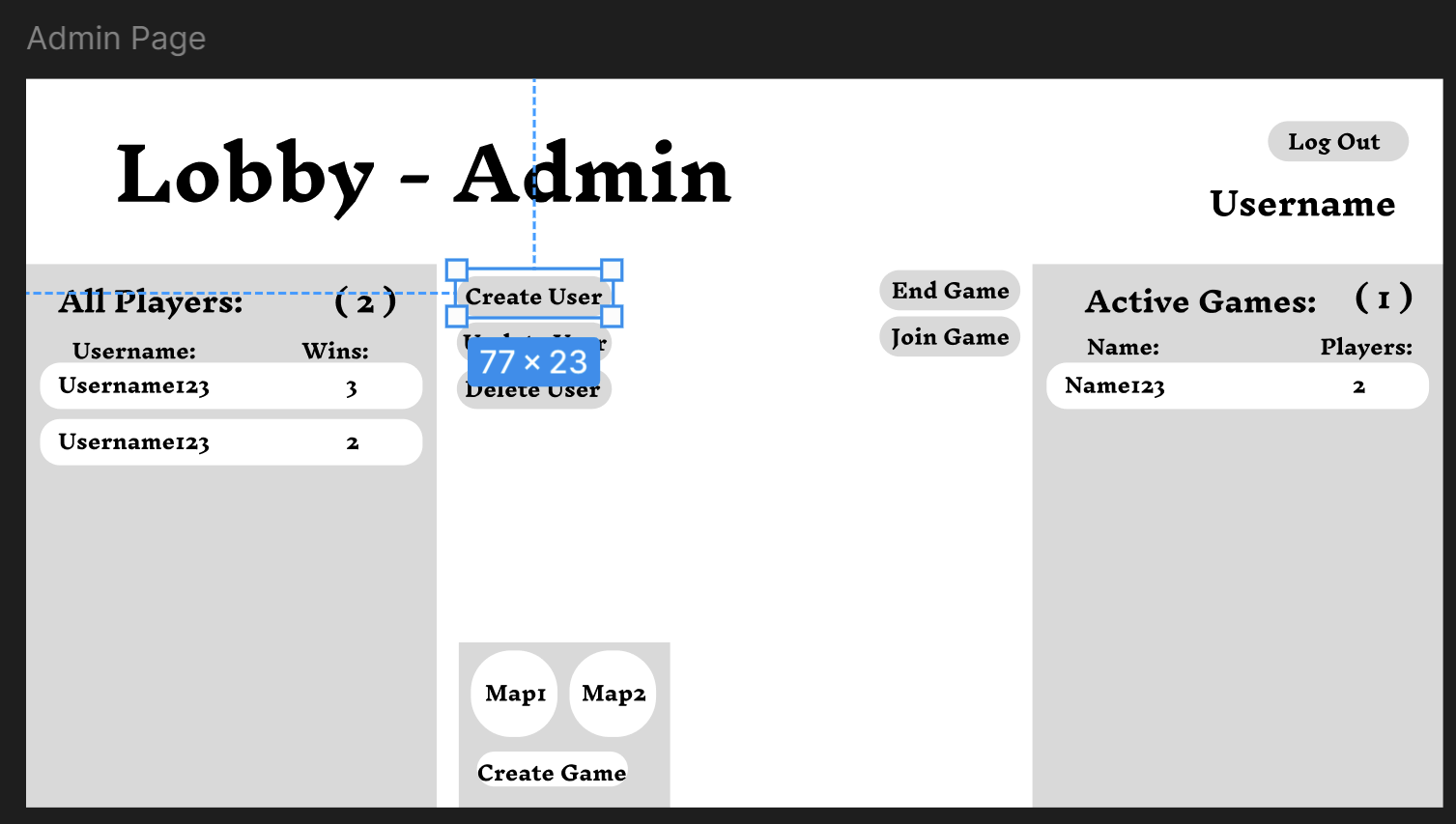
Description automatically generated

The admin could then click on the yes button on the right botton of the popup. If so the account will be deleted from the database. If they click ‘no’ then the popup will be closed and then nothing will happen.

A screenshot of a computer

Description automatically generated

### Admin – Create player account



To create a user account the admin must click on the ‘create user’ button.

A screenshot of a computer

Description automatically generated

After they clicked that button, a will face a popup which will ask for confirmation, if yes then the admin will be logged out and then the user will be sent to the registration page.

In which they will enter the new user account’s username, email and password and will then press create user and then the user will have been created successfully.

A screenshot of a register page

Description automatically generated

### Admin – Update player account

The admin will log in and be in the admin lobby. The admin will select any user on the left pane by clicking on them. They will then click on the update user button.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

If yes a window will pop up and then the admin can edit the details of the user.

A screen shot of a login screen

Description automatically generated

Then the admin once they have made their changes by changing the inputs, clicking reset password which will change the password of the user to what, then all they need to do is to click update.

### How Combat works

Combat is really simple, once you are in a tile adjacent to a monster you are able to use your attack button. You can then click attack then click the monster. The monster health and player health is displayed at the top of the screen in a popup, in this popup you can then confirm the attack or not, in which if you confirm the moster will be dealt damage equal to the player strength and the same the other way around. The interaction of combat is now done for the players turn.

A screenshot of a game

Description automatically generated

A screenshot of a video game

Description automatically generated

The player can click x and then the attack phase has ended for their turn.

### How to end your turn

To end your turn click the end turn button on the Bottom left of the game screen. You may click this button at any point on your turn. It will then skip to the next persons turn.

A screenshot of a game

Description automatically generated

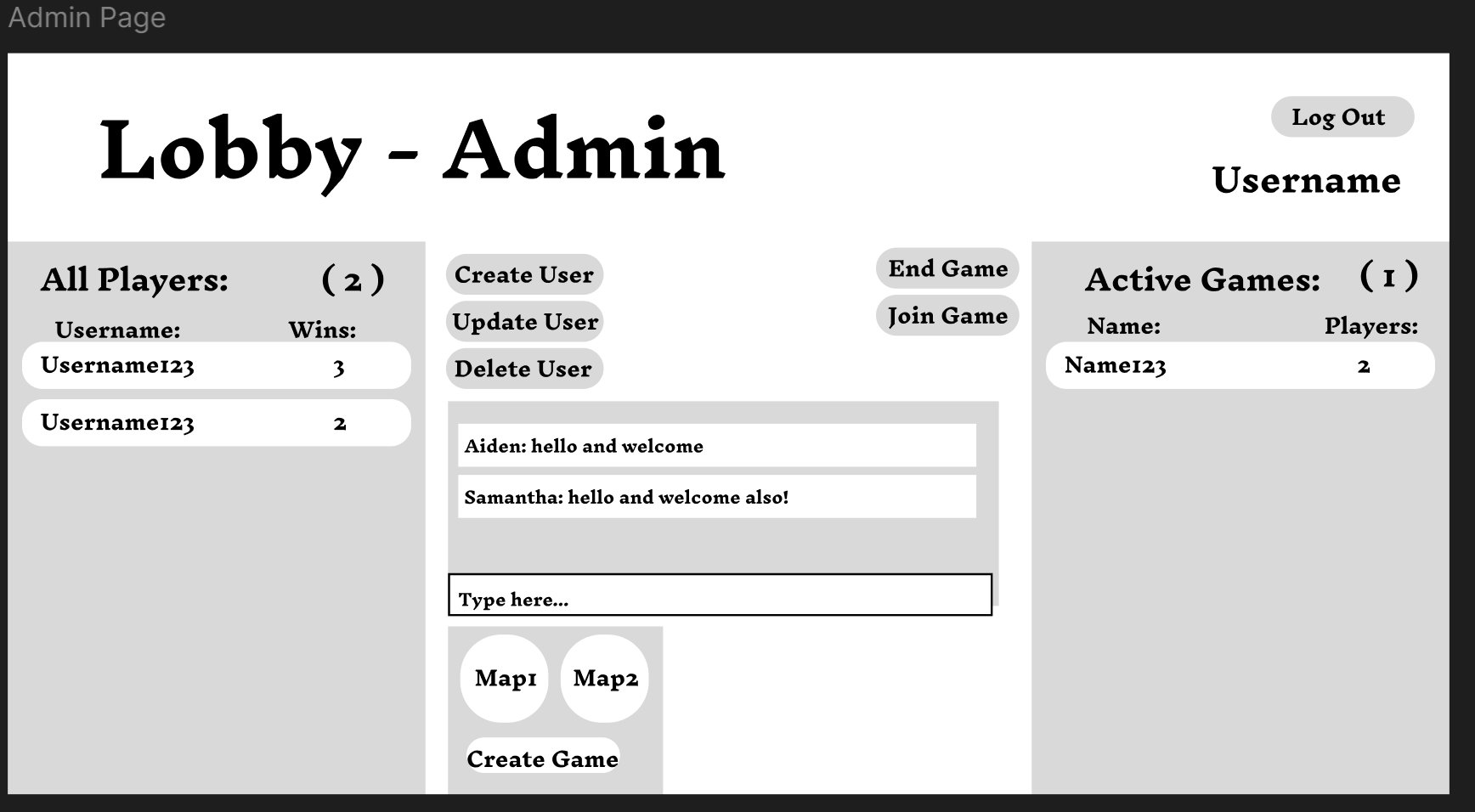
A grey rectangular sign with black text

Description automatically generated

A screenshot of a game

Description automatically generated

### How Chat works



The chat system is simple, you type your message into the input and then press the enter key on your keyboard.

Your message should show up on the listbox above for all the players in the lobby to see.

### How to log out.

On the top right side of your screen in the corner of the lobby, there is a button labelled ‘logout’. If you click this button you will then be logged out and redirected into the login page.

A screenshot of a computer

Description automatically generated

A grey rectangular sign with black text

Description automatically generated

## Entity Relationship Diagram

A screenshot of a computer

Description automatically generated

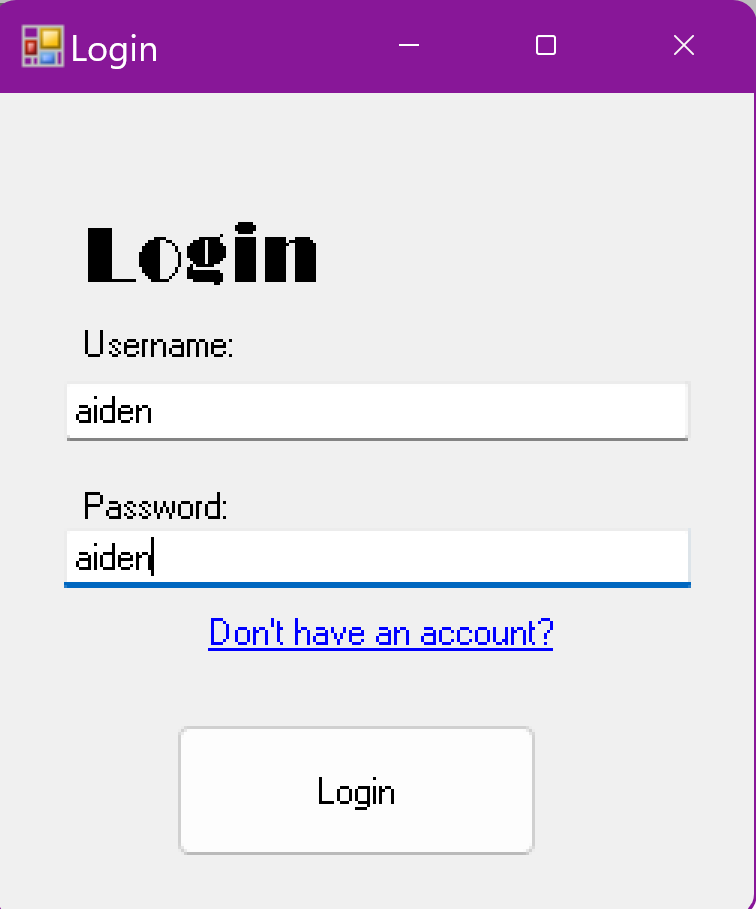
## CRUD Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Check username in DB | Reg new user | Check password against username | Lock account | Login accepted | Join existing game | Start game | Game ends | Player moves | Combat | Player quits Game | Player logs out | Admin kill game | Admin edits existing player | Picking up an item | Admin Delete User |
|  |  |
| Player |  | R | C |  |  | R | R | R | RU | R | R | R | R |  | RU | R | D |
| Player ID |  | C |  |  |  |  |  |  | R | R |  |  |  | R |  | D |
| Username | R | C | R | R | R | R | R | R |  |  | R | R |  | RU | R | D |
| Password |  | C | R |  |  |  |  |  |  |  |  |  |  | RU |  | D |
| Email |  | C |  |  |  |  |  |  |  |  |  |  |  | RU |  | D |
| Locked User | R | C |  | U |  |  |  |  |  |  |  |  |  | RU |  | D |
| Login attemps | R | C | R, U |  | U |  |  |  |  |  |  |  |  |  |  | D |
| User in lobby | R | C |  |  | U | U | R, U | RU |  |  | U | U | U | RU |  | U |
| Player Score |  | C |  |  | R |  |  | RU |  |  |  |  |  |  |  | D |
| Admin user |  |  |  |  | R |  |  |  |  |  |  |  |  |  |  | D |
| Tile ID |  | C |  |  |  | C | C | D | RU | R | RU |  |  |  | R | D |
| Game ID |  | C |  |  |  | C | C | D | R |  | RU |  |  |  | R | D |
| Health |  | C |  |  |  | C | C | D |  | RU | R |  |  |  | RU | D |
| Strength |  | C |  |  |  | C | C | D |  | R | R |  |  |  | RU | D |
| Game |  |  |  |  |  |  | R |  |  |  |  |  |  | D |  |  |  |
| Game ID |  |  |  |  |  | R | C | D |  |  |  |  | D |  | R |  |
| Map |  |  |  |  |  | R | C | D | R |  |  |  | D |  | R |  |
| Game Status |  |  |  |  |  | R | C | D |  |  |  |  | D |  |  |  |
| Turn |  |  |  |  |  | R, U | C | D | R |  | RU |  | D |  | R |  |
| Item |  |  |  |  |  |  | R | C | D |  |  |  |  | D |  |  |  |
| Item ID |  |  |  |  |  | R | C | D |  |  |  |  | D |  | R |  |
| Tile ID |  |  |  |  |  | R | C | D |  |  |  |  | D |  | R |  |
| Status |  |  |  |  |  | R | C | D |  |  |  |  | D |  | RU |  |
| Monster |  |  |  |  |  |  | R | C | D |  |  |  |  | D |  |  |  |
| Status |  |  |  |  |  | R | C | D |  |  |  |  | D |  |  |  |
| Health |  |  |  |  |  | R | C | D |  |  |  |  | D |  |  |  |
| Strength |  |  |  |  |  | R | C | D |  |  |  |  | D |  |  |  |
| Tile ID |  |  |  |  |  | R | C | D | R |  |  |  | D |  |  |  |
|  |  | Loading messages | Tpying and sending a message |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Chat |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ChatID | R | C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Timestamp | R | C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | text | R | C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Milestone 2

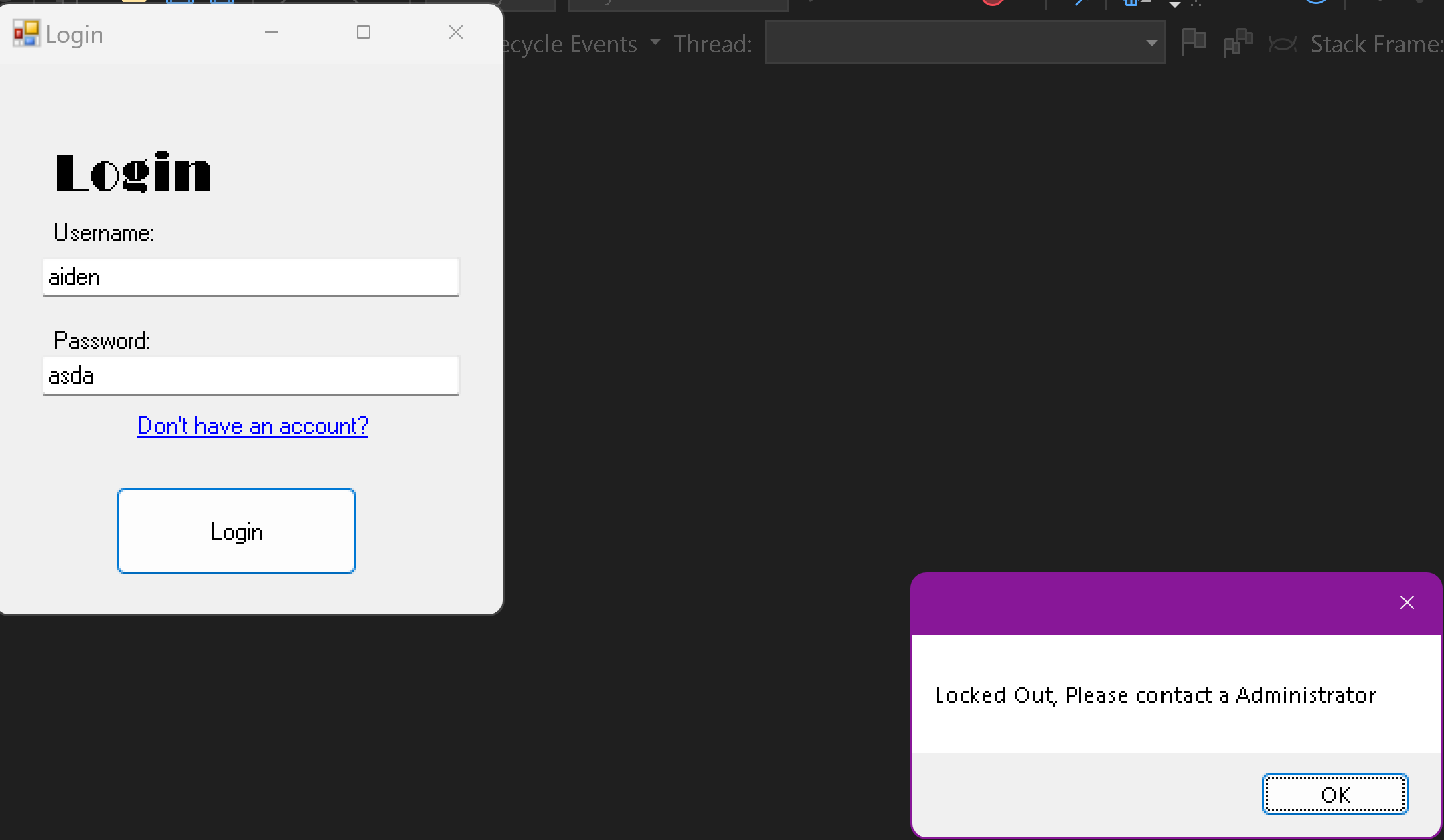
The interaction and handlers have been tested and that is recorded in the report.

## Player login, including lock out. [4]

 A screenshot of a computer

Description automatically generated

Locked out after faling login 5 times.



## Player registration,[4]





## Laying out tiles on a game board. [4]

Click on game and then join game.

A screenshot of a computer

Description automatically generated

A screenshot of a computer game

Description automatically generated

## Placing an item on a tile. [4]

A screenshot of a computer game

Description automatically generated

The Items are placed on the board when the game is created.

## Player game play movement, i.e., moving a player to a “legal” tile. [4]

A screenshot of a computer

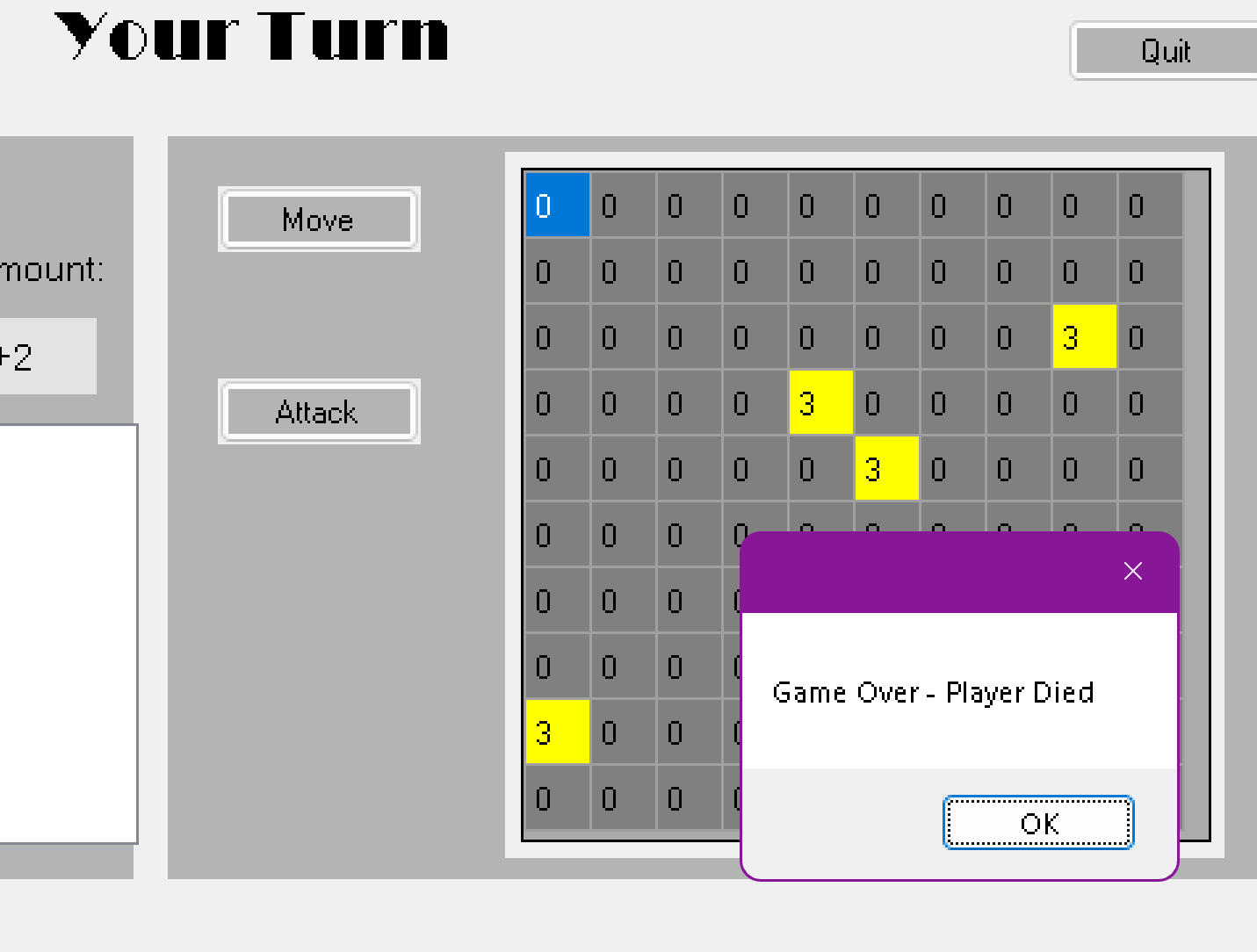
Description automatically generated

When the player wants to move, they can click the move button, this will then highlight the movable tiles to in the game which should be 1 cell around the player except on borders(feature).

Cell validation is implemented but commented out for testing functionality as it does not work as intended but is still functional. As In the player can only move diagonally without it commented out. This is because somewhere in my code I have mixed up my X and Y coordinates and the movable tiles goes in the opposite direction the player wants to move, so it has been disabled temporarily.

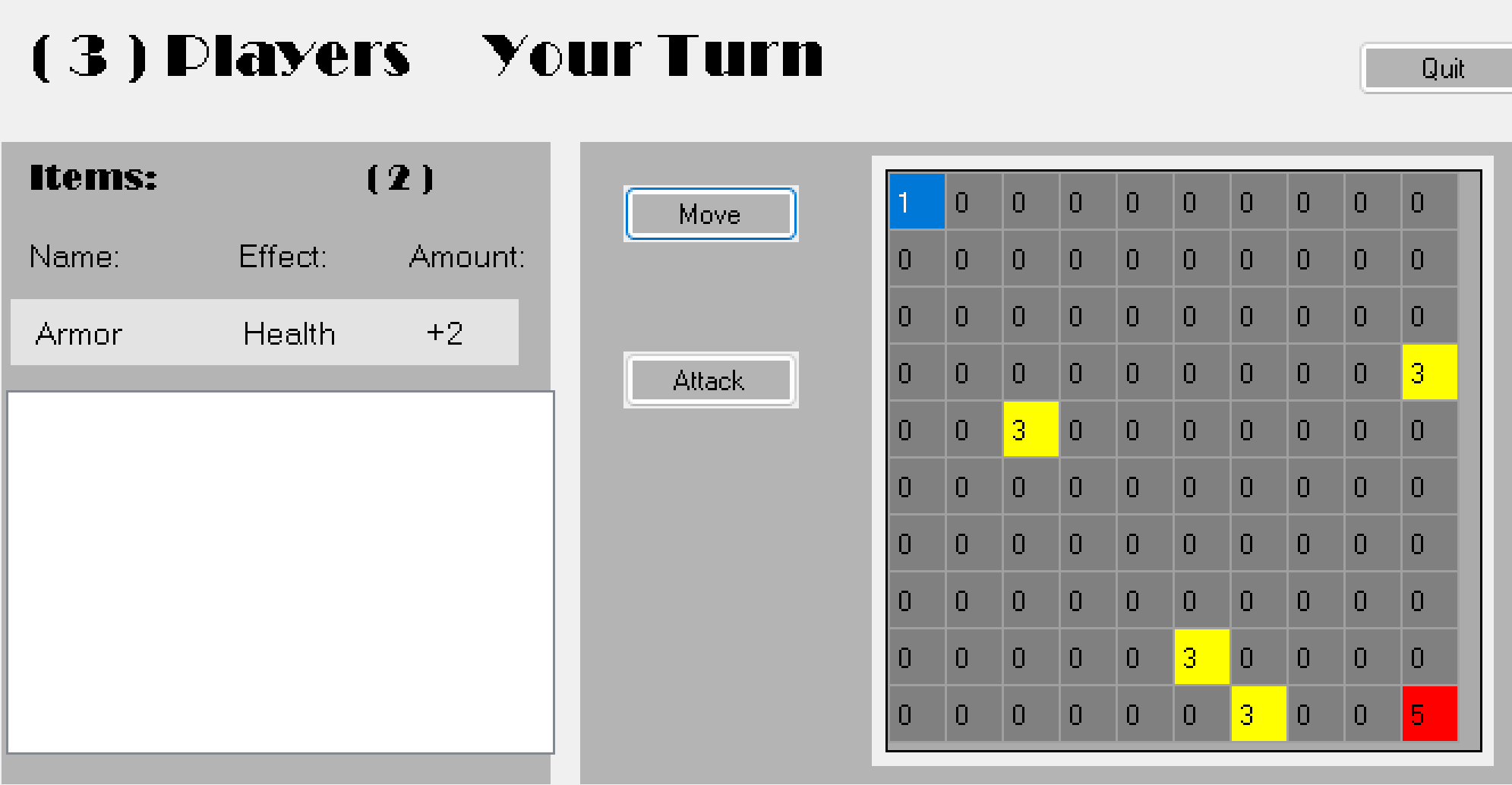
## Game play scoring. How do players gain and lose points? [4]

The player gains points for each cell they survive by moving to another cell without being eaten by the monster.

A screenshot of a video game

Description automatically generated

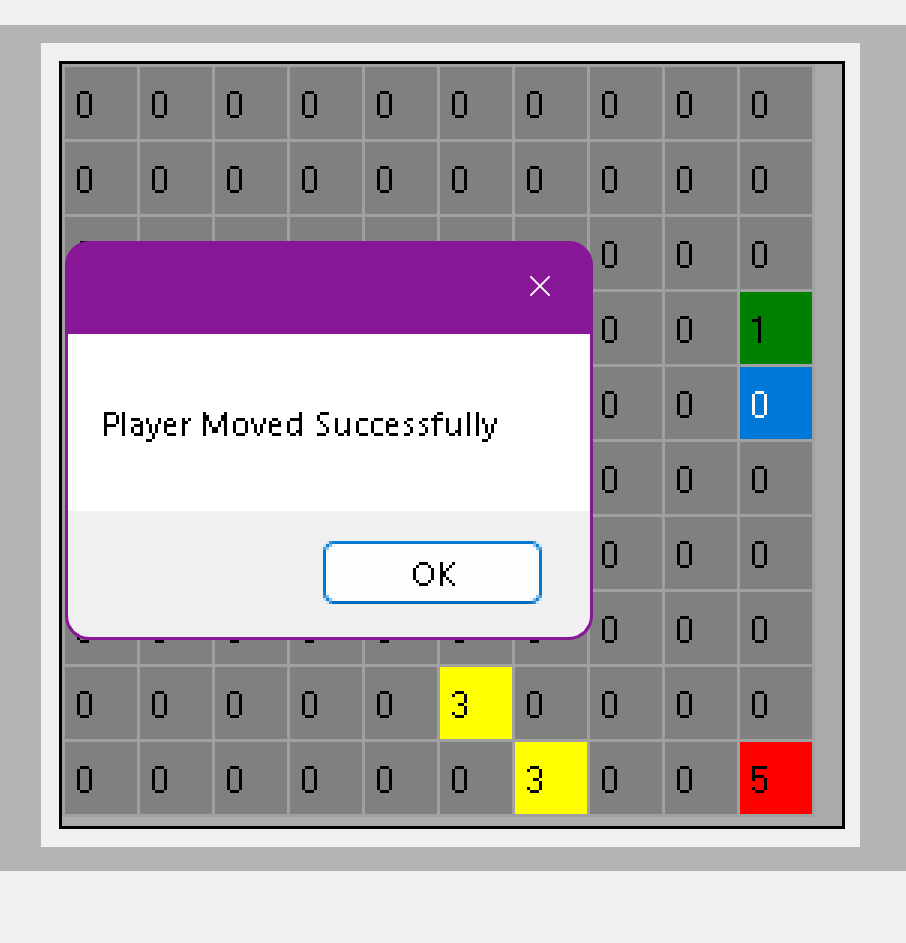
## Player game play acquiring inventory, e.g. picks up items off a tile and putting them in an inventory (bag?) [4]

The player can acquire an inventory by moving to a golden tile, when they do they will get an item that affects either their health or damage.   
A screenshot of a computer game

Description automatically generated

## Move an Item (NPC effect). [4]

Each time the Player moves to a tile that isn’t the monster, themselves or a treasure, the monsterMove function will be called and the monster will move a tile in a random cell around it, within the bounds of the grid.

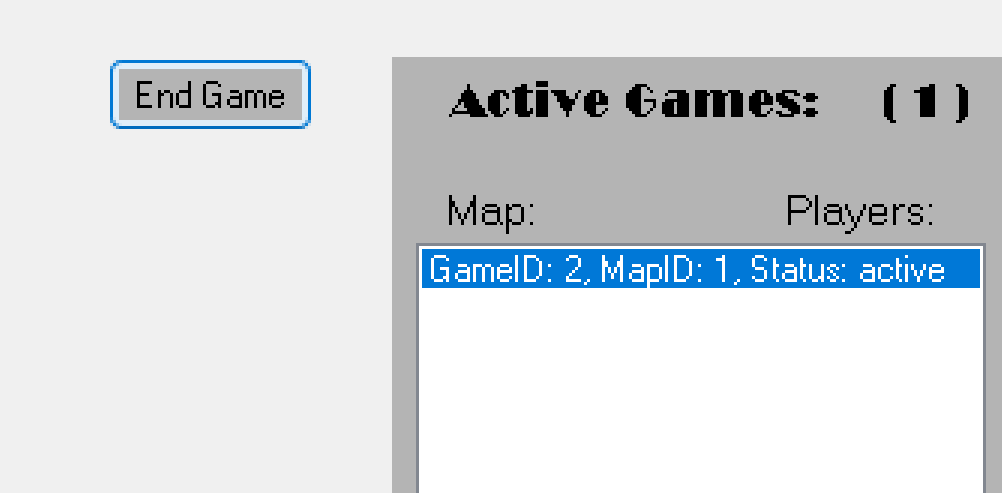
A screenshot of a computer game

Description automatically generatedA screenshot of a game

Description automatically generated

## Kill running games. [4]

To kill a game all the admin has to do is select the game and then click end game.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a video game

Description automatically generated

## Add new players. [4]

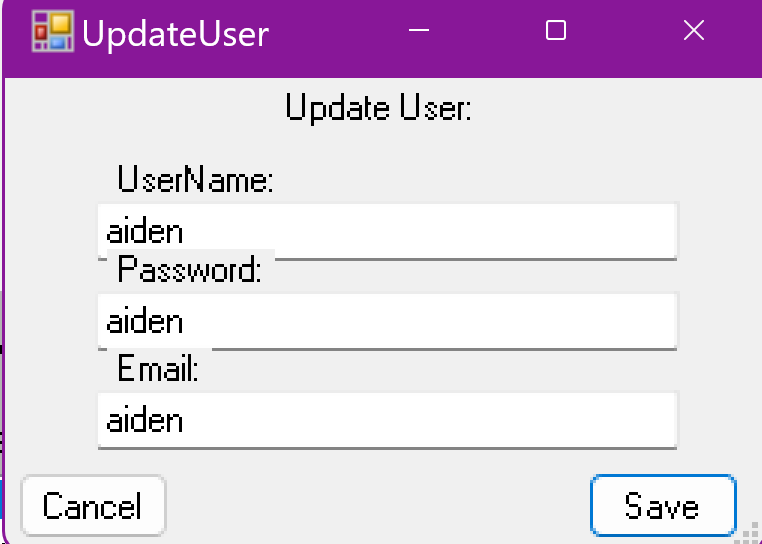
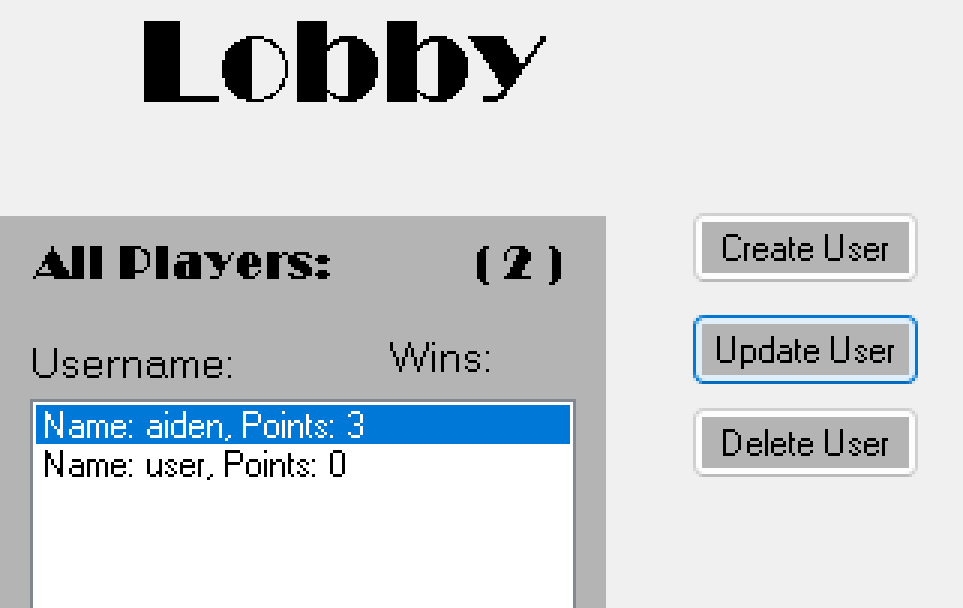
A screenshot of a computer

Description automatically generatedA screenshot of a computer login form

Description automatically generated

The admin button, create a user, takes the admin to the register page as it is easier to do this than to make another window.

## Update data of a player. [4]

To update data of a player, click on their name in the player list as an admin. Then click update, edit the details then click save then your done!  


## Delete a player. [4]

Select a player from the list and then click delete, they will ten be deleted from the list and from the database.  
  
A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generatedA screenshot of a computer

Description automatically generated

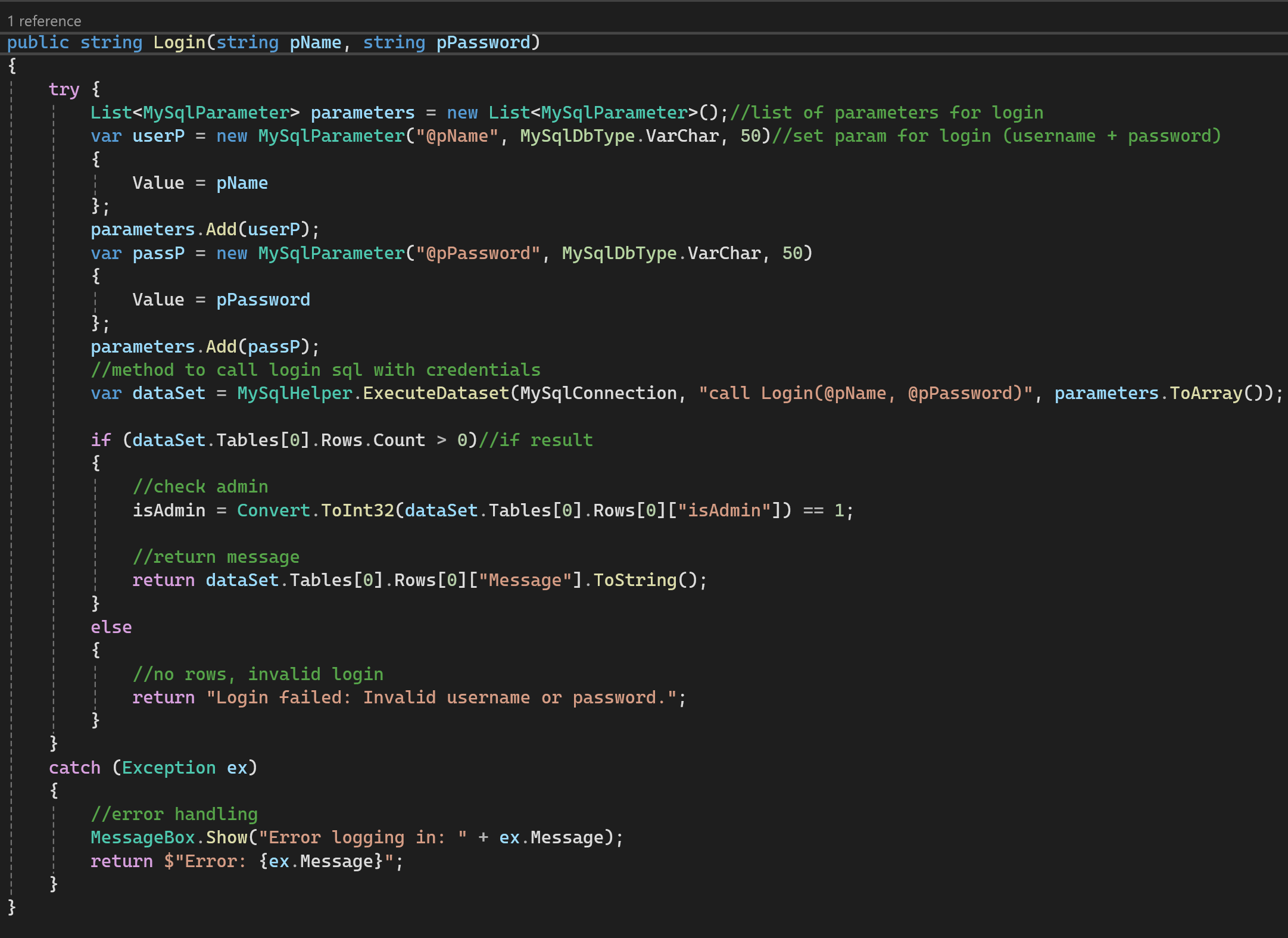
# Milestone 3

## Test Exceptions:

### Player login, including lock out. [3]

A computer screen shot of a program

Description automatically generated

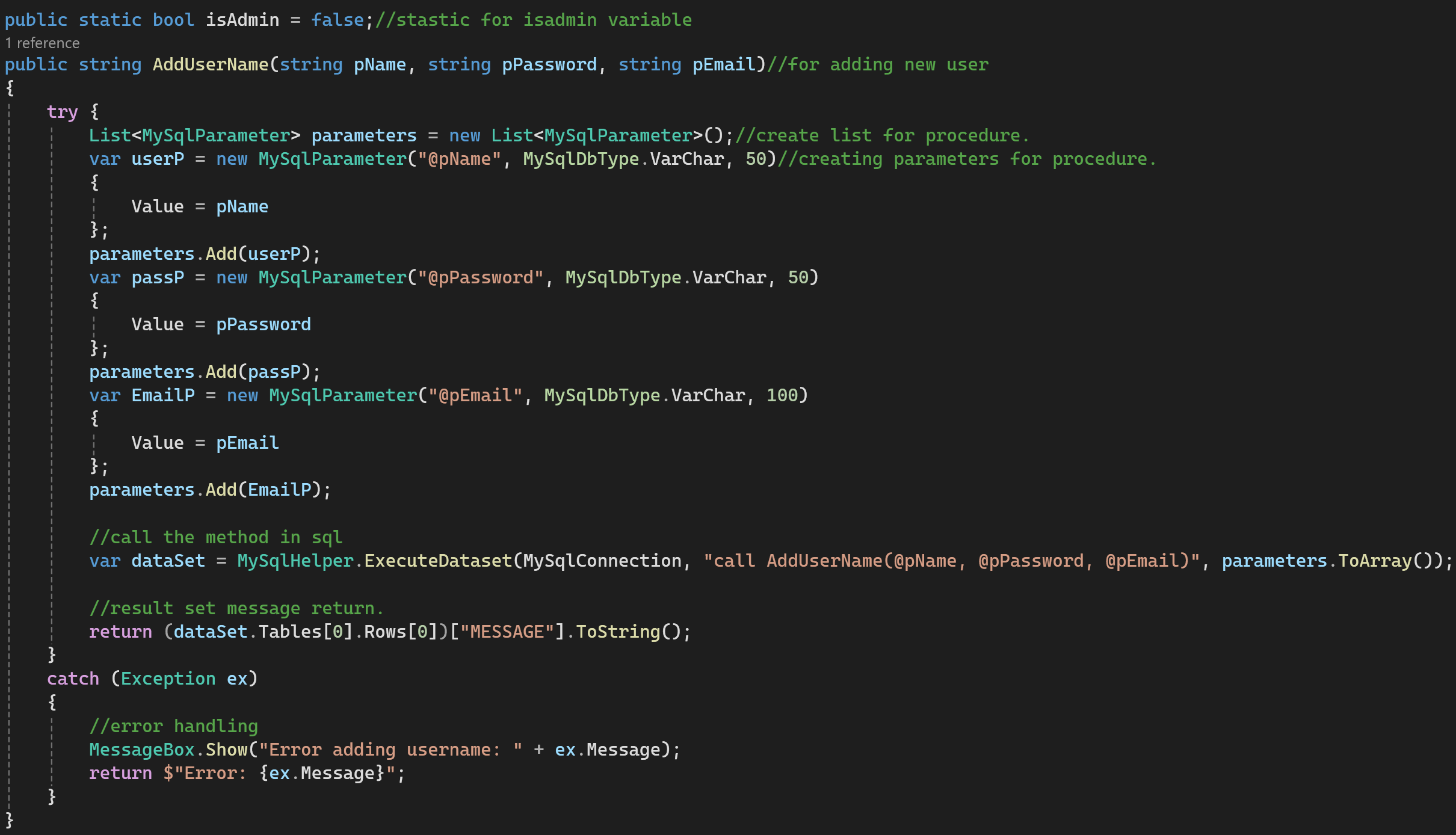


In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Player registration,[3]

A screenshot of a computer program

Description automatically generated



In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Laying out tiles on a game board. [3]

A screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Placing an item on a tile. [3]

This follows the Same Code on the method LoadGameBoard. This is where the tiles are grabbed and updated with new colours then placed onto the grid.

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Player game play movement, i.e., moving a player to a “legal” tile. [3]

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

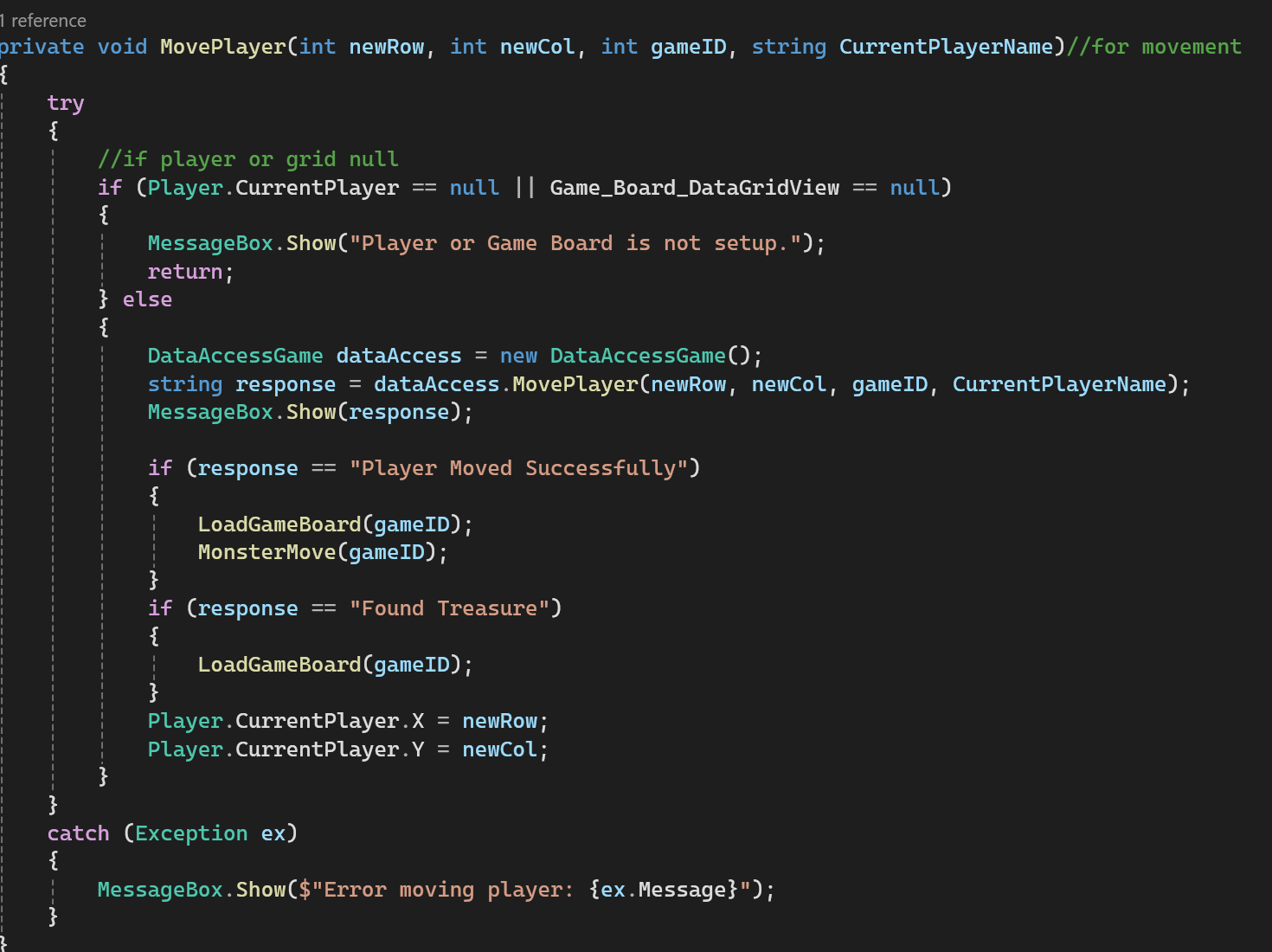
A screen shot of a computer program

Description automatically generated

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Game play scoring. How do players gain and lose points? [3]

When a player moves to an empty tile they gain a point, if the monster overlaps them in the game then they get sent back to the lobby and game is ended.



In the call to SQL it updates the score of the player through the query automatically by adding +1 each time the player moves to a valid tile.

A close-up of a logo

Description automatically generated

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Player game play acquiring inventory, e.g., picking up items off a tile and putting them in an inventory (bag?) [3]

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

When a player moves to a tile that has a treasure on it, it will give the player an item from my items table.

A computer screen shot of text

Description automatically generated

It then calls get treasure which basically gives the player a itemID to them.

### Move an Item (NPC effect). [3]

This is called whenever the player is moved to a valid tile.

A computer screen shot of text

Description automatically generated

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

### Kill running games. [3]

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

A screen shot of a computer code

Description automatically generated

A computer screen with text on it

Description automatically generated

### Add new players. [3]

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

A screen shot of a computer program

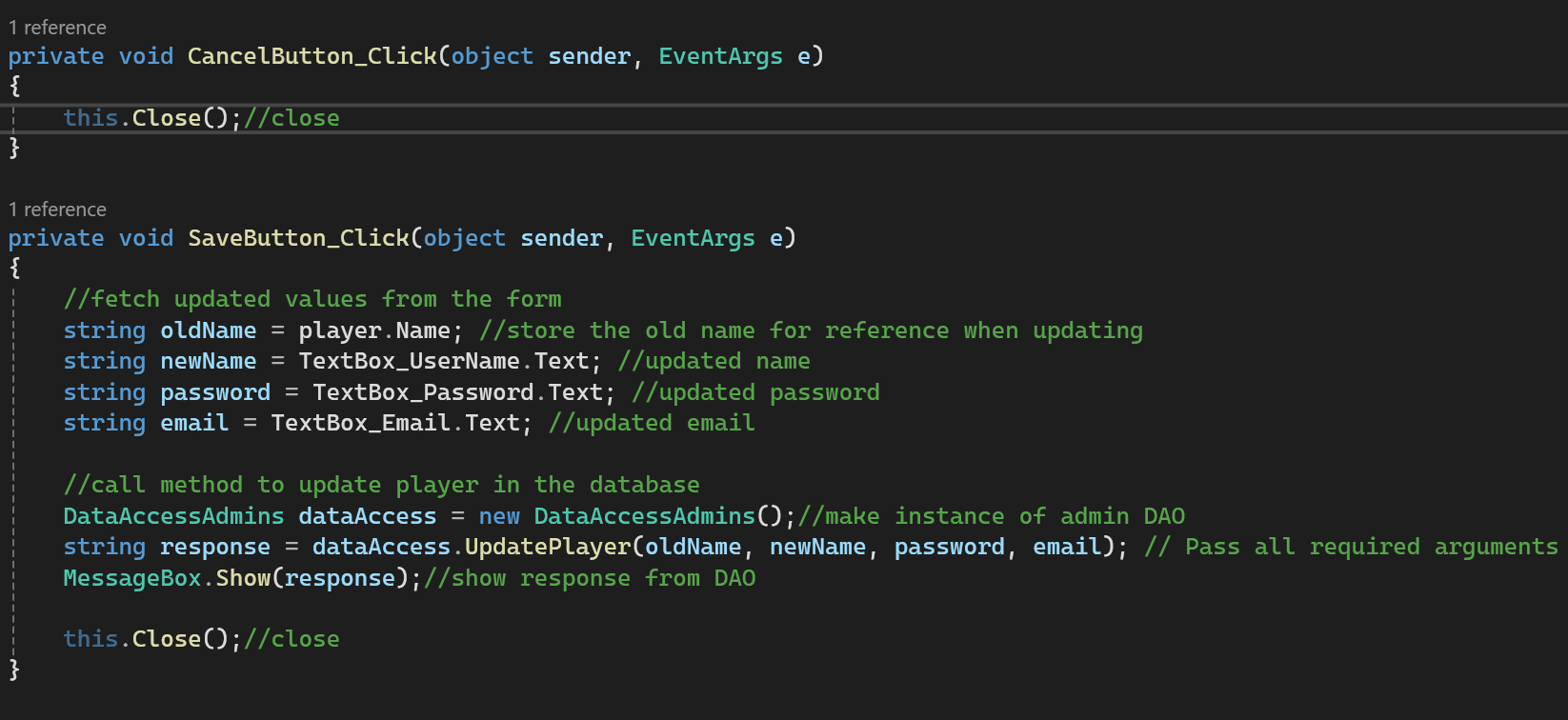
Description automatically generated

A screen shot of a computer program

Description automatically generated

### Update data of a player. [3]

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

s

A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated

### Delete a player. [3]

In this code I use try catch to know if there is an error in my code when its been run. These have been added to all DAO Methods AND all GUI Methods.

A screen shot of text

Description automatically generated

A computer screen shot of text

Description automatically generated

## Concurrency Management:

Compare effective concurrency management in MySQL and.Net C#. **LO3**

Concurrency is important when dealing/managing a multi-user environment where multiple users attempt to modify the same data all at once. Without having concurrency in a high-volume environment, data could be overwritten, and problems could arise that would undermine the integrity or your system and your data. To ensure that data consistency and integrity is maintained, we ensure that we use proper transaction handling for when there a multiple entries and records at the mercy of your transaction.

When it comes to handling concurrency in MySQL code in regard to an environment which has multiple users trying to modify and update the same sets of data simultaneously. The use of transactions and its isolation levels (Read Uncommitted, Read Committed, Repeatable Read, and Serializable) can choose how the transaction level is processed by other transactions (E.G. Serializable is the highest isolation level that locks whole sets of data while the transaction is in process until its committed) as this is essential if the data being changed/modified does not produce a merge or update error that creates an internal conflict in the table/dataset.

(*MySQL Isolation Levels and How They Work — PlanetScale*, 2024)

When it comes to handling concurrency in C# code in regards of a multi-user environment, concurrency is good if data is trying to be updated and changed simultaneously. There are a couple of ways you can use C# concurrency, through Tasks and (parallel) LINQ queries. Tasks can work by utilizing an asynchronous approach when running code so that the user has to wait for a certain method to complete for another to run. With PLINQ it will automatically partition the data and will process it concurrently with multiple threads. It also uses the Task Parallel Library to manage when queries are run synchronously.

(*Supercharge Data Processing with Parallel LINQ (PLINQ) | LinkedIn*, n.d.)

Optimistic vs pessimistic approaches.  
Pessimistic is when the data you are trying to change gets locked while you are editing it. It has better integrity than optimism as its easier and reduces the chance of dirty reads but this way can result in deadlocks of processes where two or more processes are waiting for each other to finish. (solution is to add a protocol for when this happens – specific to the methods being run)

An optimistic approach in C# is where you change a row and then you could assign its value a hash, when it comes back and it’s the same hash, that means it hasn’t been changed, and you can then update/continue to the next row. If it has been changed, it is considered a “Dirty Read” and will have to be rolled back and the values will go to their previous values.(*Database - Optimistic vs. Pessimistic Locking - Stack Overflow*, n.d.)

For my Conclusion, in regards to concurrency, MySQL has Isolation levels that allow the dev to change and modify what changes can be made concurrently which could be good for low level trafficked applications. For C# It allows developers to choose between an optimistic or pessimistic approach when it comes to multi threading and concurrency. A pessimistic approach could be good for a site that needs data integrity like a bank, however optimistic could be good for a faster more user-experience based application like a game by lowering ping(server response).

# REFERENCES:

*Database—Optimistic vs. Pessimistic locking—Stack Overflow*. (n.d.). Retrieved 14 November 2024, from https://stackoverflow.com/questions/129329/optimistic-vs-pessimistic-locking

*MySQL isolation levels and how they work—PlanetScale*. (2024, January 8). https://planetscale.com/blog/mysql-isolation-levels-and-how-they-work

*Supercharge Data Processing with Parallel LINQ (PLINQ) | LinkedIn*. (n.d.). Retrieved 14 November 2024, from https://www.linkedin.com/pulse/supercharge-data-processing-parallel-linq-plinq-amir-doosti/