

DAT602

Project Documentation

Xander Crocker

## Table of Contents

Milestone One .....	7
1. Game Description .....	7
1.1 Game Objective: .....	7
1.2 Gameboard / Movement: .....	7
1.3 Player: .....	7
1.4 Items:.....	7
1.5 Scoring: .....	7
1.6 Login / Registration .....	8
1.6.1 Player Login:.....	8
1.6.2 Player Registration:.....	8
1.6.3 Player Lockout:.....	8
1.7 Administrative Functionality .....	8
2. Storyboards .....	9
Storyboard 1 – Login .....	9
1.1 Username: .....	9
1.2 Password: .....	9
1.3 Login Button: .....	9
1.4 Register Button: .....	9
Storyboard 2 – Login Error .....	10
2.1 Error Message: .....	10
2.2 Yes Button: .....	10
2.3 No Button:.....	10
Storyboard 3 – Registration .....	11
3.1 First Name: .....	11

3.2 Last Name: .....	11
3.3 Email: .....	11
3.4 Username: .....	11
3.5 Password: .....	11
3.6 Register Button: .....	11
3.7 Login Button: .....	11
Storyboard 4 – Locked Account .....	12
4.1 Account Locked Message: .....	12
4.2 Close Button: .....	12
Storyboard 5 – Main Window.....	13
5.1 New Game: .....	13
5.2 Join Game: .....	13
5.3 Admin Controls: .....	13
5.4 Logout:.....	13
Storyboard 6 – In Game Player Gameboard .....	14
6.1 Gameboard (10x10 Grid):.....	14
6.2 Player One: .....	14
6.3 Player Two:.....	14
6.4 Your Score: .....	14
6.5 Opponents Score:.....	14
6.6 Home Button:.....	14
Storyboard 7 – Player Win.....	15
7.1 Message: .....	15
7.2 Home Button:.....	15
Storyboard 8 – Player Loss .....	15

8.1 Message: .....	15
8.2 Home Button:.....	15
Storyboard 9 – Administrator Screen .....	16
9.1 Current Active Games: .....	16
9.2 Kill Current Active Game Button:.....	16
9.3 Registered Users: .....	16
9.4 Create Button: .....	16
9.5 Edit Button:.....	16
9.6 Delete Button: .....	17
Storyboard 10 – Administrator Editor Screen .....	17
10.1 First Name: .....	17
10.2 Last Name: .....	17
10.3 Email:.....	17
10.4 Username: .....	17
10.5 Password: .....	18
10.6 Lock Account Checkbox: .....	18
10.7 Administrative Permissions Checkbox:.....	18
10.8 Update Button: .....	18
2.1 Storyboard Design Descriptions .....	18
2.1.1 Login & Registration Screens:.....	18
2.1.2 Main Window: .....	18
2.1.3 In Game Player Gameboard:.....	19
2.1.4 Administrator Console & Editor:.....	19
3. Entity Relationship Diagram.....	20
3.1 Entity Relationship Diagram Relationships.....	20

3.1.1 Player, Session, Game Table Relationship .....	20
3.1.2 Game, BoardTile Table Relationship .....	20
3.1.3 BoardTile, Item Table Relationship.....	20
4. CRUD Table .....	21
4.1 CRUD Analysis .....	21
4.1.1 Login Player.....	21
4.1.2 Check Username in Database .....	21
4.1.3 Check Password in Database .....	21
4.1.4 Register Player .....	21
4.1.5 Login Attempts .....	22
4.1.6 Player Logout.....	22
4.1.7 (Admin) Lock Out of Account.....	22
4.1.8 (Admin) Kill a Running Game .....	22
4.1.9 (Admin) Create a New Player.....	22
4.1.10 (Admin) Edit a Players Details .....	22
4.1.11 (Admin) Delete a Player .....	22
4.1.12 Start a New Game .....	23
4.1.13 End the Current Game .....	23
4.1.14 Join an Existing Game .....	23
4.1.15 Move Player .....	23
5. SQL.....	24
5.1 DDL .....	24
5.1.1 Player Table .....	24
5.1.2 Game Table.....	25
5.1.3 Session Table: .....	26

5.1.4 Board Tile Table: ..... 26

5.1.5 Item Tile Table: ..... 26

# **Milestone One**

## **1. Game Description**

### **1.1 Game Objective:**

The objective of the game is to gather more items than your opponent. Each item collected by a player will give them a point. Items will be randomly placed on the gameboard one at a time, each player will need to get to the square before their opponent to collect the item and receive a point. The game will end when one player has reached the point limit (10).

### **1.2 Gameboard / Movement:**

The gameboard will be a 10x10 grid of squares. Players will be able to move up, down, left, and right along the gameboard. Players will not be able to move diagonally along the grid. The player's movement will be controlled with the four arrow keys.

### **1.3 Player:**

When the game starts players will be placed in either the top right or bottom left corners of the gameboard. Once the game starts players will be able to move freely around the gameboard using the arrow keys.

### **1.4 Items:**

Items will be randomly placed on the gameboard one at a time. The next item will appear when the current item is collected by a player. To collect an item a player must get to and stand on the tile where the item has spawned before their opponent.

### **1.5 Scoring:**

Scores are calculated by the number of items each player collects. Each item collected will give the player a point. The game will end when one player has 10 points.

## **1.6 Login / Registration**

### **1.6.1 Player Login:**

When opening the application users will be directed to a login screen. This screen will allow users to input their login details, email, username, password, etc. The login details will then be checked by the database to see if the user is registered. If a player is not currently registered in the database the user will be asked if they would like to register, if yes, the user will be redirected to the registration screen.

### **1.6.2 Player Registration:**

When a user is redirected to the registration screen the user must input their details, email, first and last names, username, and password to register their new account. Once the new player is registered, they will be redirected back to the login screen to login to their new account and start playing the game.

### **1.6.3 Player Lockout:**

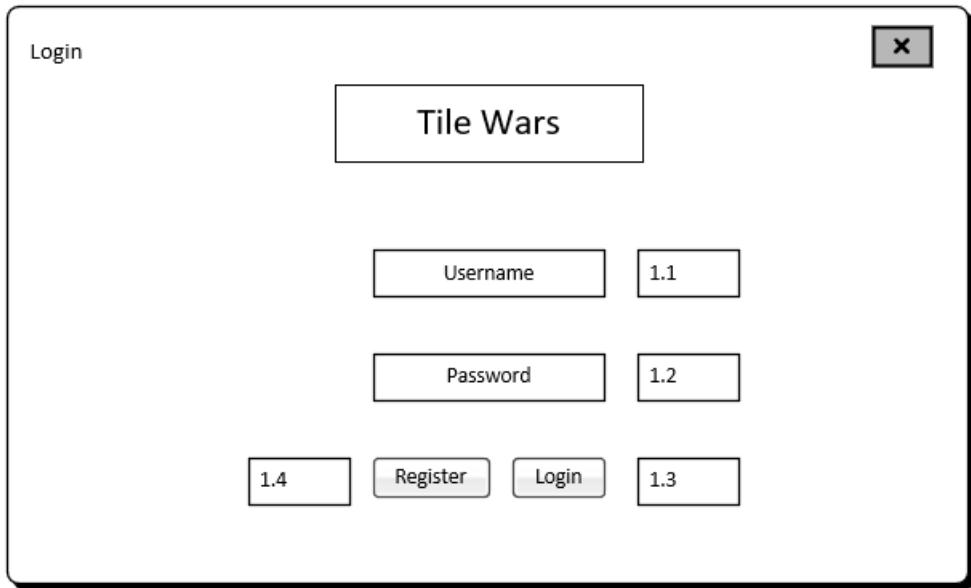
A user will have 5 attempts to login to their account from the login screen. If a user is unable to login to their account within these 5 attempts their account will be temporarily locked.

## **1.7 Administrative Functionality**

Administrators will have special privileges; these privileges will allow them to create new users, edit existing users' details, delete existing users from the database, and lock or unlock accounts.

## 2. Storyboards

### Storyboard 1 – Login



#### 1.1 Username:

Textbox for username input.

#### 1.2 Password:

Textbox for password input.

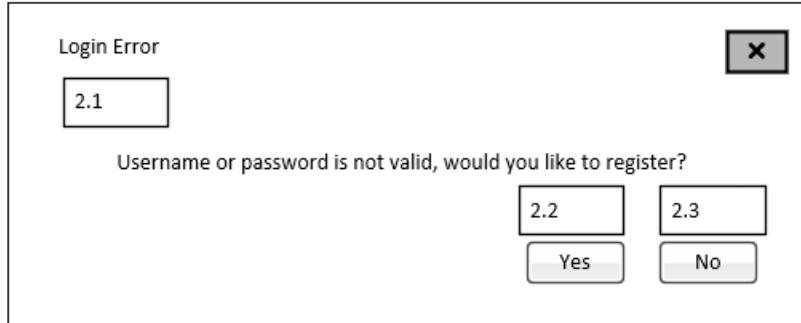
#### 1.3 Login Button:

Button that when clicked checks if username and password exist in the database.

#### 1.4 Register Button:

Button that sends the user to the registration screen (storyboard 3).

## Storyboard 2 – Login Error



### 2.1 Error Message:

Message that appears if the username or password does not match an existing entry in the player table in the database.

### 2.2 Yes Button:

The yes button when clicked will send the user to the registration screen (storyboard 3).

### 2.3 No Button:

The no button when clicked will close the error message and the user will then be able to attempt to login again.

## Storyboard 3 – Registration

The registration screen is titled "Register" at the top left and features a close button at the top right. The title "Tile Wars" is centered above the input fields. The form consists of seven input fields arranged in two columns. The first column contains "First Name" (3.1), "Last Name" (3.2), "Email" (3.3), and "Username" (3.4). The second column contains "3.1" (aligned with First Name), "3.2" (aligned with Last Name), "3.3" (aligned with Email), and "3.4" (aligned with Username). Below these is a "Password" field (3.5) with "3.5" aligned to its right. At the bottom are three buttons: "3.7" (aligned with First Name), "Login" (aligned with Password), "Register" (highlighted in grey), and "3.6" (aligned with Password).

### 3.1 First Name:

Textbox for the user to input their first name.

### 3.2 Last Name:

Textbox for the user to input their last name.

### 3.3 Email:

Textbox for the user to input their email.

### 3.4 Username:

Textbox for the user to input the username they wish to be recognised by in game.

### 3.5 Password:

Textbox for the user to input their preferred password.

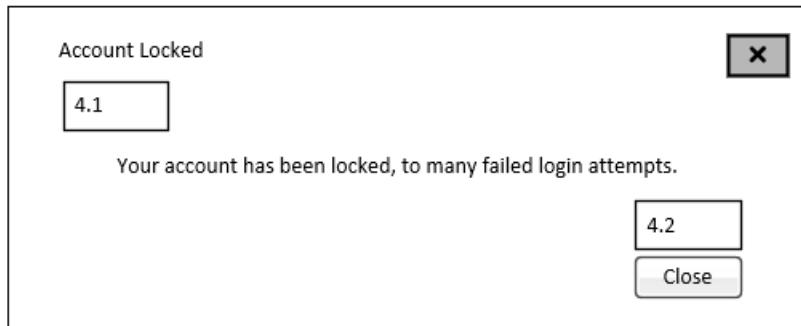
### 3.6 Register Button:

Button that when clicked will complete the registration and save the user's details in the player's table in the database.

### 3.7 Login Button:

Button that when clicked will send the user to the login screen (storyboard 1).

## Storyboard 4 – Locked Account



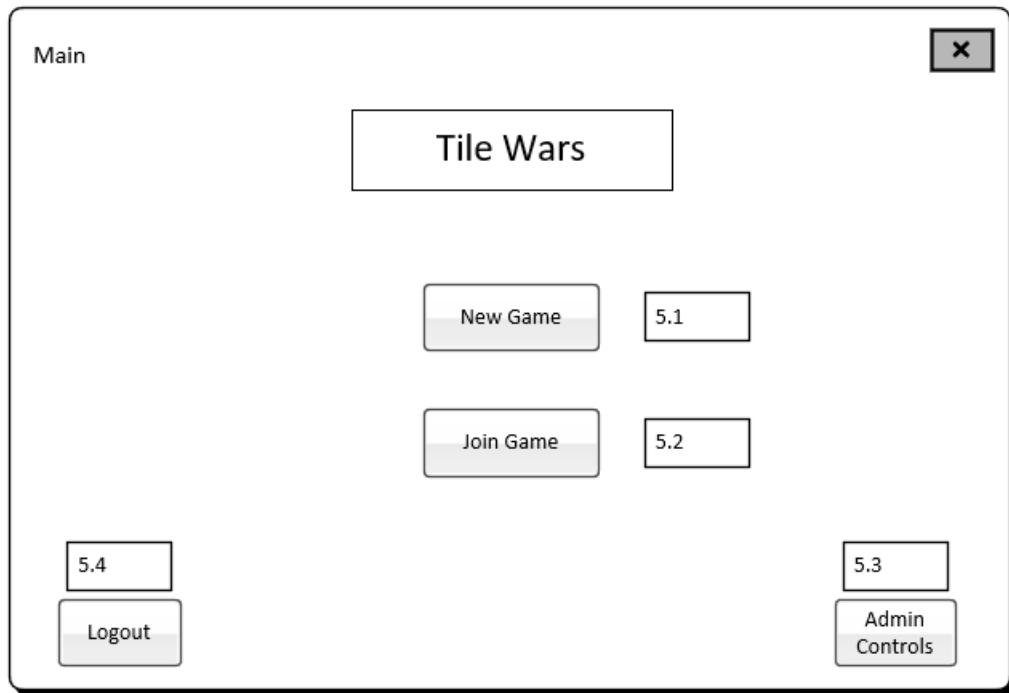
### 4.1 Account Locked Message:

Message that appears when the user's account is locked due to three failed login attempts.

### 4.2 Close Button:

Button that when clicked closes the message.

## Storyboard 5 – Main Window



### 5.1 New Game:

Button that when clicked starts a new game.

### 5.2 Join Game:

Button that when clicked connects the player to an existing game.

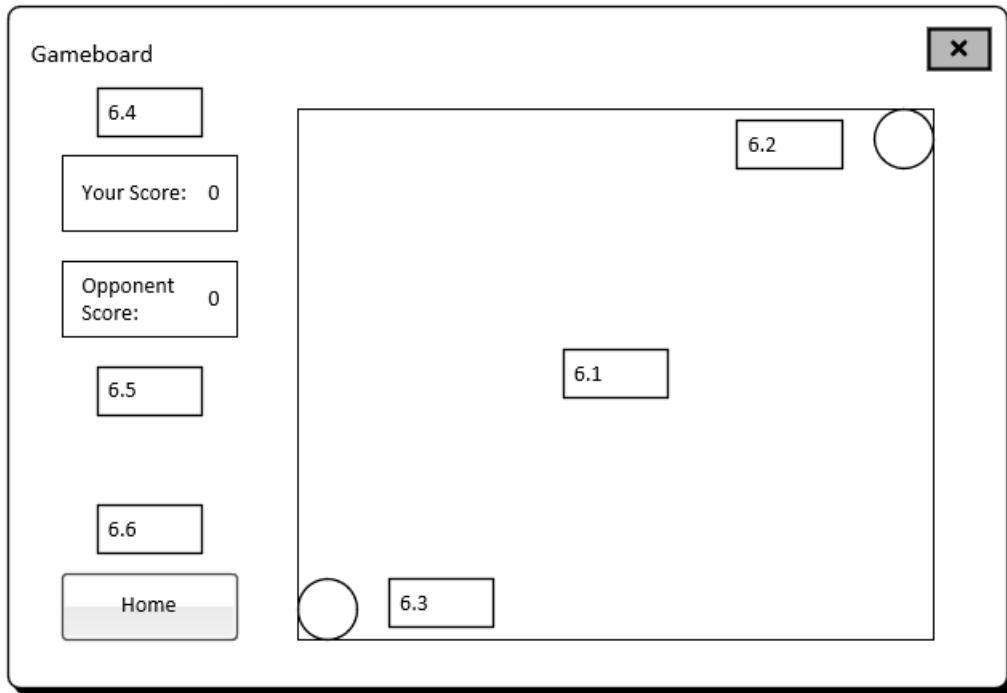
### 5.3 Admin Controls:

Button that when clicked opens the administrative controls. This button will only open the administrative console when the player has administrative permissions.

### 5.4 Logout:

Button that when clicked logs the user out of their account and sends them back to the login screen (storyboard 1).

## Storyboard 6 – In Game Player Gameboard



### 6.1 Gameboard (10x10 Grid):

This is a 10x10 grid of tiles that both players shear.

### 6.2 Player One:

Player One's starting position.

### 6.3 Player Two:

Player Two's starting position.

### 6.4 Your Score:

Displays the current user's score.

### 6.5 Opponents Score:

Displays the current opponent's score.

### 6.6 Home Button:

Button that when clicked sends the player back to the main window (Storyboard 5).

## **Storyboard 7 – Player Win**



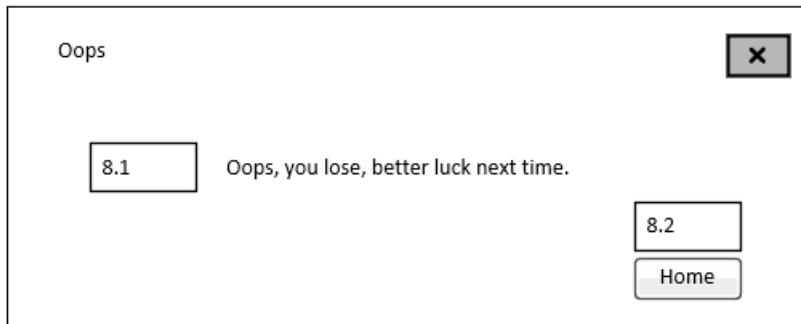
### **7.1 Message:**

Message that tells the player that they have won the game.

### **7.2 Home Button:**

Button that when clicked sends the player back to the main window (storyboard 5).

## **Storyboard 8 – Player Loss**



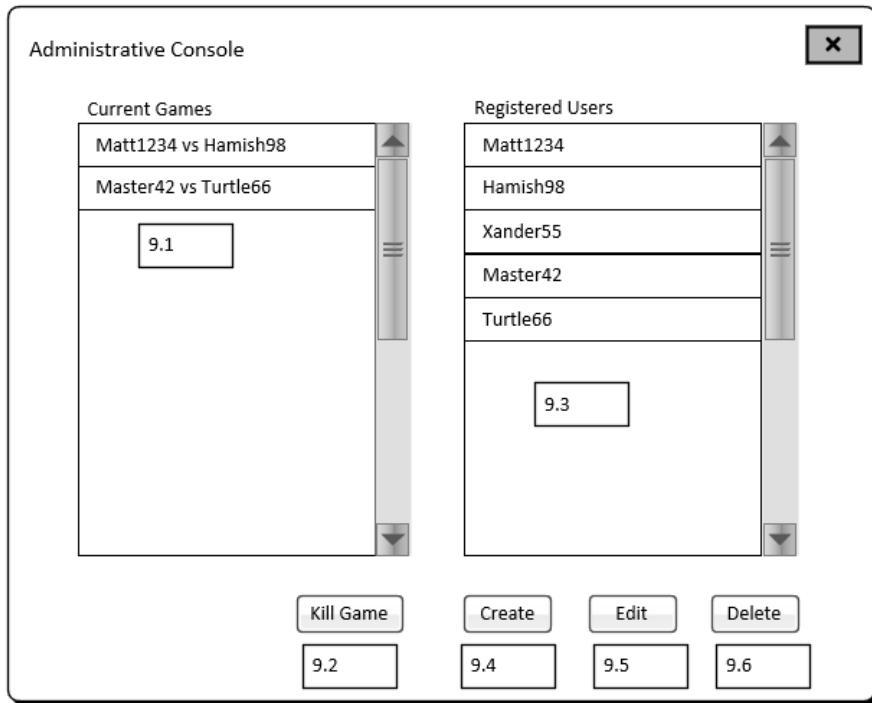
### **8.1 Message:**

Message that tells the player that they have lost the game.

### **8.2 Home Button:**

Button that when clicked sends the player back to the main window (storyboard 5).

## Storyboard 9 – Administrator Screen



### 9.1 Current Active Games:

List box displaying the current active games and the two players in each game.

### 9.2 Kill Current Active Game Button:

Button that when clicked kills the selected game.

### 9.3 Registered Users:

List box of currently registered users in the player's table of the database.

### 9.4 Create Button:

Button that when clicked opens the Administrator Editor screen (storyboard 10). From the editor screen an administrator can create a new user and the new user's details will be created in the player table of the database.

### 9.5 Edit Button:

Button that when clicked opens the Administrator Editor screen for a selected user (storyboard 10). From the editor screen an administrator can edit and update a user's details.

## **9.6 Delete Button:**

Button that when clicked deletes selected registered user.

## **Storyboard 10 – Administrator Editor Screen**

The diagram shows a rectangular window titled "Administration Console". In the top right corner is a close button (X). The window contains eight input fields arranged vertically. Each field consists of a small rectangular box on the left labeled with a number (9.1 through 9.7) and a larger rectangular box on the right labeled with a field name. Below these fields are two status indicators: "Locked" with an empty checkbox and "Administrator" with an empty checkbox. At the bottom right are two buttons: a small one labeled "9.8" and a larger one labeled "Update".

### **10.1 First Name:**

Input where an administrator can enter the user's first name.

### **10.2 Last Name:**

Input where an administrator can enter the user's last name.

### **10.3 Email:**

Input where an administrator can enter the user's email address.

### **10.4 Username:**

Input where an administrator can enter the user's username.

## **10.5 Password:**

Input where an administrator can enter the user's password.

## **10.6 Lock Account Checkbox:**

Checkbox that an administrator can check to lock or uncheck to unlock a user's account.

## **10.7 Administrative Permissions Checkbox:**

Checkbox that an administrator can check to give a user administrative permission or uncheck to remove a user's administrative permission.

## **10.8 Update Button:**

Button that when clicked will update or create the user's details in the player's table of the database.

# **2.1 Storyboard Design Descriptions**

## **2.1.1 Login & Registration Screens:**

The login screen was designed to have a simple reusable layout where users simply need to input their username and password to log into their accounts. When the login button on the login screen is clicked the username and password are then referenced in the database to see if they exist, if so, they will login, if not the application will throw the user an error asking them to register.

The registration screen follows the same simple and reusable layout allowing a user to input the required details to create a new account in the database. Once the user has registered their account, they will be sent back to the login screen to login to their new account.

When a user logs into their account they will be redirected to the main window (home screen).

## **2.1.2 Main Window:**

The main window was designed to be simplistic giving the user only four options to choose from. The user can choose to start a new game being randomly matched with another user doing the same. Secondly, the user has the option to join an existing ongoing game. The last two options are for accessing the administrative functionality and logging out of their account.

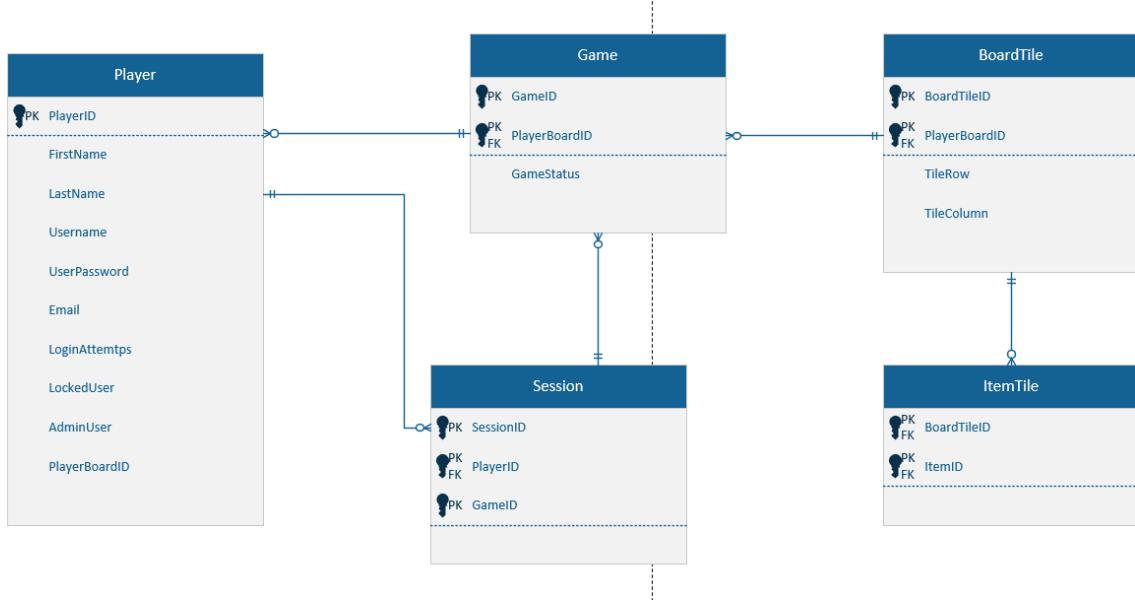
### **2.1.3 In Game Player Gameboard:**

The gameboard design was designed to show the user all the relevant information in a layout in which a player cannot make a mistake. The user can see the gameboard they are playing on (a 10x10 grid), their current score and their opponent's current score. The player also has a way of exiting the game by clicking the home button at the bottom left of the screen. This will take the user back to the main window.

### **2.1.4 Administrator Console & Editor:**

Any user with administrative permissions can access the administrative console by clicking the admin console button at the bottom right of the main window screen. From this console, the administrative user can do one of four things. They can kill a currently running game, create a new user, edit a current user's details, and delete a user from the database.

### 3. Entity Relationship Diagram



#### 3.1 Entity Relationship Diagram Relationships

##### 3.1.1 Player, Session, Game Table Relationship

Because a player can create a new game or join an existing game there needs to be a way to link the player to a game. The session table creates the join between the player and game tables. The game table gives each game an ID and provides a way of determining if a game is active or not.

##### 3.1.2 Game, BoardTile Table Relationship

This relationship between the game and board tile tables allows the database to keep track of which tile each player is on while on the board. As a game board has many tiles there needs to be a foreign key that refers the game table to the board tile table.

##### 3.1.3 BoardTile, Item Table Relationship

There can be more than one item on the board at a time but can only exist on one tile at a time. The item table determines where the item is and whether it has been collected or not and links items to the tiles on the board.

## 4. CRUD Table

		C = INSERT R = SELECT U = UPDATE D = DELETE																					
Table	Player	Player ID	Player Name	Player Email	Player Password	Admin User	Game ID	Game Status	Player ID	Game ID	Board ID	Player ID	Board ID	Game ID	Board ID	Tile ID	Board ID	Tile Row	Tile Column	Game ID	Board ID	Tile ID	Status
Process																							
Login Player	R	R	R	R	R			R	R														
Check Username in DB	R	R	R																				
Check Password in DB	R	R		R																			
Register Player	C	C	C	C	U																		
Login Attempts	RU				U																		
Player Logout	R							U						U									
Admin Lock Out Account	RU	U	U	U																			
Admin Kill Running Game						U		RU	U														
Admin Create New Player	C	C	C	C	C																		
Admin Edit Player Details		U	U	U	U																		
Admin Delete Player Account	R	R	R	R	D			D	D			D	D							D	D	D	
Start New Game	R					C	U	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	
End Current Game	R							U		U													
Join Existing Game	R					C	U	CR	CR	R	R	R	R	R	R	R	R	R	R	R	R	R	
Moving Player	R									RU							RU	RU	RU	RU	RU		

### 4.1 CRUD Analysis

#### 4.1.1 Login Player

If a player logs in successfully they will be directed to the main window of the game where they can start playing by creating a new game or joining an existing game.

#### 4.1.2 Check Username in Database

When a player is logging in their username will be checked in the database to see if it currently exists in the player table. If the username does not exist, the player will be asked to register.

#### 4.1.3 Check Password in Database

When a player logs in with their username their password will also be checked if it exists in the database and matches the username being used. If the user's password does not match the records in the player table of the database, the user's login attempts will be increased by one. When a user's login attempts increase to three their account will be locked and will have to be unlocked by an administrative user.

#### 4.1.4 Register Player

A user can register by inputting the required details. The details will then be cross-referenced to the existing details within the database, if no records are found the account will then be created.

#### **4.1.5 Login Attempts**

When a user's login attempts reach a count of three the account will be locked.

#### **4.1.6 Player Logout**

When a player logs out of their account their details will be removed from any current games. Their player board, items, and status will disappear and become inactive.

#### **4.1.7 (Admin) Lock Out of Account**

Administrative users will have the capability to unlock accounts that have reached three incorrect login attempts from the edit players details screen.

#### **4.1.8 (Admin) Kill a Running Game**

An administrative user can kill a game while it is running and force the players back to the main window.

#### **4.1.9 (Admin) Create a New Player**

A new player can be created by an administrative user. When the new player's details are placed within the required fields the details will be cross-referenced to any existing data in the player table of the database. If there is no record of the user, the player will be registered. If there are any conflicts an error will occur.

#### **4.1.10 (Admin) Edit a Players Details**

From the administrators window an administrative user can edit a player's details; this will update the player table in the database with the user's new information. Before the process is complete the new information will be cross-referenced with any existing data in the database, if any data already exists an error will occur, if not the details will be updated.

#### **4.1.11 (Admin) Delete a Player**

An administrative user can delete a user from the player table in the database.

#### **4.1.12 Start a New Game**

When a player starts a new game the game table will create a new game with a unique ID in the game table. The board tile table will then populate the board table with tiles and items will be placed in random tile locations. When this happens, the player will join the game via the unique ID, and the player board and session tables will then update and will link the player to the game through the game board table.

#### **4.1.13 End the Current Game**

When one of the two players reaches a score count of ten (collected ten items) the game will end resulting in a win or loss.

#### **4.1.14 Join an Existing Game**

If a player joins an existing game the joining player will take over the previous player's score and board information. The board information and score will be retrieved from the existing tables from the previous player and joining them to the new player's game board.

#### **4.1.15 Move Player**

When a player moves around the board from tile to tile the game table needs to be updated to show that the player has moved. If a player moves onto a tile with an item on it the item table needs to be updated to show that a player has collected that item and give them a point for doing so.

## **5. SQL**

Attached SQL file (DAT602\_TileWars\_XanderC\_2023.sql)

### **5.1 DDL**

#### **5.1.1 Player Table**

The player table is used to store the user's registration information.

##### **Player ID:**

The player ID is the primary key and is used to identify users within the database and game. The player ID is also used as a foreign key in other tables. This field is set to auto-increment each user as a new registration is created.

##### **Player First Name:**

The first name field is used to store the user's first name within the database. In the database, this field is set to NOT NULL and is set to VARCHAR(45).

##### **Player Last Name:**

The last name field is used to store the user's last name within the database. In the database, this field is set to NOT NULL and is set to VARCHAR(45).

##### **Player Email:**

The Email field is used to store the user's Email within the database. In the database, this field is set to NOT NULL and is set to VARCHAR(60).

##### **Player Username:**

The username field is used to store the user's username, this is what players will use to recognise other players in game. This field has a unique constraint so that users cannot have the same username. In the database, this field is set to NOT NULL and is set to VARCHAR(45).

##### **Player Password:**

The password field is where users store their password for their account. In the database, this field is set to NOT NULL and is set to VARCHAR(45).

### **Player Login Attempts:**

Login Attempts is a field that checks how many login attempts the user still has. This field is set to have 5 default attempts before their account is locked.

### **Player Lockout:**

Lockout is a field that checks if the user is locked out of their account due to many failed login attempts. This field is set to a Boolean value and is treated as a true or false statement.

### **Player Is Admin:**

Login Attempts is a field that checks if the user is locked out of their account due to many failed login attempts. This field is set to a Boolean value and is treated as a true or false statement.

### **Player Board ID:**

Player board ID is a unique identifier that is used to track which game the user is playing currently. This is used as a foreign key in other tables to identify if a user is in a game, if not the field will be NULL.

## **5.1.2 Game Table**

The game table is used to store information about the game whether it is ongoing and which players are in said game.

### **Game ID:**

This is an auto-incremented identifier that is used as a primary key for each new or existing game.

### **Game Status:**

The game status is used to show if a game is active or not.

### **Player Board ID:**

This is a foreign key used to identify which players are in which game.

### **5.1.3 Session Table:**

The session table is used to link players to games.

#### **Session ID:**

The session ID is the primary key that is auto-incremented.

#### **Player ID:**

This is a foreign key to identify which players are currently in a game or not.

#### **Game ID:**

The game ID is a foreign key that is used to identify which game is currently ongoing.

### **5.1.4 Board Tile Table:**

This table creates the gameboard and lays out a 10x10 grid of tiles.

#### **Board Tile ID:**

This is the primary key for each tile.

#### **Player Board ID:**

This is a foreign key that lets the game know which tile a player is on at any one time.

#### **Tile Row:**

This field creates the tiles for the X-axis of the grid.

#### **Tile Column:**

This field creates the tiles for the Y-axis of the grid.

### **5.1.5 Item Tile Table:**

This table is used to track the items that will be placed on random tiles as the game goes on.

**Item ID:**

The item ID is the primary key for each item that is used in the game.

**Board Tile ID:**

This is a foreign key that determines where the item is on the 10x10 grid.