

**Project Report**

**Amazing Adventures**

*DAT602*

*By Travis Byrman*

Contents

[Milestone One 4](#_Toc68112530)

[Game description 4](#_Toc68112531)

[Idea 4](#_Toc68112532)

[Gameplay 4](#_Toc68112533)

[Login and Registration 4](#_Toc68112534)

[Lobby 5](#_Toc68112535)

[Administration Interface 5](#_Toc68112536)

[Storyboard 6](#_Toc68112537)

[Storyboard 1: Gameplay 6](#_Toc68112538)

[Storyboard 2: Login 7](#_Toc68112539)

[Storyboard 3: Login – Invalid Attempt 8](#_Toc68112540)

[Storyboard 4: Login – Account locked 8](#_Toc68112541)

[Storyboard 5: Login – New Username Detected 9](#_Toc68112542)

[Storyboard 6: Register Form 9](#_Toc68112543)

[Storyboard 7: Lobby 10](#_Toc68112544)

[Storyboard 8: Administrative settings 11](#_Toc68112545)

[Storyboard 9: Administrative player settings 12](#_Toc68112546)

[Storyboard 10: Player settings 13](#_Toc68112547)

[Storyboard 11: Global chat 14](#_Toc68112548)

[Storyboard 12: Character creation 14](#_Toc68112549)

[Storyboard 13: New game form 15](#_Toc68112550)

[Storyboard 14: Error prompt 15](#_Toc68112551)

[Design Choices 16](#_Toc68112552)

[Logical Entity Relationship Diagram 17](#_Toc68112553)

[ERD Analysis 18](#_Toc68112554)

[tbl\_tileID → tbl\_tileAsset 18](#_Toc68112555)

[tbl\_tile → tbl\_character 18](#_Toc68112556)

[tbl\_character → tbl\_backpack 18](#_Toc68112557)

[tbl\_character → tbl\_game 18](#_Toc68112558)

[tbl\_game → tbl\_tileAsset 18](#_Toc68112559)

[tbl\_player → tbl\_chat 18](#_Toc68112560)

[tbl\_player → tbl\_game 18](#_Toc68112561)

[tbl\_game → tbl\_character 18](#_Toc68112562)

[tbl\_item → tbl\_backpack 18](#_Toc68112563)

[tbl\_item → tbl\_tileAsset 18](#_Toc68112564)

[CRUD Diagram 19](#_Toc68112565)

[CRUD Analysis 20](#_Toc68112566)

[SQL Script 22](#_Toc68112567)

[DDL Script 22](#_Toc68112568)

[tbl\_tile 22](#_Toc68112569)

[tbl\_tileAsset 22](#_Toc68112570)

[tbl\_item 22](#_Toc68112571)

[tbl\_character 22](#_Toc68112572)

[tbl\_backpack 23](#_Toc68112573)

[tbl\_player 23](#_Toc68112574)

[tbl\_chat 23](#_Toc68112575)

[tbl\_game 23](#_Toc68112576)

# Milestone One

## Game description

### Idea

I plan to create a single player multicharacter game where characters move around the map via tiles, collecting items and competing with other characters. To achieve this, I will be using a mySQL database along with C# to create an intuitive frontend and efficient backend.

### Gameplay

* The aim of the game is to collect as many items as possible by moving around a tiled map. While doing this, characters must cautiously move around the map, avoiding trapped tiles. If a character moves onto a trapped tile the character will die and lose the game.
* Upon entering into the game, the character will start on the home tile (alternative tile if taken by an existing character).
* Each tile has a maximum of one item or trap. Items and traps are spawned randomly onto tiles.
* The overall game will end when all the characters have quit or died.
* Characters can move up, down, left, or right using arrow keystrokes or the buttons provided.
* If a character leaves the game and the game is still open, their progress will be saved, and they can resume.
* If a tile is occupied by a character, another character cannot be on the same tile.
* When a player tries to join a game, they will be asked to configure their character before joining.
* The game has a maximum time duration. If the game has not been completed before the time runs out, the character scores are calculated, the character with the highest score wins the game.
* Certain players can have administrator abilities. These abilities do not give the player a gameplay advantage, but give the player the ability to manage/monitor other players.

### Login and Registration

* Upon logging in, if a player tries to log in with a username that is not established in the database, they will be given the option to register.
* If the player does have an existing account, they will be given the option to try to login again with the correct credentials.
* If the player attempts to login with a valid username, but the password is incorrect, they will be given 5 attempts before the account will be locked.
* If a player account is locked the player will have to ask an Administrator to unlock the account.

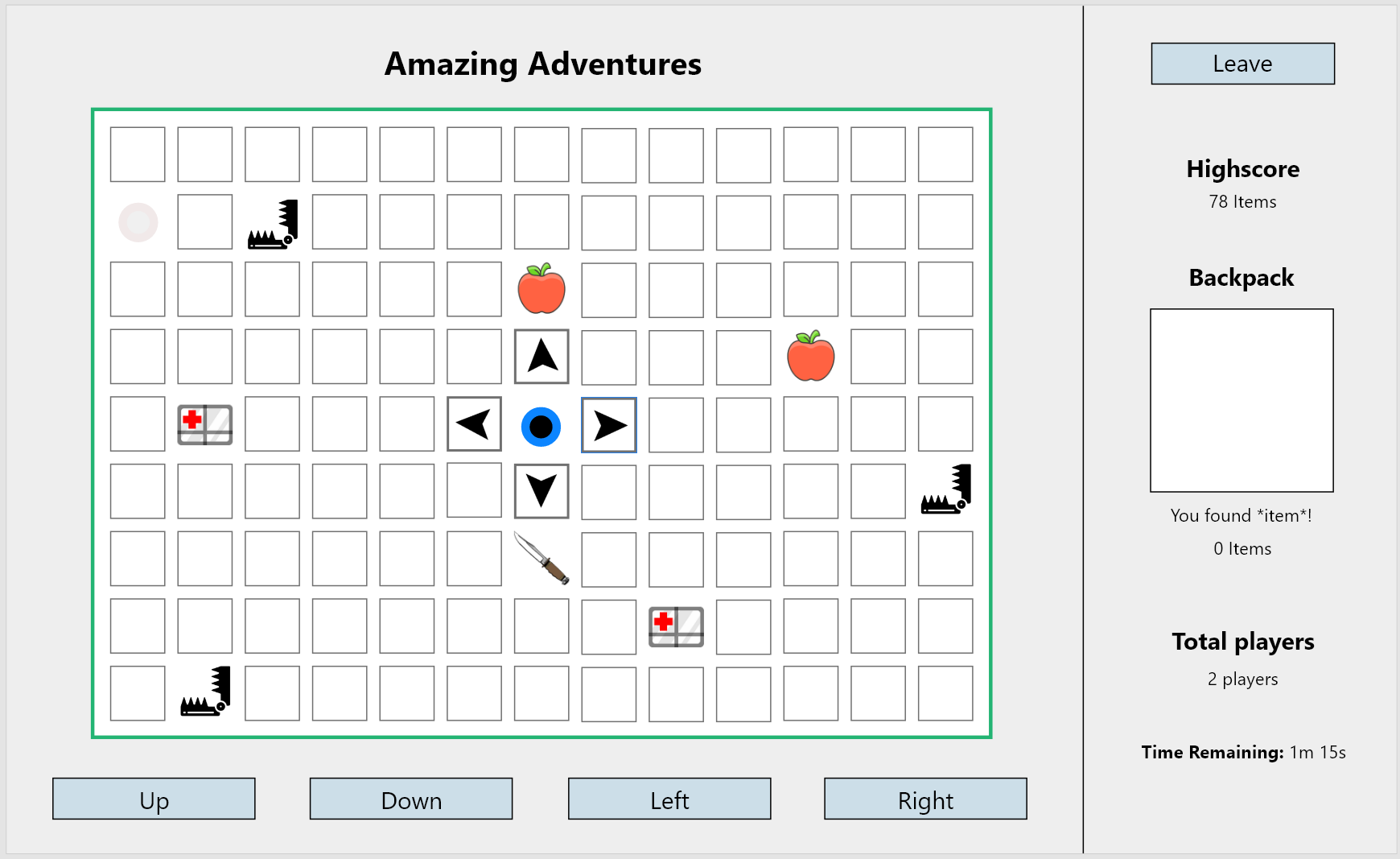
### Lobby

* Inside the lobby, players will be shown the total amount of players and available games they can join.
* Global chat is also available for any player to use.
* Players can also create a game where other players can join.
* If a player is an administrator, they will have access to admin settings.
* Players also have access to their settings, where they can change their username, password etc.

### Administration Interface

* After selecting the admin console, administrators will be shown a total list of players and active games.
* From here administrators can lock and change player accounts and delete active games.
* Only certain players will be given administrator abilities.

## Storyboard

Storyboard 1: Gameplay

1.2

1.9

1.6

1.4

1.8

1.0

1.7

1.5

1.3

1.1

1.0 **Leave button** - Click to exit the game. Player is returned to lobby.

1.1 **Trap** – If a character moves to this tile they die and can no longer play.

1.2 **Other character** – This is another player.

1.3 **Current character** – This is the character that the player is in control of.

1.4 **Latest item** – This shows your total items as well as the latest item you have found.

1.5 **Item** – This is an item, if a character lands on this tile, they collect the item.

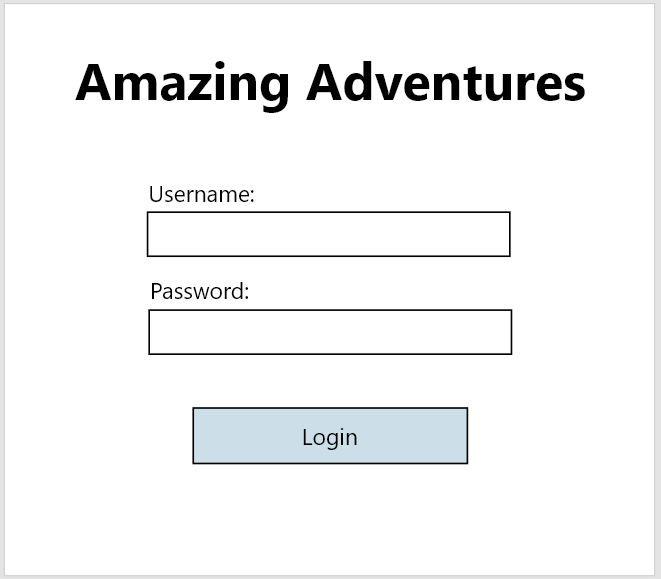
1.6 **Total player count** – This shows the total players in the game.

1.7 **Character movement** – These buttons can be used to move the character.

1.8 **Highscore** – This shows the user their highest score that have received.

1.9 **Time remaining** – This shows the player how much time is left.

### Storyboard 2: Login



2.0 **Username field** – Input field where a user inputs their player account username.

2.1 **Password field** – Input field where a user inputs their player account password.

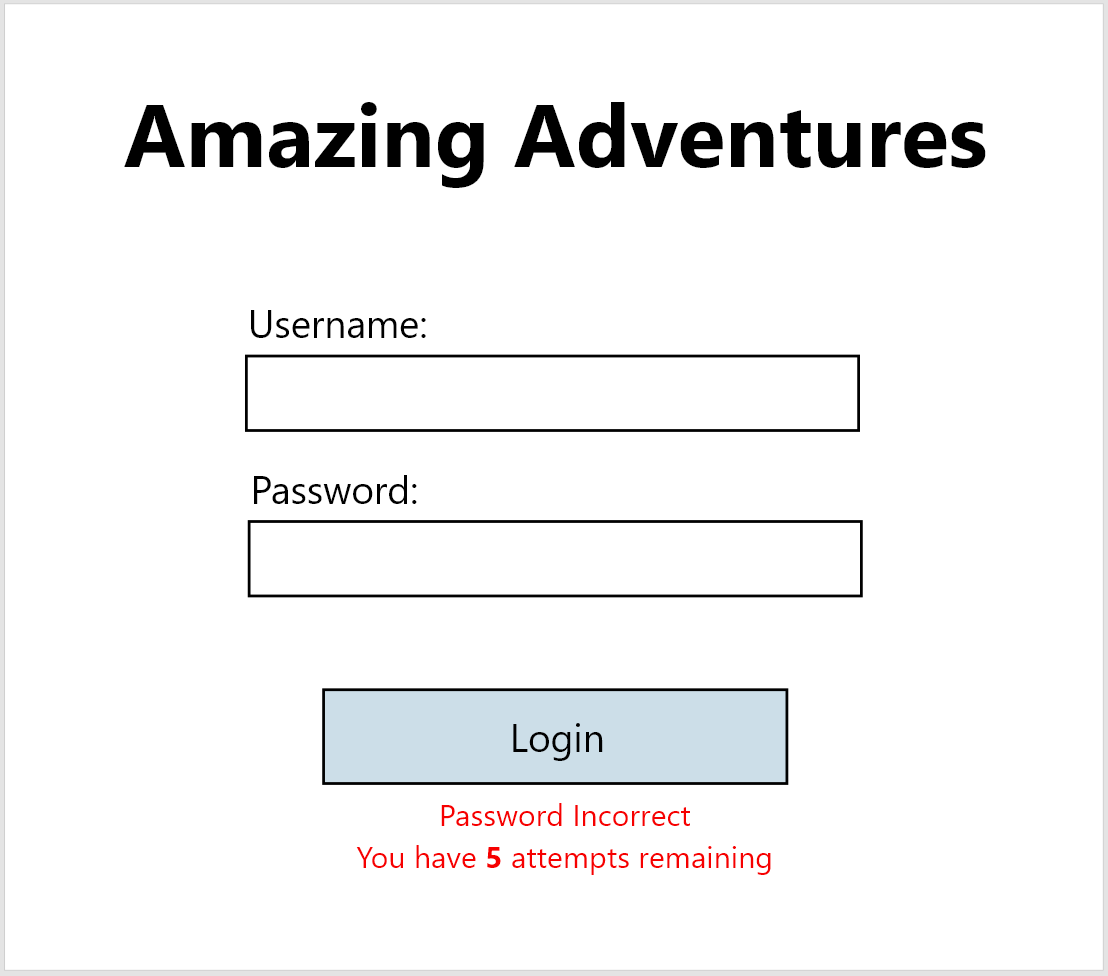
2.2 **Login button** – User clicks this button to login. If username or password is invalid login will be unsuccessful. If username is not a valid player account username, user is given the option to register. User has 5 login attempts before account is locked.

2.2

2.1

2.0

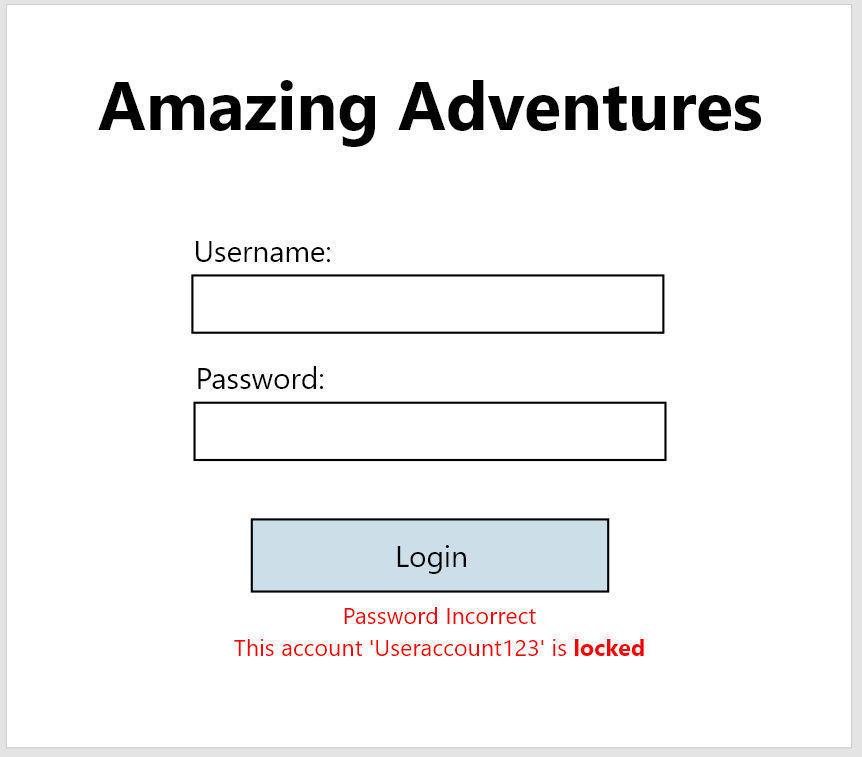
### Storyboard 3: Login – Invalid Attempt



3.0 **Prompt** – Indicates user how many login attempts remaining.

3.0

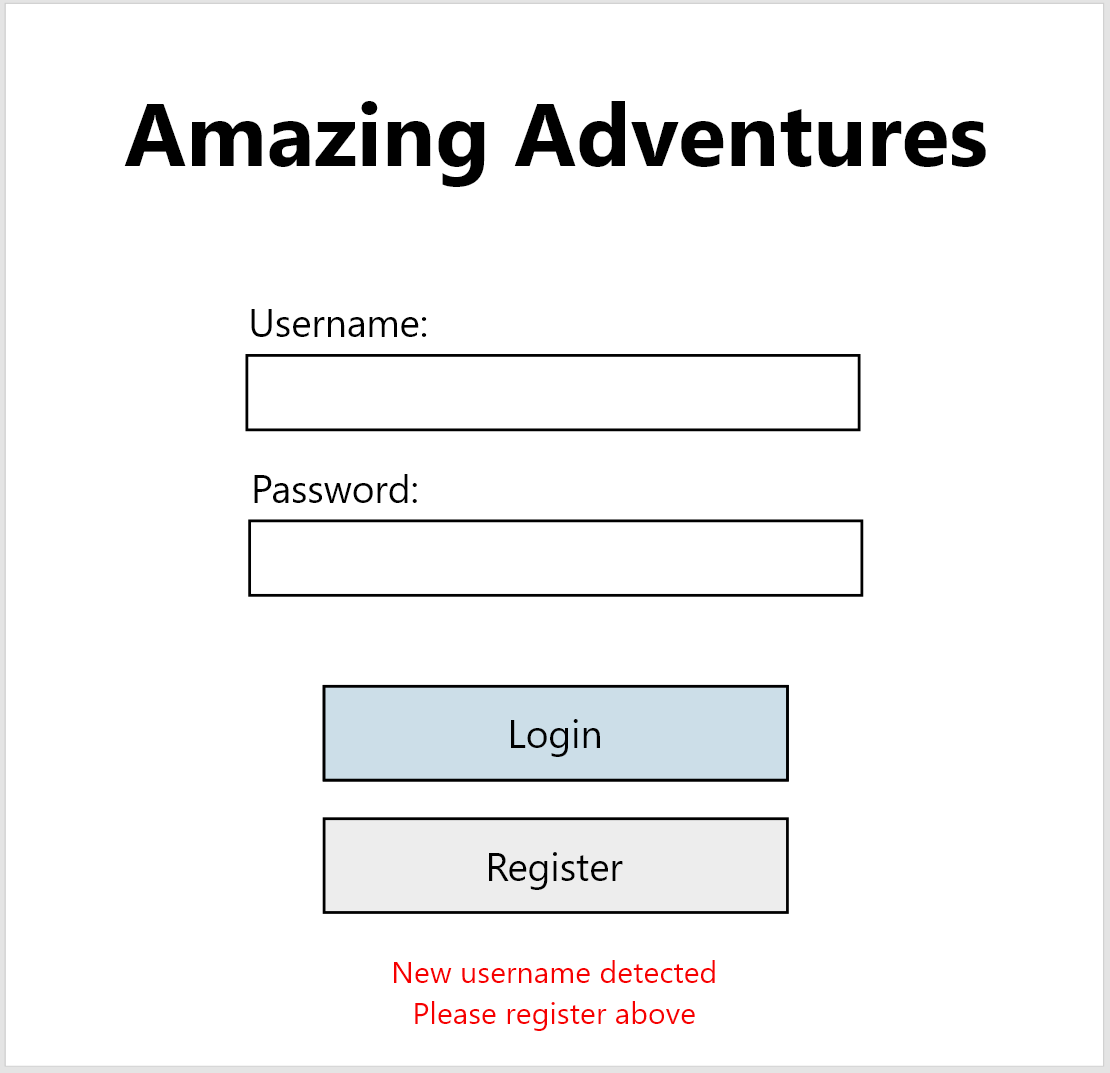
### Storyboard 4: Login – Account locked



4.0 **Prompt** – Indicates to user that the account they tried to log into has now been locked.

4.0

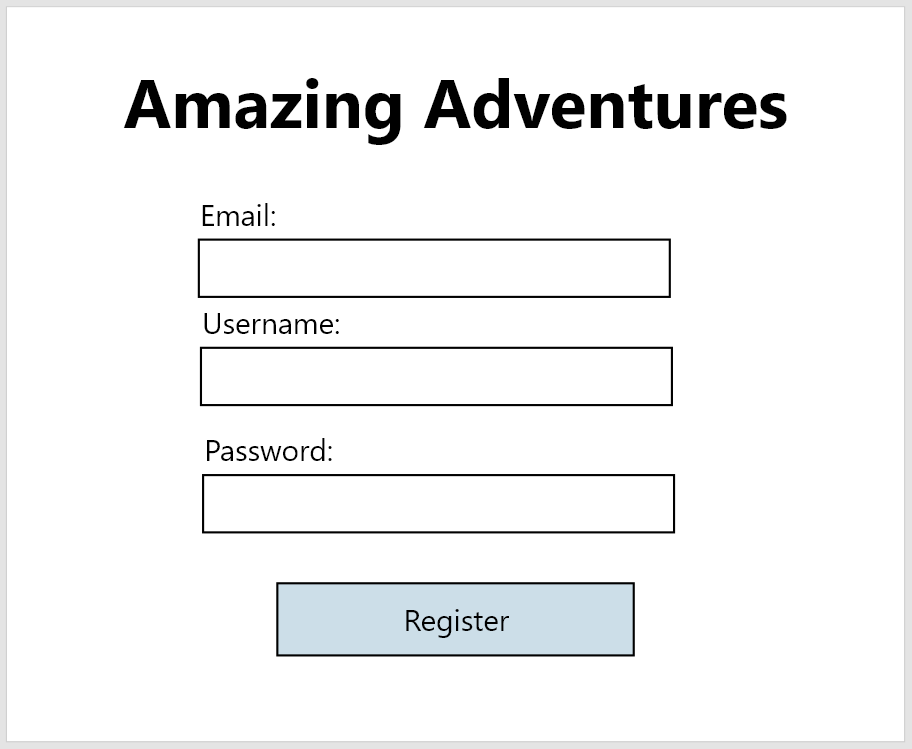
### Storyboard 5: Login – New Username Detected



5.0 **Prompt** – Indicates to user that the username they entered is not recognised in the database. Registration button is shown, given the user the option to register a player with that username.

5.0

### Storyboard 6: Register Form



6.0 **Email field** – Input field where a user inputs their player account email address.

6.1 **Username field** - Input field where a user inputs their player account username.

6.2 **Password field** - Input field where a user inputs their player account password.

6.3 **Register button** – User clicks this button to register the account. If username, password, or email address is invalid, registration will be unsuccessful.

6.3

6.2

6.0

6.1

### Storyboard 7: Lobby



7.7

7.5

7.4

7.3

7.2

7.6

7.8

7.1

7.0

7.0 **Current games list** – This shows the current games that players can join.

7.1 **Online players list** – This shows the list of players that are online.

7.2 **High-score** - This shows the players high-score (most items they have collected in a game).

7.3 **New game buttons** – Players can press this button to create a new game.

7.4 **Chat button** – Players can press this button to open the global chat.

7.5 **Setting buttons** – Players can press this button to view their player settings.

7.6 **Join game button** – Players select a game from the current games list and press this button to join it.

7.7 **Admin console button** – Players can press this button open administrator settings. Only players with administrative access can see this button.

7.8 **Logout button** - Players can press this button logout of their account. User is returned to login screen.

### Storyboard 8: Administrative settings



8.1

8.2

8.4

8.3

8.0

8.7

8.5

8.6

8.0 **Active games list** – This shows the current games that players can join, as well as the total player count to the left.

8.1 **Total player list** – This shows the list of player accounts created.

8.2 **Player settings button** – Administrators can press this button after selecting a player from the player list to view that players account settings.

8.3 **Clear chat button** – Administrators can press this button to remove all messages from the global chat.

8.4 **Refresh button** – Administrators can press this button to refresh the total player and active games lists.

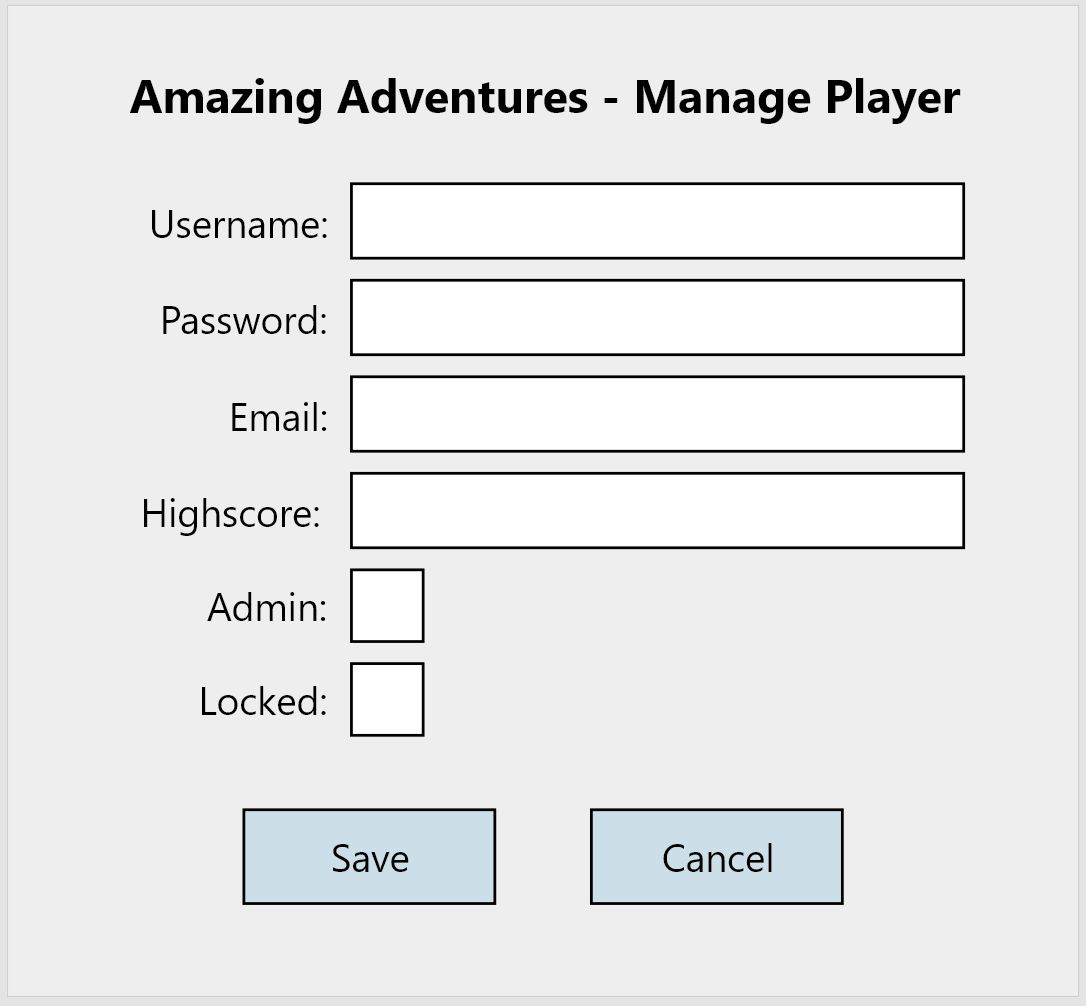
8.5 **Close game button** – Administrators can press this button after selecting an active game from the list to close/remove the game.

8.6 **Return to lobby button** – Administrators can press this button to exit the administrator settings and return to the lobby.

8.7 **Delete button** – Administrators can press this button after selecting a player account from the list to delete the player account.

**Storyboard 8: Administrative settings**

### Storyboard 9: Administrative player settings



9.0 **Username field** – Input field where a user inputs their player account username.

9.1 **Password field** – Input field where a user inputs their player account password.

9.2 **Email address field** – Input field where a user inputs their player account email address.

9.3 **Highscore field** – Input field where a user inputs their player account highscore.

9.4 **Admin tick box** – Tick box field where a tick represents the user has administrative access.

9.5 **Locked tick box** – Tick box field where a tick represents the players account is locked.

9.6 **Save button** – Administrators click this button to save the players details. If details are incorrectly added, administrator is prompted.

9.7 **Cancel button** – Administrators click this button to cancel the players detail form. This will close the form and return the admin to the administrative settings.

9.6

9.7

9.5

9.4

9.3

9.2

9.1

9.0

### Storyboard 10: Player settings



10.0 **Username field** – Input field where a user inputs their player account username.

10.1 **Password field** – Input field where a user inputs their player account password.

10.2 **Email address field** – Input field where a user inputs their player account email address.

10.3 **Save button** – Players click this button to save their players details. If details are incorrectly added, player is prompted.

10.4 **Cancel button** – Players click this button to cancel their players detail form. This will close the form and return the player to the lobby.

10.3

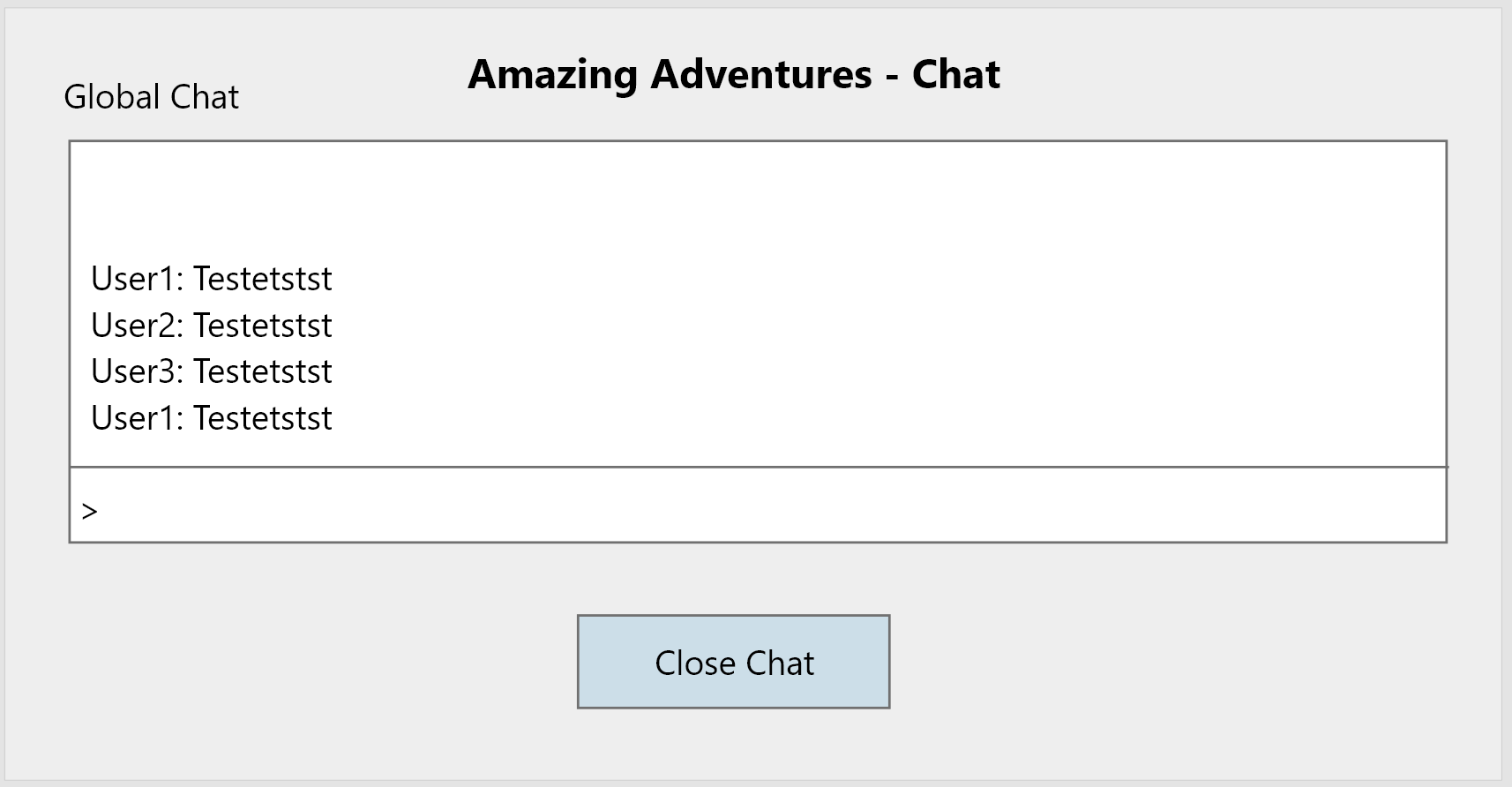
10.4

10.2

10.1

10.0

### Storyboard 11: Global chat



11.0 **Global chat text** – Players will see every player’s text chat and date/time of publish.

11.1 **Global chat input box** – Input field where a user inputs the text they wish to add to the chat. Once finished, press keystroke enter to submit.

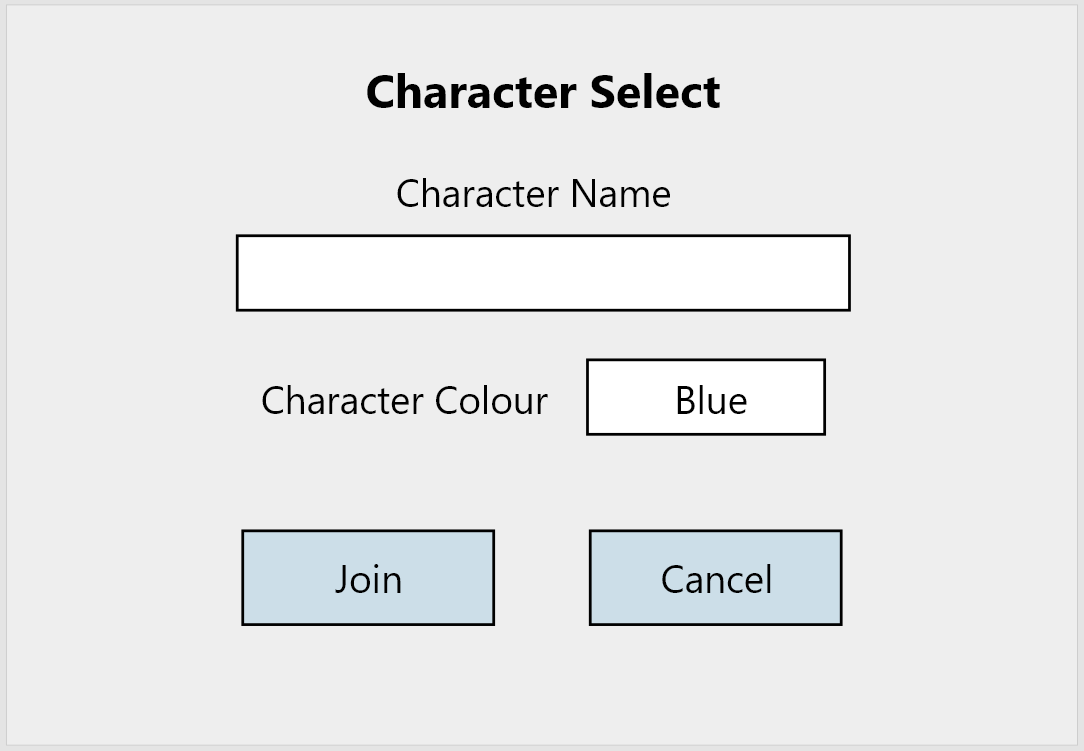
11.2 **Close chat button** – Players click this button to close the global chat form. This will close the form and return the player to the lobby.

11.0

11.1

11.2

### Storyboard 12: Character creation



12.0

12.1

12.3

12.2

12.0 **Character name input box** – Input field where a player inputs their character name.

12.1 **Character input colour** – Input field where a player chooses their character colour.

12.2 **Join button** – Players click this button to join the game. This will close the form and will add the player to the game.

12.3 **Cancel button** – Players click this button to close character creation form. This will close the form and return the user to the lobby.

### Storyboard 13: New game form



13.3

13.2

13.1

13.0

13.0 **Game name input box** – Input field where a player inputs their game name.

13.1 **Game duration input box** – Input field where a player chooses their games time/duration.

13.2 **Create button** – The player clicks this button to create the game. This will close the form and the player will be asked to create a character.

13.3 **Cancel button** – Players click this button to close the new game form. This will close the form and return the user to the lobby.

### Storyboard 14: Error prompt



14.0

13.0 **Ok button** – Players click this button to acknowledge the error.

When an error occurs, this form will pop up notifying the user of their error. The message in red will change depending on the error.

## D**esign Choices**

I choose to have a colour theme that mainly uses the colours grey, white and black. I decided to do this as I found the colours work well with each other and make the features stand out. I found that by using the colour blue and red to show errors and buttons make those features stand out against the grey background. These colours draw the user’s attention.

As shown in the gameplay screenshot above, I decided to add buttons that move the player around the map. I chose to do this as it allows users who have a touch screen device to play the game. I also made the text and buttons large enough so that people with disabilities and can easily view them. It also allows a large variety of devices to be able to play this game without any difficulties due to resolution.

I have also decided to use combo box when necessary (such as for character colours) as it limits the users range of how many colours they can choose.

## Logical Entity Relationship Diagram

## ERD Analysis

### tbl\_tileID → tbl\_tileAsset

The tile table contains the location of each tile. Each tile asset needs to be on a tile. One tile can only have one asset, one asset can only be on one tile.

### tbl\_tile → tbl\_character

The tile table contains the location of each tile. A character needs to be on a tile. One tile can only have one character, one character can only be on a tile.

### tbl\_character → tbl\_backpack

The character table contains information about the character, such as character name, colour, and score. A character can have multiple items inside the backpack table. A character can have only one item at a move, only one item can be claimed by a character on a tile.

### tbl\_character → tbl\_game

The character table contains information about the character, such as character name, colour, and score. A character can only be in one game, one game can have many characters.

### tbl\_game → tbl\_tileAsset

The game table contains information about the game, such as who created the game, game duration and name of the game. One game can have many tile assets, many tile assets can be in one game.

### tbl\_player → tbl\_chat

The player table contains information about the player, such as username, password, email is it an admin etc. Many players can make submit many texts in the chat table. Many texts can be submitted by many players.

### tbl\_player → tbl\_game

The player table contains information about the player, such as username, password, email is it an admin etc. Only one player can create a game, the same game can only have one creator (player who creates that exact game).

### tbl\_game → tbl\_character

The game table contains information about the game, such as who created the game, game duration and name of the game. One game can have many characters, many characters can only be in one game.

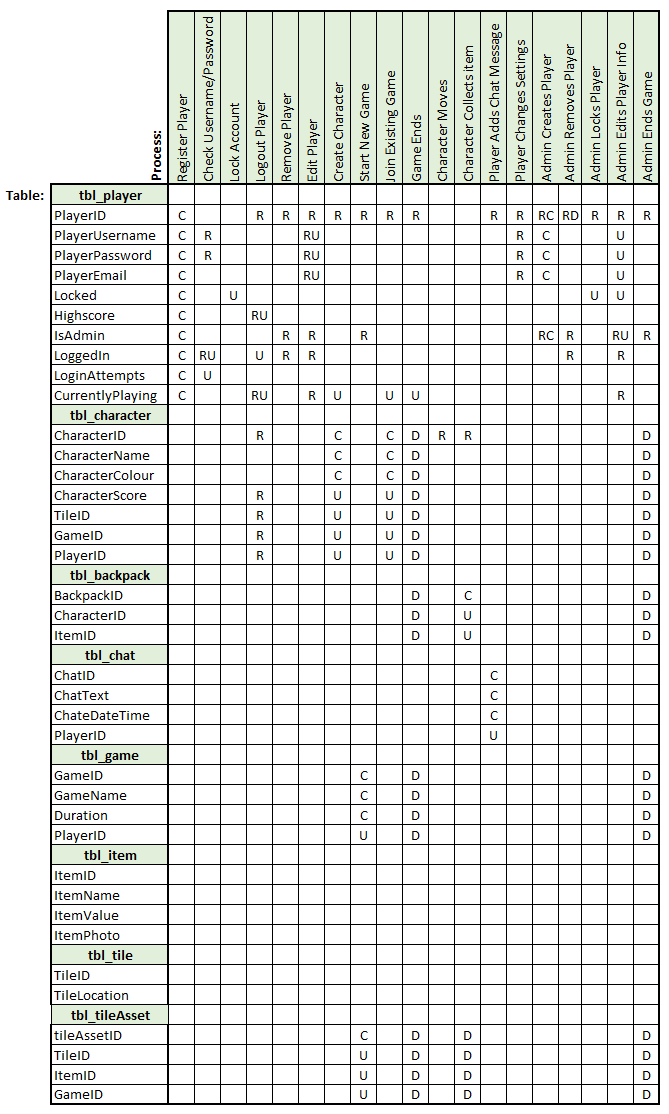
### tbl\_item → tbl\_backpack

The item table contains information about the items, such as item name, item value and item photo. Many items can be in many backpacks, many backpacks can have many items.

### tbl\_item → tbl\_tileAsset

The item table contains information about the items, such as item name, item value and item photo. The same item can be many tile assets, many tile assets can have the same item.

## CRUD Diagram



## CRUD Analysis

By creating the logical ERD diagram first, I was able to use it as a reference to create the tables and fields for my CRUD diagram. After analysing the assessment requirements and schedule, as well as my storyboards, I would be able to create the processes needed. Here is what I found.

**Register Player**

When a player enters a username into the login fields, that is not recognised as existing in the database, the user will be prompted. A register button will appear, as well as a label explaining to the user that they can create and account if they do not have an existing account.

After clicking the register button, a registration form will show, getting the users information to create an account.

**Check username/password**

When a user submits the login forms, checks will be made with select statements to decide whether the username and password that has been entered is correct. If not correct and the username is recognised, the user will be prompted, as well as the amount of login attempts, they have remaining to access the account.

**Lock account**

If the user has tried 5 times to log into the same account, the account will become locked. This means that any user will be unable to log into this account without having an administrator unlock it. Once unlock users will have another 5 login attempts until the same thing occurs again.

**Logout player**

If a user logs out of their account, by clicking the logout button in the lobby, the lobby form will close, and the user will be presented with the login form. The account will no longer be shown to be logged in.

**Remove player**

If an administrator decides to delete another user’s player account, they will be asked to confirm this action, then the account will be removed. If the user is currently logged in, they will be logged out immediately. If a user wants to delete their player account they will be prompted to confirm, then will be logged out.

**Edit player**

If an administrator wishes to change the settings of another player, they can do so by choosing a player from the player list in the administrator settings. A button will only be visible to administrators on the lobby form. After an admin selects the user, they can press player settings and change the players settings.

**Create character**

When a player wants to join a game, they will be prompted to create a character. Here they can choose their characters name and colour before entering. Once entered the game will be shown, and they character will have the corresponding features.

**Start new game**

When a player chooses to create a new game, they will be prompted with a form and will need to decide on the games name and duration. If incorrectly submitted, the user will be promoted with the corresponding error message.

**Join an existing game**

When a player joins an existing game, they will be prompted to create a character (explained above). Then they will have three seconds before the game starts for themselves.

**Game ends**

When the game ends the player will be given their score. If the score is greater than their highscore, then this new score will be their highscore. Once the game ends, all characters are deleted, and players are returned to the lobby. The game will no longer exist, and players will not be able to join.

**Character moves**

A player can move their character by using the keystroke arrows or buttons provided on the game play. When moved the database will save the players current location and add any items to their backpack if an item was on the tile.

**Character collects item**

When a character moves to a tile with an item on it, the item will be removed from the tile and added to their backpack. Depending on the item, the number of points will be determined by it.

**Player adds chat message**

When a player creates a chat message, it will be shown in the global chat where every player can read it. The players name, chat text and date/time will be provided. Chat messages can be cleared at admins choice in the administration settings.

**Player changes settings**

If a user wishes to edit their player account details, they can do this by clicking the settings button in the lobby. Here they will be given access to edit the details of their player account. After clicking the save button, the player details will be saved. If incorrectly inputted, the user will be prompted, and the corresponding error message will be shown.

**Admin creates player**

If an admin wishes to create a player account, they can do so in the administrator settings. After clicking the button, they will be shown a form where they can enter the details. If the details are incorrect the admin will be promoted the account will not be created until errors are corrected.

**Admin removes player**

If an admin wishes to delete a player than can in the administrator settings by selecting the player and clicking delete.

**Admin locks player**

If an admin wishes to lock another player account, they can do so in the administrator settings. This will make the account unavailable to any user.

**Admin ends game**

If an admin wishes to delete/end a game they can do so in the administrator settings by selecting the active game and clicking close game. When they do this, all characters in the game will be deleted and the players will be prompted, that the game ended by an admin. The scores will be shown, and they will be returned to the lobby.

## SQL Script

Separate file attached titled AssessmentD DL.sql

## DDL Script

### tbl\_tile

TileID – Primary key that is an auto-incremental number. This is used to link the tiles to the tileAsset table, and character table.

TileLocation – Integer, used to hold the corresponding value in which the tile can be found in an array.

### tbl\_tileAsset

tileAssetID – Primary key that is an auto-incremental number.

This also table consists of three foreign keys, TileID, ItemID and GameID. This table is used identify where and item is, what it is, in what game.

### tbl\_item

ItemID - Primary key that is an auto-incremental number. This is used to uniquely identify each item and to link the item to the tileAsset table. This is nalso used to link the items to the backpack table.

ItemName – Varchar field, used to identify each item by name.

ItemValue – Integer field, used to hold the value of each item.

ItemPhoto – Varchar field, used to hold the link to the item photo in the database.

### tbl\_character

CharacterID - Primary key that is an auto-incremental number. This is used to identify each character. This is also used to link the character table to the backpack.

CharacterName – Varchar field, used to identify each character by name.

CharacterColour – Varchar field, used to give each character a colour.

CharacterScore – Int field, used to hold the characters score.

CharacterLocation – Varchar field, used to store the characters current location in the database.

There are also three foreign keys, TileID, GameID and PlayerID. These are used to link the tile, game and player tables to the character.

### tbl\_backpack

BackpackID - Primary key that is an auto-incremental number.

CharacterID – Used to link the character to a backpack that is held by an item.

ItemID – Used to link an item to the backpack that is held by a character.

### tbl\_player

PlayerID - Primary key that is an auto-incremental number. This is also used to link the player to the character that they create per game. It is also used to link the player to a game they create. Chat messages from the chat table are also linked to the PlayerID

PlayerUsername – Varchar field, this is used for users to log into their player account.

PlayerPassword Varchar field, this is used for users to log into their player account securely.

PlayerEmail – Varchar field, this is used for users to associate themselves to their player account securely.

Locked – Boolean, used to identify when the player account is locked or not.

Highscore – Int, used to store the highest score a player gets using their character.

IsAdmin – Boolean, used to identify whether the player has administrator access/powers.

LoggedIn – Boolean, used to identify whether the player is online.

LoginAttempts – Int, used to identify how many times the user has attempted to log into their account.

CurrentlyPlaying – Boolean, used to identify whether the player is in a game or not.

### tbl\_chat

ChatID - Primary key that is an auto-incremental number. Used to uniquely identify each chat message.

ChatText – Varchar, used to store the players chat message.

ChatDateTime – Varchar, used to store the date and time that the message was submitted.

PlayerID – Used to link the chat message to the player. Makes the chat message identifiable and traceable back to the player.

### tbl\_game

GameID - Primary key that is an auto-incremental number. Used to link the game table with the tileAsset and character table.

GameName – Varchar, used to identify each game by the name given by the user who created the game.

GameDuration – Int, used to store the duration of the game, how long it needs to run for, and how long the duration is left.

PlayerID – Used to identify which player created the game.