DAT602 Project

Milestone 1

Kenneth-John Williams-Stockdale

NMIT Student ID: 13477293

National Student Number: 133006701

Contents

Milestone 1

Project Outline/Brief

This project is guided and undertaken throughout the course. To design and implement a database that allows for the creation of a multi- user "point and click" game application based on a tiled map layout. A prototype will be developed.

Game brief

The game is a "point and click" (possibly with "keystroke") controlled game that lets two or more players play against each other at the same time on different computers through the network. Players' turns are to be stored in the database in such a way that the database can be used

to implement live game play. The system that will be implemented is to support the running of multiple copies of the game. When a player is logged in, the system displays a list of other online players and their highest score. Play is started by selecting an opponent who is not currently playing.

Functional requirements

- Players move around the map from one tile to another, collecting items, competing with other players who are playing on the same tiled map. Note they are not running the same install of the App on a device.
- A player is logged in and registered.
- Players move from one tile to another by clicking on neighbouring tiles.
- Players start on the “Home Tile”. Apart from the “Home tile” only one player can be placed on each tile at time. The player who achieves the first “click” on an empty tile moves to that tile. Once on a tile the player can click on the items on that tile to gain or lose points. When a player moves from the tile, the tile becomes empty.
- When a player leaves the game, their current state is kept in the database.
- When the player returns to the game the player, if the tile they were on is currently empty, they return to the tile on the map they were on when they left the game, otherwise the player must choose a different neighbouring tile to continue playing the game.
- Players accumulate assets (inventory) that may be used in the game play. The database keeps track of all the assets and the state of the player
- As the player moves, the position of the player is to be stored in the database.
- As game assets move, the position of the asset is to be stored in the database.
- Players can communicate with other characters through text chat
- A player can “delete” their account.
- A player can be an Administrator

Objective

The game is called “Coin Hunt”. The game objective is to collect as many coins as possible before the opposing player does.

The objective is to move around the game board in the most efficient way possible. When a player moves to a tile either nothing will happen, or a message is displayed stating that the tile has coins on it and how many the player has collected. A tile can have different values of coins on it so players can strategically seek out higher value coins first to increase their score higher quickly. The coins collected will be added to the player inventory to be displayed as the player score.

There will be some obstacles that players must move around to get to the coins. Whichever player collects the last coin ends the round. The player with the most coins at the end of the round wins the game. This will be displayed via pop up message.

Login process

Upon accessing the game for the first time a user is presented with the game application title screen. Pressing the play button will present the user with the login screen. From here the user inputs their username and password. At this point the database checks to confirm that there is an existing record of this username and password. Or a user can register as a new user.

If the username is not found an error message is displayed and users must re-enter their credentials again.

If an existing user inputs their password wrong, they will be presented a message saying they have entered their details incorrectly and have a limited number of attempts. If they fail all the attempts, they will be presented with another message saying they have used up all their attempts and will be taken back to the title screen. When the user tries to login in again with existing credentials/account that has been locked they will be meet with a matching error message.

If both username and password credentials are correct and match the database, then the user will re direct to the home/player lobby.

A new player who has no account can register a new account by selecting an email, username and password which will then be added to the database. If the player has entered a username that's already in the database, then they will be prompted to choose another. From there the player will re direct to the home/player lobby.

Home/player lobby

On successful/confirmed login players are re directed to the applications home/player lobby screen. This displays relevant information such as the players in the lobby their high scores and current games. Other navigation buttons are also present. These include buttons for the "Account"," Admin", "Logout", "Join game", and "New Game".

Users can if they so choose to join an existing game with the join game button or they may choose to start a new game by picking the new game button.

If a user clicks on the account button, they can access to limited options with their account they can delete their current account and reset their password.

If a user decides to delete their account, there will be a prompt asking the user to confirm their choice. This is to ensure no slip click or accidental deletion. Once deleted the user will be redirected to the title page.

Should a user need to reset their password they can access this functionality through the account manager and follow the prompt to input a new password to update the database. The user will then be redirected to login with the updated details.

Users can check if they have admin access. Once the button is pressed the database is checked to confirm that the user has admin rights. If they do, they are re directed to the admin screen. If the user does not have admin rights, they are shown a pop-up message stating they are not administrators.

The logout button logs out users back to the title screen.

Administrators

Users can have administrator access and rights. When in the game/player lobby screen a user can click the admin button, and this will check their access level against the database and if admin status is true the admin screen is displayed. Admin options are to stop all gaming sessions which immediately stops all gaming sessions. A prompt will display to let the user confirm the action about to take place.

Other admin functionality is the ability to add a new user, update the details of an existing user or delete a user.

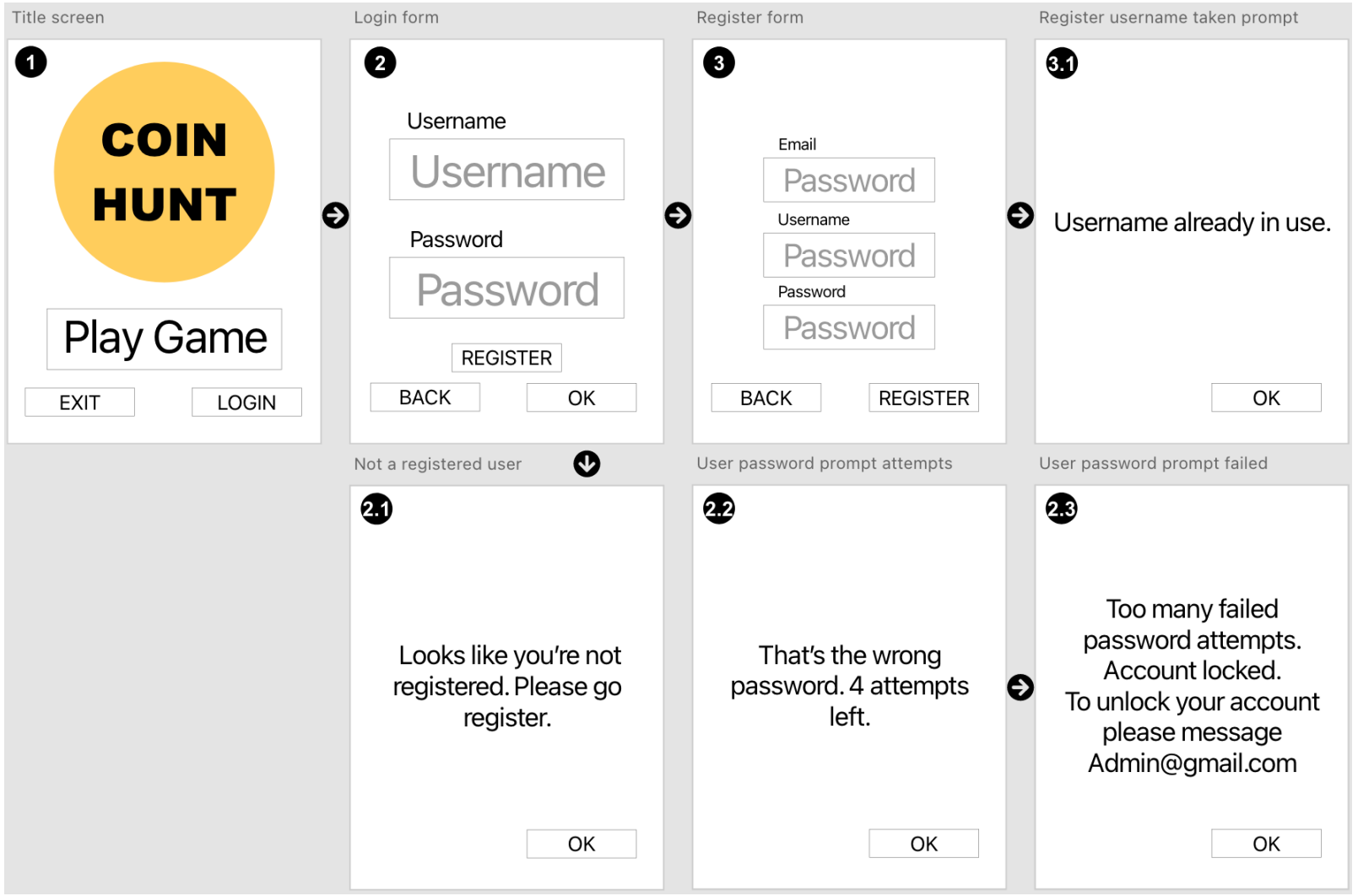
The add a user button screen follows the examples of the login process as previously mentioned adding a user to the database.

The update user button will follow suit in a similar nature but will read information and then update as per the admins request. There are extras fields that will need or optionally can be filled out that detailed in the story boards below.

The delete button follows the example of the delete functionality in the player account section where a prompt will be presented asking to confirm the action before actioning.

Storyboards and descriptions

Login and registration

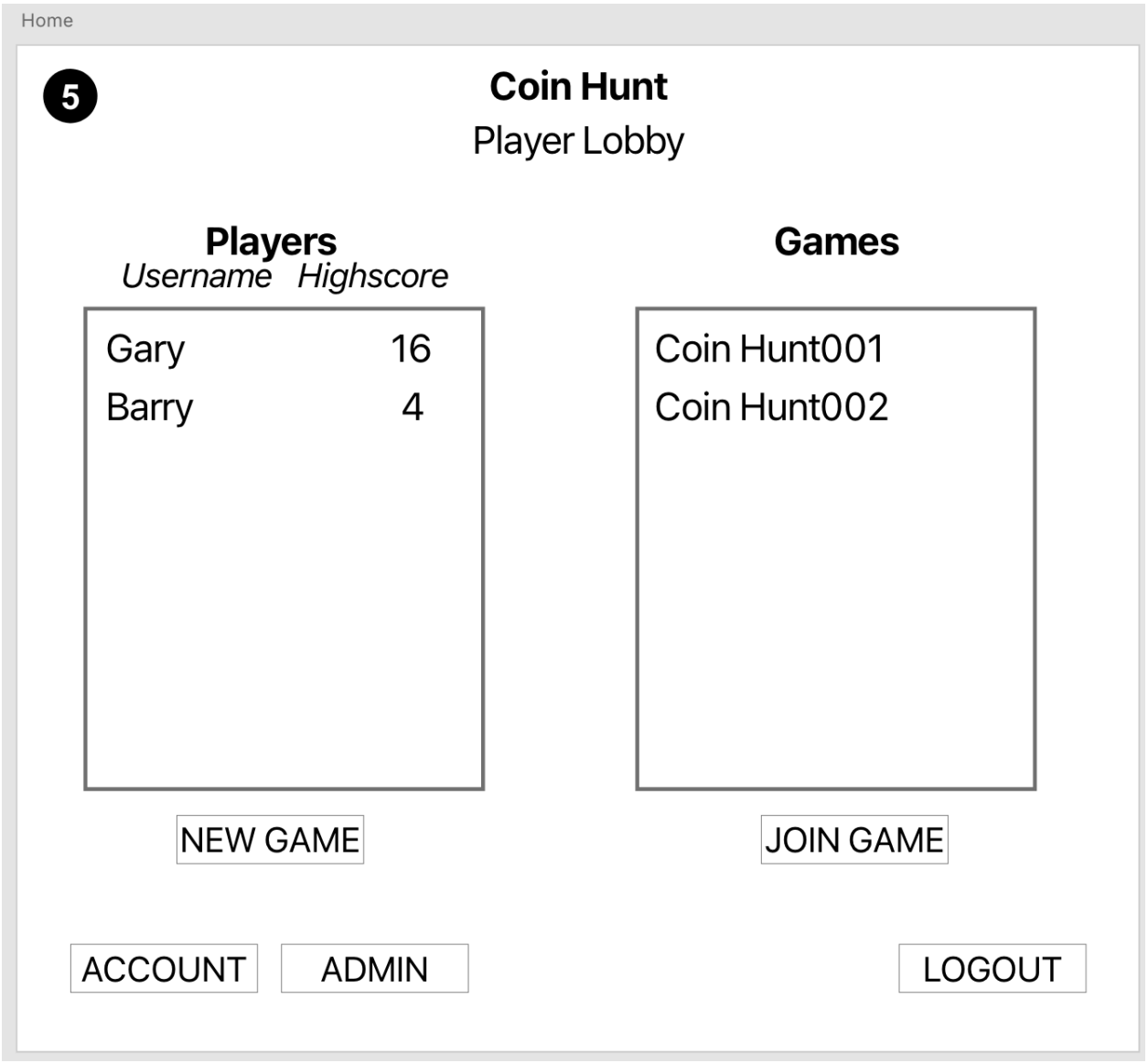


These are the first set of screens that the user will encounter upon launching the application. The 1st screen to the 3rd screen shows the basic flow on how the user will interact and either login or register within the app. These screens have multiple input fields and options to use.

The designed / intended flow of the application is as follows:

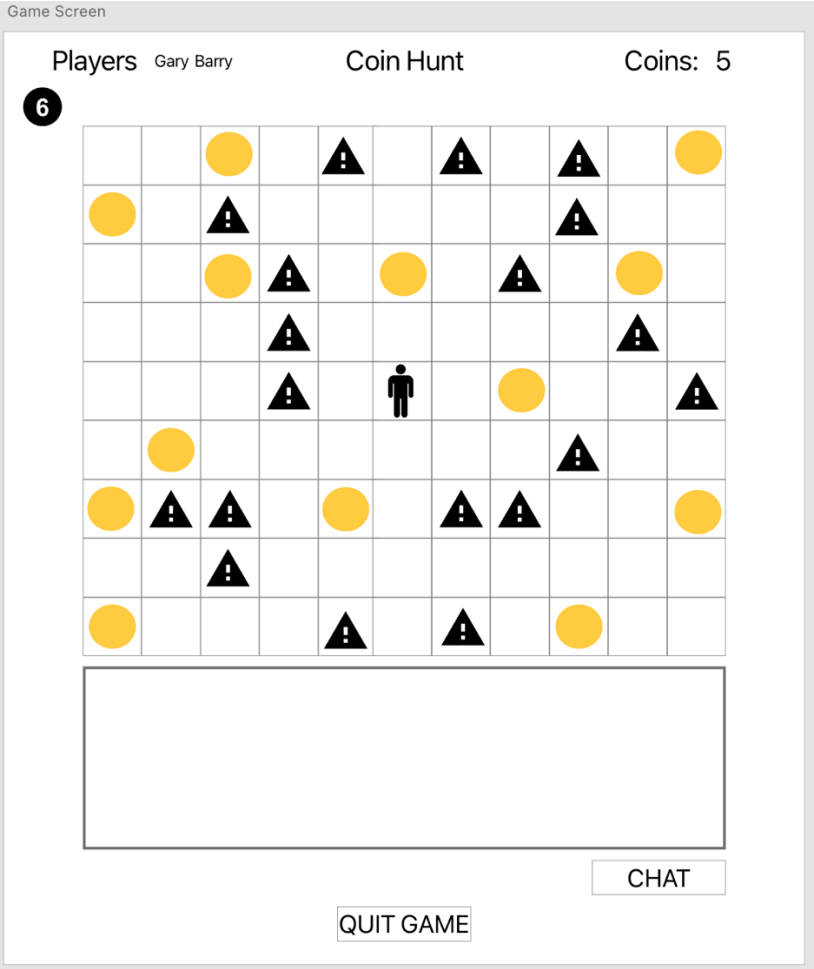
- **1.** The user upon launching application will arrive at the title screen and then press the play game button which will allow the user to play the game as a guest and redirect to the player lobby. But upon quitting the game or application users will lose all access to that progress. If the user decides to log in it will redirect the user to log in/ register screen.
- **2.** The user will enter in a username and password in the input boxes. Once inputted a username and password and the user has selected the "OK" button a check is made to the existing database to check for a match that there is an existing username and password matching the inputted username. If correct the user will directed to the player lobby.
- **2.1.** In this case if the username is not found then the user is not registered and should register to continue using the application.
- **2.2.** If the user enters their password incorrectly, after five attempts the user will be locked out and redirected to the title screen, screen 1.
- **2.3.** After five failed password attempts the user will be locked out and redirected to the title screen, screen 1.
- **3.** If the username does not match any usernames existing in the database, the user is then taken to screen 3 where they are prompted to register as a new player. They will enter a new username and password that will be uploaded/created and stored in the database. Then the user will be redirected to the player lobby.
- **3.1.** If the username is already taken, then the user will be re directed to choose another.
- Users also have the options to navigate between screens as required in-case of or if the wrong input happens etc.

Player lobby / Main screen



- **5. Player lobby/ Main screen** – Once the users/ players have logged in or registered they are then directed to the player lobby. From this screen players can see theirs and other player's names and high scores in the players view area. The list of games is also present, along with new game or join game buttons. Players can also logout back to the title screen, access their account or if admin have admin privileges.

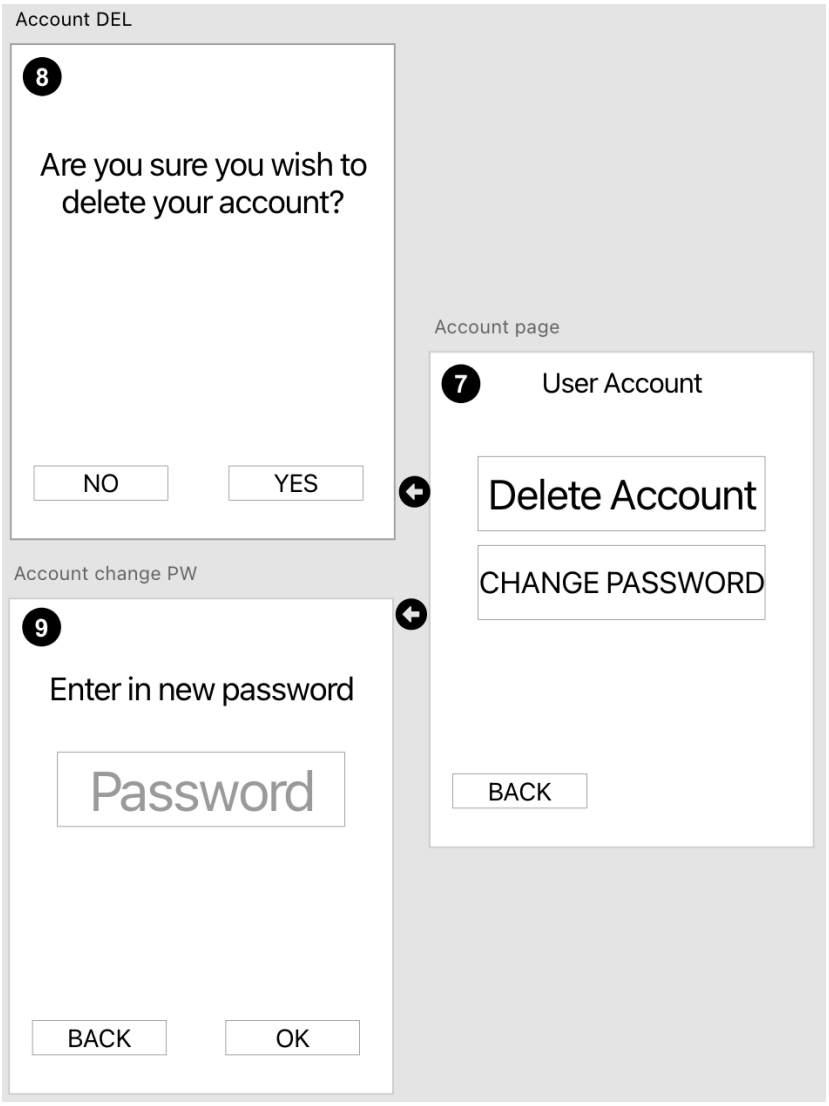
Game screen



- **6.Game screen** – This screen is a mock-up design show casing the general idea of the game objective. The interface shows the players currently in the game along with the users / player score denoted in coins. Also, there is a chat system for players to use and communicate with one another. Players can quit the game back to the main player lobby.

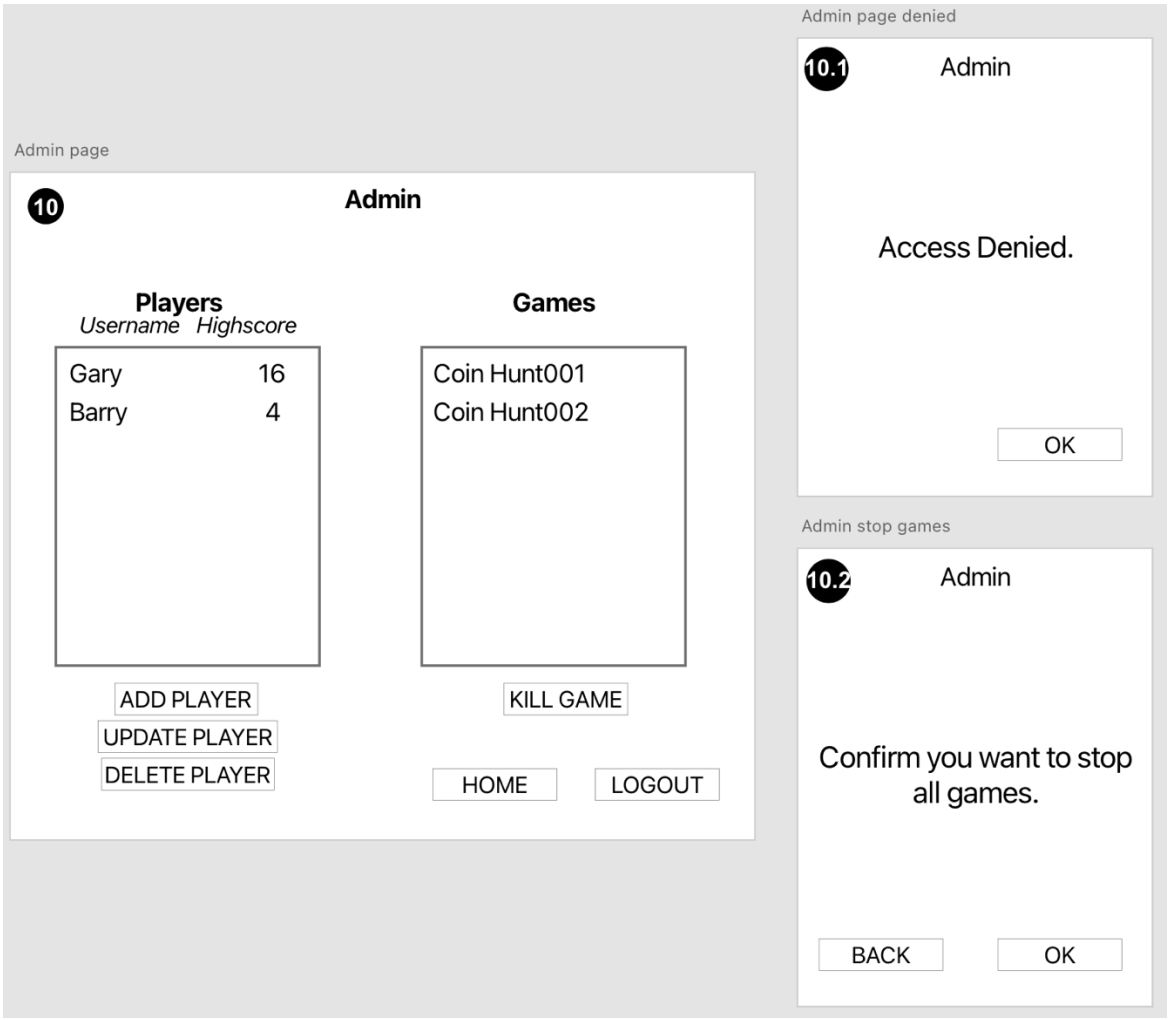
  The layout of the game shows the player titles or map grid. On this grid is the coins marked in a gold/yellow the obstacles using a placeholder image of triangles with exclamation marks. The player icon in the middle is the starting home plate. Players start at their home tile or start position.

Player account



- **7, 8, 9. User account** – Players also can access their account and delete their accounts if required. A pop-up will occur to confirm their action of deleting their account so there is no accidental deletion.
  Note: There is a password change feature for players, but this is a nice to have feature at this stage in development as it is not a requirement.

Admin



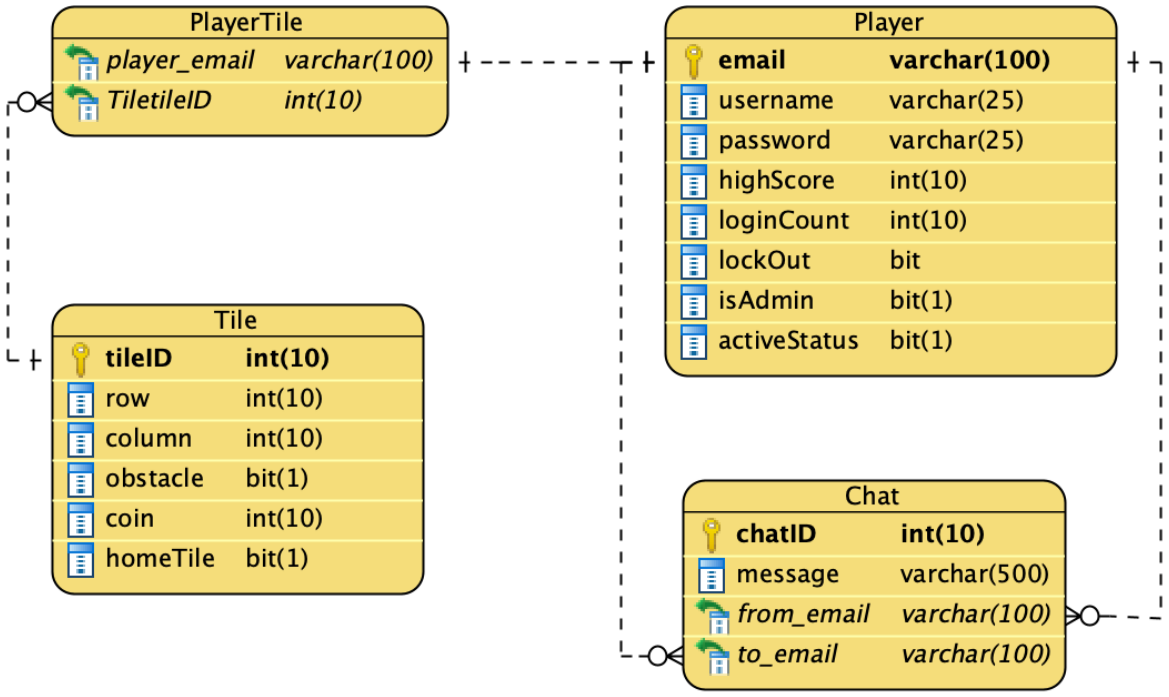
- **10. Admin** – If a player has admin privileges, they can access the admin menu where they can stop all game sessions or add, update, or delete player information.
- **10.1** – Should a player try to see if they can access the admin page and not have admin privileges, they will be met with a pop up of access denied and redirected to the home screen/ player lobby.
- **10.2** – Upon selecting if the admin would like to stop all games, they are presented with a pop up, that details if the admin would like to confirm this action or reconsider and go back to the admin page. Upon selecting ok all current games from the database will be stopped and deleted.

Storyboard interaction list

User Path	Function being tested	Initial app state	Input	Expected output
Login/create/check credentials	Application checks user credentials The username does not already exist, and a new account/user is created The username already exists but the password is incorrect display error message.	Login screen is displayed	User inputs username and password then selects the “Login/OK” button	Login successful: Redirect to player lobby Username created Redirect to player lobby Login unsuccessful: Login unsuccessful redirect to login page. Lock out after 5 attempts.
Logout	Logs out user from multiple interfaces and application	User is currently logged in	User selects the “Logout” button on various interfaces available	User logged out of the app. Directed back to main title screen/page
Exit application	Currently in application. Stop application running	Application is currently running	User selects the “Exit” button or the “X” on any interface tab	Application exits and stops completely
Player lobby	Gain access to player lobby	User completing log in/registration phase	User inputs login/registration information	User’s information is correct, and user successfully logs in or registers and redirects to player lobby.
Player account	Access and display player account information	User on player lobby screen interface	User selects the accounts buttons	Open player account page and displays relevant information
Player deletes account	User can delete existing account	User on account screen interface	User selects “Delete” button	Deletes the current users account so they start fresh or must create a new account.
Game screen	User able to access and play the game	User on player lobby screen interface	User selects new game or join game button	Open game screen and user begins to play the game.

Chat system	Users able to chat to one another while in game	Game open, Username active on chat screen	User types a message in chat window next to their username	Message displays on chat window for all current active users on the same game screen to see.
Admin screen	User with admin privilege can access admin screen/panel.	User on player lobby screen	User sees and can select “admin” button on player lobby screen	Directs the user to admin page and displays the relevant information.
Admin Stop games	User with admin privilege can stop game	User/admin on admin main screen	User selects kill all games button on main screen	A prompt will confirm the action and upon selecting yes, all games are stopped.
Admin player information	Admin able to edit, create and delete player information	User/admin on player information screen	Admin clicks on either add button or delete button or can edit player details directly.	Player information has been either created, updated, or deleted.

Logical ERD



This ERD design is a relational modelling diagram. It is developed to both improve the system and how it is visualised and communicated. It details events that take place and the data required.

In this ERD the brief overview of its' design is to construct a player character with all the necessary data needed as per the project requirements. The linking tables include the "PlayerTile", "Tile" and "Chat" tables. These relate to the player in a few different ways.

The "Player" table has all the information needed to identify the player. This is useful for login, logout, registration, and admin privileges. An active status will be made to show the status of a player if they are available to play or not.

The "PlayerTile" table is a join table between both the player and tile tables. With this it allows the player identifier, which is the player email, and the tile id from the tile table to identify where the player is on the game board. Also, it will allow for other uses when it comes to game play and interaction with objects on the board.

The "Tile" table has information all about the gameboard, its layout, size, starting positions, and objects that the player will eventually interact with.

The "Chat" table has information about any chat messages that are sent and received to players. This will be used for player communication.

C.R.U.D Table

CRUD for tile based game.																				
		C = INSERT	R = SELECT	U = UPDATE	D = DELETE	R/U = Select and Update			R/D = Select and Delete											
		Post	Get	Put	Delete															
	Table	Player								Tile				Player Tile		Chat				
	Process	Email	Username	Password	High Score	Login Count	Lockout	Is Admin	Active Status	ID	Row	Column	Obstacle	Coin	Player Em	Tile ID	ID	Message	From Player Email	To Player Email
	Register a player	C	C	C	C	C	C	C	C						C	C				
	Login Player	R	R	R	R	R/U	R	R	U											
	Login Check Credentials	R	R	R		R	R													
	Failed login Account lock	R					U													
	Logout Player	R							U											
	Delete Player	R/D	D	D	D	D	D	D	D						D	D				
	Join Game	R							U						R	R				
	New Game	R							U	C	C	C	C	C						
	End Game	R							U				D	D	R	U				
	Chat to Player	R															C	C	C	C
	Player Location	R													R	R				
	Player Movement	R													R	R/U				
	Check Tile									R	R	R	R	R						
	Player Picks up Coin	R			R/U									U						
	Enter Admin Screen	R						R	U											
	Admin Add Player	C	C	C	C		C	C												
	Admin Update Player	R/U	R/U	R/U	R/U		R/U	R/U												
	Admin Delete Player	R/D	D	D	D	D	D	D	D											
	Display High Score	R			R															

This CRUD is based off the project outline/ game brief descriptions in conjunction with the logical ERD design. It details the processes users will make upon interacting with the application. When these processes are used or activated the CRUD visually indicates how the database will respond to these interactions.

DDL- DML SQL

Check out GitHub here: https://github.com/Kenny-WilliamsStockdale/DAT602-Project.git

Revisions and notes

The current design has undergone changes since initial conception.

- Single level board.
- Player email is the unique identifier, instead of an ID.
- Player is connected to tile through the player title join table. It identifies email along with tile ID which can see the player position on the board.
- The chat system will be moved to the player lobby for ease of starting up games and communication. This I feel adds more flow to the overall design and user experience
- There will be only one game session available at a time that will enable players to join or host. This will still adhere to the multiplayer experience.

General notes:

- The admin add, update and remove buttons/ forms are not present in the storyboards mentioned in this report. However, they will be available in the final build.
- Other changes may occur during the development of this game application.

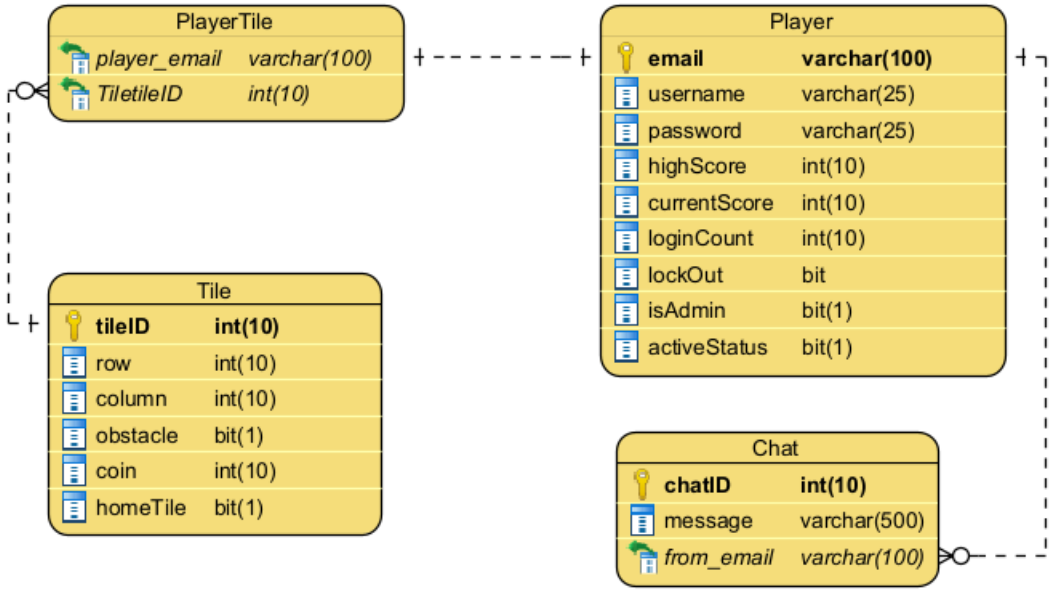
Milestone 2

Logical and physical design alterations and rationale

The logical design created in milestone one was re-designed during the creation of SQL procedures for milestone 2. In turn the DDL, DML and physical design has been altered to match. The alterations made are primarily to get the necessary dataflow to and from the database.

The most notable changes added is a single column to the player table which is 'currentScore'. This alteration may be changed again in the future of the project. The reason for this change is there is a requirement as part of the game logic that requires players to accumulate score during the current game. The current score is used to present the winner of the game, then the score is compared to their current high score. If the current score is more than the players overall high score, then the player's high score is adjusted to reflect that change.

Another change is from the chat table where the 'to_email' foreign key has been taken away. This is done to simplify the chat system into a broad global chat system. This will instead have messages sent to a global application chat. The players viewing the chat will be able to see which player has sent the message (via email identification) but the message will be viewable by any player viewing the chat instead of any direct messaging to an email/player identifier.



SQL procedures and transactions

Login

Returning players will need to login to the game. A login procedure is created to allow for registered users to login to the game. The procedure checks if the current username and password exist and if matching credentials of the database, logs in the user. If the credentials are not correct there is a

max number of 5 attempts to login. If users unsuccessfully login after 5 attempts, it will lock the user's account. The user will have to wait for an admin to unlock their account.

If the users have logged in successfully a message is provided and the user's account active status is set to 1, which means being active. Active status shows other players that a player is online/ eligible for a game/match.

Logout

This procedure will log the user out of their account and will change their active status to 0, in an offline state.

Register player

This allows the user and admins to create a new account within the game application. It allows use of the system and is required if there is no matching email already existing within the system. The procedure currently takes in an email, username and password provided by the user, then inserts them into a new record ad provides other default values as necessary.

Delete account

Administrator and users can both use this procedure to delete their accounts or in the case of the admin also delete user accounts of their choosing. Inserting an existing username into the procedure will prompt deletion of the account or it will return that the user does not currently exist in the system.

Get all players

There are two separate procedures in the SQL script that gets all active status players and all users. Getting all active status players is useful for populating a list inside the application for all users using the app to see which players are online and or eligible for a game/match. Upon logging in the active status is triggered to be active/online.

The get all users procedure can be used for admins in gathering all the information about current users registered in the application. It will be used for admin purposes.

Generate map

In this lengthy procedure a dynamic 10 x 10 grid or map is generated via loops to represent the tiles for the game map.

Included in the procedure is other updates as part of the project requirements. Firstly, one of the updates is to set the home tile (starting positions) for all players.

Next, on the game map loop over and generate random positions for the obstacles that sit in the way of players and the coins the players are to collect. Obstacles will not generate on the player home tile.

Lastly, loop over and generate coins over the game board where there are no obstacles or a player home tile.

Join game

At this stage of development there is only one game available, and many players can join if they so wish. If players decide to join the current game, all players will start in the centre of the map

Player move

Players will move around the game map to collect coins. Players can move up, down, left, and right around the current tile they inhabit. Before the player moves to an adjacent tile multiple checks are made on the neighbouring tile.

Firstly, to check if the tile is close and able to be moved to. Does the tile contain and obstacle, coin, or is it empty. If the tile has an obstacle, then the player cannot move to that tile. If the tile has a coin, then the player can move to the tile, collect all available coins, add the coin value to the players current score and delete the coin off the map at the current tile location. Or, if the tile is empty with no obstacle and or coin then the player can move.

Score

In each movement that the player does to a tile where a coin is found and collected, a call is made to another procedure. This procedure deals with examining the players current score and overall high score from the total matches the player has played. If the current score that the player has accumulated in game is bigger than the player's overall high score, then after the player's movement the player's overall high score is changed to reflect the larger score.

Chat

This a simple procedure that allows players in the game lobby to send a message to a shared table which serves as a global chat room. The messages will show from which user they were sent from, and the message content will be able to be seen by all users that are currently active in the application.

Finish game

In this procedure when called the game will finish. It will check all players and their current scores. The player with the largest current score will be shown as the winner of the game. All players in game current scores will be reset and the position on the map will be reset back to the starting home tile location.

Admin update player information

The admins will be able to have access to player information where the admin can alter player account information. This includes information such as username and password.

Multiplayer support and ACID

ACID

ACID stands for Atomicity, Consistency, Isolation, Durability

Atomicity

Is an all or nothing proposition. This means there is only two possible outcomes of a transaction, either success or failure of the transaction. There is no in between. The focus of this property is to prevent data loss or corruption occurring if some how a service stopped operating mid-way through. (*What Are ACID Transactions?*, n.d.).

An example of this would be the player join game. The player joins the map and is placed at the home tile, or the procedure fails, and the player does not join the game. This way there is no confusion the player is either on the game map or they are not.

Consistency

Ensures that a transaction never leaves the database in a half-finished state. It means that if part of a transaction fails, all if any changes made up until that point will be rolled back to their previous state. This leaves the database in a state as it was before the transaction took place. (Jepson, n.d.).

A state of example from the current project is the player and chat records. When deleting the player account to keep consistency the players chats should be deleted before the player is deleted as the chat records are marked as a child of the player table. This would be an example of consistency but for the projects current design the chat is left in as a history for other players/users to see even if that player who sent the chat message has been deleted.

Isolation

Is the purpose of keeping transactions separated from each other until they are finished. An example is to make sure any read or write will not be impacted by other reads or writes of separate transactions. However, it does not mean two operations can't happen at the same time as long as the transactions have no way of occurring at the same time. (Watts, 2020).

Durability

Durability ensures that changes made to the database such as transactions stay permanent. Even in the case of failures, crashes, and outages. The database can access a transaction log if terminated in an abnormal way to retrieve completed transactions and apply them back to the database.

Using backup strategies such as software and hardware will allow the system to recover with the correct information from such terminations.

Multiplayer support

As per the design of milestone one, Coin Hunt has only one game session where multiple players can join to play (*subject to change as development progress*). The map is generated currently and exists before a player can join. This ensures isolation of generating the map procedure. Players will be able to join and start at the same time or part way through a game. The procedures have been designed in a way in the current logical and physical design that all the actions performed on that database are encapsulated in a transaction, this helps keep all actions performed on the database consistent.

Revisions and notes

The current design lacks in some areas of the project requirements and will need to be changed once more for the final milestone.

- A session table will be implemented to provide separate game sessions.
- The player table will be updated to include a session ID
- Login procedure needs to prompt user to register if username credentials do not exist.
- User register procedure needs to have admin separated so users cannot instantly make themselves an admin.
- Player join game procedure will change to meet the requirements of the session table
- Finish game procedure will change to meet session requirements and have a check to ensure that all coins on the map have been claimed before ending game.
- Admin procedures need to be fleshed out to meet project requirements.

References

Jepson, B. (n.d.). *Web techniques: PostgreSQL vs. mysql*. People.apache.org. Retrieved May 25, 2022, from https://people.apache.org/~jim/NewArchitect/webtech/2001/09/jepson/index.html#:~:text=ACID%20is%20an%20acronym%20that

Watts, S. (2020, April 24). *ACID: Atomic, consistent, isolated, & durable*. BMC Blogs. https://www.bmc.com/blogs/acid-atomic-consistent-isolated-durable/

What are ACID transactions? (n.d.). Databricks. https://databricks.com/glossary/acid-transactions