Phillip David Shields
DAT602 Milestone One
Todd Cockrane
March 23, 2022

Milestone One

Analysis and design

Description of Game

This game will be a multi-user 2-D adventure tile game that will be playable via a point and click game system connected with a simple database. This will be a rudimentary prototype that can be used for the future development of a much more complex version.

Basic functional requirements

1. Registration of new players require:
    a. User name
    b. Password
2. Users are given three attempts at entering their passwords
3. Online players are visible to one another
4. Admin accounts will have the following abilities:
    a. Adding new players
    b. Updating existing players
    c. Deleting existing players
    d. Ending players game states
    e. Deleting current games
    f. Locking and unlocking player accounts
5. Players can start a new game or continue their previous games
6. Players start on a predefined home tile
7. Multiple different games can be played at the same time
8. Players move around the board by moving only onto adjacent tiles to their current position
9. A players location is stored in the database after every move
10. Tiles contain Elements that will increase or decrease the players' score
11. A players game, location, and score will be saved upon logout
12. A players game, location, and score will be restored when logging back in

Game concept

The game will be based on the idea of intergalactic travel. A game instance will occur within a galaxy where players will move around in search of elements to mine. Essentially, a galaxy is represented by a tiled gameboard. Each tile represents a location within the galaxy and potentially contains some type of resource or element to be collected. A single tile could possess a single element, multiple elements, no elements, or even destructive elements.

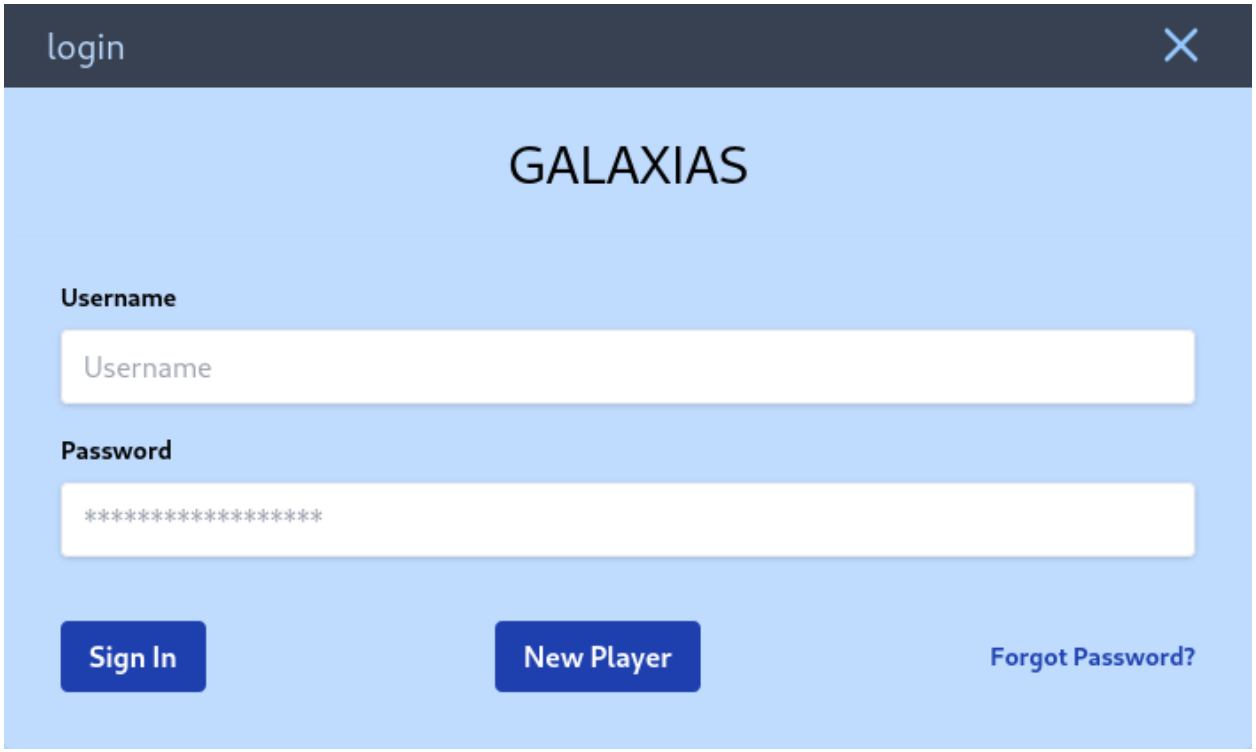
Elements will have different value points that increase the player's points and destructive elements will decrease the player's points. There may even be apocalyptic elements that kill the player and remove all points.

A galaxy will be considered explored and depleted if all elements, except destructive elements, are mined and collected by players.

Multiple players may explore the same galaxy, and each time a player starts a new game a new and random galaxy will be generated.

STORYBOARDS

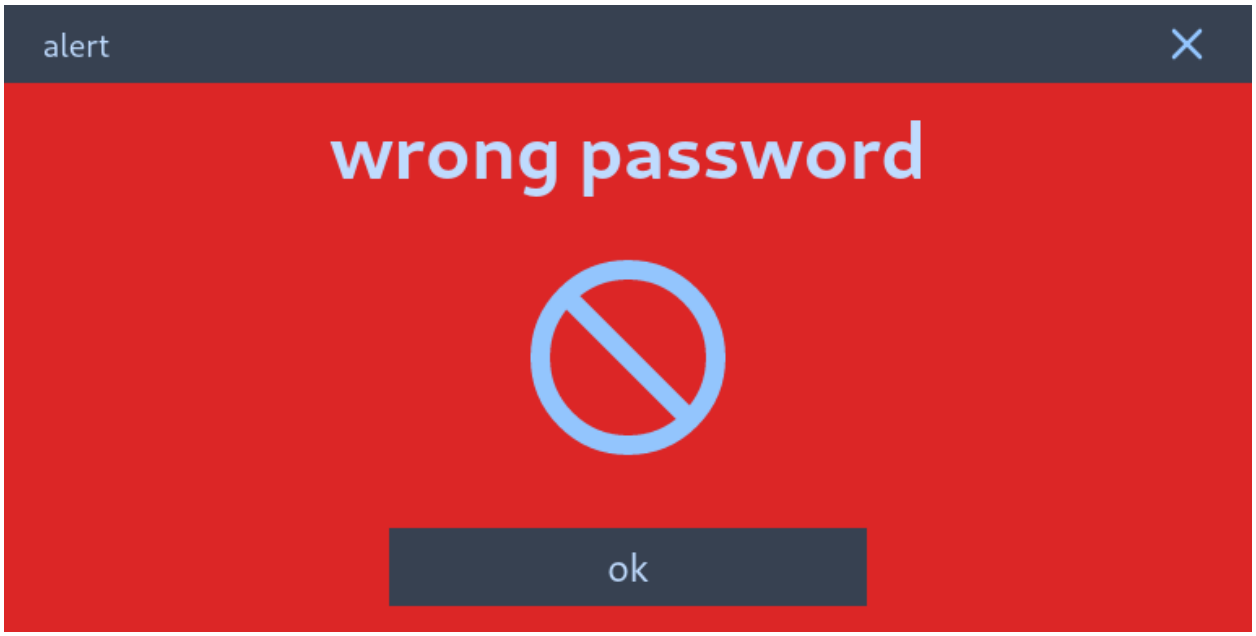
Login



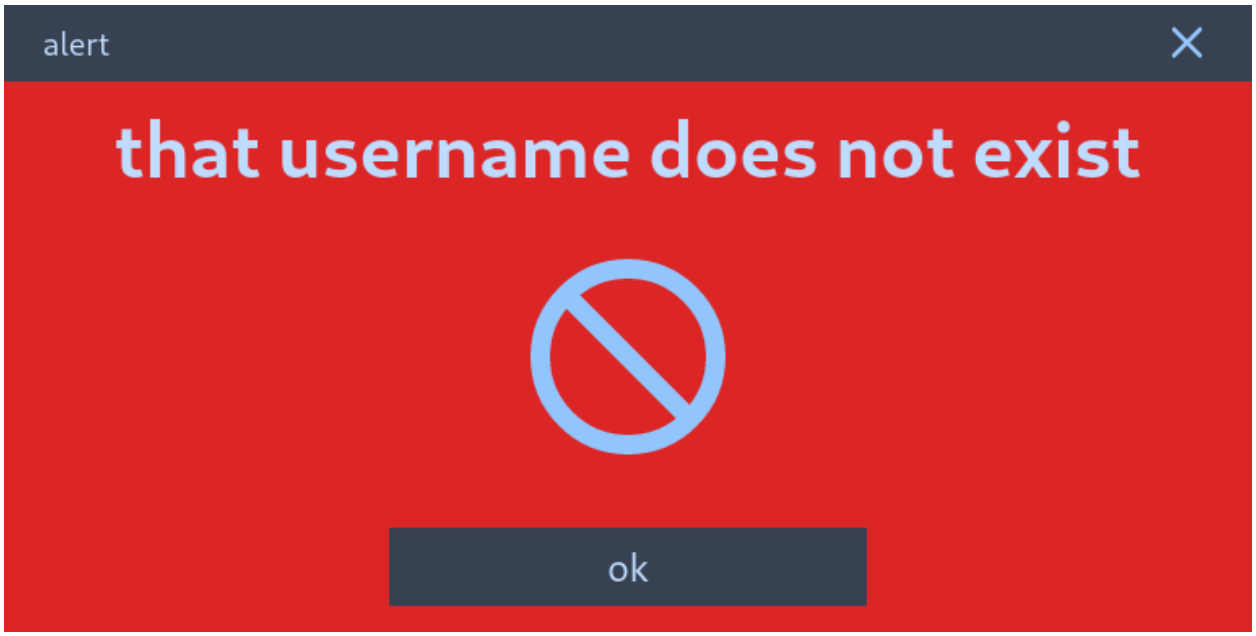
The login screen consists of a username input, a password input, a “sign in” button that submits the entered username and password to a database for verification, a “new player” button that opens the registration screen, and a forgot password link that would open some sort of menu to help the user recover or create a new password. For the scope of this project, password recovery won't actually be implemented.

Login Alerts

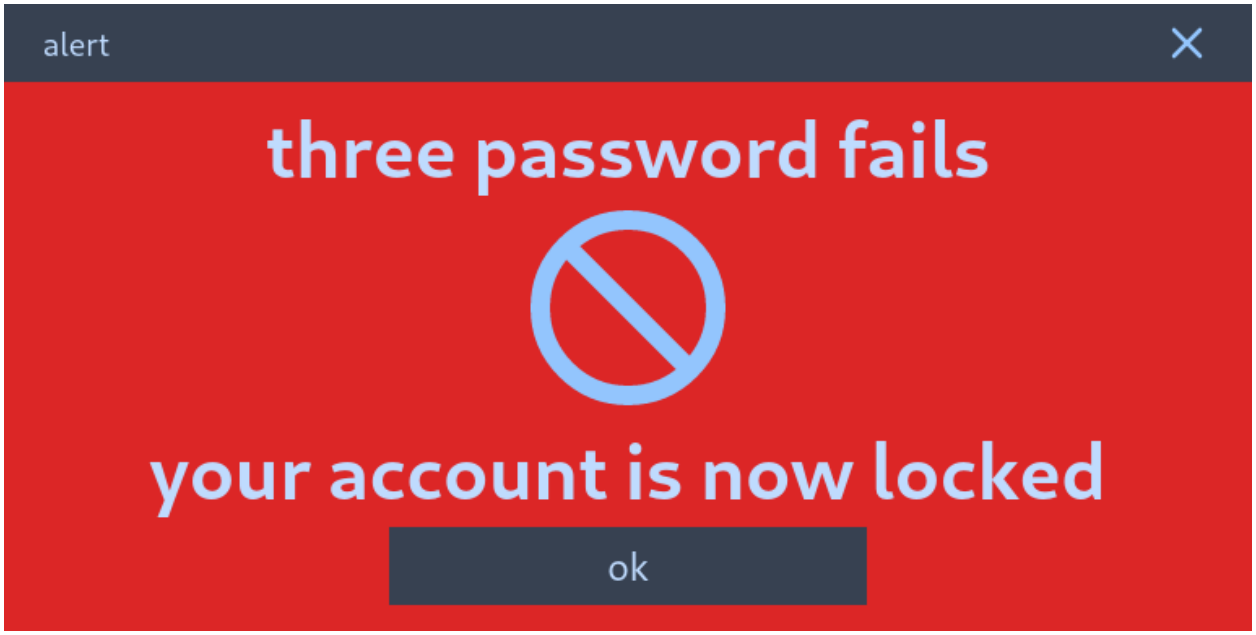
The following alerts are specific to invalid actions that occur via the Login menu.



This alert will trigger when a user tries to log in with a registered username and an incorrect password. The button returns to the login screen.

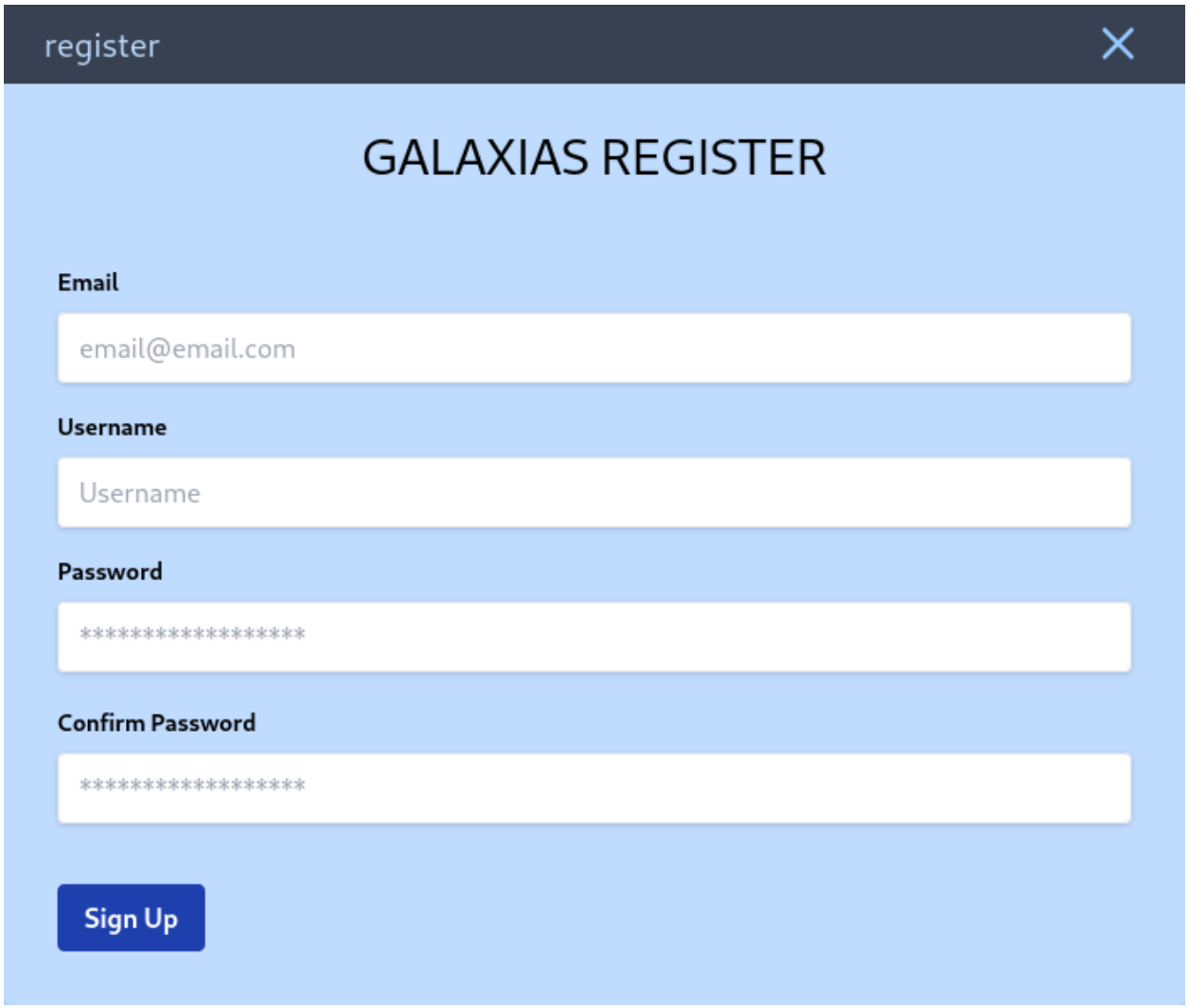


This alert will trigger when a user tries to log in with a username that is not yet registered.



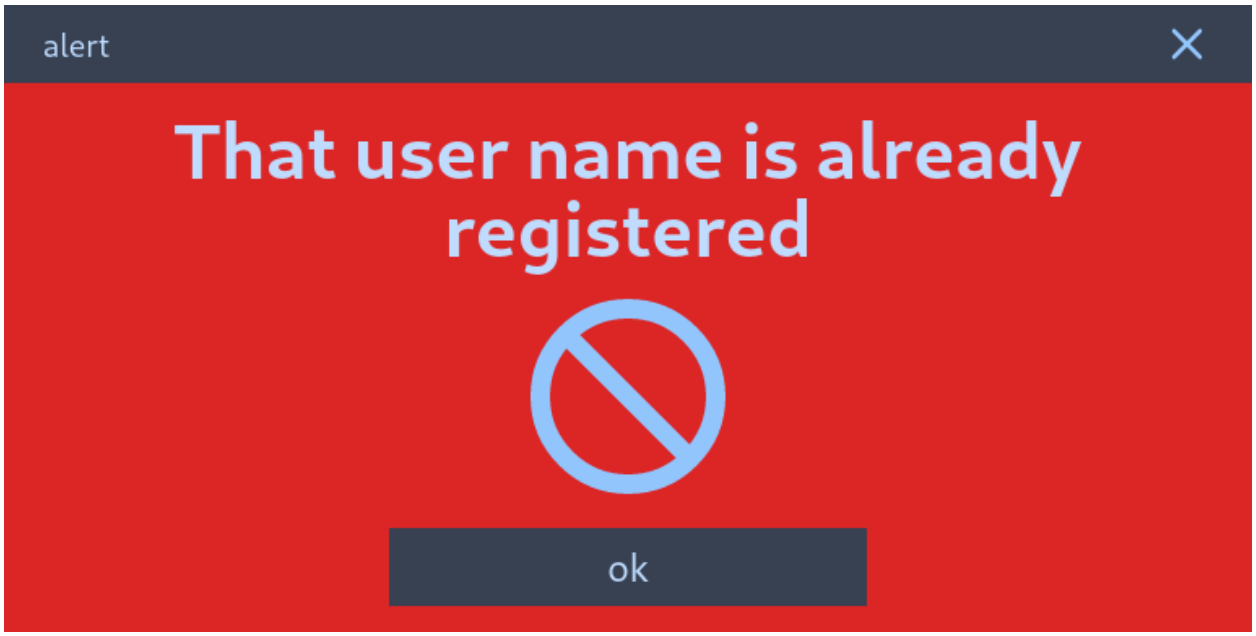
This alert will trigger when a user enters the wrong password for a registered username three times.

Register



The register menu consists of an email input, a username input, a password input, a password confirmation input, and a “sign up” button that submits the info to the database for user creation.

Register Alert



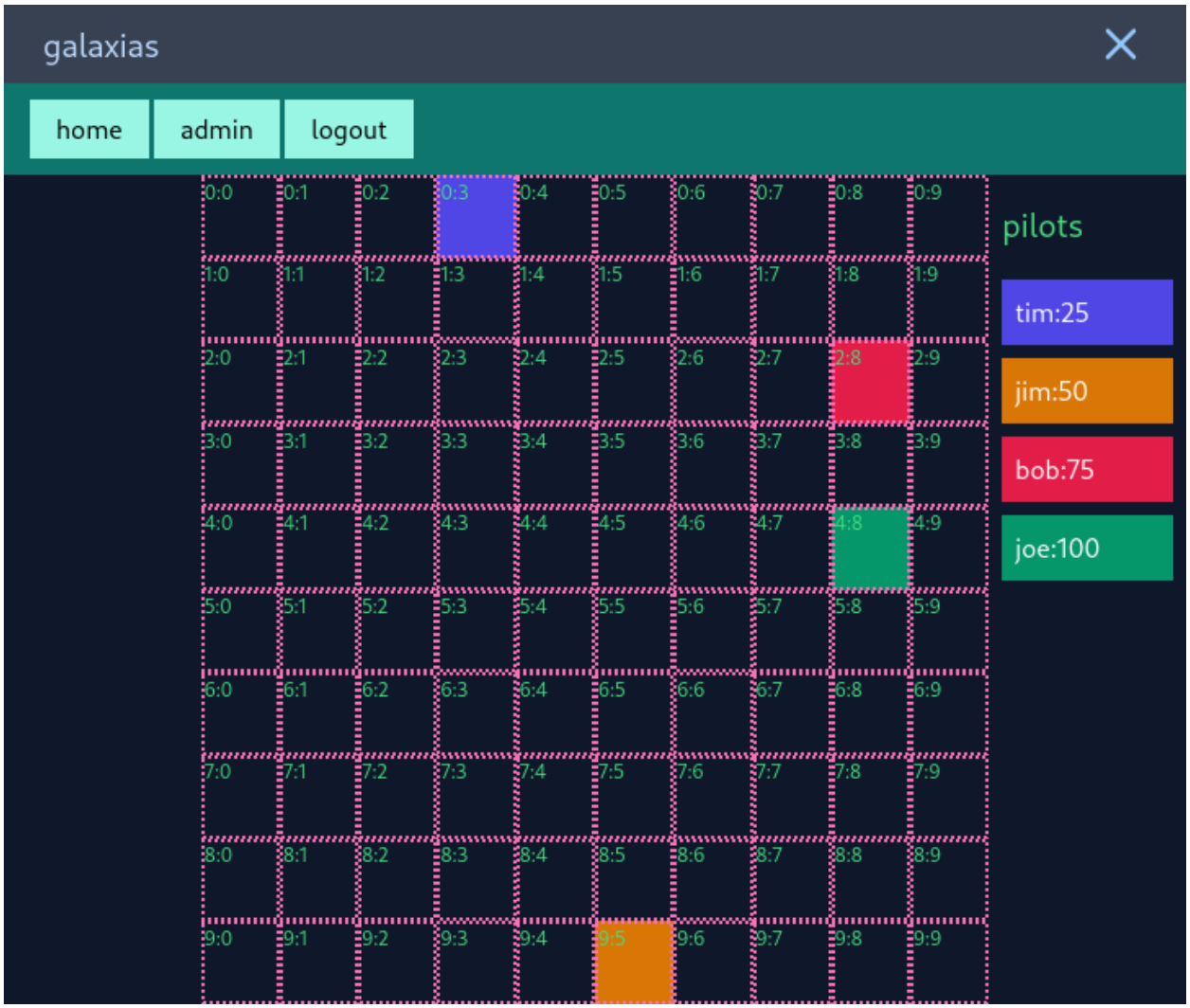
This alert will trigger when a user tries to create a new account with a username that already exists. Clicking the “ok” button will return to the registration screen.

Home Menu

The home menu consists of a top row of navigation buttons: a home button that returns to this screen, an admin button that is only clickable by administrators and opens an admin menu, and a logout button that logs the user out of the game and closes the menu.

Below the navigation buttons, are two displays that show current ongoing games via the “Open Galaxies” display box, and a list of other players currently online via the “Free Pilots” display box. Below the Open Galaxies display box is an “enter galaxy” button that allows the player to join a current game, while the “new galaxy” button creates a new game.

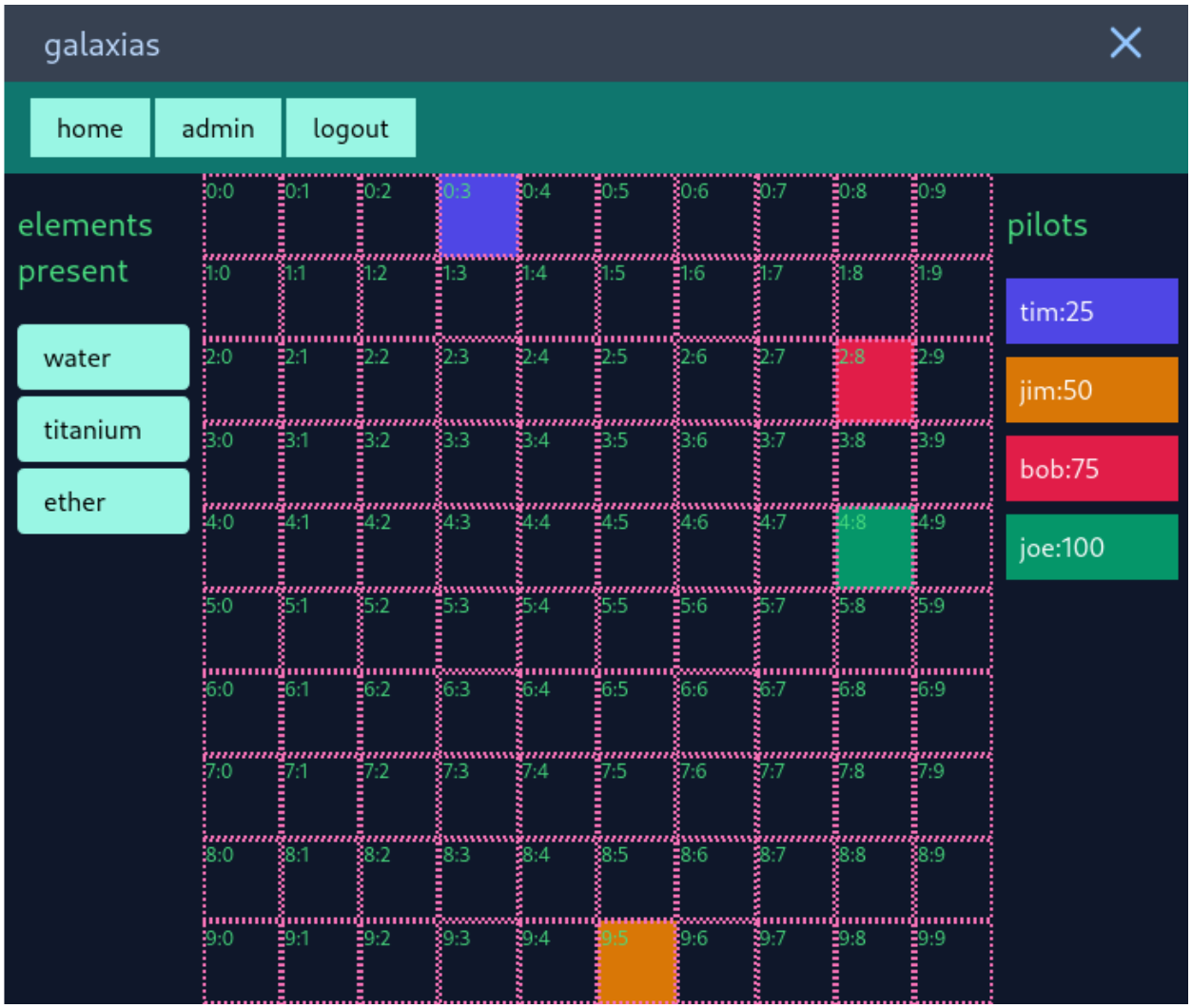
Main Game Display



The game display contains the same navigation menu as the login screen. The lower section of the game menu screen contains three main sections: on the left-hand side is an element selection menu bar that is blank in the above screenshot. In the middle is a grid that displays the galaxy or game board. And on the right is a list of players currently in this galaxy and their current scores.

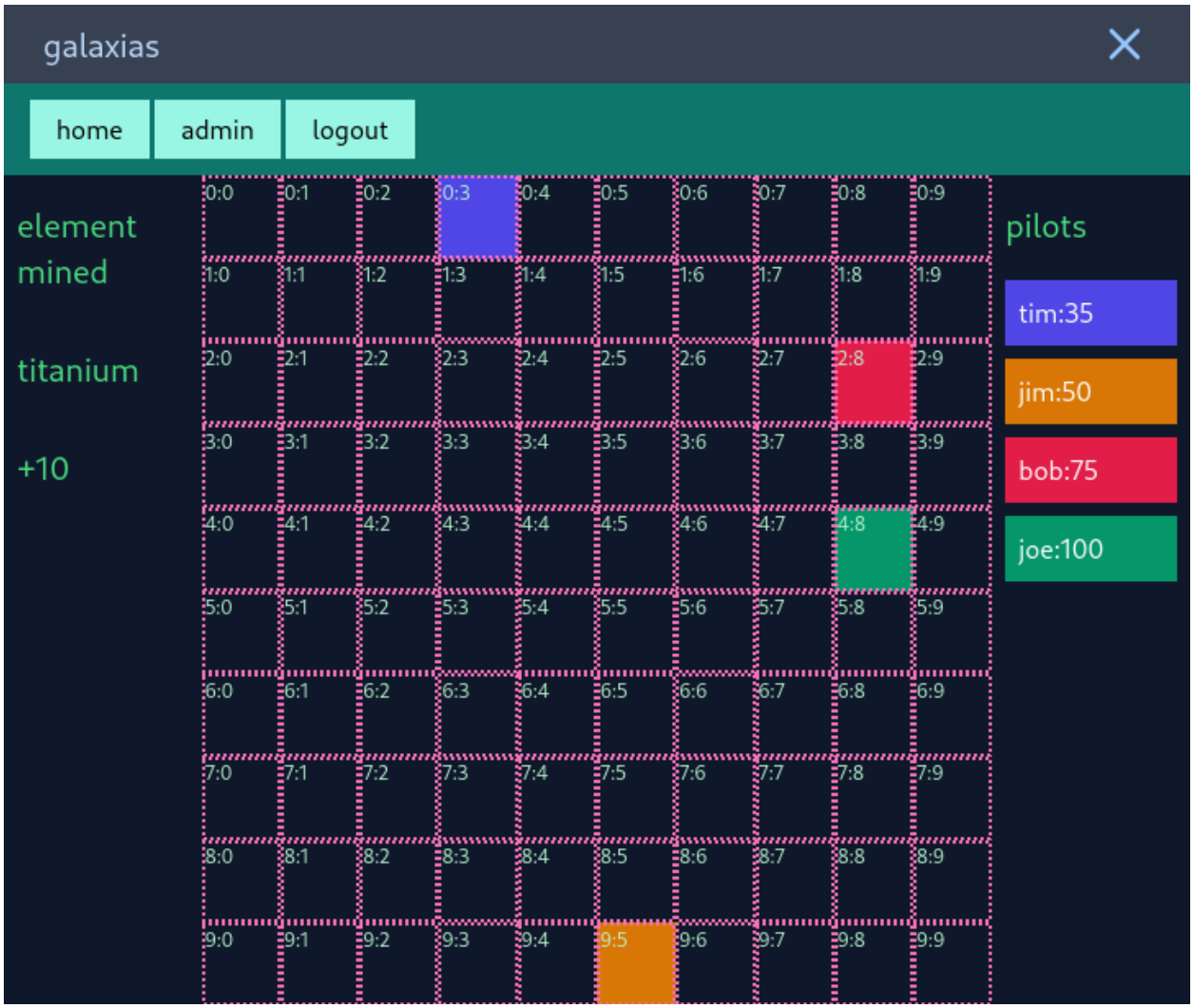
The galaxy grid has each vector's coordinates displayed in the upper left corner of the square. Colored vectors represent each pilot's location.

Main Game Display - Element Selection



The game menu screen above shows the element selection menu when a player lands enters a vector with multiple elements present. A player can only select one element at a time, but may leave and then return to a vector to mine multiple different elements.

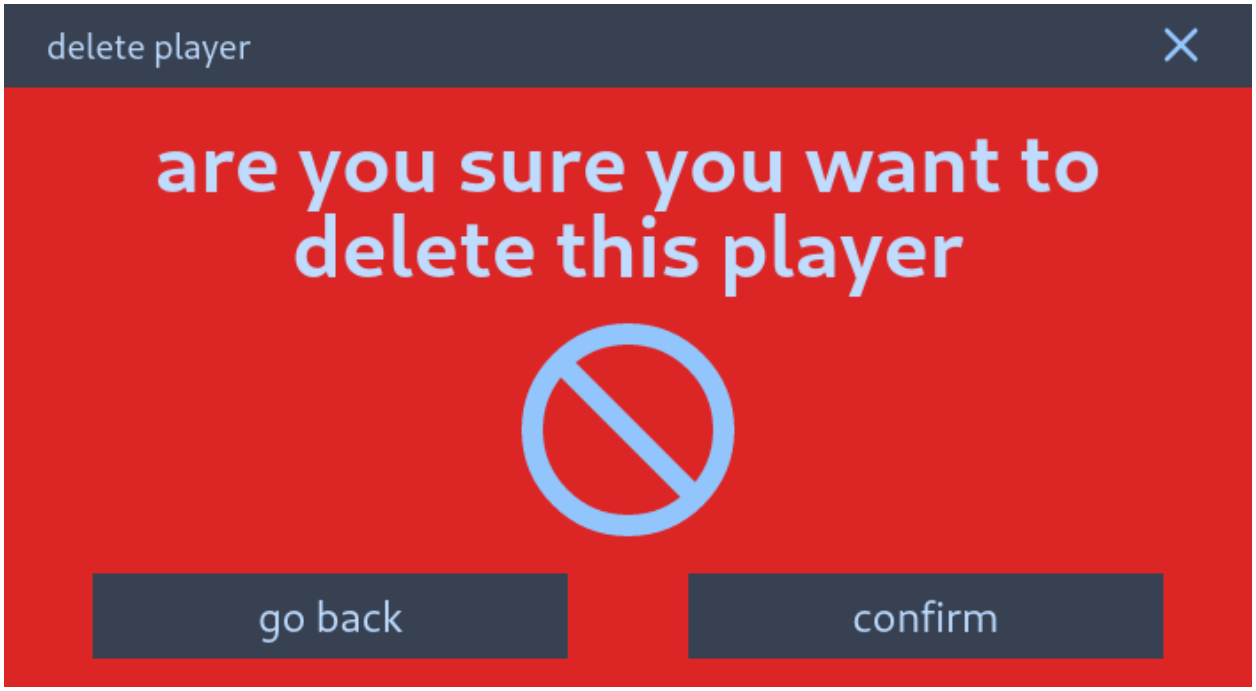
Main Game Display - Element Mined



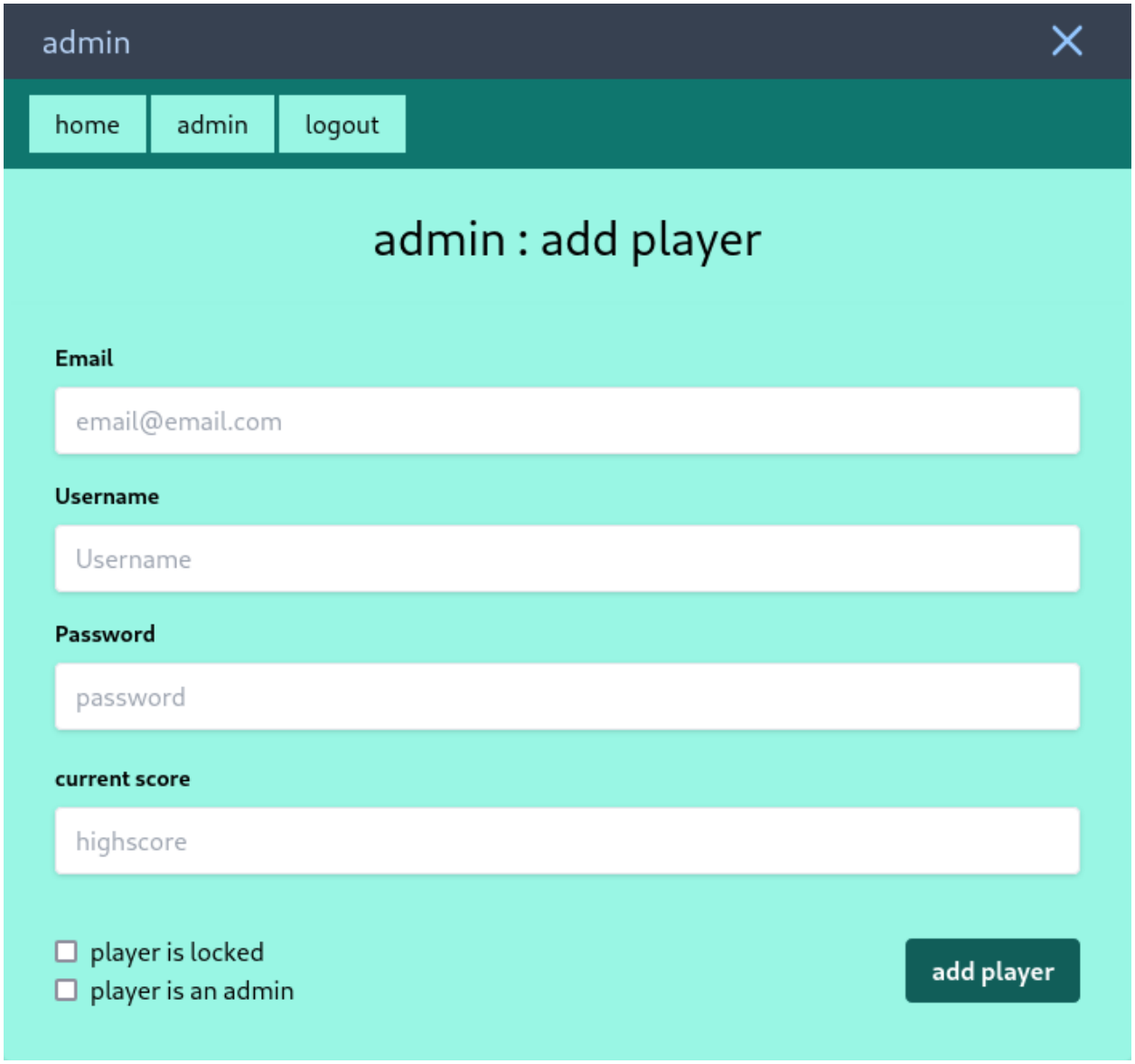
The game display above shows the element has been mined, the points added, and an updated score for the player tim to 35 points. The element menu on the left will not revert to an element selection menu until the player has moved into a different vector.

Admin Menu

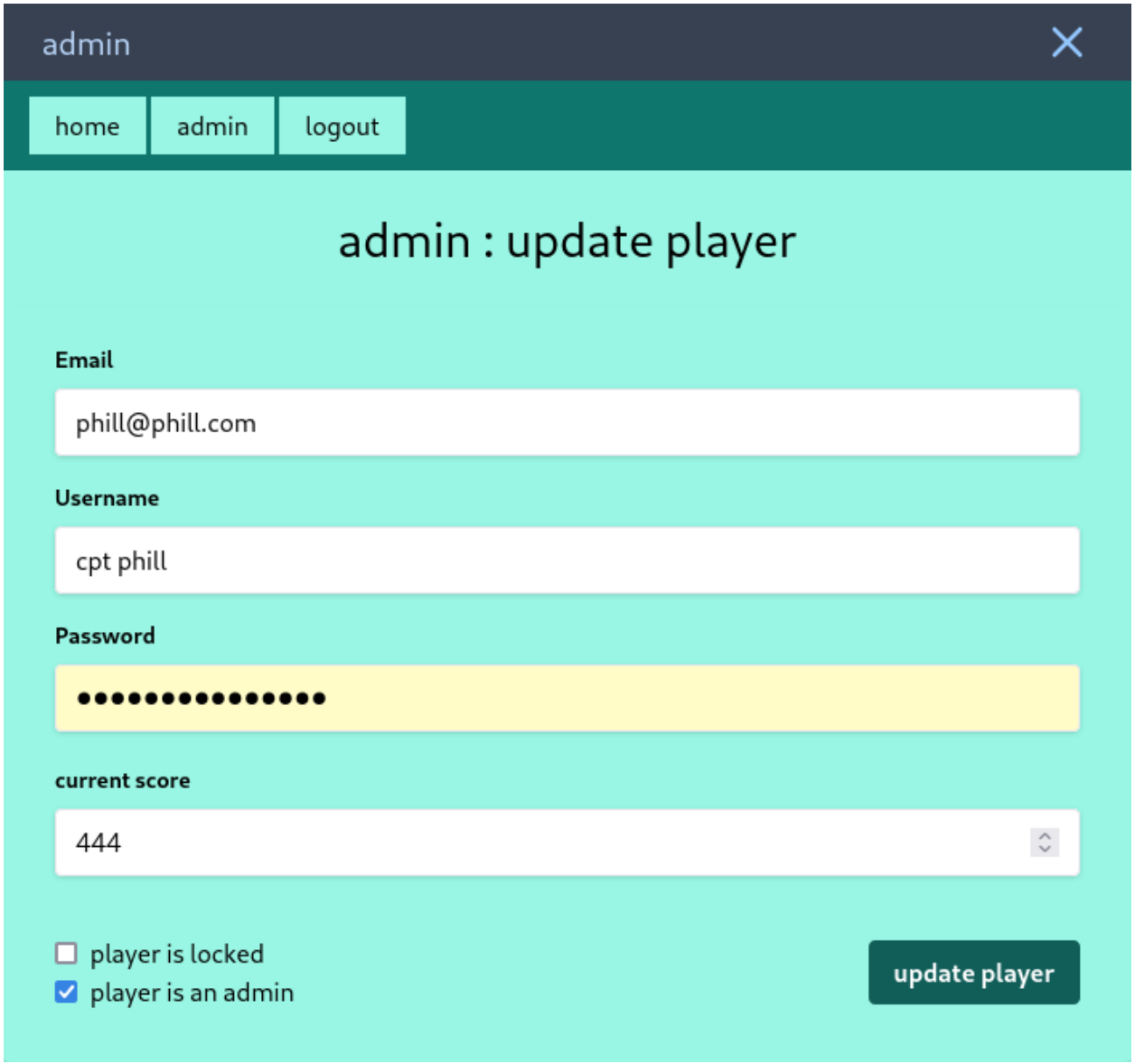
The admin menu shows the same information as the home menu regarding open galaxies and pilots currently in exploration. Below the display boxes are the admin buttons for deleting a current game instance, adding a new user, unlocking a user, and deleting a user.



This alert menu will trigger when a player is selected and the “destroy pilot” button is clicked. Pressing the “go back” button will return to the admin menu without any modifications to the player or database. Pressing the confirm button will delete the player from the database and return to the admin menu.



The admin add player menu consists of an email input, a username input, a password input, a current score input, a checkbox to lock or unlock the player, a checkbox to declare if the player is an administrator with admin privileges, and an “add player” button that will submit the new player to the database.



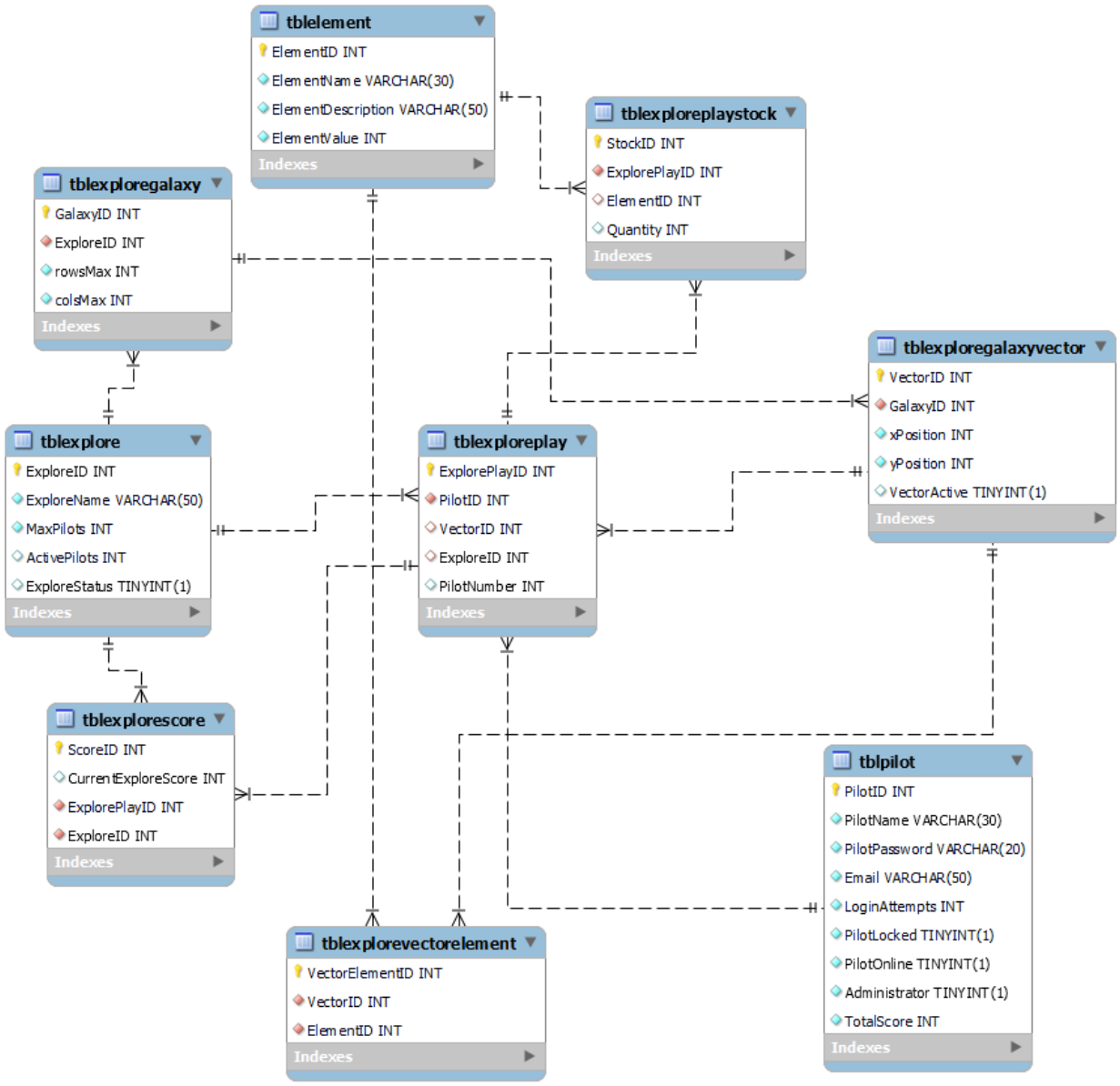
The update player menu has all the same elements as the add player menu, but the elements will be filled with the data of the player that is being modified. Clicking on the “update player” button will update the modified information in the database.

Menu Design Explanation

The game design storyboards provide a rough draft layout and general theme for the base of the game prototype. Some changes are expected, especially after user testing and feedback. An emphasis will be made to maintain simplicity and reusability of different menu components, this

will make the UI easy to use and the use of reusable components will shorten the time required for game development.

Entity Relationship Diagram



Entity Relationship Diagram Explanation

tblPilot

The Pilot table stores individual pilots' data.

tblExplore

The Explore table stores records of each exploration being played

tblExplorePlay

The ExplorePlay table bridges a pilot to an exploration.

tblExploreScore

The ExploreScore table updates and stores a score for each pilot in exploration.

tblElement

The Element table stores all the potential elements that can be discovered and collected during an exploration

tblExplorePlayStock

The ExplorePlayStock table holds all elements collected by a pilot in each exploration

tblExploreGalaxy

The ExploreGalaxy table creates an instance of a galaxy for each exploration.

tblExploreGalaxyVector

The ExploreGalaxyVector table holds all the vectors for each galaxy in each exploration.

tblExploreVectorElement

The ExploreVectorElement holds references to the various elements and their vectors within each galaxy in each exploration

CRUD Table

	pilot registration	pilot log in	lock pilot	delete pilot	pilot log out	join galaxy	new galaxy	pilot moves	admin menu	destroy galaxy
Pilot	C	R		D	RU	R	R		R	
PilotID	C			D		R	R		R	
PilotName	C	R		D		R			R	
PilotPassword	C	R		D						
Email	C			D						
LoginAttempts	C	RU		D						
PilotLocked	C	R	U	D						
PilotOnline	C			D	U	RU				
Administrator	C			D		R				
TotalScore	C			D	RU	R				RU
Explore					U	R	C	U	R	U
ExploreID					U	R	C		R	
ExploreName						R	C		R	
MaxPilots							C			
ActivePilots					U		C	U		
ExploreStatus					U				R	U
ExploreGalaxy							C	R		
GalaxyID							C			
ExploreID							C			
rowsMax							C			
colsMax							C			
ExploreGalaxyVector							C	R		
VectorID							C	RU		
GalaxyID							C	R		
xPosition							C	R		
yPosition							C	R		
VectorActive							C	RU		
ExplorePlay				D	U	R	C		R	D
ExplorePlayID				D	U	R	C		R	D
PilotID				D		R	C		R	D
VectorID				D	U	R	C	U		D
ExploreID				D	U					D
PilotNumber				D	U		C			D
ExplorePlayStock				D			C			
StockID				D			C			
ExplorePlayID				D			C			
ElementID				D			C			
Quantity				D			C			
ExploreScore				D		R	C			R
ScoreID				D		R	C			
CurrentExplorerScore				D	R	R	C	U		R
ExplorePlayID				D		R	C			
ExploreID				D			C			
ExploreVectorElement							C	RU		
VectorElementID							C	R		
VectorID							C	R		
ElementID							C	R		
Element							R	R		
ElementID							R	R		
ElementName							R	R		
ElementDescription							R	R		
ElementValue							R	R		

CRUD Table Explanation

Pilot Registration

Trigger an insert command with related pilot info to populate and add a pilot to the Pilot table.

Pilot Login

Retrieve login-related info, reset login attempts to zero, and update PilotOnline to true..

Lock Pilot

After three unsuccessful login attempts the PilotLocked field will be set to true and the pilot will not be allowed to log in until reverted by an admin.

Delete Pilot

Delete all records from the pilot table and cascade to any foreign key references in other tables.

Pilot Log Out

Update PilotOnline to false, update the pilot's score and current exploration-related fields.

Join Galaxy

Retrieve the user information and all current exploration information from other current galaxies being played. This is found via the ExploreStatus field being true for an exploration. Then retrieve all pilot, galaxy, and exploration-related tables and data.

New Galaxy

Starting a new galaxy will create a new entry into the explore table and create related table entries need to start a new game.

Pilot Moves

Update the pilot location and score according to the last move.

Admin Menu

A pilot must have their Administrator field set to true to access this menu. This menu will retrieve all current exploration, galaxy, and pilot information.

Destroy Galaxy

Revert ExploreStatus to false and update all current pilot score data.

Milestone One Summary

For this milestone, a great many elements of the game were planned and developed. Great care and effort were made to consider all the broad strokes needed to develop this game. Various technologies were used, including:

- Mysql workbench
- Visual studio code editor
- SvelteJs (storyboard designs)
- GitHub
- Microsoft Excel

The first part developed was the game concept and functional requirements. A general idea of what the game is and what sort of basic functionality were needed before anything further could be developed. I have always enjoyed learning about space and playing video games involving space exploration, so a similarly themed game was an easy first choice. I used the assessment outline to align my game's basic functionality with the course requirements.

The second part was creating the table UML needed for the game. I sketched a rough draft and then used the MySQLworkbench GUI to create the tables, during this process, I fine-tuned the tables and necessary key constraints required for the game. Then I used MySQLworkbench to generate both the DDL and ERD for me. Using the GUI first to visually piece together the tables made a lot more sense to me and saved me a lot of time and frustration. Using the built-in features to generate the DDL and ERD also allowed me to quickly recognize issues and design flaws, then make the necessary fixes to the database infrastructure.

After the tables, DDL, and ERD were complete I wrote the required test queries for each table.

The third part was spent creating the storyboard designs. For this part, I used SvelteJs with TailwindCSS, which for me is way more comfortable and enjoyable than using other wireframe

software. I can simply load a live server, then use HTML with utility classes to quickly prototype my screen designs. I am aware that my screen designs will not be consistent with the final game designs considering C# and windows forms will most likely be used to complete the project.

To complete the final part of this milestone, I used Microsoft Excel to create the CRUD table. The CRUD table outlines the interactions with the database that are required to complete various actions during the game's life cycle.

The time required to complete this milestone was immense and unfairly took large chunks of time needed for other classes, again. There was not much learned during this milestone because most of this planning process has been covered ad nauseam in prior classes. There are little to no relevant resources for creating a game in this manner online, and finding helpful and focused resources on the moodle course section added to frustrations and time lost.