WHO DID

CIS 1512: Software Engineering

Professor Hadi Nasser

FINAL REPORT

By “VEJN” Team Development AKA “Group 5”

Viktor, Ejmen, Nathan, Johnathan, Joey

12/03/23

Table of Contents

Introduction 1.0

Goals and Objectives 1.1

The game will be user-friendly, ensuring ease of use and accessibility for individuals of all ages. It will provide a straightforward way for users to connect to their accounts through our database. Our primary objective is to establish trust with our users and offer them an entertaining experience through an engaging game that embarks them on a quest to accumulate as much gold as possible. Additionally, we aim to encourage them to develop strategic thinking skills that can be applied not only in the game but also in real life.

Functional requirements:

- *The user should be able to create an account.*
- *The user should be able to set a PIN for their account.*
- *The user should have the option to label monsters with names.*
- *Instructions should be displayed on a separate page accessible via a button.*
- *Optional in-game music that the user can enable or disable.*
- *The "Start" button will transition the user to the next screen*
- *All pages should feature a button for returning to the previous screen or exiting the game.*
- *Navigation should be labeled with arrow buttons for different directions.*
- *The user can utilize a "Drink" button to collect energy and collect gold if available.*
- *A "Help" button can be used by the user to maintain their in-game energy.*
- *When a monster appears in one of the pictures, the user can choose to "Run" or "Fight."*
- *The "Try Again" button will keep the user on the Login page and start over*

Non- functional requirements:

- *Able to handle a growing user base and increased data without performance degradation*
- *The app should continuously monitor and optimize its performance to ensure fast response times and efficient resource utilization*
- *The app should have high availability with minimal downtime.*

- *Regular backups of user data should be maintained. (USB Usage or Hard drive for now)*

Statement of Scope 1.2

The scope of our app development is to create a game that is entertaining for everyone, engaging players in the knowledge that achieving a good ranking requires employing a sound strategy to stay in the game for as long as possible while maintaining sufficient energy to change direction. There will be one or more monsters present, whose purpose is to deplete your energy and impede your progress in the rankings.

This app will feature different screens that aid in navigating the game. Initially, there will be a welcome page with various ads aimed at improving the project's quality and growth. Following that, you will proceed to a page where to play, you will input a username and set a PIN to save your credentials securely. Once these credentials are added, a table displaying your ranking will be shown.

Within the game itself, as you move from side to side, the character's images will change to reflect the environment you are in. You will also have options available to assist you if you are on the verge of losing your energy. The game will conclude when all your energy has been depleted.

Software Context 1.3

Today, the gaming industry stands as one of the most profitable sectors globally. The introduction of streaming has revolutionized the way games are enjoyed, captivating not only the players themselves but also a worldwide audience. Game companies from around the globe continually innovate, striving to entertain users and engage people everywhere. The game we are developing promises to bring excitement as users compete against each other in unique ways to prolong their stay and collect gold, ultimately improving their ranking. Having competition in the game intensifies user interest. By going in different directions in the game, players can find tools to collect stuff based on where they go and switch paths to find more opportunities to win. The longer you play, the more chances you have to collect gold and compete with others in the game. Our game encourages players to think strategically and become the best.

Major Constraints 1.4

These simplified functionalities and constraints provide a clear focus on the core aspects of the project, emphasizing the user interface design, Java application development, and database integration.

- ➔ *Figma-Based User Interface Design*
- ➔ *Java Application Development*
- ➔ *Third-Party Library Restrictions*
- ➔ *Platform Independence*
- ➔ *Development Environment*
- ➔ *Code Quality and Style Guidelines*
- ➔ *Testing Constraints*
- ➔ *Database Integration*
- ➔ *Security Measures*

Usage Scenario 2.0

User Cases 2.1

This use case highlights the pivotal actions players take while engaging with the adventure game, from the initial login process to crucial in-game interactions that enhance the overall gaming experience.

User login information	*User login*
Actor	*Action*
Player	*Enters credentials, Adds PIN*
Program	*Match credentials with the one saved or not in the database*

Finding game instructions	*Instructions*
Actor	*Action*
Player	*Select the “Instructions” button*
Program	*Move the user to another interface to show the instructions*
Player	*Select the button to go back*
Program	*It returns to the preview page*

Start the game	*Start Button*
Actor	*Action*
Player	*Select the “Start” button*
Program	*Shows User raking and gold collection*
Player	*Select “Continue” button*
Program	*Enters the game*

Collecting energy and gold	*Drink Button*

Actor	*Action*
Player	*Select the "Drink" button*
Program	*The program collects and adds the gold to the user profile*
Player	*Moves to another direction*
Program	*Brings back three new images*

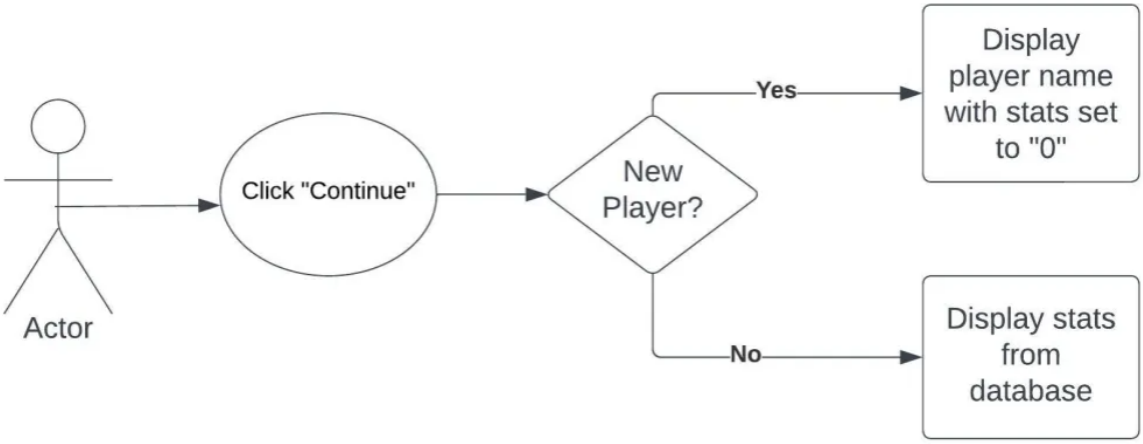
Avoiding the monster	*Run Button*
Actor	*Action*
Player	*Select the "Run" button when a monster appears*
Program	*If the user chose to run energy point will be subtracted*
Player	*Move to another direction*
Program	*Will back three new images*

Add more energy	*Energy Button*
Actor	*Action*

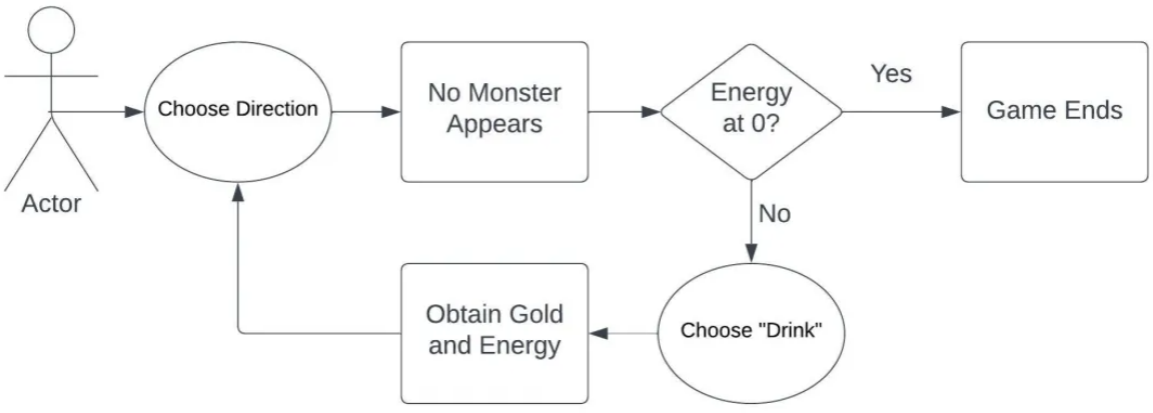
Player	*Select the “Help” button*
Program	*The program shows a picture in the help box to show that the user is taking more energy. Add the energy in the energy section, used only twice then vanishes.*

User Case-Diagram 2.2

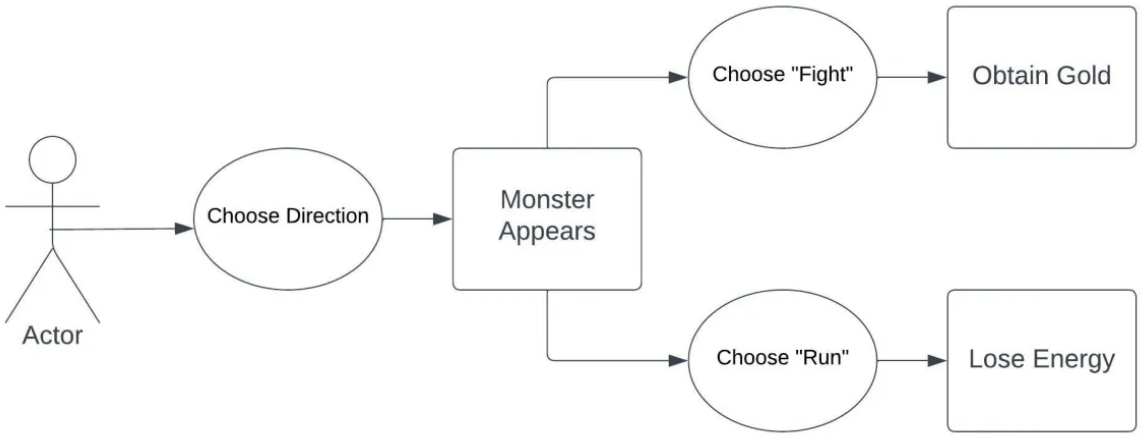
—---------- 1



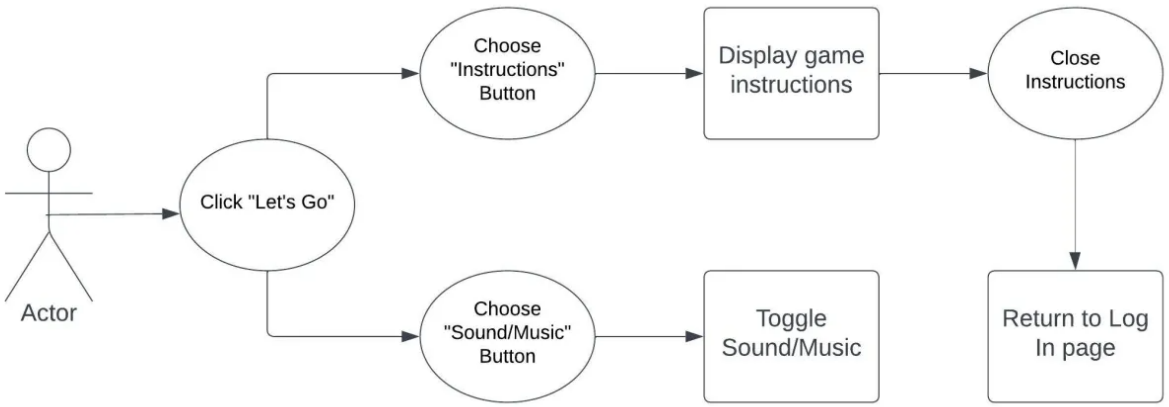
—---------- 2



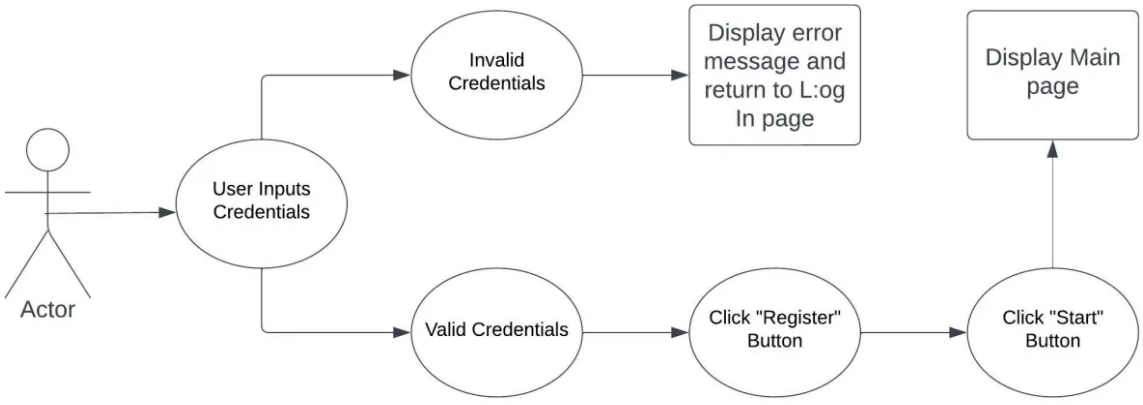
3


Actor
Choose Direction
Monster Appears
Choose "Fight"
Obtain Gold
Choose "Run"
Lose Energy

4


Actor
Click "Let's Go"
Choose "Instructions" Button
Display game instructions
Close Instructions
Choose "Sound/Music" Button
Toggle Sound/Music
Return to Log In page

5


Actor
User Inputs Credentials
Invalid Credentials
Display error message and return to L:og In page
Valid Credentials
Click "Register" Button
Click "Start" Button
Display Main page

Data Design 3.0

Internal Software Data Structure 3.1

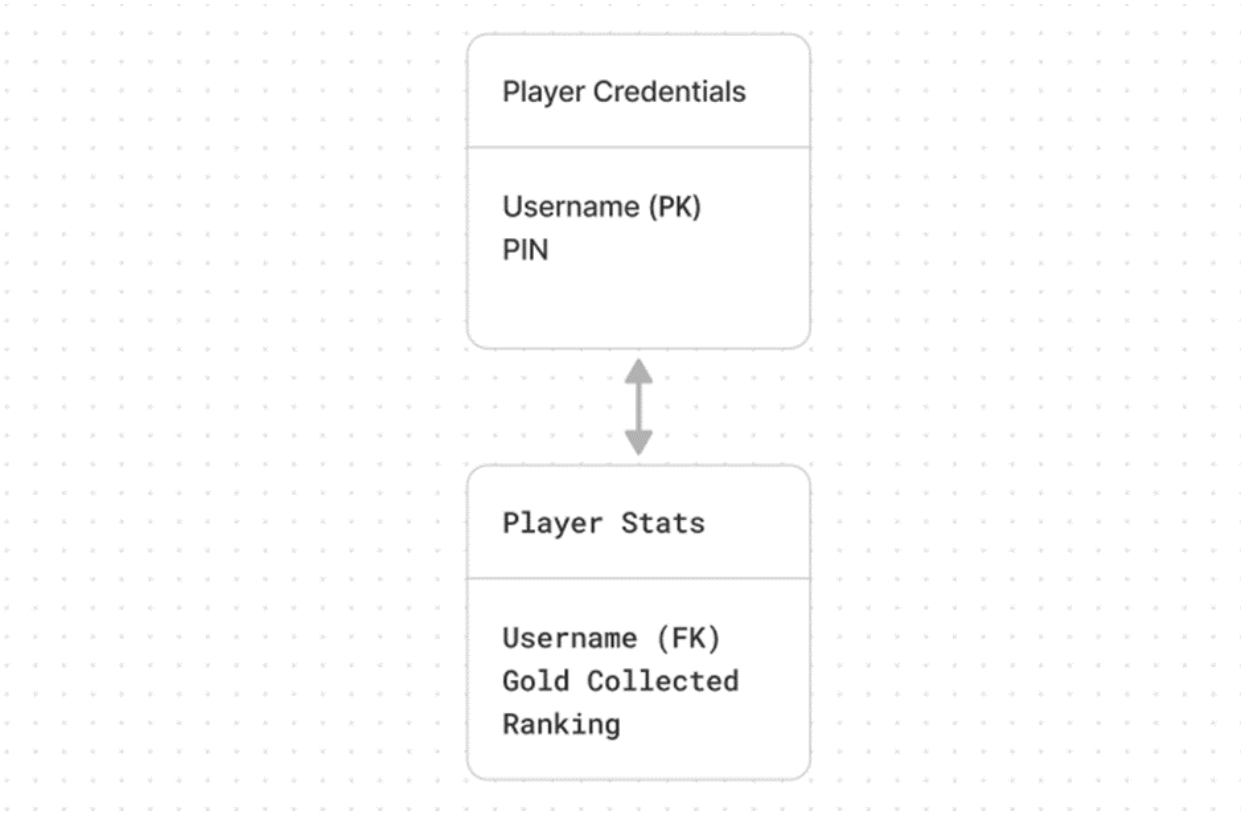
In our game project, we'll employ various data organization techniques to enhance player engagement. This includes arrays, ArrayLists, Enum for efficiently managing game data. We'll also utilize primitive data types like int and text data with Strings. Additionally, custom-designed objects will play a key role in implementing specific game features and functionalities, contributing to a polished and professional user experience.

Database Description 3.2

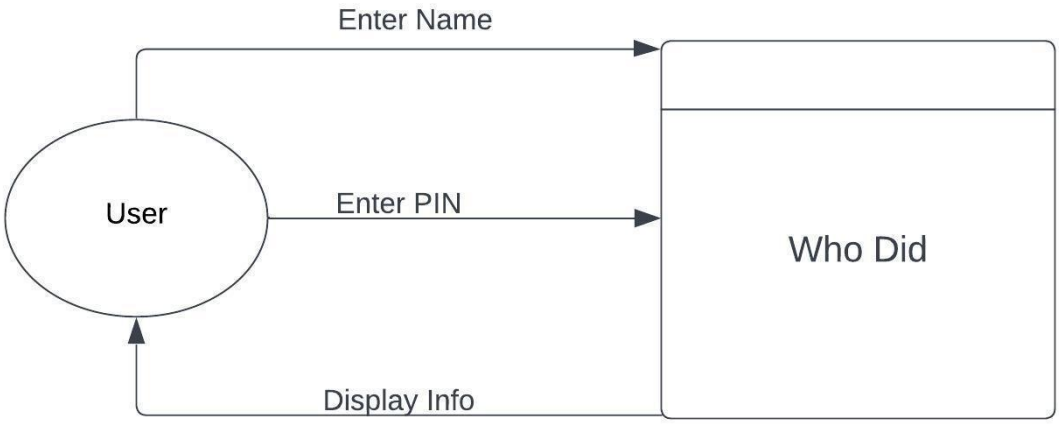
This game’s database consists of two primary tables:

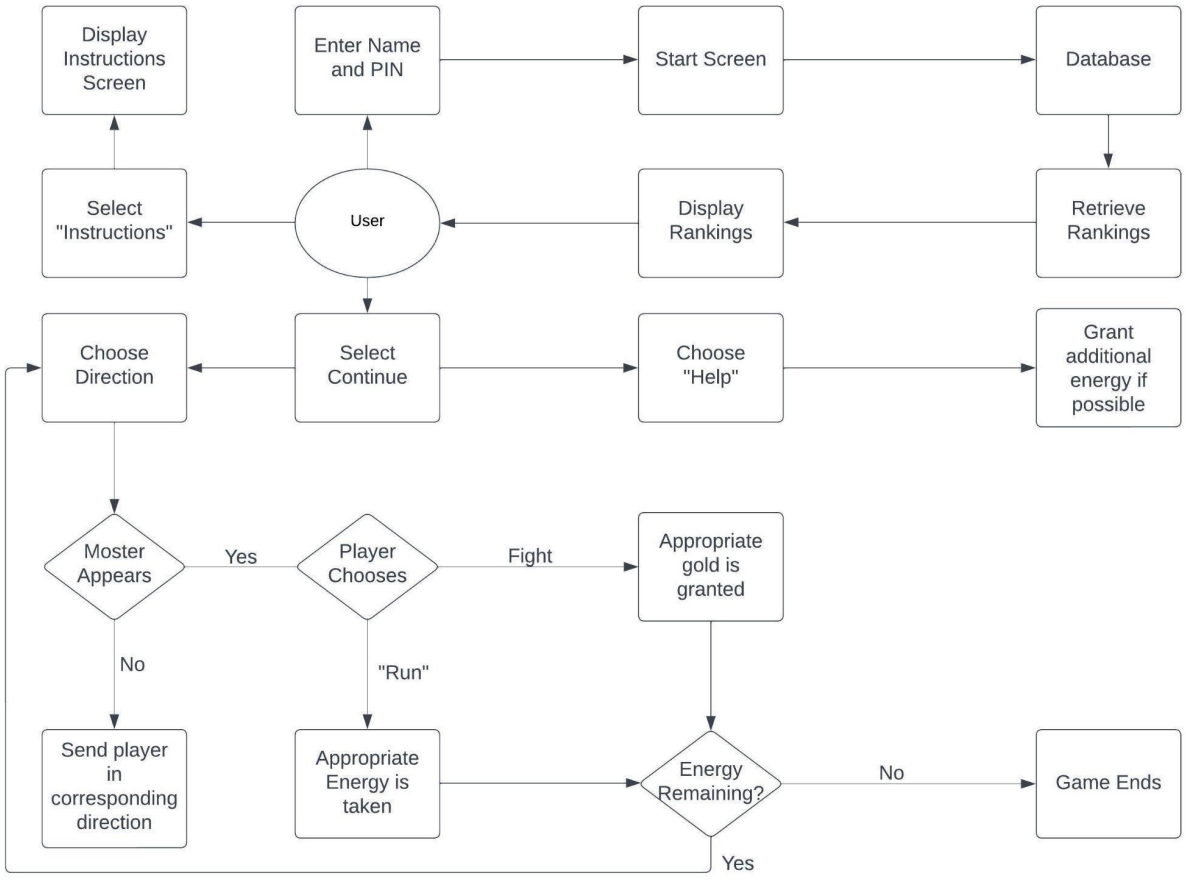
1. ***Player Credential Table:***
   STORES PLAYER CREDENTIALS
   - *Username(Primary Key): Player’s unique username.*
   - *PIN: For Authentication purposes*
2. ***Player Stats Table***
   TRACKS PLAYER RANKINGS AND GOLD
   - *Username(Foreign Key referring Player Credential): Linking the player stats*
   - *Rank: Player’s ranking within the game*
   - *Gold Collected: The amount of gold collected by the player.*

Data Model 3.3



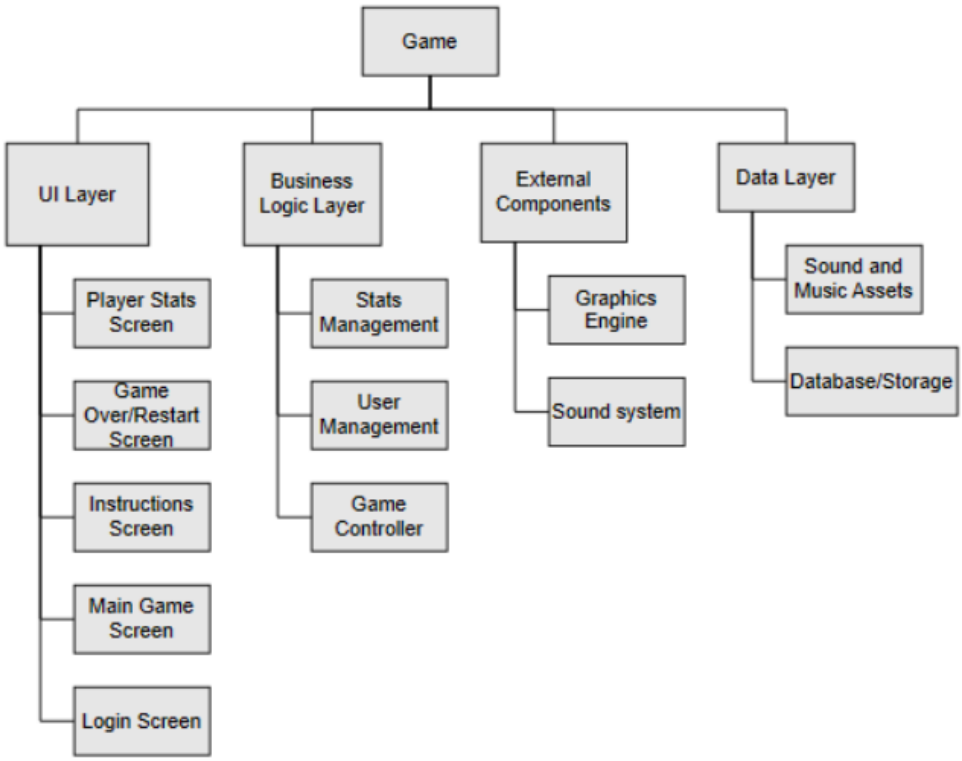
Data Flow Diagram 3.4





Architectural- Component-Level Design 4.0

Program Structure 4.1



Functional Model & User Interface Design 5.0

Software Interface Description 5.1

The game interface seamlessly integrates with a secure database system, allowing users to create personalized accounts with unique usernames and PINs for secure access. Upon login, users can view their rankings and accumulated gold, which are retrieved from the database, providing a tailored gaming experience. After accessing their account details, users are directed to the main game interface, where they interact with the game. This database connectivity enhances user engagement and personalization within the game.

External machine interfaces 5.1.1

This application will operate independently without any external device attachments or dependencies.

External system interfaces 5.1.2

This application will not be linked to external systems.

Human interface 5.1.3

The user interface is designed to offer users a significantly improved and distinct experience compared to the initial game launch, featuring a cleaner design and more streamlined navigation for a more enjoyable and extended gameplay experience.

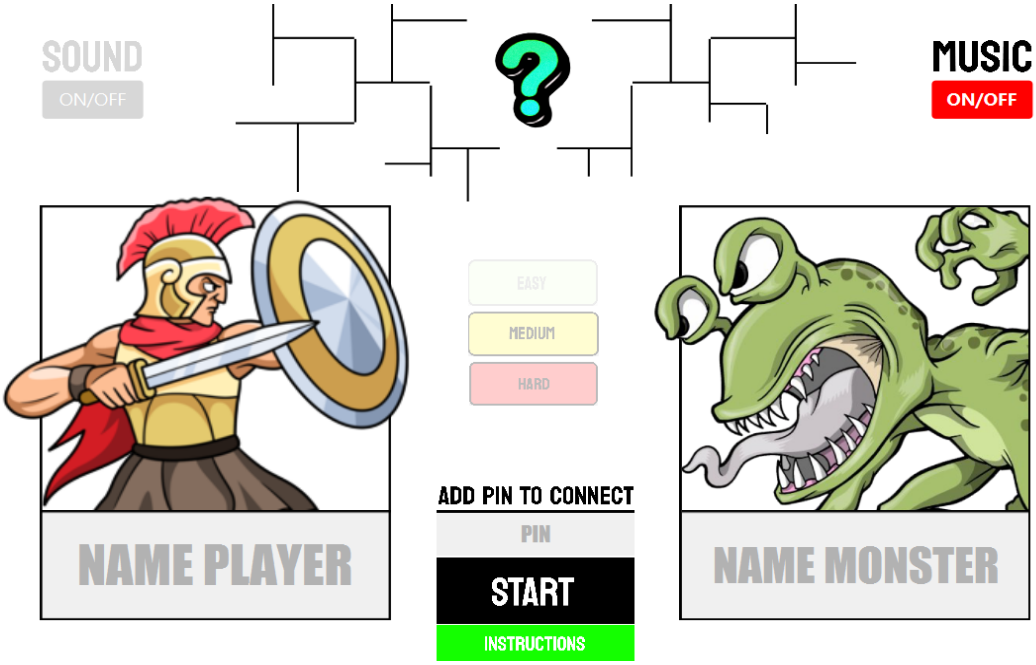
Description of the User Interface 5.2

1. *Welcome Screen*
    - *THE INITIAL INTERFACE WHEN THE GAME IS LAUNCHED*
2. *User Input Screen*
    - *THE SCREEN FOR THE USER CREDENTIALS TO REGISTER*
3. *Instruction Button*
    - *ACCESS TO A PAGE THAT EXPLAINS THE GAME’S RULES*
4. *Rank and Gold Screen*
    - *THE PLAYER RANKING WITHIN THE GAME*
5. *Main Page*
    - *SHOWS THE STATS THE NAVIGATION AND THE IMAGES)*
6. *End game*
    - *SHOWS THE TABLE AND OPTION TO START OVER*

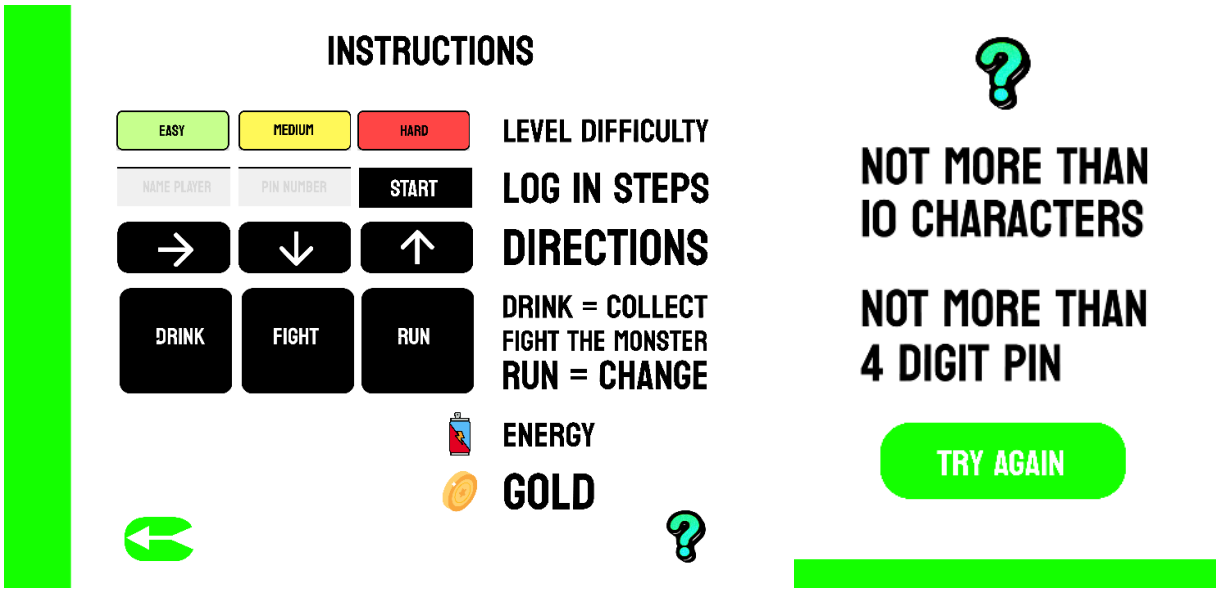
Image of Screens 5.3

—---------- *Welcome Screen*

—----------*Login Screen*



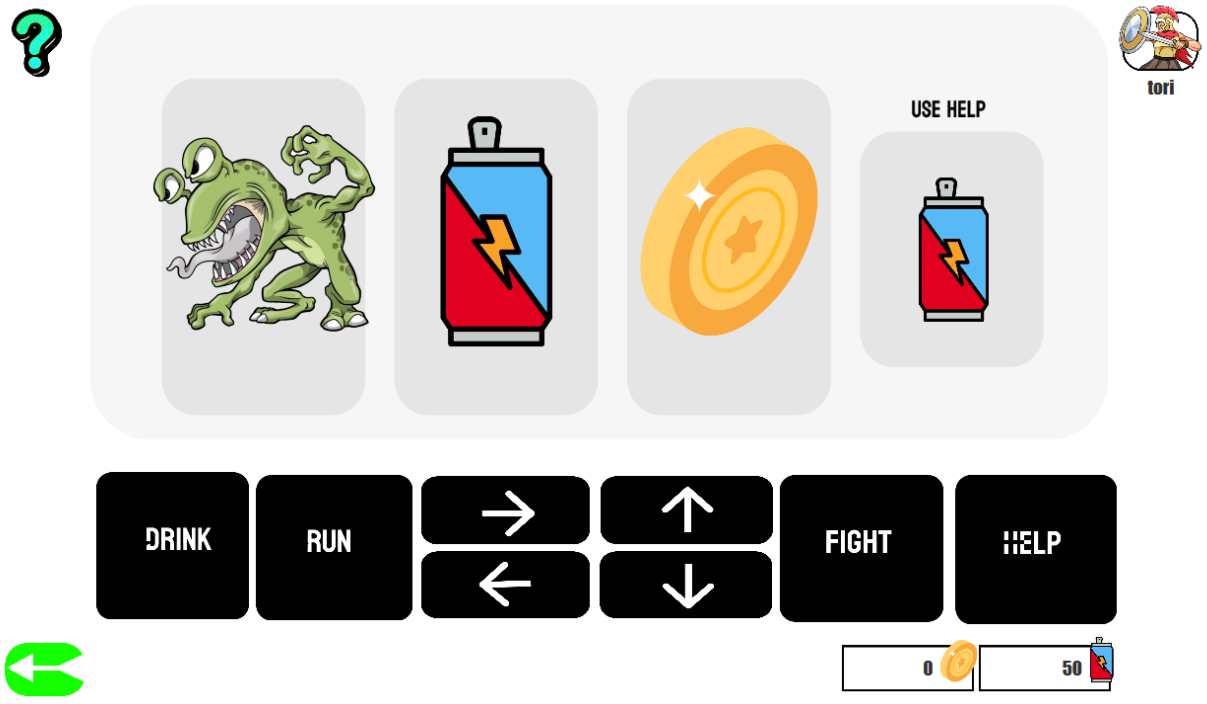
—----------*Instruction Button & Bad Login*



—----------*New User and Existing User*

—----------Rank and Gold Display

—----------Main Page



—----------*End Game*

Objects and Actions 5.4

SCREEN	OBJECTS	ACTIONS
Welcome Screen	*- Logo and Title* *- Background visuals*	*- Button to proceed to the Input Screen*
Input Screen	*- Username and PIN input fields* *- Monster Name (OPTIONAL)* *- Easy, Medium, Hard buttons* *- Instructions button* *- Start button* *- Sound and Music toggle*	*- Input fields for username and PIN* *- Selection buttons for difficulty levels* *- Open the instructions page when the Instructions button is clicked* *- Proceed to the next page when the Start button is clicked* *- Toggle sound and music on/off*
Instructions Page	*- Game rules and instructions* *- Back Button*	*- Return to the Input Screen when the Back button is clicked*

Rank and Gold Screen	*- Player ranking and gold labels* *- Continue button*	*- Proceed to the Main Page when the Continue button is clicked*
Main Page	*- Labels* *- Text fields* *- Buttons* *- Images*	*- Interact with in-game elements and features* *- Navigate to different game modes*
End Game Screen	*- Image* *- Text Field* *- Restart button*	*- Restart the game when the Try Again button is clicked*

Interface Design Rules 5.5

1. *Use large and legible fonts to ensure information is easily readable, even on smaller screens.*
2. *Implement appropriately sized and labeled buttons for seamless navigation, ensuring users can easily move between screens and functions.*
3. *Present instructions and help content in a well-organized manner, ensuring users can quickly find the information they need.*
4. *Include a readily accessible back button to allow users to navigate backward within the interface at any point.*
5. *Keep the interface simple and free from clutter, focusing on a straightforward design that users can easily understand.*
6. *Clear error messages and guidance for users when they encounter issues or mistakes within the interface.*
7. *Include icons and visual style to identify and recognize the game easily.*

Components Available 5.6

➔ ***Text Labels:*** *Used for displaying game titles, instructions, and in-game information.*

- ➔ ***Buttons:*** *Interactive elements for navigating between screens and triggering actions.*
- ➔ ***Input Fields:*** *Fields for players to enter their username and PIN during the login process.*
- ➔ ***Images:*** *Visual elements for displaying game graphics, characters, and backgrounds.*
- ➔ ***Icons:*** *Symbolic representations for in-game actions or features.*
- ➔ ***Toggle Switches:*** *Controls for enabling or disabling in-game sound and music.*
- ➔ ***Backgrounds:*** *Graphics or colors to set the visual theme for each screen.*

Behavioral Model and Description 6.0

Description of Software Behavior 6.1

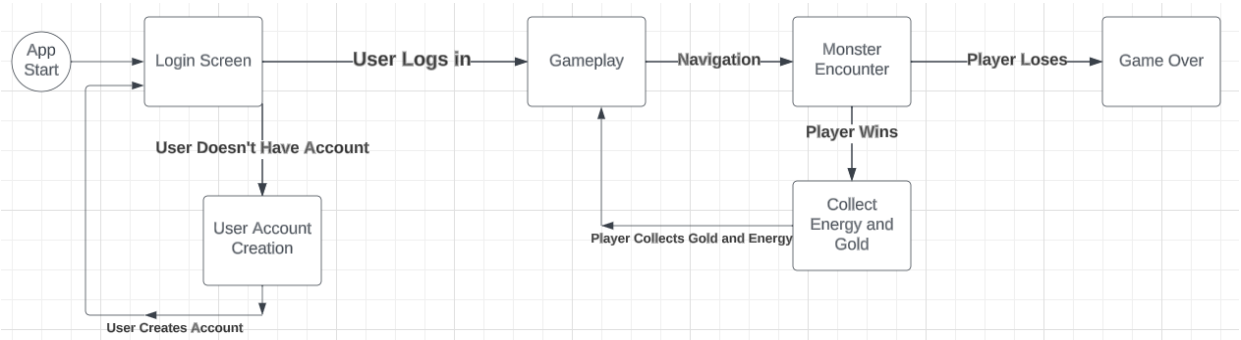
Events 6.1.1

- ***User Account Creation***
- ***User Login***
- ***Game Start***
- ***Navigation***
- ***Collect Energy and Gold***
- ***Monster Encounter***
- ***Game Over***

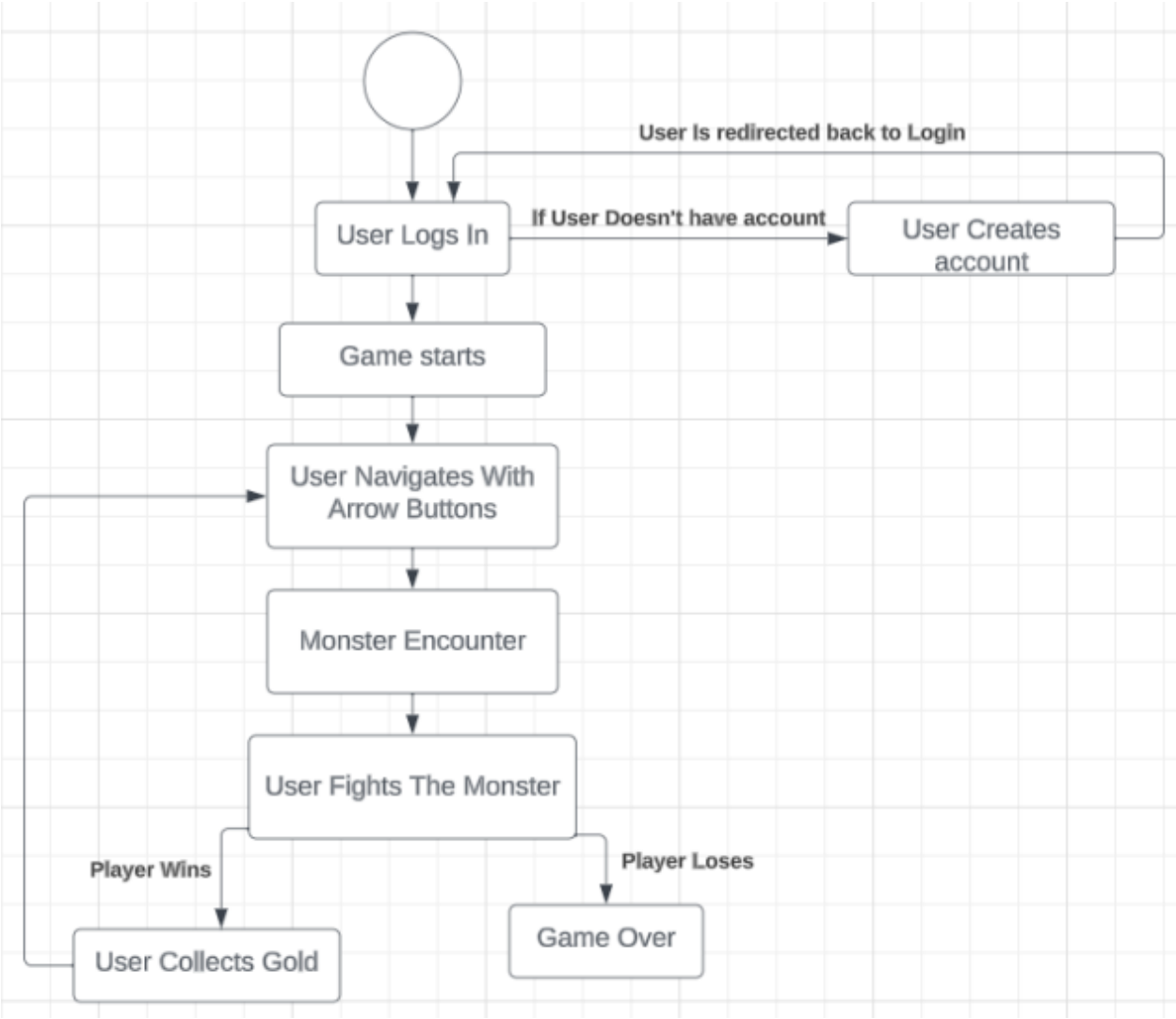
States 6.1.2

- ***Login Screen:*** *The initial state where users can create accounts or log in.*
- ***Gameplay:*** *The primary state where users navigate with arrows, collect resources, and encounter monsters.*
- ***Game Over State:*** *When a user's energy is depleted or they choose to start over, the game transitions to this state, displaying a game over message.*

State Transition Diagrams 6.2



Control specification 6.3



Restrictions, Limitations, and Constraints 7.0

---

1. System Requirements

MINIMUM REQUIREMENTS:

- *Windows 7 (32-bit) (compatible with macOS and Linux)*
- *4 GB RAM memory*
- *Intel Graphics*
- *1GB available storage*

2. Network Connectivity:

- *The game does not require an internet connection for any feature, it can be played offline.*
- *Updates may be downloaded when an internet connection is available.*

3. Input Methods:

PLAYER NAVIGATION

- *Keyboard: To input user credentials.*
- *Mouse: To press the button on the screen.*

Validation Criteria 8.0

Classes of tests 8.1

- ***Functional Testing:*** *Ensure that all game features work as intended.*
- ***Usability Testing:*** *Evaluate the user-friendliness of the game.*
- ***Security Testing:*** *Verify the security of user data, creation, and log in.*
- ***Performance Testing:*** *Assess the game's performance, including load time, and user input.*
- ***Regression Testing:*** *Test to ensure that new changes do not break existing functionality.*
- ***Database Testing:*** *User account data, rankings, and gold collections are accurately stored.*

Expected software response 8.2

User will be able to:

- **Create a Profile:** Enter and store a name and PIN into the database.
- **View Instructions:** Select “Instruction” to see game controls and return to the main page.
- **Start the Game:** Select “Start” to view the current ranking, the “Continue” to begin the game.
- **Move:** Select and travel in a direction.
- **Collect items:** Select “Drink” to add gold and energy to the player profile.
- **Obtain Extra Energy:** Select “Help” to check if additional energy is available to the player and collect it.

Test Issues 9.0

Classes of Tests 9.1

1. ***Functional Testing*** *- All game components, including buttons, and actions, work as expected.*
2. ***Visual Testing*** *- Images are displayed correctly with good resolutions, look for inconsistencies in graphics.*

3. ***Music and Sound Testing*** *- Check if they work any time someone tries to turn off or on the music or sound.*
4. ***Performance Testing*** *- Verify that the game runs smoothly without having to wait for a long time to start the game.*
5. ***User Acceptance Testing*** *- Involve a small group of players or testers to play through the game, and use the input to identify issues.*
6. ***Security Testing*** *- Make sure the Database connection works as expected and the credentials allow the user to continue.*

Test Cases 9.2

Test Case ID	*Description*	*Input*	*Expected Outcome*	*Actual Outcome*	*Pass/Fail*
T-1	The program should run without any errors, displaying the "Welcome" page on the first run.	Run Command	No errors on the first execution, The Welcome page displayed	Welcome page displayed without errors	Pass
T-2	The game should automatically start with sound in each button and background music upon execution.	Run Command	Every time the game starts, the background music will begin automatically, and pressing the "Let’s Go" button will trigger the clicking sound.	The user launches the game, and the music starts simultaneously. When the user clicks the "Let Go" button, a clicking sound is heard, indicating that the button has been pressed.	Pass
T-3	Clicking the "Let's Go" button on the initial	Press the “Let’s Go” button	Displays Log In page	Log In page appears after	Pass

	page should transition to the "Log In" page.			the button click	
T-4	On the Log-in page, users can toggle sound and music on/off, as we included these features automatically in the program.	Toggle buttons “Music” and “Sound”	Toggling turns off/on sound and music	On the Log-in page, the music and sound are initially on. To mute them, the user clicks the adjacent buttons for sound and music.	Pass
T-5	The "Instruction" button on the Log-in page should lead to a page explaining the game, with an option to return to the Log-in page.	Press the "Instruction" button	Pressing this button is expected to navigate the user to another page explaining how the game works. In the same window, there's a button to easily return to the Log-in page.	Clicking the Instruction button on the Log-in page hides it and takes the user to the instructions. After reviewing, the user can return to the Log-in page using the arrow button.	Pass
T-6	Testing user registration with credentials exceeding 10 characters for the username and 4 digits for the PIN.	Add credentials and press "Start"	After entering credentials with more than 10 characters for the username and more than 4 digits for the PIN, it is expected that a table will appear, reminding users to review their credentials.	On the Log-in page, if the user inputs credentials with more than 10 characters for the username or 4 digits for the PIN and presses start, a table will appear, indicating an	Pass

				issue with the credentials.	
T-7	Testing program response for new users with correct credentials	Add credentials and press "Start"	After a new user enters the correct credentials and presses start, a table will confirm the credentials but remind the user that they are new to the database.	After the user enters new credentials and presses start, a table appears, reminding the user that these credentials are new to the database. Two buttons are provided, one to retry and the other to register the credentials.	**Pass**
T-8	Testing the table that saves credentials using the "Register" button.	Click the "Register" button	For new users, before it shows the new “Stats of the Player” table we expect that the program asks the user to register first by using the Register button after that, the table will close and after pressing Start the “Stats of the Player” table will show up.	Using the Register button on the new table saves the data in the database, simultaneously assigning a new ranking to the user. Upon pressing Start, the first thing that will appear for the new user is a table displaying the ranking and gold collection.	**Pass**

T-9	Testing the connection between the "Player Stats" table and the "Main" table where the game is based.	Press the "Continue" button	Before reaching the Main page, every user (new or existing) sees a table displaying player ranking and gold collection (set to "0" for new users). Pressing the Continue button then takes them to the game on the Main page.	After obtaining the Player Stats table, users review the stats from the database. Then pressing the Continue button directs the user to the Main page, where the game is ready to start playing.	**Pass**
T-10	Personalize the Main page with the correct username, energy, and gold labels.	Press the "Continue" button	It is expected that after reviewing stats, the Main page will personalize by displaying the username beside the player image. Simultaneously, the labels for energy and gold will initialize to 100 and 0, respectively.	After the user presses continue and the Main page appears, it will display the username that they logged in next to the player image, along with the initialized points for energy and gold available in one game session.	**Pass**
T-11	Testing direction buttons on the Main page.	Press direction buttons	It is expected that to start the game, when the user presses one of the direction buttons, three different images will display, indicating what can be collected,	The game is based on a labyrinth where the user selects a direction to go, and in that direction, they will find	**Pass**

			whether there is anything to collect, or if they are dealing with a monster.	gold, energy, or a monster that will try to stop them. When it comes to directions in the game, the user presses the direction button, and the table will show what that direction includes.	
T-12	Testing the "Drink Button" to collect energy or gold.	Click the "Drink" button	It is expected that when there are no monsters in the images, the user can collect gold and energy. But it is expected that when there are monsters this button will be disabled to collect till you fight the monster first. The only button enabled will be the directions buttons to move to another direction.	When the direction is pressed and there is no monster in all three images, the user can click the Drink button to collect gold and energy. The amount collected will then be shown on the respective labels for gold and energy. However, this button won’t be clickable when there is a monster.	**Pass**
T-13	Testing the "Run Button" for escaping without collecting.	Click the "Run" button	It is expected that every time the user runs away to regain energy, it is not anticipated to	For this button, if the user has chosen a direction and there are no	**Pass**

			collect from the table, only to be able to choose a new direction.	coins or energy but only monsters, they can choose to run from that direction in exchange for energy.	
T-14	Testing the "Fight Button" for battling monsters.	Click the "Fight" button	It is expected that while the monster is present in one of the images, the user cannot Drink, but they can Fight the monster. Here, there is an opportunity for the user not to lose any energy, and after the fight, they can then choose to collect gold or energy.	After the direction button is pressed, if the user wants to collect energy or gold but encounters a monster, they have to fight to be able to collect. Fighting the monster saves energy for the user, allowing them to collect afterward.	Pass
T-15	Testing the "Help Button" for gaining extra energy twice.	Click the "Help" button	This button is designed to help the user collect more gold during the game for a better ranking. It is expected to be usable only twice during one game session and will be automatically disabled.	In the game, when pressed by the user, this button gives back energy and displays it in the energy label. It is never the exact amount rather, it is a random number	Pass

				added to the user’s energy. After the user presses this button twice, it is disabled.	
T-16	Testing when the user runs out of energy, ending the game.	The user loses all energy points	At one point in the game, the user will end the game by losing all their energy, and it is expected that the game will stop automatically, showing another table that concludes the game by telling the user "Game Over." The collected data are supposed to be saved in the database.	When the player plays the game, the energy decreases until it reaches 0. The user is then notified by a table that the game is over. When they log back into the game, the collected gold will be added to their stats.	**Fail**
T-17	Restarting the game after a "Game Over" with the "Try Again" button.	Click the "Try Again" button	In the new table that will show up after the user loses all the energy, there is going to be a button that will help the user start over the game with the same 100 energy and 0 gold.	After the user ends up with 0 energy and the table shows up, there is a "Try Again" button. When the user presses the button, redirect them to the login page to restart.	**Pass**

BLACK-BOX Testing
Focused on the functionality of the application without knowing its internal code structure. This method is suitable for testing the user interface, game mechanics, and overall user experience.
USER ACCEPTANCE Testing (UAT)
Invited volunteers from different backgrounds, ages, and gaming experiences
DIVERSE USER GROUP
Invited volunteers from different backgrounds, ages, and gaming experiences to participate in UAT. This ensures a wide range of perspectives and feedback.
BLIND Testing
Participants were not informed about the specific aspects being tested to avoid biasing their responses.
CONCLUSION
The combination of black-box testing and UAT, along with a diverse group of testers, ensures a thorough and unbiased evaluation of the game. This approach helps identify usability issues, bugs, and areas for improvement from both a technical and user perspective.

Project Estimates 10.0

See below for COCOMO II settings and estimates.

Results

Software Development (Elaboration and Construction)

Effort = 1.5 Person-months
Schedule = 4.2 Months
Cost = $15181

Total Equivalent Size = 500 SLOC
Effort Adjustment Factor (EAF) = 1.10

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.1	0.5	0.2	$911
Elaboration	0.4	1.6	0.2	$3643
Construction	1.2	2.6	0.4	$11538
Transition	0.2	0.5	0.3	$1822

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.0	0.1	0.0
Environment/CM	0.0	0.0	0.1	0.0
Requirements	0.0	0.1	0.1	0.0
Design	0.0	0.1	0.2	0.0
Implementation	0.0	0.0	0.4	0.0
Assessment	0.0	0.0	0.3	0.0
Deployment	0.0	0.0	0.0	0.1

Risk Management 11.0

Risk Name	*Risk Description*	*Impact Probability (High, Medium, Low)*	*Mitigation*	*Contingency Plan*
Out Sick	*Team member falls ill and is unable to work on the project*	*Medium*	*The sick team member is assigned the appropriate extra work to offset the time missed*	*The remaining team members work together to cover the missing portion*

File Corruption	*Project file(s) become corrupted*	*High*	*Team members save work often*	*Team members create multiple backups to ensure progress is not compromised*
Compile Errors	*Compiler finds an error in the code*	*High*	*Locate the code causing the error and rectify it*	*Team members consult each other to find a solution to the issue*
Change in program design	*Something may cause the program design to require tweaking*	*Medium*	*Frequent communication between team members to catch and implement possible necessary changes*	*More time will be dedicated to any required changes that cannot be handled by a single team member*

Project Schedule 12.0

—----------

Sprint 1					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase1.1 - Project Proposal	- Create Discord Group - Pick the project to work on - Assign Roles - Schedule Zoom meetings	09/09/23	09/11/23	All team members	100%

	- Plan what needs to be done in advance				
Final Proposal for the project	Find the right name				100%
Zoom Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
	Questions to be asked				
Sprint 2					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase 1.2 SRS/SPMP	- Use Google docs to include what is needed for the project. - Use Figma to use details to create the project so all the members understand - Separate the work between each other	09/15/23	09/25/23	SCRUM Masters and UI members	100%
Figma Design	Finish the idea and share with the others			UI Members	100%
Database Design	Create the database with all the entities that are needed			DEV Members	100%

Research code	Find similar project that will make the work faster			DEV Members	100%
Diagrams	Creating Diagram with different use cases			SCRUM Masters	100%
Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
Sprint 3					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase 1.3 Prototype Preparation	- PowerPoint Preparation - Finish the details needed to present the game - Prepare the speech	09/27/23	10/02/23	IU and SCRUM Master Members	100%
Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
Sprint 4					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase 2 SDD	- Use Google docs to include what is	10/03/23	10/10/23	All team members	100%

	needed for the project. Remember: - Make sure nothing is missing in the design part -Everything is included in the document that we have to upload				
Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
Sprint 5					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase 3 Implementation	-Ensure program works correctly	10/11/23	10/30/23	All team members	100%
Debug the software	-Implement major bug fixes			DEV members	100%
Review if everything required is working	-Test software for necessary fixes			All team members	100%
Database included	-Ensure database is connected			DEV members	100%
	After class meeting	Every week		All team members	100%

Meetings	Second Meeting	Review before class		All team members	100%
Sprint 6					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase 4.1 V&V Testing	-Ensure all components function as expected	11/01/23	11/10/23	All team members	100%
Test Database	-Keep in mind that the login part should function perfectly for the game to move one			DEV/UI members	100%
Test Buttons	-Each button is a priority everything should work before the project is ready to be presented			All team members	100%
Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
Sprint 7					
Task	Subtask	Time Start	Time End	Resource	Done %
Final Project Report	-Create final project document which includes all expected	11/11/23	11/19/23	All team members	100%

	design and development details				
Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
Sprint 8					
Task	Subtask	Time Start	Time End	Resource	Done %
Phase 4.3 Post-Mortem (Evaluation)	-Evaluate fellow team members	11/30/23	12/01/23	All team members	100%
Meetings	After class meeting	Every week		All team members	100%
	Second Meeting	Review before class		All team members	100%
Sprint 9					
Task	Subtask	Time Start	Time End	Resource	Done %
Presentation Final Project (DEMO)	-Present final project	12/02/23	12/09/23	All team members	100%
Meeting	LAST MEETING (Thank you meeting)				100%

Maintenance 13.0

Future Enhancements 13.1

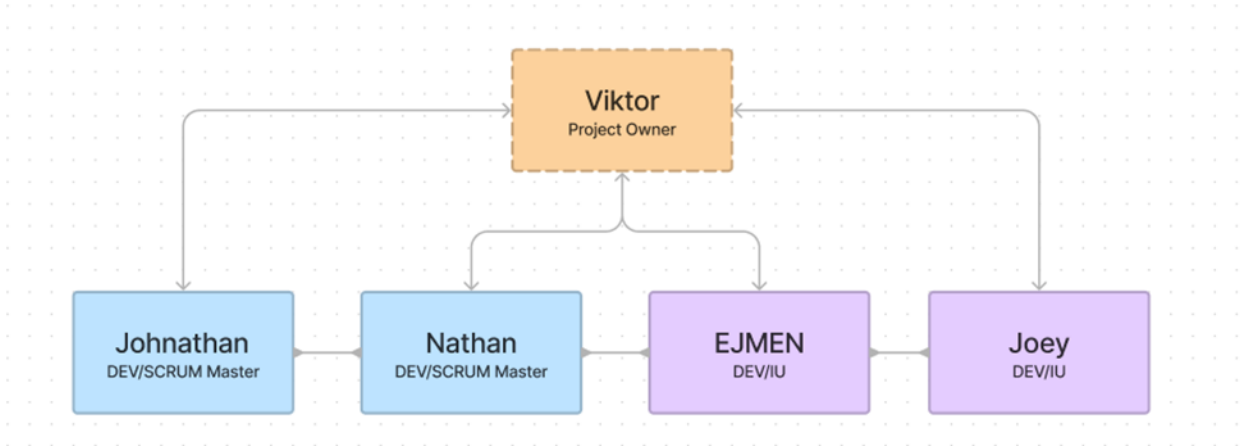
In the future, we aim to connect our game to the internet, allowing users worldwide to play together. We also plan to add different difficulty levels to make the game more interesting for all players. To enhance the user experience, we'll introduce animated elements like dynamic buttons. These improvements are geared towards creating a global, engaging, and visually appealing gaming platform.

Refactoring & Lessons Learned 13.2

In both the MainClass and LoginPage, our primary focus during development was completing the game functionality rather than emphasizing code quality. As a result, the code lacks consistency in naming conventions, leading to confusion and difficulty in locating specific components. In future iterations, we recognize the importance of prioritizing clean and well-organized coding practices to enhance readability and maintainability. This experience highlights the significance of striking a balance between feature development and code craftsmanship for a more cohesive and sustainable project.

A lesson we learned as a group is the importance of code review and documentation. With a new member joining our group, having clear documentation made it easier for everyone to understand and prevented anyone from feeling isolated within the team.

Staff Organization 14.0



Appendices 15.0

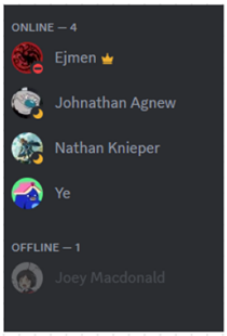
Configuration Management section 15.1

1. *Google DOCS*

***Document**: WHO DID SRS & SPMP*

2. *Discord Group*

Team Communication

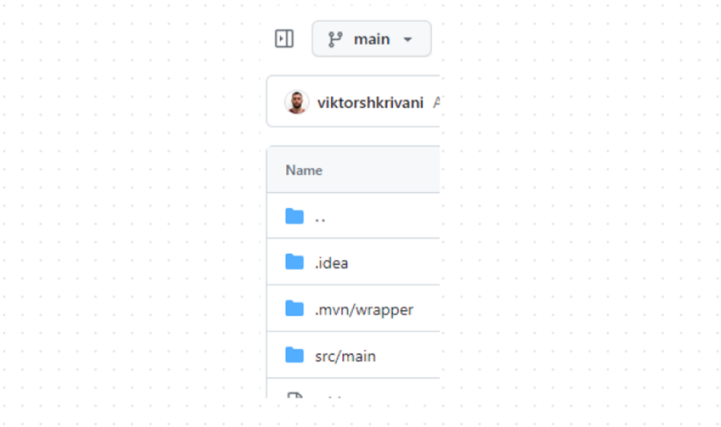


3. *Figma Design*

Model tracking and Design

4. *GitHub*

Source code tracking



Reference List 15.2

GitHub
https://github.com/viktorshkrivani/Who-did.git
SRS
http://www.rspa.com/docs/Reqmspec.html
SPMP
http://www.rspa.com/docs/Projectplan.html
COCOMO
http://softwarecost.org/tools/COCOMO/data/COCOMO_September_26_2023_22_19_07_450452.txt