A Pirate's Life Project Report

Prepared by
Michael Chau, Jon Heikkinen, Alen Peter, Alejandro Cabral
***for use in* CS 440**
at the
University of Illinois Chicago

Table of Contents

I Project Description 8

1 Project Overview 8

2 The Purpose of the Project 8

2a The User Business or Background of the Project Effort 8
2b Goals of the Project 9
2c Measurement 9

3 The Scope of the Work 10

3a The Current Situation 10
3b The Context of the Work 11
3c Work Partitioning 12
3d Competing Products 13

4 The Scope of the Product 13

4a Scenario Diagram(s) 14
4b Product Scenario List 17
4c Individual Product Scenarios 18

5 Stakeholders 19

5a The Client 19
5b The Customer 20
5c Hands-On Users of the Product 20
5d Maintenance Users and Service Technicians 21
5e Other Stakeholders 21
5f User Participation 22
5g Priorities Assigned to Users 22

6 Mandated Constraints 23

6a Solution Constraints 23
6b Implementation Environment of the Current System 24
6c Partner or Collaborative Applications 24
6d Off-the-Shelf Software 25
6e Anticipated Workplace Environment 25

7 Naming Conventions and Definitions 25

7a Definitions of Key Terms 25

7b UML and Other Notation Used in This Document 25
7c Data Dictionary for Any Included Models 27

8 Relevant Facts and Assumptions 27

8a Facts 27
8b Assumptions 27

II Requirements 29

9 Product Use Cases. 29

9a Use Case Diagrams 29
9b Product Use Case List 31
9c Individual Product Use Cases 34

10 Functional Requirements 61

11 Data Requirements 63

ID# D-1 - Player-account object 63
ID# D-2 - Player-character object 63
ID# D-3 - Friendly NPC object 64
ID# D-4 - Enemy NPC Object 64
ID# D-5 - Consumable Object 64
ID# D-6 - Offense Equippable Object 65
ID# D-7 - Defensive Equippable Object 65
ID# D-8 - Ship Object 65
ID# D-9 - Island Object 66
ID# D-10 - Town Object 66

12 Performance Requirements 66

12a Speed and Latency Requirements 66
12b Precision or Accuracy Requirements 67
12c Capacity Requirements 68

13 Dependability Requirements 68

13a Reliability Requirements 68
13b Availability Requirements 69
13c Robustness or Fault-Tolerance Requirements 69
13d Safety-Critical Requirements 69

14 Maintainability and Supportability Requirements 69

14a Maintenance Requirements....69
14b Supportability Requirements....70
14c Adaptability Requirements....71
14d Scalability or Extensibility Requirements....72
14e Longevity Requirements....72

15 Security Requirements....73

15a Access Requirements....73
15b Integrity Requirements....73
15c Privacy Requirements....73
15d Audit Requirements....74
15e Immunity Requirements....74

16 Usability and Humanity Requirements....74

16a Ease of Use Requirements....74
16b Personalization and Internationalization Requirements....75
16c Learning Requirements....75
16d Understandability and Politeness Requirements....75
16e Accessibility Requirements....76
16f User Documentation Requirements....76
16g Training Requirements....76

17 Look and Feel Requirements....76

17a Appearance Requirements....76
17b Style Requirements....77

18 Operational and Environmental Requirements....77

18a Expected Physical Environment....77
18b Requirements for Interfacing with Adjacent Systems....77
18c Productization Requirements....78
18d Release Requirements....78

19 Cultural and Political Requirements....78

19a Cultural Requirements....78
19b Political Requirements....79

20 Legal Requirements....79

20a Compliance Requirements....79
20b Standards Requirements....79

21 Requirements Acceptance Tests 80
21a Requirements – Test Correspondence Summary 80
21b Acceptance Test Descriptions 84

III Design 94

22 Design Goals 94

23 Current System Design 94

24 Proposed System Design 95
24a Initial System Analysis and Class Identification 95
24b Dynamic Modelling of Use-Cases 97
24c Proposed System Architecture 98
24d Initial Subsystem Decomposition 99

25 Additional Design Considerations 100
25a Hardware / Software Mapping 100
25b Persistent Data Management 100
25c Access Control and Security 101
25d Global Software Control 101
25e Boundary Conditions 101
25f User Interface 102
25g Application of Design Patterns 105

26 Final System Design 105

27 Object Design 105
27a Packages 105
27b Player and Environment Subsystem 106

IV Project Issues 108

28 Open Issues 108

29 Off-the-Shelf Solutions 108
29a Ready-Made Products 108
29b Reusable Components 108
29c Products That Can Be Copied 108

30 New Problems 109

30a Effects on the Current Environment....109
30b Effects on the Installed Systems....109
30c Potential User Problems....109
30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product....109
30e Follow-Up Problems....109

31 Migration to the New Product....109

31a Requirements for Migration to the New Product....109
31b Data That Has to Be Modified or Translated for the New System....109

32 Risks....110

33 Costs....110

34 Waiting Room....111

35 Ideas for Solutions....113

36 Project Retrospective....113

V Glossary....114

VI References / Bibliography....114

VII Index....115

I Project Description

The main objective of this project is to create a massively multiplayer online roleplaying game (MMORPG) that will be centered around the genre of pirates. Players will be able to select whichever region they wish to be immersed in and each region will have a unique atmosphere and environment to it. The reason why this project needs to be produced is because the pirate theme is very underrepresented in the video games industry. The product would be purchased by both pirate connoisseurs and people who have any interest in the genre. The customers who have purchased the product will be able to continuously play the game as long as their membership lasts and be immersed in the environment as well as making progress towards the endgame. The game will be designed using the newest Unreal Engine in order to create the most hyper-realistic environment possible and platform release is tentative depending on if any exclusive partnership opportunities arise.

1 Project Overview

As mentioned above, the project is based on the development of a hyper realistic online multiplayer role playing Pirate simulation game. The game in its entirety will focus on bringing the user the most immersive experience possible. In the current market there are very few games that are the same if not similar in the genre the team are trying to target, however, they do not truly give the most immersive experience, nor are they as hyper realistic. The game will begin where the player chooses what server to start off in, where this location decides the player's experience until they unlock the other region. Once a server is joined the adventure begins and the player must create their character, where once finished they can finally explore the world and experience the game.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

The idea for this project arose from an observation about the state of pirate games in general. The team noticed that there is a distinct lack of representation for pirate games, particularly realistic pirate games, and that the truly great pirate games are so few and far from what it can be. The team sees great potential for a new hit game to take over a significant portion of the market with not many competitors. Not only is there a lack of pirate games, but there is a lack of games that can truly keep their target market engaged long enough to reap the benefits of a devoted fan base. This is why the team decided to create a game that uses a subscription based payment system, which will allow us to consistently fund new content that will keep the players coming back for more. Once this project has been completed, players will have a game that they will be able to play for as long as their heart desires, and should they choose to leave for a time, they will be able to return at a later date to more content for them to dive into.

The last truly great "realistic" pirate game (depending on who one asks) can be traced all the way back to 2013 with *Assassin's Creed: Black Flag*. But this was a single player game and thus it eventually runs out of content for players to do, but this will not be the case with *A Pirate's Life*. Nowadays the project's main competitors are more cartoony versions of pirate games, and thus the team believes when *A Pirate's Life* is first announced it will be a game unlike anyone has seen for many years. The development of this game will not only address a lack of representation in the gaming industry, but it will bring a brand new experience that blends the best aspects of previous pirate games to deliver a AAA experience.

2b Goals of the Project

The goal of this project from the client's viewpoint is to provide an exhilarating experience that players will be able to continuously revisit time and again. This experience will drive players to subscribe and renew their subscriptions for as long as they are playing the game. With this product, the team would like to capture a majority of the pirate game market, as well as attract gamers from all genres to give the game a try. The main objectives are as follows

1. Cultivate a devoted fanbase that enjoys not only playing the game but also interacting with any media related to the game.
2. Pave the way for an expanded universe that can extend beyond just the initial game. Books, movies, shows, merchandise etc.
3. Maintain the fanbase by providing consistent updates to the game, thereby keeping the subscription numbers at a healthy level.

2c Measurement

Each objective mentioned in the previous section can be measured so that the team is able to determine when these goals have been reached.

1. **Cultivate a devoted fanbase**

   This objective is hard to measure exactly, however the team believes that two measurements will be able to tell us if the goal has been met. The first will be player count, the project's main competitor, *Sea of Thieves*, maintains an average of 15-25k players on any given day via their steam charts. If A Pirate's Life can achieve 30-45k players on an average day within the first few months post release, the team will consider this a major success. The other measurement is time played, although play time will not affect the fact that they are subscribed, players that play more are much more likely to resubscribe. And thus, the team would strive for an average player to play 10-15 hours a week. If both goals are met, this can be considered massive success.

2. **Pave the way for an expanded universe**
   This objective will take time to be measured, but the way to measure it is quite clear. If the team can expand into a different business within the first two years, be it books, movies, shows, or merchandise, this objective could be considered a success.

3. **Maintain a fanbase**
   To measure whether the team is maintaining a well sized fanbase, the team will need to wait until a time after release in order to determine whether players are sticking with A Pirate's Life or not. To disregard the initial excitement of the game, the team will need to measure the player base after 6 months and compare it to the launch month player base. If A Pirate's Life can achieve a 75% retention rate for players in this first 6 months, this goal can be considered a success.

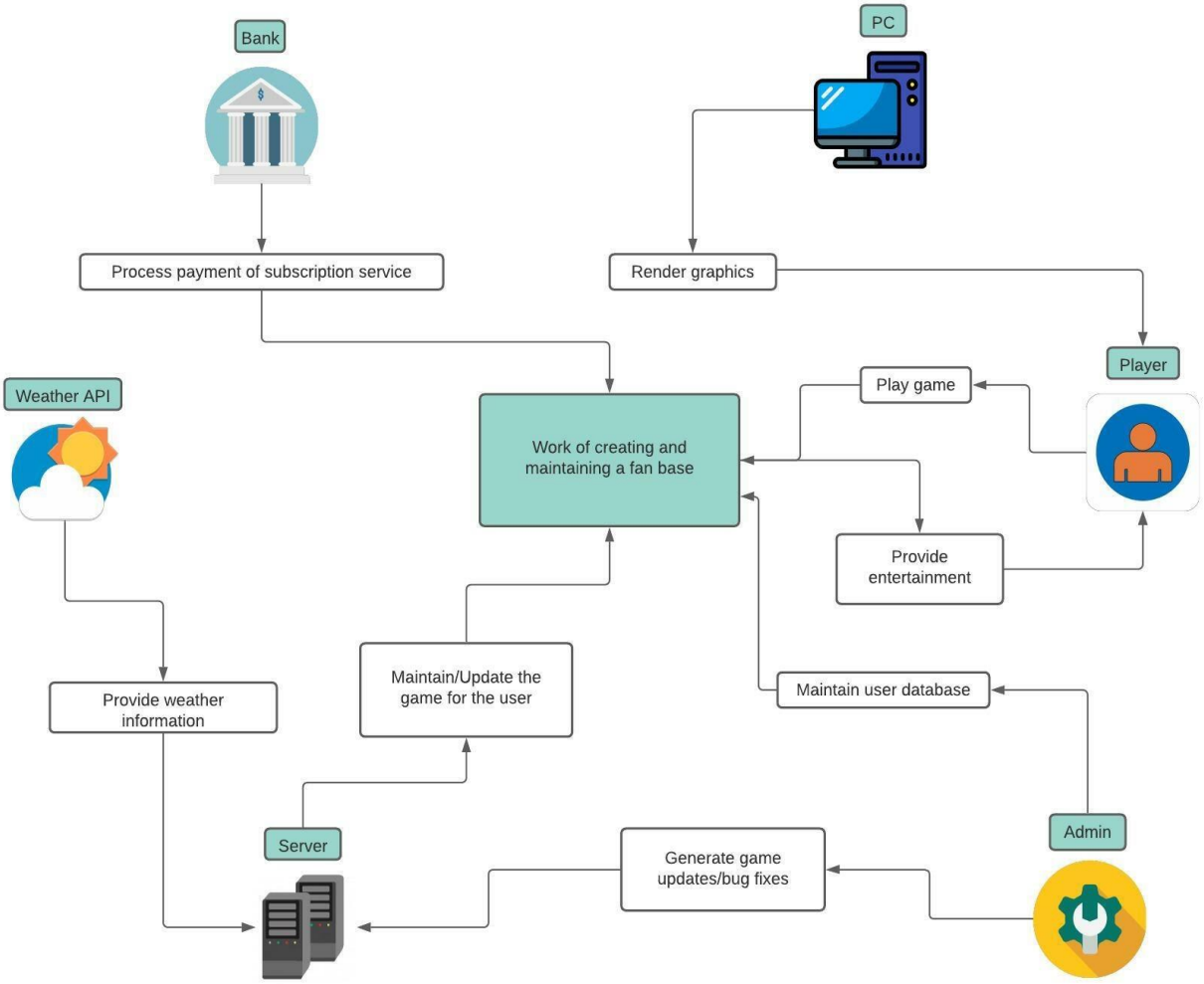
3 The Scope of the Work

The scope of the work for the project is very broadly aimed at players of all interests as well as well-known publishers. While the overall theme will be sailing the sea as pirates, there will be a constant flow of updates that will introduce all sorts of new features that will involve gardening, building, fishing, and more to keep players engaged.

3a The Current Situation

Currently, the state of video games in the pirate genre is very little in terms of what can deliver a true pirate experience. If players enjoy the pirate genre, they are forced to play from the small pool of games that currently exist. Players that have already played all these games are currently forced to just play them again as opposed to having the extra option of playing the project "A Pirate's life".

3b The Context of the Work



3c Work Partitioning

Event Name	Input and Output	Summary
1. Provide weather information	Weather API (in)	Server will use weather information to update in-game weather.
2. Generate game updates/bug fixes	Admin (in)	Administrators will be generate updates for the game and fix bugs
3. Process payment of subscription service	Bank (in)	The bank will handle all methods of payment regarding users.
4. Maintain/update the game for the user	Server (in)	The server for the game will be constantly updating information for players
5. Engage the player	Work of creating and maintaining a fanbase (in)	The fanbase that is currently made will aid in engaging the player while the game is being played.
6. Render Graphics	PC (in)	The computer will handle rendering the graphics for the player.

7. Maintain user database	Admin (in)	The admin will handle the database information for the user.
8. Provide Entertainment	Work of creating and maintaining a fanbase (out)	The fanbase created will entertain all of the players as well.

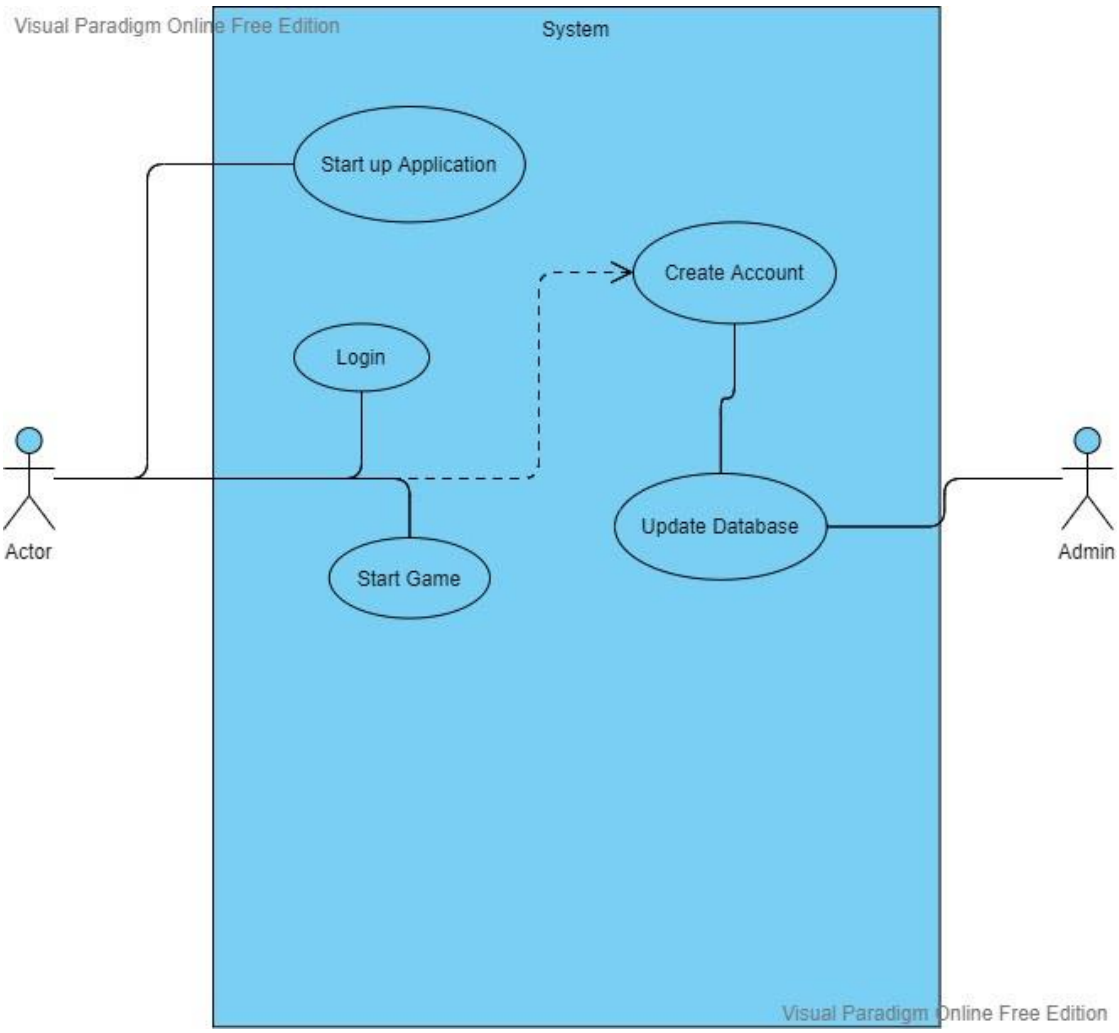
3d Competing Products

Currently the competing products include *Sea of Thieves*, *Atlas*, *New World* and "*Assassins Creed Black Flag*". The reason why the project still needs to be made is because all the competitors cater more towards the fantasy side of MMORPG's whereas A Pirate's Life would be aiming to focus more on a realistic type of environment and gameplay. A Pirate's Life is also aiming to release content at a much faster pace than any of these competitors with just as much content to keep the players more engaged.

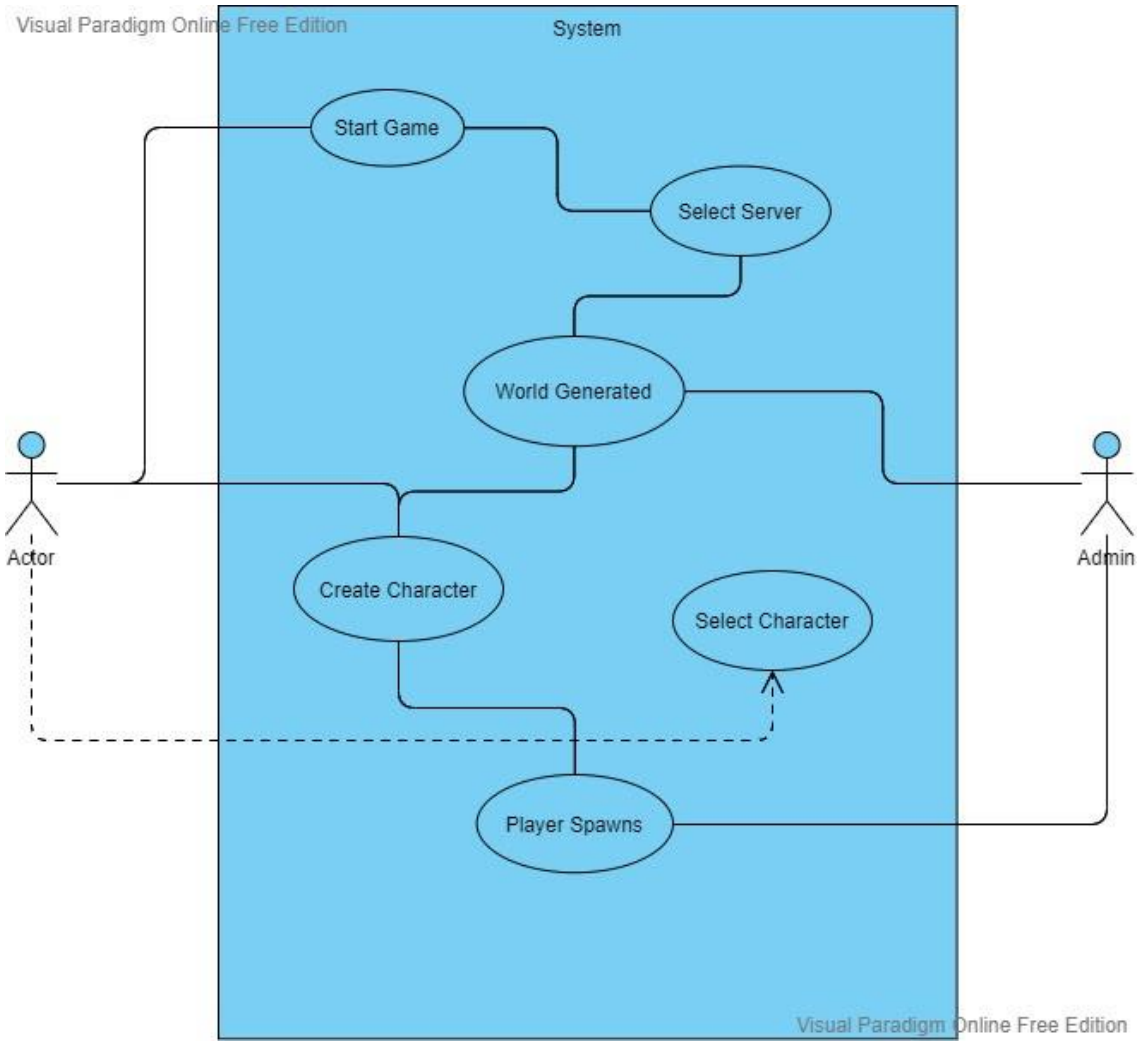
4 The Scope of the Product

This section provided different scenarios of the product along with diagrams and stories that are likely to happen. This application will primarily involve the interactions from the player, with the game communicating with the admin/server, which will communicate with the weather API for additional information to load within the application and the final interaction will be of communicating with the bank to process payments from the user to the game. The application will also have a database that will contain the user information of the players that will be maintained and updated by the admin.

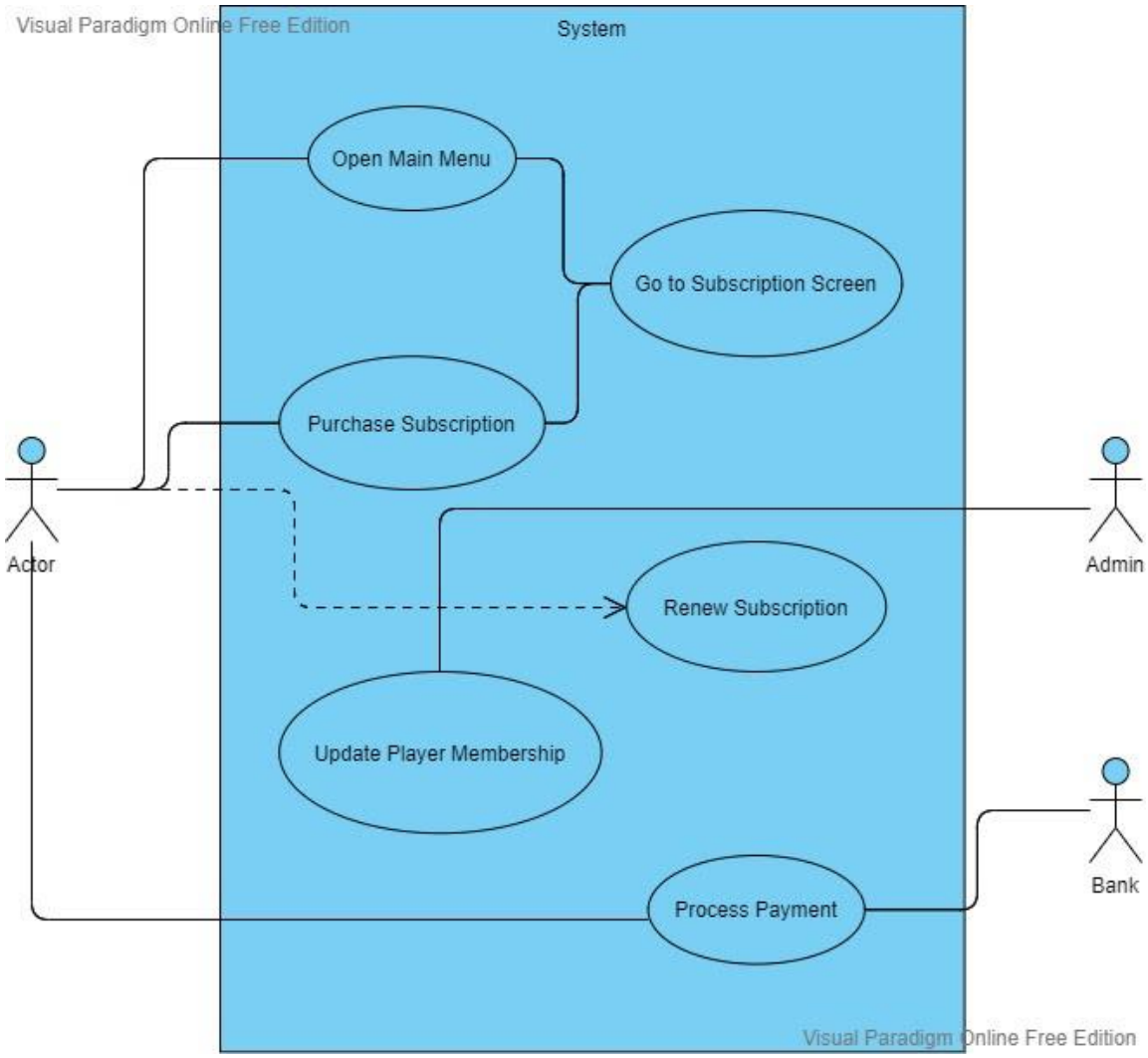
4a Scenario Diagram(s)



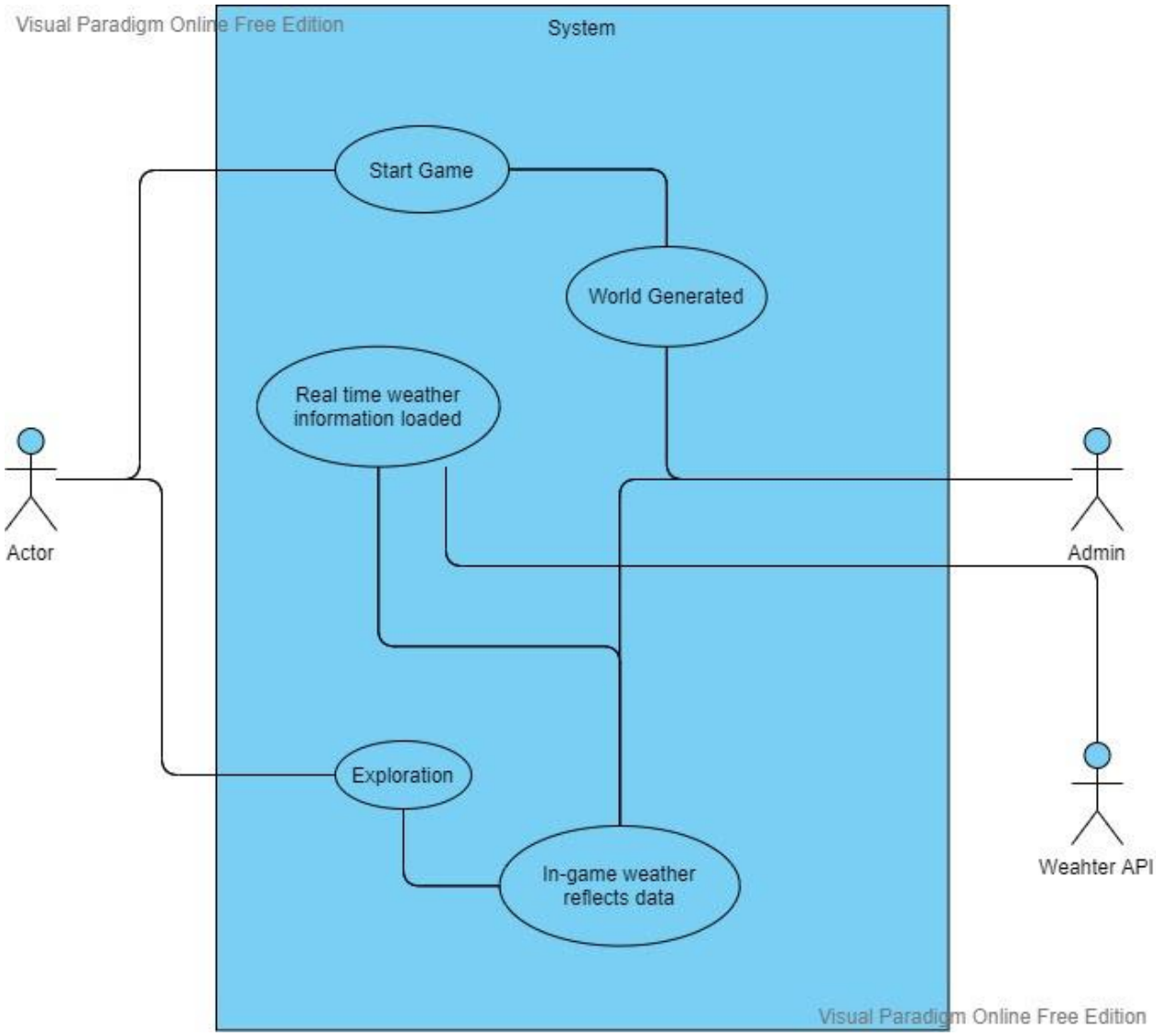
[Scenario: Start application]



[Scenario: Player logs in/Creates new character]



[Scenario: Player purchases subscription/Renews Subscription]



[Scenario: General Gameplay/Exploration]

4b Product Scenario List

Figure 2 - List of scenarios for the product

SL #	Scenario name	External Actors
1	Player starts the game	Player, server, hardware

2	Player renews subscription	Player, bank
3	Player enters new region	Player, server, hardware

4c Individual Product Scenarios

Player starts the game: Jimmy decides to play *A Pirate's Life* with his friends. The first thing he will need to do is launch the executable file which was installed from the installer. Once he has run the executable, he will be greeted by a launch window. This window will prompt him to log in to his account. After Jimmy has entered his account name and password the game will launch. Once the game has fully launched, he will be at the server selection screen, which will contain all different servers and their respective time period and region (Caribbean or South China Sea). After selecting a server, the software will then show the character selection screen, which will display all of the characters Jimmy has created on that server. Once he has chosen a character, he will be able to press the play button, which will then connect him to the server and start his game. Once everything has finished loading, his character will spawn into the world in the last location he logged out. If he did not log out in a valid location, then he will be spawned at the nearest harbor.

Player renews subscription: Timmy sees that all his friends are playing *A Pirate's Life* and he decides that he wants to play with them as well. Timmy launches the game and is greeted by the launch window. When Timmy tries to login after entering his username and password, he will see a message stating that he does not have an active subscription on his account. The message will be accompanied by a link to the team's main website, particularly the subscription page. Once Timmy clicks this link, he will be redirected to the subscription page of the website. Here, he will log in with his username and password. Once he has logged in, he will be able to hit the "Renew subscription" button. Once he has clicked this button his browser will display a new page containing a form where he can enter his payment information. There will be an option to save payment information for ease of use, but Timmy did not do this. Once Timmy has entered in all his payment information, he will be able to send a subscription request by hitting the "Submit" button at the bottom of the form. Once this submit button has been clicked, a request will be sent to the servers, which will then route that request to the bank. The bank will then verify the payment information and return either a success or failure. Fortunately for Timmy his subscription renewal was successful, and he will be able to play *A Pirate's Life* with his friends.

Player enters a new region: Ginny has been sailing around the beginner seas of the Caribbean of *A Pirate's Life* for the past few hours. Her raids have been going very well and she has decided she is ready to head into a more dangerous region. Ginny begins to head for the edges of the map which point to the next regions. She lands on the North-west region, and thus she adjusts her bearing. Once she has reached the edge of the map, she will hit an area that will send her to a loading screen. During this loading screen the entire area will be rendered, and the weather will be dynamically changed to match the weather setting that the server has set for that region, which will be dynamically sourced from the weather API every 10 minutes to replicate rapidly changing conditions on the seas. Once the loading screen is over, Ginny and her crew will now be in the new more dangerous area, ready to cause a load of mischief for anyone unfortunate enough to cross her path.

5 Stakeholders

5a The Client

Publishers - Video game publishers that will help endorse and market the game. Ideally these publishers will allow the development team to freely work on the product without any sort of issues with the monetization ideas that the team has.

Ubisoft

Ubisoft has already placed their foot in the door by creating one of the rivals to this idea. If they wanted to further pursue and explore the limits of their original idea, this idea can achieve that.

Amazon

Recently just released New World, a MMORPG that has similar features to this idea. If they wanted to branch out and go more depth into a specific genre of game, this idea could help them be the frontrunner.

Bethesda

Bethesda is known to make games that can vary between realism and fantasy such as the Fallout series and Elder Scrolls. The publisher would be a good fit for A Pirate's Life since they have already published games that have similar traits.

Microsoft

Microsoft has a good standing history of publishing well-made games and allowing development teams to manage themselves which is what would be ideal for A Pirate's Life. Since Microsoft owns Windows and the Xbox series it would allow the product to have multiple platform releases maximizing profits.

5b The Customer

Players - Individuals/groups of individuals wanting to play the game. These players will be either interested in the genre or avid pirate enthusiasts which would lead them to purchase A Pirate's Life. The players themselves will vary from handicapped players, younger players, and older players.

5c Hands-On Users of the Product

Figure 3 - Table of Hands-On Users

Username	**User Role**	**Subject Matter**	**Technology Experience**	**Other characteristics**
Streamer (Influencers)	Person who streams the game	Journeyman	Journeyman	Variation of everything from gender, age, handicap status, etc.
Historian	Provides accurate historical descriptions of the area.	Master	Journeyman	Any gender, handicap status does not matter, ideally loves games.
Oceanographer	Provides feedback on environment designs and how realistic they are.	Master	Journeyman	Any gender, handicap status does not matter, ideally loves games.

5d Maintenance Users and Service Technicians

The people who can maintain and service the project will include customer service representatives, the development team, and database administrators/experts, and moderators.

Customer Service Representatives - Will handle all ticket submitted issues that users may submit that can be resolved without too much difficulty. These representatives will be trained to handle customers in a very polite and professional manner since they represent the entire company as well. More complicated tickets may be forwarded towards either the development team or the database administrators depending on who's expertise would be more useful.

Development team - The team will aid in maintaining the game by providing a healthy flow of content to the player base of the game. It is important that the content released will provide more meaningful gameplay to the user and set new milestones for the players to achieve while providing entertainment or providing quality of life (QOL) changes.

Database Administrators - In charge of handling all database related features and will constantly be ensuring that no issues arise during the development of new features that may possibly affect the database.

Moderators - Ensures that players in game may have the safest and most enjoyable experience by making sure that players that behave inappropriately are punished or removed from the game. These moderators will be monitoring the in-game chat and making sure no bugs or exploits are being abused in game.

5e Other Stakeholders

The other stakeholders for the project will include the historians who aid in making sure the game is as true to history as possible to help keep the game realistic and historically accurate.

Oceanographers will also be referred to as the game will mainly be based around being overseas. Since oceanographers are familiar with what species of plants and animals live in each respective region of the ocean, they can be referred to when creating species in game to help ensure that it could be realistic.

Below are other shareholders who help in building the project:

- Beta testers
- Meteorologists
- Marketing experts
- Legal experts
- Translators
- Business partners

- IT professionals.

5f User Participation

Players - Closed alpha keys will be distributed to users who sign up for the test after the product is announced for release after 2 weeks. The test periods will be weekends to maximize the amount of playtime these users may get and will give them a feedback slip after each period ends so the team can read the feedback and make any changes necessary before the official release.

Historians - Historians will constantly be in contact with the development team as the team adds new content and areas to ensure that most of the content is historically accurate. There may be times where something is slightly inaccurate, and the historians will be referred to see how outlandish the idea may seem.

Oceanographers - The team will also be in contact with oceanographers as well as some of the key features may be linked to the environment the player is in. The team will need the aid of oceanographers to properly model the life forms that the player will encounter throughout their game. The behavior of the life forms will also need the feedback of oceanographers to portray the most realistic experience for the players.

5g Priorities Assigned to Users

Key users:

- Dedicated players: Players that play the game 30+ hours a week
- Influencers: Streamers and content creators

Key users will be the audience that we most try to cater the game to. These are the players that are very dedicated and most likely to interact with anything to do with the game, as well as keep their subscription constantly updated. Influencers such as streamers and content creators will be extremely important to the mission, as they can provide a wide range of exposure to communities of gamers. As such, we will take feedback from these users with the utmost importance. These content creators will prove invaluable in the months leading up to the game launch, as they will be the best way for us to build hype for the product.

Secondary users:

- Standard players: Players that play the game between 10-30 hours a week
- Casual players: Players that play the game less than 10 hours a week.

Secondary users are still a very important part of the audience as they will likely make up a large portion of the player base, however they are also more likely to unsubscribe and move on to other games. As such, the team will still consider their feedback, but it will not be as crucial as information gained from the key users.

6 Mandated Constraints

6a Solution Constraints

Description: The game will be required to be played on standard internet connection for requirements such as payments, rendering game and easy communication between players and the admin.

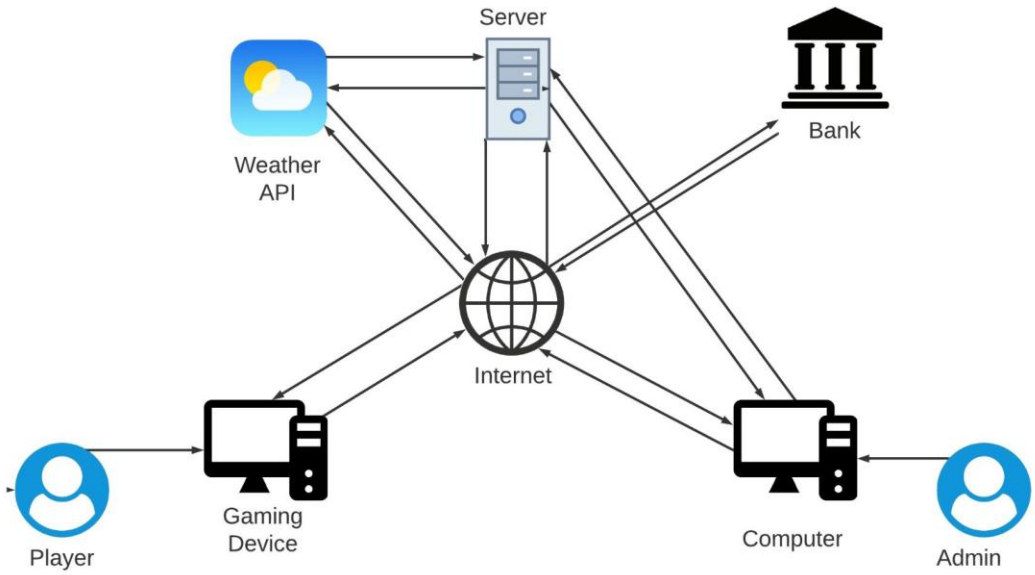
Rationale: The player will pay the subscription fees to the game publisher through the bank with the third-party application's online payment feature. The rendering of the game to the player by the publisher is also done via active internet connection. Also, communications between players and the admin/publisher are also made possible through the internet. Therefore, if standard internet could not be used by the game, then it will delay/stop the working of the game with proper notification being sent to other users of the game.

Fit-criterion: The game will regularly check for Internet connection of at least 3 Mbps download speed and 0.5 Mbps - 1 Mbps upload speed.
Description: Players are required to make successful payments to the publisher through a third-party payment application that is accepted by the publisher.

Rationale: The subscription fees for the game are paid through a third-party application accepted by the publisher such as PayPal, payTM or ChasePaymentech

Fit criterion: If the publisher finds the player is not subscribed to the game with payment, he will be rejected to have access to the game by the publisher.

6b Implementation Environment of the Current System



6c Partner or Collaborative Applications

The game is expected to be played by many players with a subscription. Also, they would be provided with individual accounts to identify their gaming history. Therefore, to accomplish efficient retrieval and storage of data relating to subscription and accounts, effective communication with publisher database

The game provides maximum possible real gaming experience to the player. This includes providing actual weather data for the region selected by the player. To accomplish this, effective communication with an external weather data API is required. The selection of external weather data API depends on the targeted expense of the game. Based on this, the team can provide access to either free or paid weather data API available online.

The players are required to make the payment to be subscribed in the game. These payments meet up the overall expense of the game in general. Therefore, these payments to the publisher by the player can be made through a third-party application such as PayPal, payTM or ChasePaymentech.

6d Off-the-Shelf Software

The off-the-shelf software that would be useful to the product would be Nvidia's software since it would enable features such as Ray Tracing and HairWorks in the game which would ultimately increase the realism and immersion the players may experience.

6e Anticipated Workplace Environment

This game is mainly operated from a computer, enhanced with other electronic devices such as simple headsets, keyboards and mouse. The game also provides an option to be played with VR devices. Therefore, considering the electricity requirement of the game, it is suggested to be played at a place where the power source is easily accessible. Similarly, since the game requires interactions through the internet it is mandatory that the location where the game is played should be internet accessible.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

MMORPG - An acronym for a genre that the game will encompass which stands for Massively Multiplayer Online Role-Playing Game.

NPC - An acronym for Non-Playable Characters

AI - An acronym for artificial intelligence

Exp - An acronym for Experience, which player's gain in order to level up

Infamy - A player's reputation, which affects their ability to purchase better armaments, ships, and venture into more dangerous regions. This can be increased by raiding, fighting ships, and other various activities.

7b UML and Other Notation Used in This Document

Scope of Work Diagram [3b]: This diagram shows within the application which individuals/groups will send information and which will receive it. Starting at the Admin, they generate updates to the game and fix any bugs, while also maintaining the player database. Going clockwise, the Server applies the updates from the admin to the game for the user, while also receiving the weather information from the weather API. The weather API's sole purpose is to send the weather information of the area to the server. The Bank will process payments for the game from the users who purchase the subscription or renew it. The PC is the player's computer that will

be used to play the game, and it will render the game's graphics to the user. Lastly, the Player will play the game, and at the end be provided with entertainment.

Scenario Diagram [Start Application]: This diagram shows the process for when the user starts the application. Once the user starts, the main screen will pop up allowing the user to either login or create an account. If the user has no current account, then they press the account button following the process there. Once the user has an account they will be brought back to the main screen where they can login now. Additionally, once the account is created, the admin will update the database, entering the information that the user just entered to acknowledge an account was created.

Scenario Diagram [Player logs in/Creates new character]: This diagram illustrates the process of when the user chooses to login and create a new character or select a created one. Once the player logs in, they are given the option to select which server they would like to join, where each server is different in region and story. Once the player chooses a server the world is generated by the server. Once they load into the server if they have not made a character for the selected region, they are then prompted to create a new character. If the player selects a previously made character, then they will load where they last left off. For the new characters, once they finish creating their character, the plaster will spawn in the game.

Scenario Diagram [Player buys subscription/Renews subscription]: This diagram illustrates the process the player will take if they want to purchase or renew the subscription to their account. From the game's main menu, the player will navigate to the subscription screen of the menu. Once there the player will click on purchase subscription where the screen will allow for the player to enter their payment information. Once the player submits the payment the Bank will process the payment, and the administrator will update their membership.

Scenario Diagram [General Gameplay/Exploration]: This diagram shows the process of how the weather API delivers data to the game to add in game effects that the player will encounter. Once the player starts the game and the world is generated, the server will be communicating with the weather API to grab accurate weather information. The acquired weather data will be sent to the server so that the in-game weather will reflect the real conditions, both on land and at sea.

7c Data Dictionary for Any Included Models

SQL Database Player Information: Name, UserName, Password, Date of Birth, Age, playerID.

SQL Database Moderators Information: Name, UserName, Password, Date of Birth, Age, playerID.

List Database for all creatures in server (one for each server): Creature Name, location/s

List Database for all NPCs in server (one for each server): Name, location/s

8 Relevant Facts and Assumptions

8a Facts

Pirate - an individual who robs and attacks ships at the sea
Piracy - the act of robbery by using a ship at sea, stealing the valuables of other ships, or cargo.
Players will face real diseases that may give the characters side effects/debuffs the same as it would in real life. Leaving some untreated will result in death.
Weapons and technology will be historically accurate for the time period.

8b Assumptions

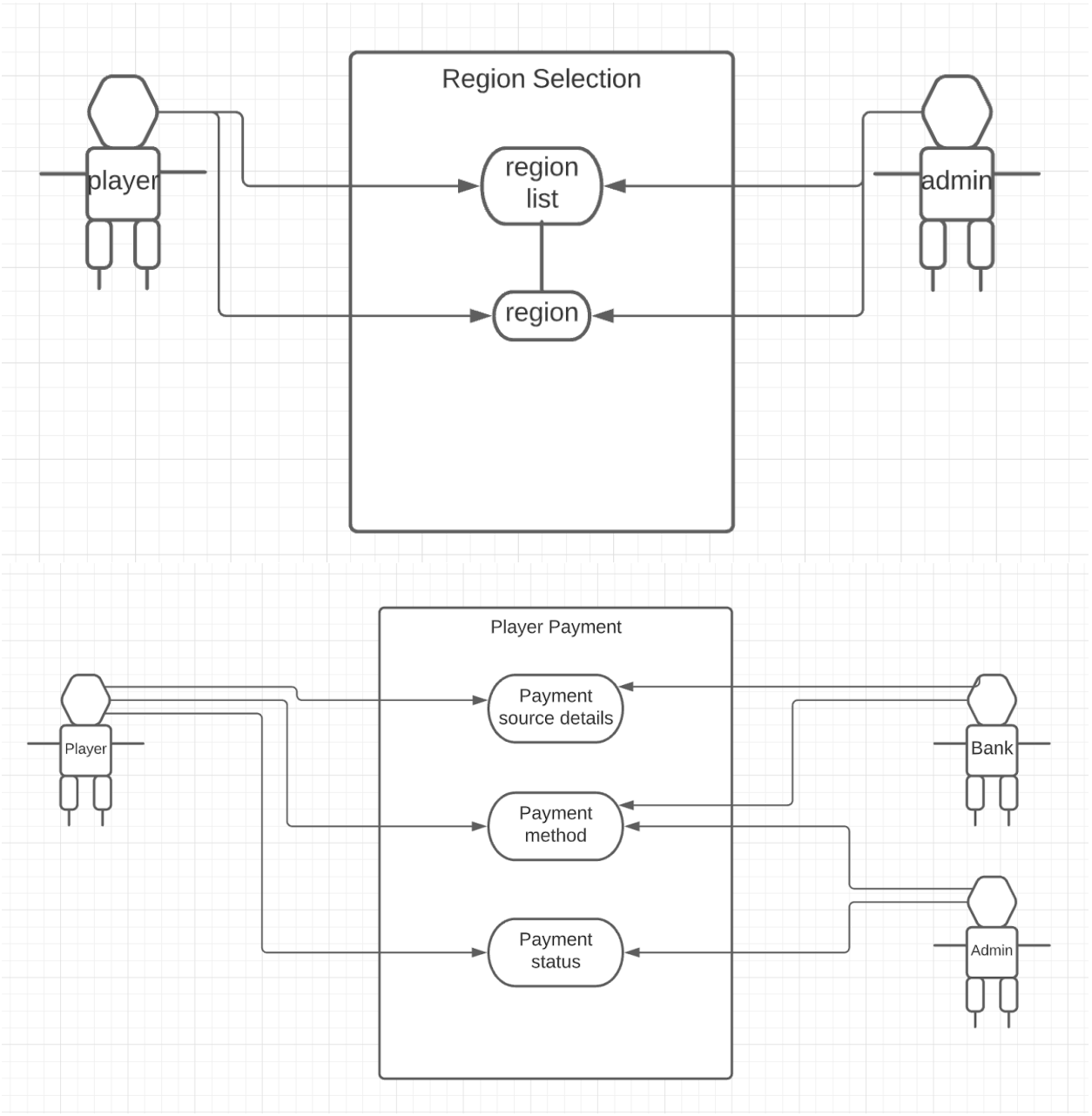
Assumptions:

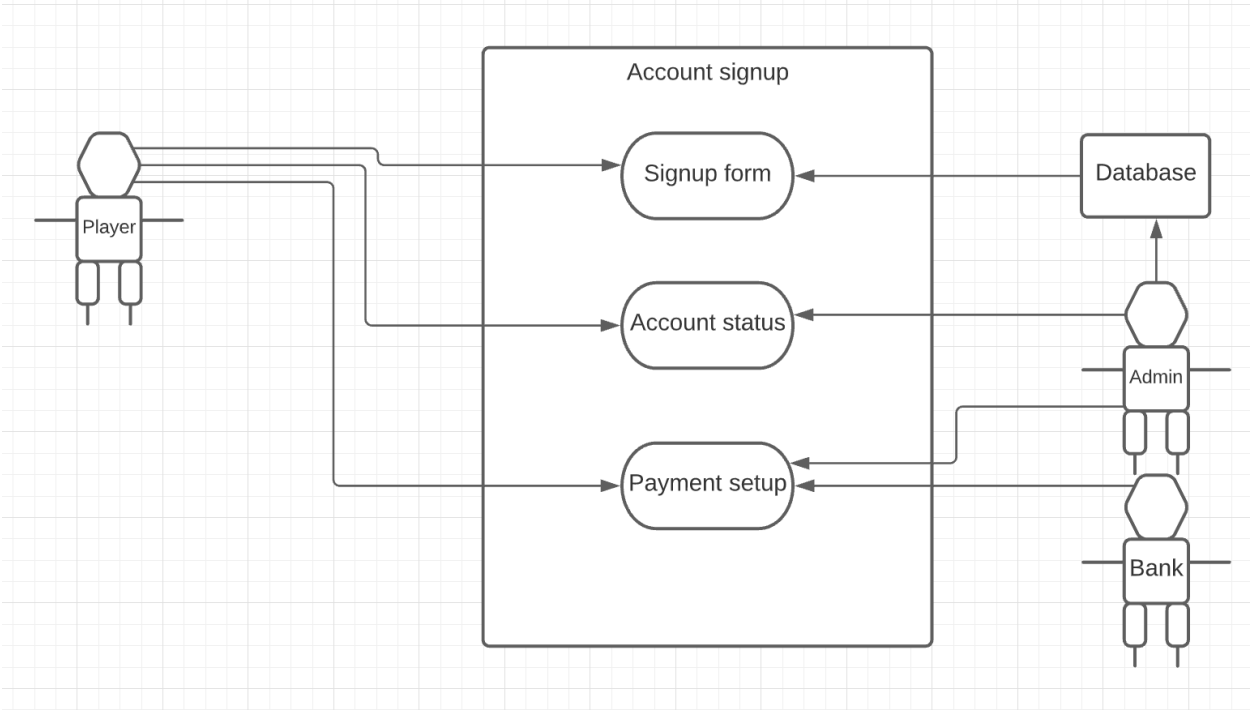
- Product will be used on the operating system and console: Windows 10, Xbox, and PlayStation. Product will work on previous releases of the software that are still supported from the operating system, and consoles used.
- Players will have compatible computer hardware components to be able to run the game.
- Mobile versions of this project are not planned at the moment.
- Users will have a stable internet connection
- Users will have an understanding on how to navigate the internet and websites
- Users will have understanding on account creation via websites, and paying for subscriptions (step by step tutorials can be included to help make the process as easy as possible)

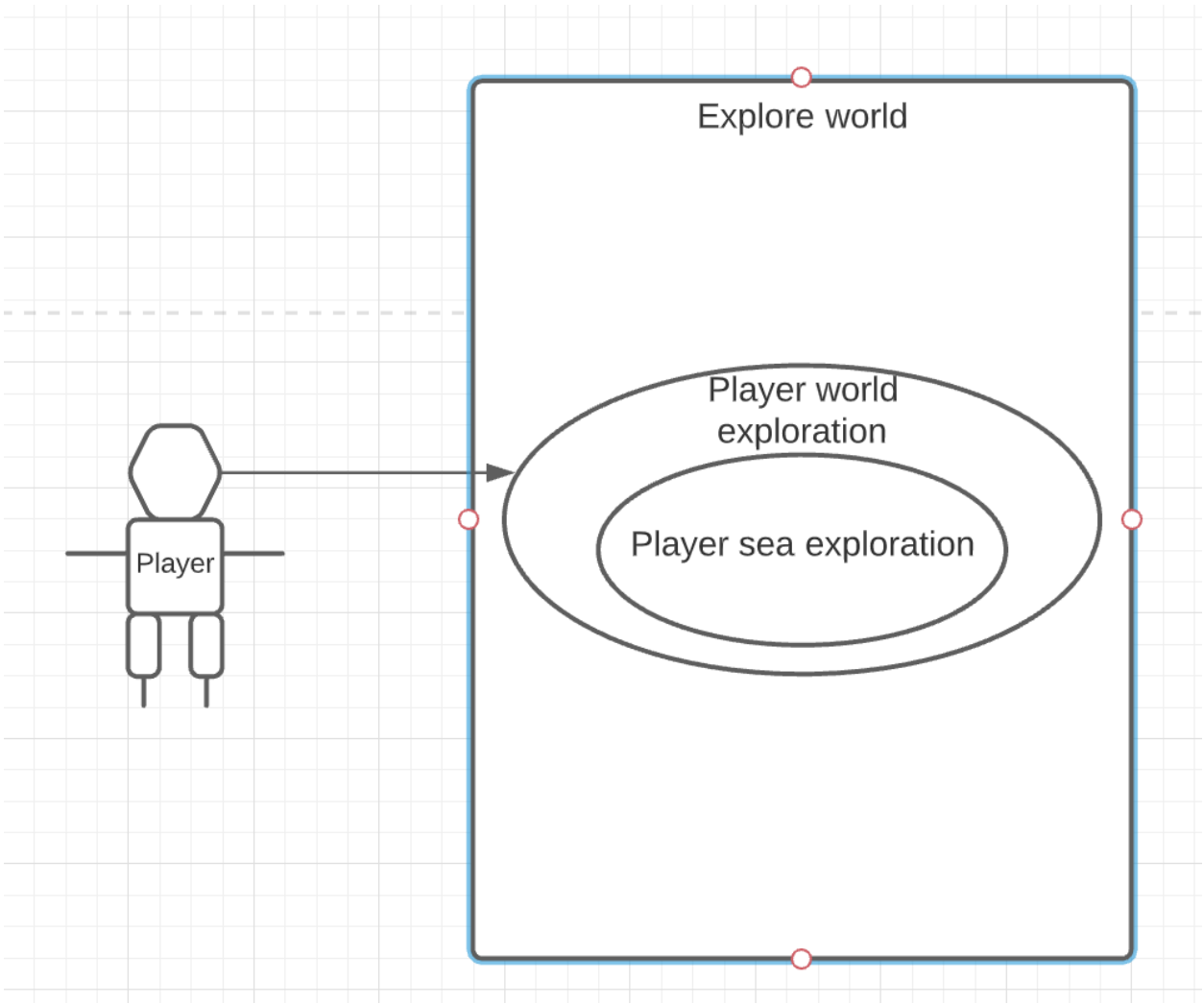
II Requirements

9 Product Use Cases.

9a Use Case Diagrams




Account signup
Signup form
Account status
Payment setup
Player
Database
Admin
Bank



9b Product Use Case List

Figure 4 - List of use cases for Player actor

Event (Account sign up)	**Use case**
Player fills account form	Player sign up
Player checks the account status	Player checks account
Event (Player payment)	**Use case**

Player decides and sets payment method	Set payment method
Player checks payment status	Verify payment
Event (Create Character)	**Use case**
Player picks what features will be on character	Character customization
Player selects world server	World server load
Event (Explore World)	**Use case**
Player walks through generated world (wild, cities, towns, etc.)	Player world exploration
Player sails the sea	Player sea exploration
Event (Item management)	**Use case**
Player collects items to use in-game (weapons, armor, maps, clothes, hats, etc)	Player manages equipment
Event (Combat)	**Use case**
Player battles other players, NPCs, creatures	Player combat

Figure 5 - List of use cases for Admin actor

Event (Maintain player database)	**Use case**
Admin updates database for new player accounts	New account creation
Admin updates database for changed information	Account data values modified
Admin removed account information	Player account deleted
Admin updates player subscription status	Account upgrade
Event (Server update)	**Use case**
Send game updates to server	Server update
Event (Generate game models)	
Generate World from player selected server	Generate World

Generate player character model	Player character creation/load
Generate in-game object models (trees, animals, etc.)	Generate in-game object models
Generate NPC models	NPCs generated/loaded

Figure 6 - List of use cases for Weather API actor

Event (Grab weather data)	**Use case**
Get the weather data	Store weather data for use in-game
Event (Load weather data)	
Load weather data into server so in-game weather reflects data	Send weather data to server

Figure 7 - List of use cases for bank actor

Event (Player payment process)	**Use case**
Bank checks the payment method chosen by the player.	Payment source validation.
Bank collects information for player's source of payment and transfer money	Money transfer
Bank checks whether the transaction is complete and sends notification to the player and admin based on completion of the transaction.	Transaction status update to player and admin.
Event (Game payment reports)	**Use Case**
Bank reports periodic transaction summary regarding transactions held with bank related to the game to admin.	Transaction report submission

9c Individual Product Use Cases

Use case ID: #1 Name: Player sign up

Pre-conditions: Player has valid email address

Post-conditions: Player has account

Initiated by: Player

Triggering event: Create account

Additional actors: Admin

Sequence of events:

1. Player goes to the account creation page
2. Player enters the required information
3. System ensures email and username is valid and not already taken

Alternative:

4. If system finds any invalid information, system notifies player

5. Player finalizes account creation “create account button”

Alternatives: Player exits account creation before pressing “create account”

Exceptions: System is down, does not take account information or send new account to admin

Use case ID: #2 Name: Player checks account

Pre-conditions: Player has successfully signed up

Post-conditions: Player will be able to log on

Initiated by: Player

Triggering event: Player logs on after account creation

Additional actors: Admin

Sequence of events:

1. Player logs in for the first time after creating account
2. System ensures the player has verified the email

Alternative:

3. Player has not verified email

4. System does not allow the player to continue and asks for verification code

5. Player types in verification code

6. Player is allowed to continue to the next menu and begin character customization

Alternatives: Player logs off after not verifying email, Player types in wrong verification code.

Exceptions: Server is down, Player's computer turns off, unexpected program crash occurs.

Use case ID: #3 Name: Set payment method

Pre-conditions: Player navigates through the menu to set up payment methods.

Post-conditions: Player successfully updated payment method

Initiated by: Player

Triggering event: Pressing "Set up payment methods" button.

Additional actors: Bank, Admin

Sequence of events:

1. System displays a screen with information that needs to be filled out
   2. Player fills out the information out

Alternative:

   3. Player types in false information
      4. System declines the process and makes them reenter information
   6. Player types in correct information.
      7. System accepts payment method and completes transactions

Alternatives: Player exits payment method screen, Player closes client

Exceptions: Player has not sufficient funds, Server is brought down for maintenance, Bank declined transaction.

Use case ID: # 4 Name: Verify payment

Pre-conditions: Player has already set up a payment profile

Post-conditions: Player's payment status is known to the player

Initiated by: Player

Triggering event: Player checks

Additional actors: Bank

Sequence of events:

1. Player goes to check if subscription has been paid/next payment
2. System displays status or next payment

Alternatives: N/A

Exceptions: System is down, player can't check status

Use case ID: #5 Name: Character customization

Pre-conditions: Player has an account in the game

Post-conditions: Player has a character to play the game.

Initiated by: Player

Triggering event: Character creation

Additional actors: Admin

Sequence of events:

1. Player opts to create a character to play the game.
2. The admin provides character creation elements to the player based on player's current subscription.
3. The player designs and sets a character for him to play.
4. The player's character is saved in

Alternatives:

Exceptions:

Use case ID: #6 Name: World server load

Pre-conditions: World server picked

Post-conditions: World server loaded for player

Initiated by: Admin

Triggering event: Player selects which world server

Additional actors: Player

Sequence of events:

1. Player selects world server after login/character creation
2. Admin receives which server was picked and loads the selected world for player
3. World around the player is loaded

Alternatives: N/A

Exceptions: Player had no prior character to immediately select server, then player must create character

Use case ID: # 7 Name: Player world exploration

Pre-conditions: Map is fully loaded, and player has control of character.

Post-conditions: Map tracks explored locations and saves them to the player.

Initiated by: Player

Triggering event: Player logging on and selecting server

Additional actors: Admin

Sequence of events:

1. Player moves character freely around the world
   2. System updates the player map to flag the area as revealed and save it
3. Player chooses to talk to NPC's
   4. System will archive NPC as visited and save information regarding it in archives
5. Player enters a city
   6. System displays text "Safe Area" and the city is saved to map.

Alternatives: Player logs off, Player runs into collision

Exceptions: Server for game goes offline, Player runs out of membership while online.

Use case ID: # 8 Name: Player Sea exploration

Pre-conditions: Map is fully loaded, and player has control of character.

Post-conditions: Player can venture on ship on the sea

Initiated by: Player

Triggering event: Player enters ship

Additional actors:

Sequence of events:

1. Player enters their ship
   2. Game loads the ship and ship controls
2. Player can now sail the sea to go to any part of the water or sail to another land

Alternatives: Player exits ship

Exceptions: N/A

Use case ID: # 9 Name: Player manages equipment

Pre-conditions: Player opens bag/inventory

Post-conditions: Player managed equipment

Initiated by: Player

Triggering event: Player opens bag

Additional actors: System

Sequence of events:

1. Player opens bag
2. Player chooses to equip/discard/move items in their inventory
    3. System deletes item if discarded, equips item if item equipped, or item moved within inventory or to external storage where system remembers location

Alternatives: Player chooses to only view inventory not interact with any item

Exceptions: Player closes inventory

Use case ID: #10 Name: Player combat

Pre-conditions: Player comes into contact with hostile entity

Post-conditions: A character has won the battle

Initiated by: Player

Triggering event: Player has attacked enemy.

Additional actors: Admin

Sequence of events:

1. Player hits attack key
2. Player strikes with weapon

Alternative:

3. Player misses

4. System ensures the enemy takes no damage

5. Player lands attack

6. System subtracts health from the enemy depending on the player's weapon

6. Enemy Attacks

Alternative:

7. Enemy misses

8. System ensures the Player takes no damage

9. Enemy lands attack

10. System subtracts health from the player depending on the strength of the enemy.

11. Player or enemy dies

12. Battle ends

Alternatives: Player runs away, Player is too far from enemy when striking, Player logs off

Exceptions: NA

Use case ID: #11 Name: New account creation

Pre-conditions: Player created new account

Post-conditions: Accounted created and stored in database

Initiated by: Admin

Triggering event: Player creates new account

Additional actors: Player

Sequence of events:

1. Once a new account is created from player side, that information is sent to the admin
2. Admin then updates the database for players to hold that information

Alternatives: N/A

Exceptions: Player did not create new account
System did not send the new information to admin

Use case ID: #12 Name: Account data values modified

Pre-conditions: Player information must already be in database
Player requests information change

Post-conditions: Player information is modified

Initiated by: Player

Triggering event:

Additional actors: Admin

Sequence of events:

1. Player requests information change
2. Admin searches for player within the database and updates the information based on new changes made.

Alternatives: Player made invalid update, hence admin rejecting modification

Exceptions: N/A

Use case ID: # 13 Name: Account deleted

Pre-conditions: Player requested deletion of account

Post-conditions: Admin has deleted the requested player account

Initiated by: Player

Triggering event: Player requests account deletion

Additional actors: N/A

Sequence of events:

1. Admin types in player ID to search for through archive
2. System returns matched results
3. Admin selects player account
4. Admin clicks delete
5. System deletes accounts from the database of player accounts.

Alternatives: No matched results for player ID

Exceptions: NA

Use case ID: #14 Name: Player account upgrade

Pre-conditions: Player has account, valid payment profile, and requests to upgrade account
Player's account must not be upgraded already

Post-conditions: Player account upgraded

Initiated by: Player

Triggering event: Player request account upgrade

Additional actors: Admin

Sequence of events:

1. Player requests to upgrade account
   2. System checks if player's payment profile is valid
   3. Player is charged upgrade amount
4. Admin locates account in database
5. Admin updates account upgrade status

Alternatives: Player's payment is invalid therefore not upgraded

Exceptions: N/A

Use case ID: #15 Name: Server update

Pre-conditions: Scheduled server update is about to arrive

Post-conditions: Server is updated with changes

Initiated by: Admin

Triggering event: Admin scheduled server update

Additional actors: System, Player

Sequence of events:

1. Server logs off all players currently online for the game
   2. Players are displayed a message mentioning server maintenance expected duration
3. Admin turns off server
4. Admin applies updates to the server
5. Admin ends server maintenance
6. Server brought back online for players to log on
   7. Players required to patch game in order to log on

Alternatives: NA

Exceptions: NA

Use case ID: #16 Name: Generate World

Pre-conditions: Player has selected server to join

Post-conditions: World has been generated for the player and the player may play the game

Initiated by: Player

Triggering event: Player selects server

Additional actors: System

Sequence of events:

1. System begins to load character information
2. System loads map for the player with all the proper information
   3. Player's computer loads the information from the server into the client

Alternatives: Player closes window during world generation

Exceptions: Server goes down while player is loading world

Use case ID: #17 Name: Player character creation/load

Pre-conditions: Player has selected the character to log into server with

Post-conditions: Character information and appearance is loaded

Initiated by: Player

Triggering event: Player selection of character

Additional actors: System

Sequence of events:

1. System fetches player ID
2. System retrieves character information from archive
3. System sends information to player client
4. Player's client loads information
5. Character is now loaded into the world

Alternatives: Player switches character, Player is banned

Exceptions: NA

Use case ID: # 18 Name: Generate in-game object models

Pre-conditions: Server has been started up by Admin

Post-conditions: Objects are loaded into the world

Initiated by: Admin

Triggering event: Admin starts server

Additional actors: System

Sequence of events:

1. System fetches objects that need to be loaded
2. System locates the coordinates of where it needs to be loaded
3. System loads functionalities of objects if any

Alternatives: NA

Exceptions: NA

Use case ID: # 19 Name: NPCs generated/loaded

Pre-conditions: Server has been started up by Admin

Post-conditions: NPCs are loaded into the world

Initiated by: Admin

Triggering event: Admin starts the server

Additional actors: System

Sequence of events:

1. System fetches NPCs that need to be generated and loaded
2. System locates coordinates of where NPCs need to be placed
3. System loads NPC into the world along with dialogue options

Alternatives: NPC is hidden until player meets quest requirements to spawn

Exceptions: NA

Use case ID: #20 Name: Store weather data for use in-game

Pre-conditions: Server has been started up by Admin

Post-conditions: Weather API stores correct weather information to each region

Initiated by: Admin

Triggering event: Admin starts server

Additional actors: Weather API, System

Sequence of events:

1. Weather API checks through database of each respective area
2. Weather API saves current weather for each location
3. System stores weather API information

Alternatives: NA

Exceptions: Weather API goes down

Use case ID: #21 Name: Send weather data to server

Pre-conditions: Weather API has saved current weather for each location

Post-conditions: Weather has been loaded into server

Initiated by: Weather API

Triggering event: Weather API successfully saves data for all locations

Additional actors: System

Sequence of events:

1. Weather API sends all saved information to server
2. System loads each area and loads corresponding weather

Alternatives: loads snow, loads rain, loads hail

Exceptions: weather API functionality stops working

Use case ID: #22 Name: Payment source validation

Pre-conditions: System has successfully processed information regarding player payment

Post-conditions: Bank validates or invalidates payment

Initiated by: Server

Triggering event: Server finishes loading in payment information

Additional actors: NA

Sequence of events:

1. Server sends information over to bank
2. Bank finds bank account with matching visa or account number
3. Bank checks if information entered matches the bank account
4. Bank approves the transaction payment source

Alternatives: Bank declines the transaction

Exceptions: Bank communication network is down

Use case ID: #23 Name: Money transfer

Pre-conditions: Bank has successfully validated payment source

Post-conditions: Money is transferred to admin

Initiated by: Bank

Triggering event: Bank accepts payment source

Additional actors: NA

Sequence of events:

1. Bank checks if player balance is sufficient

Alternative:

2. Player balance is not sufficient

3. Bank declines transfer

4. Player has sufficient balance

5. Bank transfers money to system

Alternatives: NA

Exceptions: NA

Use case ID: # 24 Name: Transaction status update to player and admin

Pre-conditions: Bank has successfully transferred currency

Post-conditions: both players and admin receive status update

Initiated by: Bank

Triggering event: Bank transfers currency

Additional actors: System

Sequence of events:

1. Admin receives message from bank that transaction is completed
2. Admin updates player status and unlocks requested feature
3. Player receives message from system saying transaction is completed
4. Feature that the player has bought is now unlocked

Alternatives: NA

Exceptions: NA

Use case ID: # 25 Name: Transaction report submission

Pre-conditions: Transaction has properly been updated to Admin and Player

Post-conditions: Bank submits report of transaction summary to Admin

Initiated by: Bank

Triggering event: Transaction completed

Additional actors: NA

Sequence of events:

1. Bank retrieves entire payment summary of transaction
2. Bank sends payment summary to admin
3. Admin receives payment summary

Alternatives: No payment summary at all

Exceptions: NA

10 Functional Requirements

ID# F-1 - Game launches correctly

Description: The system must be able to start the game correctly
Rationale: To play the game, it must correctly start first
Fit Criterion: User can get to the login screen after launching game executable.
Acceptance Tests: F-1

ID# F-2 User login

Description: The system must allow users to log in to their game accounts
Rationale: To play the game, logging in is a necessity
Fit Criterion: User can go from login screen to game menu screen
Acceptance Tests: F-2

ID# F-3 - Payment process

Description: The system must process player payment for subscription.
Rationale: To fit the financial needs of the admin/publisher, the system must be enabled to collect payment source information and send money to the bank to the admin/publisher's account.
Fit Criterion: Bank sends payment confirmation to the user and admin. The player will be successfully subscribed to the game.
Acceptance Tests: F-3

ID# F-4 - Form fill up

Description: The system must collect user information in an electronic form during sign up.
Rationale: Getting player information in a form is essential for account creation, player validation and various other processes.
Fit Criterion: Player receives confirmation from admin that the form is received by the admin.
Acceptance Tests: F-4

ID# F-5 - Select game region

Description: System must allow player to select between the two game regions (Caribbean and South China Sea)
Rationale: It is necessary to allow players to select a region before they can play the game
Fit Criterion: Player can select a region and move to the character select screen

Acceptance Tests: F-5

ID# F-6 - Character selection screen

Description: System must display existing characters and allow player to select one or create a new one.
Rationale: Players must be able to see their existing characters for gameplay purposes
Fit Criterion: System correctly displays character select after player selects a region
Acceptance Tests: F-6

ID# F-7 - Character creation

Description: System must allow the player to create a new character.
Rationale: Without the ability to create a character, players won't be able to play.
Fit Criterion: After finishing character creation, character object is created correctly and displayed on the character selection screen.
Acceptance Tests: F-7

ID# F-8 - Load into game

Description: System must allow the player to select a character and load into the game with that character by hitting the play button.
Rationale: To play the game, players must be able to select a character and start the game.
Fit Criterion: Correctly load into game when a character is selected, and the play button is hit.
Acceptance Tests: F-8

ID# F-9 - Render game graphics

Description: System must render graphics correctly once the player has loaded in.
Rationale: Game graphics must load for the player to be able to play the game.
Fit Criterion: The game is fully rendered for the player.
Acceptance Tests: F-9

ID# F-10 - Add player on friends list

Description: System must allow players to add friends via a friends list
Rationale: In an online multiplayer game it is a necessity to allow players to add friends and be able to play with them.
Fit Criterion: Player is able to open their friends list and add a friend.

Acceptance Tests: F-10

ID# F-11 - Connect to server

Description: System must allow players to select their preferred server to connect to.
Rationale: This will allow the player to select the server with the lowest latency to play on.
Fit Criterion: Correct server is connected to when the player selects a server to play on.
Acceptance Tests: F-11

11 Data Requirements

ID# D-1 - Player-account object

Description: Data

First Name	Last Name	Username	Password	Date of Birth	Email

Rationale: Player object will be used to access player information for each individual player
Fit Criterion: Object should hold information about a given player
Acceptance Tests: D-A

ID# D-2 - Player-character object

Description: Data

Character model	Name	Equipment	Skills	Stats	Level

Rationale: Player object will be used to access player character information for each individual player
Fit Criterion: Object should hold information about a given character
Acceptance Tests: D-A

ID# D-3 - Friendly NPC object

Description: Data

NPC Model	Name	Quests Involved	Dialogue	Location	Skills

Rationale: NPC object will be used to access NPC information for information for each individual player
Fit Criterion: Object should hold information about a given NPC
Acceptance Tests: D-A

ID# D-4 - Enemy NPC Object

Description: Data

NPC Model	Skills	Level	HP	Defense	Skills	Quests Involved

Rationale: Enemy NPC objects that players may encounter and battle
Fit Criterion: Object should hold information about a given enemy NPC
Acceptance Tests: D-A

ID# D-5 - Consumable Object

Description: Data

Item ID#	Name	Quality	Restore amount	Side-effect	Cost

Rationale: Consumable objects that players may consume for healing, and buffs
Fit Criterion: Object should hold information about a consumable object
Acceptance Tests: D-B

ID# D-6 - Offense Equippable Object

Description: Data

Item ID#	Name	Quality	ATK damage	Weight	Level requirement	Cost

Rationale: Offensive equippable objects that players may utilize for battle
Fit Criterion: Object should hold information about a given offensive equippable item
Acceptance Tests: D-B

ID# D-7 - Defensive Equippable Object

Description: Data

Item ID#	Name	Quality	Defense stat	Weight	Level Requirement	Cost

Rationale: Defensive equippable objects that players may utilize for battle
Fit Criterion: Object should hold information about a given defensive equippable object
Acceptance Tests: D-B

ID# D-8 - Ship Object

Description: Data

Ship Model	Name	Status	Quality	Capacity	Offensive upgrades	Defensive upgrades

Rationale: Ship objects that players may utilize for exploration and battles at sea
Fit Criterion: Object should hold information about a ship
Acceptance Tests: D-C

ID# D-9 - Island Object

Description: Data

Island Name	Location	# Of NPCs	List of Quests	List of NPCs	List of enemy NPCs	List of Towns/Cities

Rationale: Island objects that players may encounter on their journey
Fit Criterion: Object should hold information about the island
Acceptance Tests: D-E

ID# D-10 - Town Object

Description: Data

Town Name	Location	# Of NPCs	List of Quests	List of NPCs	List of Retailers

Rationale: Towns that players may encounter on their journey
Fit Criterion: Object should hold information about the town
Acceptance Tests: D-E

12 Performance Requirements

12a Speed and Latency Requirements

ID# P-1 – Loading map speed

Description: The product must quickly be loaded quickly so players do not feel like they have waited too long.

Rationale: Players will get impatient if it takes too long to load and feel obligated to play something else.

Fit Criterion: Loading should take no longer than 1-2 minutes

Acceptance Tests: P-1

ID# P-2 – Player latency

Description: The product must be able to take the user input and match that to output the desired outcome in-game quickly

Rationale: Players will not like if the game is not near real time user command execution due to input lag

Fit Criterion: User command input and in-game output should feel almost instantaneous

Acceptance Tests: P-2

12b Precision or Accuracy Requirements

ID# P-3 - Crosshair

Description: The product must be as precise/accurate as possible with where the player has the crosshair in relation to mouse movement

Rationale: If the crosshair is not where the player intends then it will cause frustration

Fit Criterion: User crosshair should be as seamless as possible with player aim

Acceptance Tests: P-3

12c Capacity Requirements

ID# P-4 – Server Load

Description: Product can support up to 100,000 concurrent players

Rationale: The product wants a stable connection for everyone involved and anything above 100,000 would trivialize connections.

Fit Criterion: Majority of 100,000 players have no connection issues

Acceptance Tests: P-4

13 Dependability Requirements

13a Reliability Requirements

ID# DE-1 – Account data during system failure

Description: In situations of system failure, related to an account(s) should be protected.

Rationale: Data related to an account is important because it is required to properly deliver contents to the player based on their subscription.

Fit Criterion: During system failure data related to accounts will be protected in the database.

Acceptance Tests: DE-1

ID# DE-2 – Subscription payment transfer

Description: The payment for subscription by player should be transferred to a reliable financial holder.

Rationale: All financial transactions and holding should be done through reliable institution as loss money affects the financial benefits and stability of the publisher.

Fit Criterion: In case of system failure all account data should be unaffected.

Acceptance Tests: DE-2

13b Availability Requirements

ID# DE-3 – Player should be able to play the game

Description: All subscribed players should be able to play the game all the time.

Rationale: The player subscribes to the game to play it all the time. So, it should be provided to the players all the time.

Fit Criterion: The game is available to play for the subscribers all the time

Acceptance Tests: DE-3

13c Robustness or Fault-Tolerance Requirements

ID# DE-4 - Client must still have functionality without internet connection

Description: The product must allow the user to still modify settings even if not connected to the game via the internet.

Rationale: Client should have functionality for methods that do not require internet connection.

Fit Criterion: Player can change graphics settings and volume with no

Acceptance Tests: DE-4

13d Safety-Critical Requirements

Not applicable

14 Maintainability and Supportability Requirements

14aMaintenance Requirements

ID# M-1 – Gaming error handling

Description: Gaming malfunctions reported by players resolved by program developers.

Rationale: Any malfunctioning related to gaming like simulation error, inaccessibility of subscribed gaming elements and so on should be resolved by the program developing team.

Fit Criterion: The developer notifies the player when the errors are fixed.

Acceptance Tests: M-1

ID# M-2 – Database maintenance

Description: Database malfunctioning/errors should be fixed immediately by database handler.

Rationale: The database handler checks that the database is safe and not experiencing any errors.

Fit Criterion: In case of database errors, the database handler can resolve the problems with the help of a backup.

Acceptance Tests: M-2

ID# M-3 – Payment gateway maintenance

Description: Any sort of errors with payment gateway should be resolved bank immediately.

Rationale: The bank should frequently check its payment gateway is error

Free and should resolve if any exist.

Fit Criterion: The bank should notify the admin while it has errors, and the admin can display this notification in the program until it is fixed.

Acceptance Tests: M-3

14b Supportability Requirements

ID# M-4 - Game support

Description: Customer support should be made available by publisher for subscription and gaming enquiries.

Rationale: Solving subscription and gaming enquiries are essential for uninterrupted gaming experience for the player.

Fit Criterion: A gaming support staff ensures customer concerns regarding gaming and subscriptions are solved.

Acceptance Tests: M-4

ID# M-5 – Payment support

Description: Customer support should be made available by the bank for payment related enquiries.

Rationale: Solving payment related problems are essential to avoid money loss to both player and the publisher.

Fit Criterion: A bank support staff ensures customer concerns regarding payments are solved.

Acceptance Tests: M-5

14c Adaptability Requirements

ID# M-6 – Personal computers

Description: The game is playable on personal computers.

Rationale: Account access and all the gaming features runs effectively in any hardware requirement qualified personal computers.

Fit Criterion: The program works well on a personal computer with standard operating systems like windows and mac and with standard hardware capabilities**.**

Acceptance Tests: M-6

ID# M-7 – Gaming consoles

Description: Game could be played on a gaming console.

Rationale: On different gaming consoles like Xbox, PlayStation, and Nintendo this game is could be accessed and proper agreement with the console company.

Fit Criterion: The game works best with modern console versions for best gaming experience.

Acceptance Tests: M-7

ID# M-8 – Virtual reality devices

Description: Game is accessible in VR enhanced gaming devices.

Rationale: The game would be made available to play in VR enhanced gaming devices. It would give more real like gaming experience to players

Fit Criterion: It will be mainly accompanied by VR headset and other game controlling devices making the player immersed in the game.

Acceptance Tests: M-8

14d Scalability or Extensibility Requirements

ID# M-9 – Server extension

Description: As player population increases, more servers may be added

Rationale: Server capacity may reach at a maximum, so it is important that the product can support as many players as possible.

Fit Criterion: Easy expansion of servers

Acceptance Tests: M-9

14e Longevity Requirements

ID# M-10 –Product Lifetime

Description: The product is expected to maintain a healthy player base for a minimum of 5 years, and the team aims to support it up to 10 years.

Rationale: The team will continuously create content and update the game for the active player base.

Fit Criterion: New events and content on holidays and key dates.

Acceptance Tests: M-10

15 Security Requirements

15a Access Requirements

ID# S-1 – Payment source access

Description: Payment source access only to the bank.

Rationale: The access to payment source should only be given to the bank by the program. If the publisher needs access to this information, the program should add provision to request player by the publisher.

Fit Criterion: Acknowledgement by bank for secure payment should be given to the player. Also, when the publisher needs access to payment source proper communication should be made between player and publisher for request to be made.

Acceptance Tests: S-1

15b Integrity Requirements

ID# S-2 – Anti-cheat system

Description: The team that will form a unique anti-cheat system that will prevent players from using third party software that may affect the game.

Rationale: Cheaters would ruin the experience for other players, so it is important that the matter is resolved immediately

Fit Criterion: There would be a minimal number of cheaters in the game.

Acceptance Tests: S-2

15c Privacy Requirements

ID# S-3 – Hashed passwords

Description: Passwords need to be hashed to prevent any sort of security breach resulting in the access of the users' passwords.

Rationale: Protecting the privacy of the player base should always be a top priority

Fit Criterion: All passwords are hashed

Acceptance Tests: S-3

15d Audit Requirements

ID# S-4 - Administration

Description: The servers need an auditing system managed by administrators

Rationale: Administrators may audit any payment records for players who have paid for membership or purchased payments if necessary.

Fit Criterion: Administrators may audit payments.

Acceptance Tests: S-4

15e Immunity Requirements

ID# S-5 – Malware protection

Description: The product will have a built-in malware detection that will flag any users who are infected with malware and report it.

Rationale: Immunity is maintained by protecting the game servers from any sort of malware/trojans that may possibly infect it.

Fit Criterion: Administrators are notified when malware is detected

Acceptance Tests: S-5

16 Usability and Humanity Requirements

16a Ease of Use Requirements

ID# U-1 – Customizable controls

Description: Controls will fully support customization and modifications that the player may want to change.

Rationale: Default controls may not be preferred by players and control customization is important to support handicapped players.

Fit Criterion: Fully mappable keyboard/gamepad support

Acceptance Tests: U-1

16b Personalization and Internationalization Requirements

ID# U-2 – Personalization and Internationalization Requirements

Description: The application will allow users to set their preferred language.

Rationale: This option will allow players of many different languages to enjoy the game to the fullest.

Fit Criterion: The language selection should correctly change all text in the application to the preferred language.

Acceptance Tests: U-2

16c Learning Requirements

ID# U-3 – Learning to play the game

Description: The game will contain a tutorial that will introduce the players to the different mechanics of the game.

Rationale: This tutorial will allow players to start the game with some knowledge. This will prevent players not knowing what to do in the beginning

Fit Criterion: During testing phase majority of the players will feel comfortable with the game's mechanics after the tutorial.

Acceptance Tests: U-3

16d Understandability and Politeness Requirements

ID# U-4 – Profanity filter

Description: Game will allow players to filter profanity

Rationale: This will allow players to filter out inappropriate language if they wish to.

Fit Criterion: Enabling the profanity filter will correctly censor all profanity within the game, whether from NPCs or other players.

Acceptance Tests: U-4

16e Accessibility Requirements

ID# U-5 – Third party controller support

Description: Along with mappable controls, the product will also support third party controllers that may be designed for handicapped players

Rationale: Handicapped players may have a device that is easier for them to play the game on and it is important that these type of devices are supported.

Fit Criterion: Fully supports all types of controllers and players may submit a ticket if they want a unique one supported.

Acceptance Tests: U-5

16f User Documentation Requirements

ID# U-6 – In-game wiki

Description: There will be an included in-game guide that will allow players to seek help or information if they need it.

Rationale: This will be a complex game with a lot of mechanics, and so it will be good for the players to have a database of information about the game.

Fit Criterion: Players can successfully access the wiki from the game

Acceptance Tests: U-6

16g Training Requirements

Not applicable

17 Look and Feel Requirements

17a Appearance Requirements

ID# LF- 1 – Game must look realistic

Description: The game visually must appear as realistic as possible

Rationale: The more realistic then the more immersed into the world the players will be

Fit Criterion: Players will be immersed in the game and think the game is an extension of their own reality

Acceptance Tests: LF-1

17b Style Requirements

ID# LF-2 – Servers must have unique styles

Description: The product will have two servers, and both will have unique theme and style to them that will capture a broader audience

Rationale: The two distinct styles will allow the players to choose a server that will suit their tastes.

Fit Criterion: Minimum of two servers that users may select with unique styles.

Acceptance Tests: LF-2

18 Operational and Environmental Requirements

18a Expected Physical Environment

Not applicable

18b Requirements for Interfacing with Adjacent Systems

ID# O-2 – Windows, VR, PS5, and Xbox series X

Description: To run on Windows PCs, VR systems, PlayStation 5 and the Xbox series X as these are the latest consoles available.

Rationale: Windows is the most popular system for gaming, VR is a fast-growing system, and the PS5/Xbox series X are the latest consoles. This will allow the project to reach the largest market of players.

Fit Criterion: Game works effectively on the mentioned systems.

Acceptance Tests: O-2

18c Productization Requirements

ID# O-3 – Steam official launch

Description: The product will have a big announcement of the release on Steam and will have a front page on the Steam homepage to attract as many players as possible.

Rationale: Having an Official launch through Steam will maximize the amount of exposure and player base attracted

Fit Criterion: Launches with a frontpage of Steam announcing the game.

Acceptance Tests: O-3

18d Release Requirements

ID# O-4 – Consistent updates

Description: To release consistent content updates for the game.

Rationale: This will make sure that the game stays fresh and continues to keep players engaged and subscribed.

Fit Criterion: Multiple content updates within the first year after release.

Acceptance Tests: O-4

19 Cultural and Political Requirements

19a Cultural Requirements

ID# CP-1 - Gamers

Description: The pirate game is marketed to attract all kinds of gaming communities especially simulation gamers.

Rationale: The application is expected to be widely popular among many gaming communities especially simulation gamers. It is also expected to strengthen the relationships between the people.

Fit Criterion: The game should provide fun and entertainment in communities

Acceptance Tests: CP-1

19b Political Requirements

Not applicable

20 Legal Requirements

20a Compliance Requirements

ID# LE-1 – ESRB Rated

Description: The product will be rated officially by the ESRB to meet the requirements of many countries that specify what age group is allowed

Rationale: It is required by law to put the product on store shelves to have the game rated.

Fit Criterion: Game will be rated T so players may not need to present an ID to purchase the game at official stores.

Acceptance Tests: LE-1

ID# LE-2 – Financial Privacy

Description: Financial Modernization Act of 1999 requires companies to implement safeguards to protect customer information, as well as the Financial Privacy rule which governs how to collect and disclose user information.

Rationale: Must comply with any financial privacy laws because sensitive information will be gathered from users.

Fit Criterion: Audit the protection of user's privacy and make sure that all complies with regulations related to privacy.

Acceptance Tests: LE-1

20b Standards Requirements

ID# STAN-1 – Agile Development

Description: This game will be developed using the Agile methodology.

Rationale: This will allow development to build up the game flexibly and address issues that may spring up quickly and effectively. This will allow a working product up quicker and be able to test extensively.

Fit Criterion: Development successfully follows Agile methodology.

Acceptance Tests: STAN-1

21 Requirements Acceptance Tests

21a Requirements – Test Correspondence Summary

	Requirements										
Test	Req F-1	Req F-2	Req F-3	Req F-4	Req F-5	Req F-6	Req F-7	Req F-8	Req F-9	Req F-10	Req F-11
Test F-1	X										
Test F-2		X									
Test F-3			X								
Test F-4				X							
Test F-5					X						
Test F-6						X	X				
Test F-7								X			
Test F-8									X	X	
Test F-9											X

Table 1 - Requirements - Acceptance Tests Correspondence

	Requirements																
Test	Req P-1	Req P-2	Req P-3	Req P-4	Req DE-1	Req DE-2	Req DE-3	Req DE-4	Req DE-5	Req M-1	Req M-2	Req M-3	Req M-4	Req M-5	Req M-6	Req M-7	Req M-8	Req M-9
Test P-1	X																	
Test P-2		X																
Test P-3			X															
Test P-4				X	X													
Test DE-1						X												
Test DE-2							X											
Test DE-3								X										
Test DE-4									X									
Test DE-5										X								
Test M-1											X							
Test M-2											X	X						
Test M-3											X		X	X				
Test M-4															X			
Test M-5																X		
Test M-6																	X	
Test M-7																		X
Test M-8																		X
Test M-9																		X

Table 2 - Requirements - Acceptance Tests Correspondence

	Requirements											
Test	Req S-1	Req S-2	Req S-3	Req S-4	Req S-5	Req D-A	Req D-B	Req D-C	Req CP-1	Req CP-2	Req LE-1	Req LE-2
Test S-1	X				X							
Test S-2		X			X							
Test S-3			X		X							
Test S-4				X	X							
Test S-5					X							
Test D-A						X	X	X				
Test D-B									X			
Test D-C										X		
Test CP-1											X	
Test CP-2												X
Test LE-1												X
Test LE-2												X

Table 3 - Requirements - Acceptance Tests Correspondence

	Requirements													
Test	Req LE-1	Req LE-2	Req U-1	Req U-2	Req U-3	Req U-4	Req U-5	Req U-6	Req LF-1	Req LF-2	Req O-1	Req O-2	Req O-3	Req O-4
Test LE-1	X													
Test LE-2		X												
Test U-1			X											
Test U-2				X										
Test U-3					X									
Test U-4						X								
Test U-5							X							
Test U-6								X						
Test LF-1									X					
Test LF-2										X				
Test O-1											X			
Test O-2												X		
Test O-3													X	
Test O-4														X

Table 4 - Requirements - Acceptance Tests Correspondence

21b Acceptance Test Descriptions

ID# F-1 – Games launches correctly

Description: A player will attempt to launch the game and if the game launches with no errors then the test was passed

ID# F-2 – User login

Description: A player with an account with the publisher attempts to login to the game and if it asks for subscription (depending on the subscription) or directs to the game, the test is successful.

ID# F-3 – Payment process

Description: A player with a proper payment method will use the client to purchase membership or a microtransaction and the communication with the server and bank will allow the process to be completed where it will deduct the price from the players balance and award them with the purchased item.

ID# F-4 – Form fill up

Description: After the form is filled and send by the player, if received by the publisher and the confirmation of acceptance which is sent by publisher is received by the player then this test will be passed.

ID# F-5 – Select game region

Description: A player will be able to select which region they wish to go to, and the game will move the player to the character creation screen afterwards.

ID# F-6 – Character selection screen

Description: A player will attempt to launch the game and if the game

launches with no errors then the test was passed

ID# F-7 – Character creation

Description: If the player can access all elements to create a character (according to the subscription level) and set it as his/her character in the game, then this test will be passed.

ID# F-8 – Load into the game

Description: A player will be able to load into the game after the character has been selected.

ID# F-9 – Render game graphics

Description: A player will successfully log into the game and once the map is loaded, the game will fully render the nearby objects.

ID# F-10 – Add player on friends list

Description: A player will be able to search up another player's ID and successfully add them to the friends list.

ID# F-11 – Connect to server

Description: A player will select which server they wish to go on and the server will connect the player successfully on the correct server.

ID# P-1 – Loading map speed

Description: A player will attempt to load into the specified server and the process will take no longer than 1-2 minutes.

ID# P-2 – Player latency

Description: Users will input a command such as movement or attacking and the responsiveness of the action should be almost immediately.

ID# P-3 – Crosshair

Description: This is tested by having a player aim at an object and firing a ranged weapon. The projectile of the said weapon should land accurately in alignment of the crosshair (distance may affect this).

ID# P-4 – Server load

Description: There will be a closed beta test where the server capacity will be tested in order to aid in determining whether or not the servers would be able to support a larger audience.

ID# DE-1 – Account data during system failure

Description: This will be tested by purposely causing a system failure and ensuring that no account data may be leaked because of the failure.

ID# DE-2 – Subscription payment transfer

Description: If the bank can receive and validate the payment by the player and also after this, if the bank is able to successfully send notification/confirmation to the player and the admin then this test will be passed.

ID# DE-3 – Player should be able to play the game

Description: If the player is able access all the game elements with respect to his current subscriptions and play the game without interruptions then this test will be passed.

ID# DE-4 – Client must still have functionality without internet connection

Description: A player with no internet connection will attempt to open the client and should be able to change any settings that would require no internet connection.

ID# M-1 – Gaming error handling

Description: When the player reports/enquires about gaming interruptions, then the publisher should be able to solve the issues as soon as possible. After solving the issues if the player confirms he/she can play the game with no interruptions then the test will be passed.

ID# M-2 – Database maintenance

Description: After database checkup or maintenance if the database handler can verify all the gaming related data exist in the database (sometimes after recovery from backup) then this test will be passed.

ID# M-3 – Payment gateway maintenance

Description: After fixing the payment gateway issues when the player reattempts payment, if both the player and admin receive payment acceptance and validation confirmation then this test will be passed.

ID# M-4 – Game support

Description: A player will send a support ticket and the customer support should be able to receive the ticket and see what protocols are needed to be followed to aid the player.

ID# M-5 – Payment support

Description: A player will send a support ticket and the customer support will contact the bank and provide them information to resolve the issue.

ID# M-6 – Personal computers

Description: If the game and all its features run smoothly in personal computers then this test will be passed.

ID# M-7 – Gaming consoles

Description: If the game and all its features run smoothly on gaming consoles then this test will be passed.

ID# M-8 – Virtual reality devices

Description: If the game and all its features run smoothly on virtual reality devices then this test will be passed.

ID# M-9 – Server extension

Description: This is to test whether the server could easily be upgraded in response to an influx of new players. The maximum number of players will be lowered in the closed beta for the limit to be reached. The servers will then be upgraded, and the team will determine how easy it realistically will be.

ID# M-10 – Product lifetime

Description: A player will be tested to see how long it would take them to complete an event and the team will plan content ahead of time and take into consideration how long it will take players to complete it.

ID# S-1 – Payment source access

Description: This will be tested by attempting to process a payment and if the bank receives the payment, then the test will pass.

ID# S-2 – Anti-cheat system

Description: A player will use a well-developed third party to try and modify the in-game files to change information that should not be changed. The team will test the developed anti-cheat system to see whether it detects the third-party application.

ID# S-3 – Hashed passwords

Description: This will be tested by having an administrator going through all the passwords and ensuring that all are properly hashed and incomprehensible.

ID# S-4 – Administration

Description: A payment will already be processed, and the administrator will determine whether they are able to audit any of the payments.

ID# S-5 – Malware protection

Description: This will be tested by having a player with an infected computer try to access playing the game. The administrator will monitor this player and see whether they are flagged, and the player will provide feedback whether they had a message pop-up.

ID# U-1 – Customizable controls

Description: This will be tested by having a player tester, attempt to customize their character and/or make modifications. If the player can then the test will pass.

ID# U-2 – Personalization of languages

Description: This will be tested by having a player tester attempt to set their preferred language. If the game responds by changing the language text and dialogue to the language the player chose, then the test will pass.

ID# U-3 – Learning to play the game

Description: A completely new user will be introduced to the game and will provide feedback to the administrators about their experience.

ID# U-4 – Profanity filter

Description: This will be tested by having a player tester attempt to send a message that contains profanity. If the profanity is filtered out of the message, then this test will pass.

ID# U-5 – Third party controller support

Description: This will be tested by having a variety of controllers and testing the functionality of them and seeing whether they feel fluid enough.

ID# U-6 – In-game wiki

Description: This will be tested by having a player tester attempt to open the wiki to look for help. If the wiki opens and shows relevant information on what the player wanted help with then the test will pass.

ID# D-A – Player account object

Description: Players will log on to their account and if their account exists then this test will pass.

ID# D-A – Player character object

Description: Players will log on and see whether other players are visible on their end.

ID# D-A – Friendly NPC object

Description: This will be tested by having a player go up to a friendly NPC and see that they are not attacked and have proper dialogue.

ID# D-A – Enemy NPC object

Description: This is tested by having a player go up to the enemy NPC's and see that it is possible to engage in combat when appropriate.

ID# D-B – Consumable object

Description: This is tested by having a player attempt to consume non-consumable objects and consumable objects to see whether the expected result occurs.

ID# D-B – Offense Equippable object

Description: A player will attempt to equip the equippable object and see whether they are able to use it in combat or it's intended use.

ID# D-B – Defense Equippable object

Description: A player will equip a defensive object and see whether the amount of damage they take from an enemy is properly reduced.

ID# D-C – Ship object

Description: A player will purchase a ship and see whether the appearance matches the expected and make sure it is able to be boarded and sail properly.

ID# D-E – Island object

Description: Islands will be spawned throughout the game and players will provide feedback whether they were properly filled with enemies and quests or if there were any bugs that occurred.

ID# D-E – Town object

Description: This will be tested by having players going into town and ensuring they are safe from being attacked from enemies and that models and NPCs properly are loaded in.

ID# CP-1 – Gamers

Description: This game should be distributed among all gaming communities especially simulation gamers. If any complaints regarding accessibility of the game are received, then it should be resolved as soon as possible.

ID# LE-1 – Financial Privacy

Description: This will be tested by complying to the laws of financial privacy, if all the requirements are met then this test passes.

ID# LE-2 – Agile Development

Description: This will be tested using a Agile development methodology software like JIRA where the team will be able to view the stories and sprints completed over the course of the game's lifetime.

ID# LF-1 – Game must look realistic

Description: This will be tested by the game rendering, if all the models and textures load fully and the details remain upon inspection in game then this test will pass.

ID# LF-2 – Servers must have unique styles

Description: This will be tested on game load into a server, if the game visually looks like the time-period is different than the other then this test will pass.

ID# O-2 – Windows, VR, PS5, and Xbox series X

Description: This will be tested by having executables that will run on each operating system, if the executable can run on all systems without any errors, then the test will pass.

ID# O-3 – Steam official launch

Description: This will be tested by launching the Steam client and checking if the game is listed on the front page, if it is then the test will pass.

ID# O-4 – Consistent updates

Description: This will be tested by having incremental updates that provide a better experience for the game, whether the updates include bug fixes or additional content. The test will pass with player overall feedback being positive on how the game is reviewed.

III Design

22 Design Goals

- Account information must be securely stored and protect the user information from any sort of data breach.
- User latency to servers must be in an acceptable state and the player should feel no virtual delay from inputs/outputs.
- Application must run at a minimum of 30 frames on devices and higher than 60 frames per second on more modern devices.
- Application must involve cross platform functionality to maximize playerbase for social interactions.
- There must exist some sort of endgame content to maintain the playerbase once they have finished the main quest scenario.

23 Current System Design

There is no existing system to replace.

24 Proposed System Design

24a Initial System Analysis and Class Identification

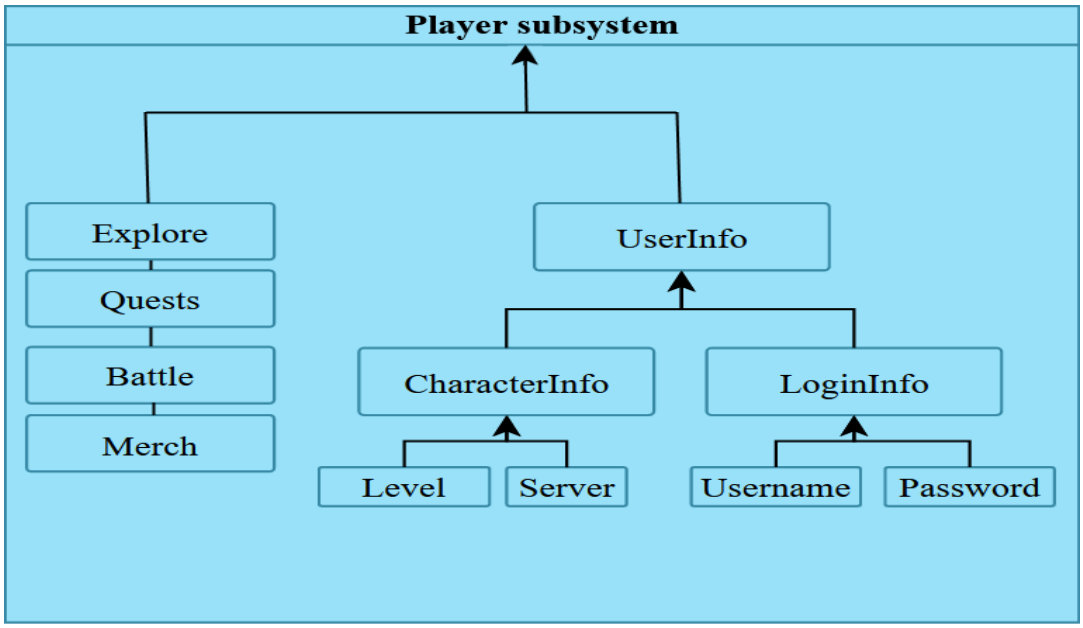


Figure 13: Player Subsystem

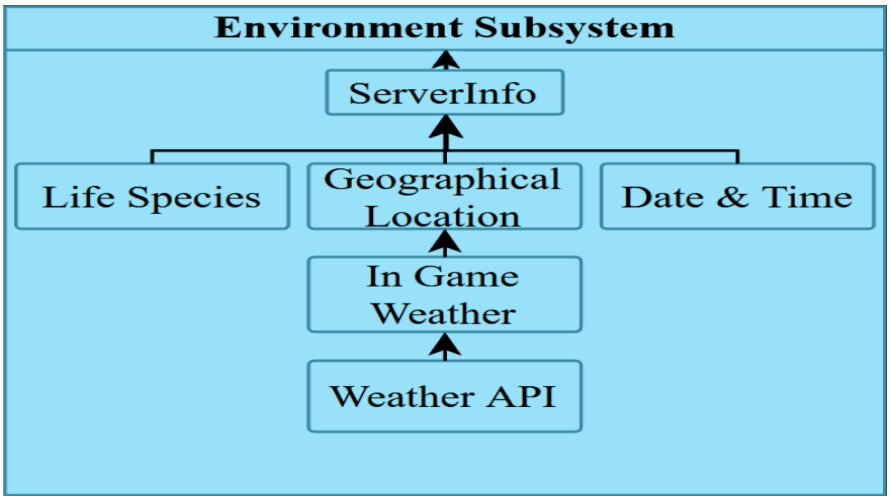


Figure 14: Environment Subsystem

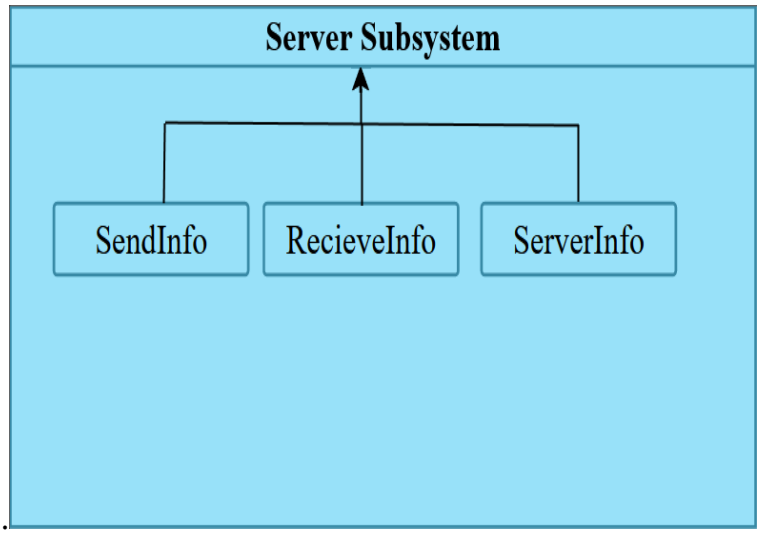


Figure 15: Server Subsystem

24b Dynamic Modelling of Use-Cases

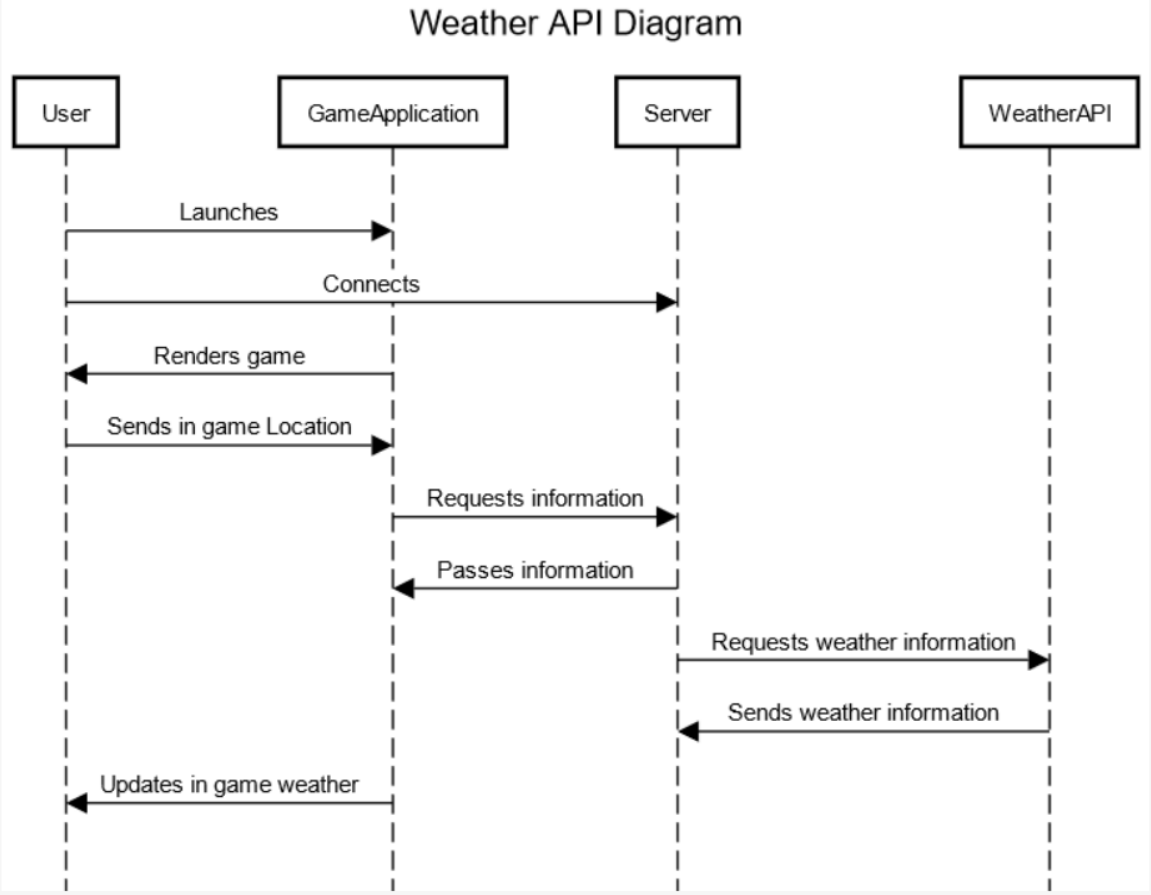


Figure 16: Weather API Dynamic Use Case

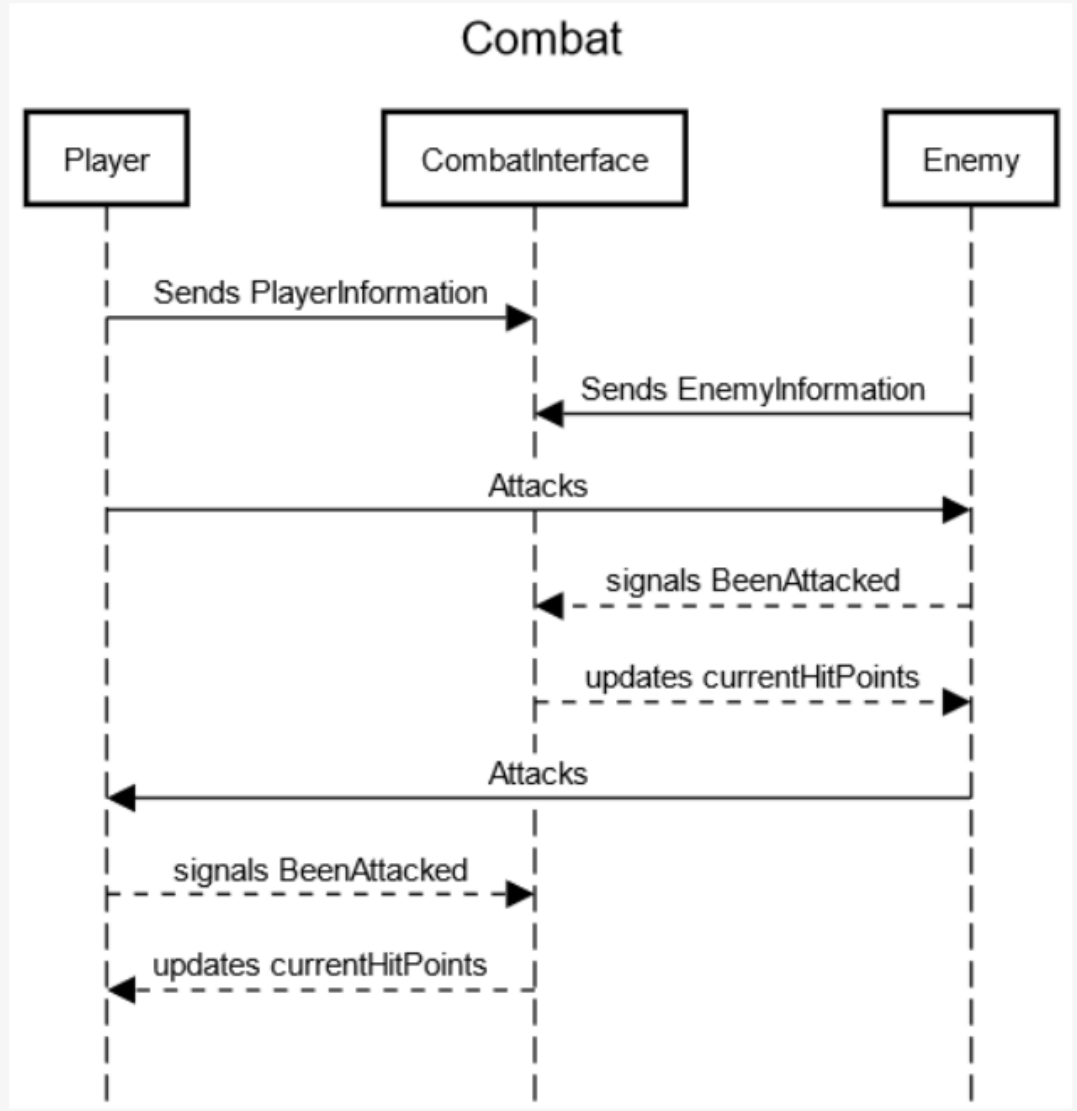


Figure 17: Weather API Dynamic Use Case

24c Proposed System Architecture

The client-server architecture is applied to the system to enable the admin control and provide the game to the gamers. Based on this model and requirement of the game, the admin acts as the server and the gamers who connect to the admin act as the clients.

24d Initial Subsystem Decomposition

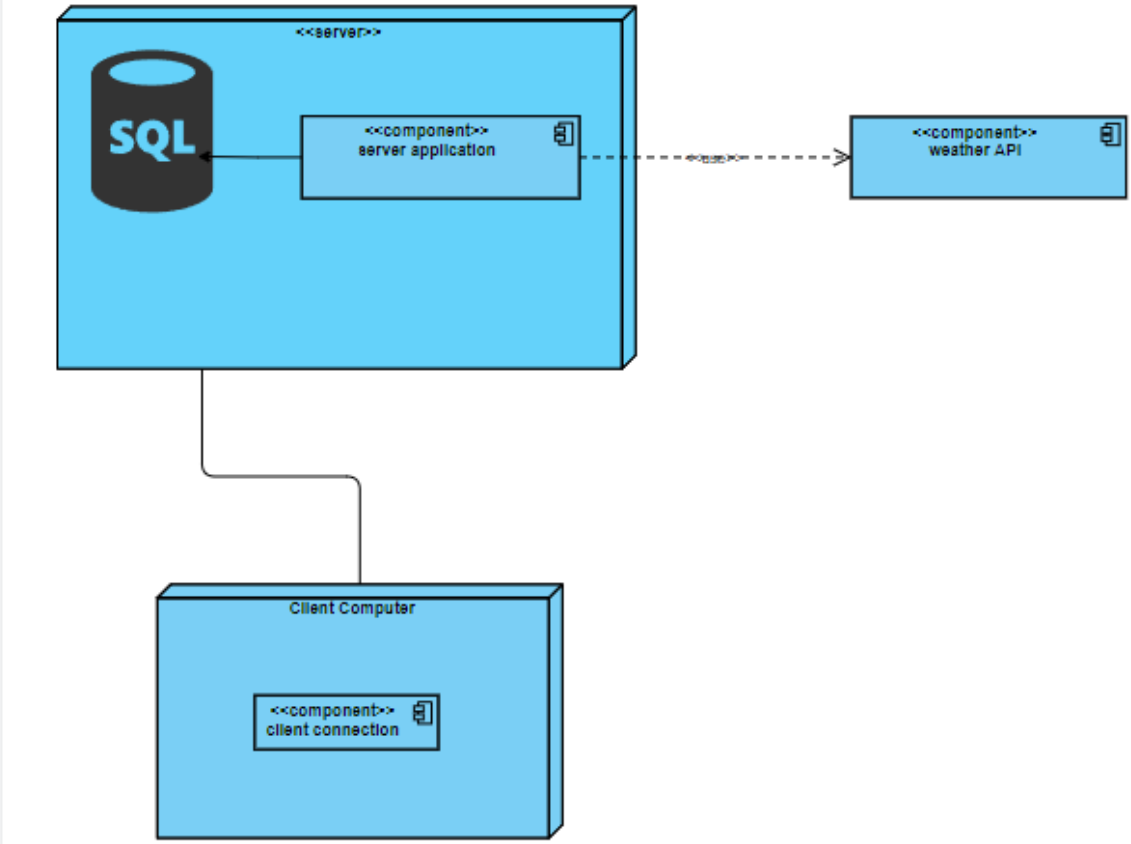
Player subsystem: All the features relating to players such as Explore, Quest, Battle and Merch are contained in this subsystem. It will also contain player related data such as character information and login information.

Environmental Subsystem: This subsystem holds data that relating to the environment in which the game is played. This includes life species, geographical location, date, and time.

Server Subsystem: This subsystem enhances the major communications between the server and clients.

25 Additional Design Considerations

25a Hardware / Software Mapping



25b Persistent Data Management

The game needs to fetch accurate weather data from a weather API. This data needs to be accessible in a fast and persistent manner to conduct proper game play.

For accurate rendering of the game to the proper user persistent data relating to users such as account information, user available items and character information should be

stored by the admin in a database that can be accessed fast and efficiently by the program.

25c Access Control and Security

The accessibility of players to certain gaming elements such as weapons, ships and regions should be based on their payment status. Therefore, proper validation should be made by the admin when players request these gaming elements.

All the entries of sensitive data such as payment information should be handled by the system with standard security features. Also, user login information like username and password should be handled in a similar manner.

25d Global Software Control

There is concern that some of the content of the game may not be suitable for other countries. And as such, we would need to design international versions of the game if we wished to tap into those markets.

25e Boundary Conditions

If a player crashes in the middle of playing we need to make sure that their progress is still properly saved, to prevent items/xp/gold etc. rolling back.

We also need to make sure the game correctly saves its state in the event of a server crash, as this would be catastrophic in terms of player progression if this were to happen and not be properly implemented. It would also cause a lot more resources to be clogged in support due to player complaints.

25f User Interface

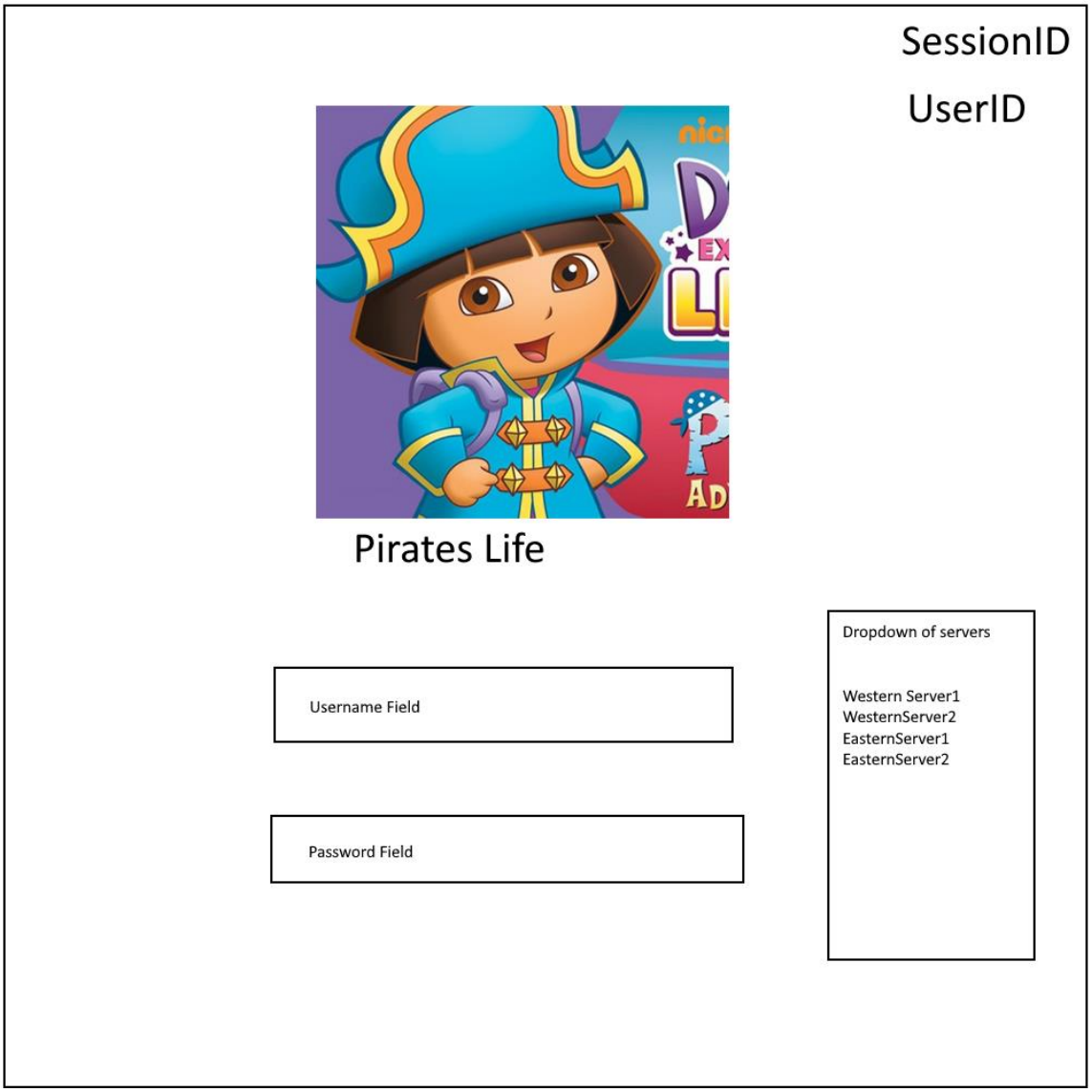


Figure 18: Login Page UI Design

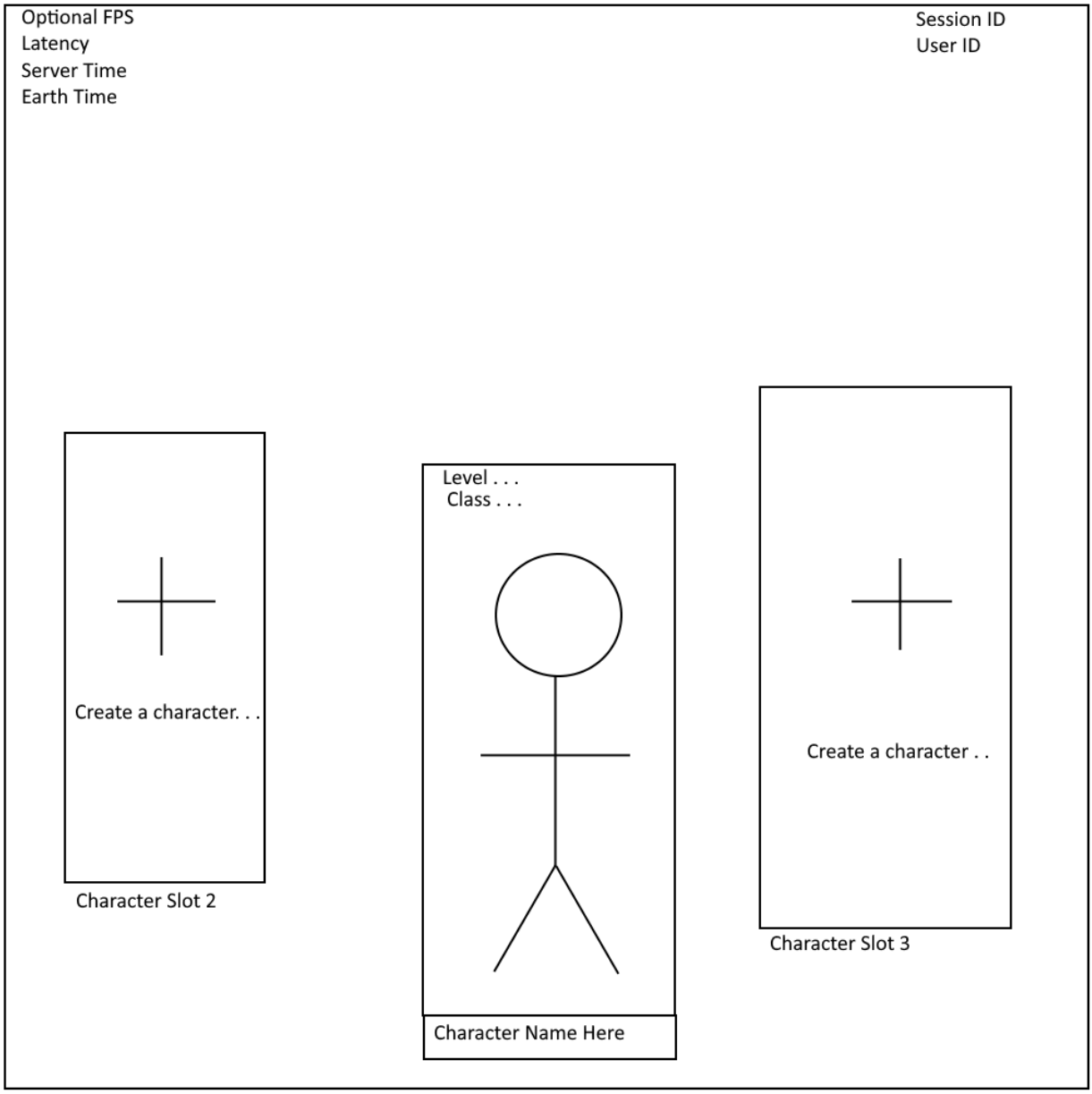


Figure 19: Character Selection Scene

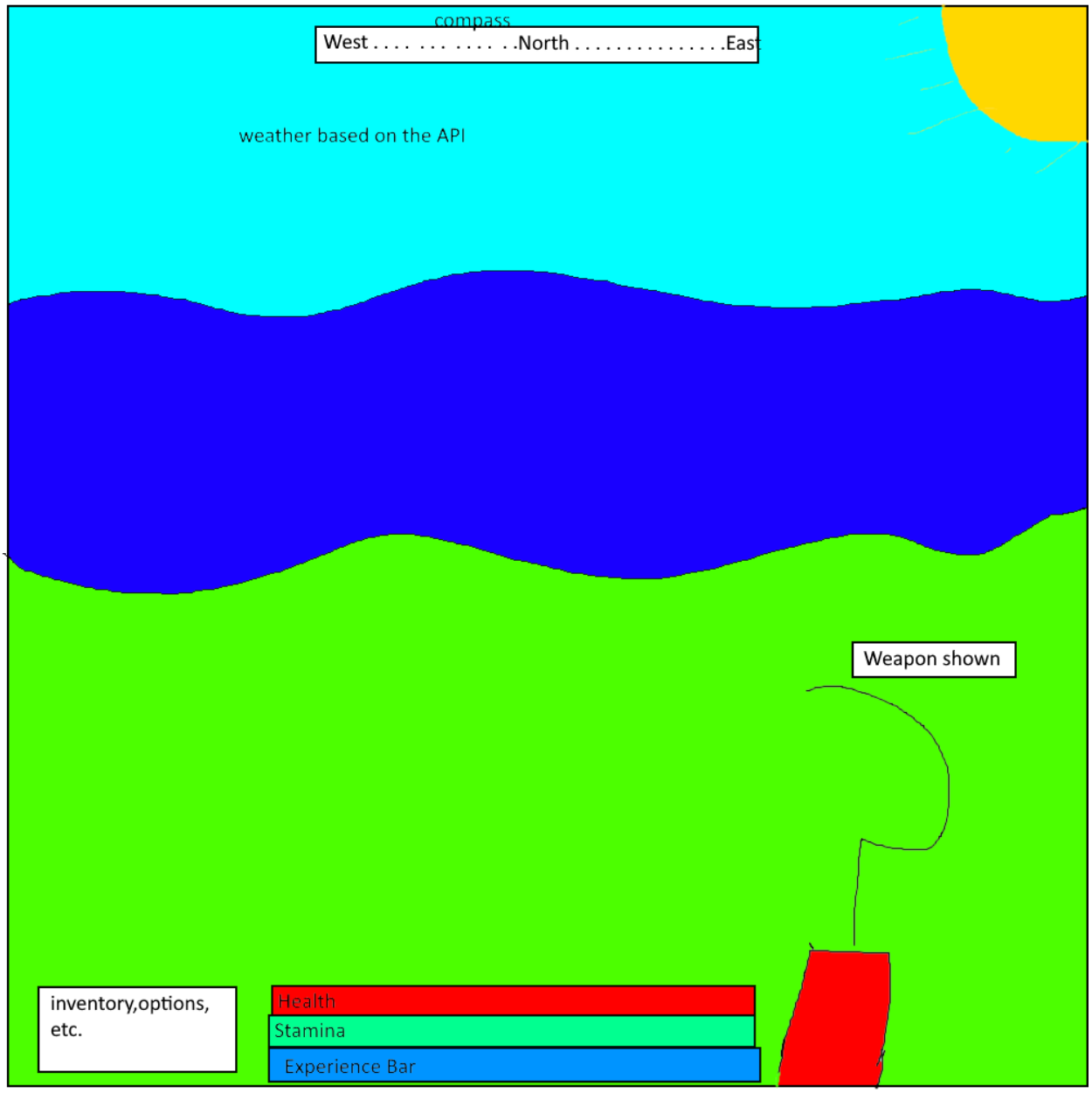
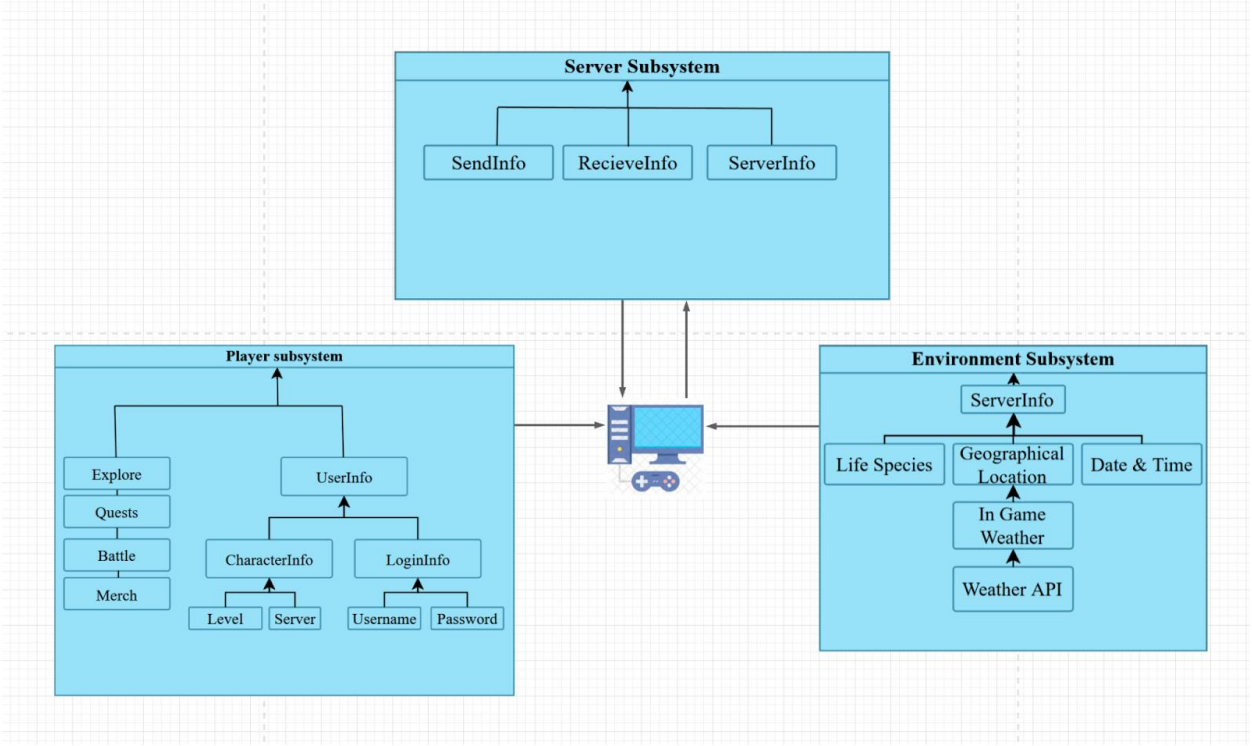


Figure 20: In game UI

25g Application of Design Patterns

The players will use the state design pattern since the player's character will react to the input of the user. The character will react when the user wants to engage in a sword fight, pressing the necessary buttons for example. Or the character will simply move based on player input movement controls.

26 Final System Design

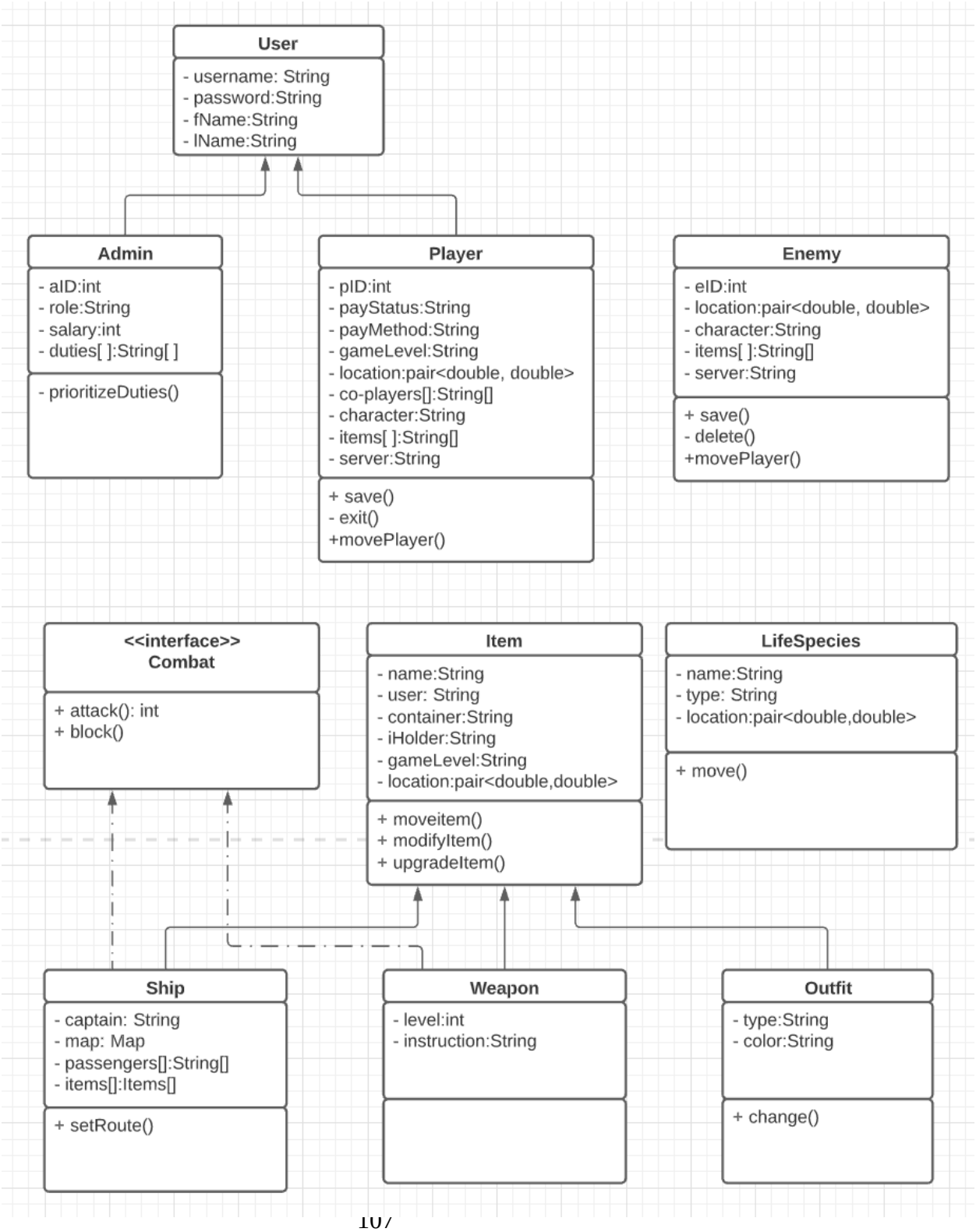


27 Object Design

27a Packages

Not applicable.

27b Player and Environment Subsystem


User
- username: String
- password:String
- fName:String
- lName:String
Admin
- aID:int
- role:String
- salary:int
- duties[ ]:String[ ]
- prioritizeDuties()
Player
- pID:int
- payStatus:String
- payMethod:String
- gameLevel:String
- location:pair<double, double>
- co-players[]:String[]
- character:String
- items[ ]:String[]
- server:String
+ save()
- exit()
+movePlayer()
Enemy
- eID:int
- location:pair<double, double>
- character:String
- items[ ]:String[]
- server:String
+ save()
- delete()
+movePlayer()
<<interface>>
Combat
+ attack(): int
+ block()
Item
- name:String
- user: String
- container:String
- iHolder:String
- gameLevel:String
- location:pair<double,double>
+ moveitem()
+ modifyItem()
+ upgradeItem()
LifeSpecies
- name:String
- type: String
- location:pair<double,double>
+ move()
Ship
- captain: String
- map: Map
- passengers[]:String[]
- items[]:Items[]
+ setRoute()
Weapon
- level:int
- instruction:String
Outfit
- type:String
- color:String
+ change()

IV Project Issues

28 Open Issues

The speed of a user's internet connection and hardware power will have a significant impact on the use of this product. If the minimum requirements are not met the game will most likely be near unplayable. This problem will depend on the users, however there should be a baseline given to consumers so they know whether what they have will be able to run the game effectively.

Minimum requirements will be acknowledged to the player and if the player fails to meet the requirements, then the player will not be able to play the game. The strongest requirements will be the hardware components of their computer to handle graphics and to have a satisfactory internet connection to be able to download content from the server.

29 Off-the-Shelf Solutions

Servers can be hosted using the currently existing Amazon Web Services.

29a Ready-Made Products

Already existing game engines can be used. This will require negotiating with the company that owns the desired game engine.

29b Reusable Components

Depending on the game engine, it should contain much of the physics engines required for the game.

29c Products That Can Be Copied

This is a new type of game that currently does not exist on the market. However that does not mean that we cannot draw inspiration from other successful games and incorporate that into A Pirate's Life. Popular MMO games such as World of Warcraft and Sea of Thieves could be used as a comparison for our product's gameplay and progression.

30 New Problems

30a Effects on the Current Environment

The only effect this game will have on the current environment of video games is illustrate the potential that pirate games, as well as simulation games, have on the open market

30b Effects on the Installed Systems

This product should not interfere with any system as it will be an application that is self-contained.

30c Potential User Problems

This application should not have any issues with users of the software. Users will simply be the players playing the game. The only potential "problems" would be players learning how to play the game for the first time, and making sure the new player experience is good.

30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

N/A

30e Follow-Up Problems

Slow internet connections may cause issues within the game both for the player to experience the problems, and for other players. It is imperative that the system is able to handle issues such as that.

31 Migration to the New Product

31a Requirements for Migration to the New Product

N/A

31b Data That Has to Be Modified or Translated for the New System

N/A

32 Risks

Inadequate sales: It is possible that the game will not sell enough in order to cover the costs of development.

Insufficient funds: There is potential that the game may not be able to be completed before funds are used up, which would require even more capital to be raised.

Game life is short: The game could end up dying within the first few months if player reviews are bad or players quickly lose interest in the game. This would effectively kill the income stream as the monetization will be predicated on monthly subscriptions.

Low quality product: Product could end up being of a lower quality than promised, which would quickly lower player intrigue, and potentially spell death for the product.

Lack of accountability: With a game this large it is possible that teams do not hold up their end of the work, which would cause headaches for every team and put the project behind the timeline.

Pushing back release date: In the event of setbacks, it may be needed to push back the release date of the game despite advertising money being spent for the original release.

Releasing unfinished product: In the event of setbacks, the publisher could still desire to release the game at the originally stated date. This would mean the game would be released in an unfinished state, which most likely would not go over well with the player base.

33 Costs

	By Complexity			Total
Source	**Low**	**Avg**	**High**	
Logic file	2 x 8 = 16	4 x 11 = 44	4 x 8 = 32	92
Interface files	3 x 5 = 15	3 x 7 = 35	6 x 4 = 24	74
External Input	2 x 3 = 6	3 x 5 = 15	5 x 2 = 10	31
External Output	2 x 4 = 8	3 x 6 = 18	3 x 5 = 15	41
External Interface Files	4 x 5 = 20	5 x 10 = 50	5 x 8 = 40	90

- Number of input and output flows on the work context - 30
- Number of business events - 10
- Number of product use cases - 10
- Number of functional requirements - 30
- Number of nonfunctional requirements - 40
- Number of requirements constraints - 31
- Number of function points - 328

The total number of cases we have is 479 and the team expects that each should be tested for roughly 8 hours so the total number of hours expected to be put into testing should be 3832 hours. Assuming wages are 20 per hour, the team has predicted this project to cost roughly $76,64

34 Waiting Room

- **Paid DLC**

  Description: Players will be allowed to purchase microtransactions or cosmetics.

  Rationale: This will help monetize the overall game and allow the development

  team to continue to create more content. Will also satisfy shareholders to know the ability of the game to generate more revenue.

  Fit Criterion: Users must currently have an active membership and a character created to purchase microtransactions.

- **More Future Endgame Raids**

Description: The game will continue to release endgame raids for the players.

Rationale: There must exist more future plans of content for players who have reached the endgame of Pirate's Life in order to incentivize the players to continue playing the game.

Fit Criterion: Players must reach a certain level and specific quests to unlock the raid.

- **Player vs Player content**

  Description: The ability to battle other players may be implemented in later versions.

  Rationale: Playerbases in massively multiplayer online games traditionally are split between Player vs Enemies (PVE) and Player vs Players (PVP) communities, the development team plans to cater towards both audiences.

  Fit Criterion: Players must be in a specific region to have PVP enabled and be a certain level.

- **Community Events**

  Description: Specific holidays will be celebrated throughout the game

  Rationale: On days such as Halloween or Christmas, the team will implement events that will be specially themed for the occasion and reward players for being involved and doing the event.

Fit Criterion: Players must be a certain point through the story/main quest in order to be able to participate.

35 Ideas for Solutions

Language

- Back-end: C#
- Front-end: Unity, UnrealEngine
- Database: MySQL

IDE

- Back-end: Unity, VSCode
- Database: MySQL WorkBench

Framework: LibGDX

Testing: xUnit

Storage: private server

36 Project Retrospective

What Worked:

- Splitting the work up evenly.
- Brainstorming project ideas and agreeing on one we all felt good about.
- Creating a group chat for us to quickly communicate throughout the semester.

Improvements for next time:

- Meet a bit more often in order to discuss the project.
- Focus on planning for future projects

- More in-depth understanding of class diagrams and flowchart diagrams to better understand the processes.

V Glossary

User - any person who uses the product

Player - an individual who uses the product to play the game associated with the project

Administrator - an individual who manages the project system, applying the bug fixes, and maintaining the project.

Ship - the player's means of transportation within the project's game in the sea.

Weather API - an API that will continuously get the weather information and send the information to the server.

Server - system which will control the project's game for each player, updating in-game information and sending information to the administrator.

VI References / Bibliography

[1] Robertson and Robertson, Mastering the Requirements Process.

[2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.

[3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

[4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

VII Index

Definitions..... 25
Functional Requirements..... .58
Project Description.....5
Project Issues 92
Purpose..... 6
Requirements 28, 61
Scope.....81
System Design 91
Use Case.....30
Design 86, 98
Mandated Constraints 22
Project Goals..... 7
Project Overview 6
Relevant Facts..... 27
Scenario Diagrams..... 13
Stakeholders..... 18
Test Correspondence..... 76
User Interface..... 90