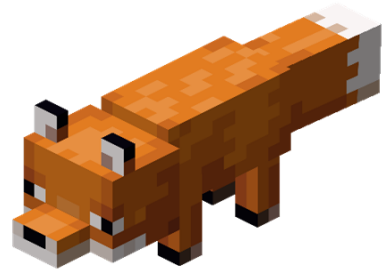
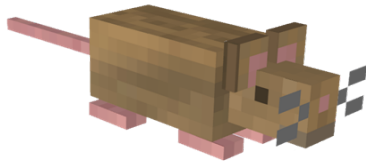


# Pac-Rats



# Pac-Rats



## **A Browser Based Implementation of Multiplayer Pac-Man**

**Prepared by  
Jonathon Repta, Jennifer Alonso, Edward Liang, Amal Syed  
at the  
University of Illinois Chicago**

**October 2020**

## Table of Contents

List of Figures	5
List of Tables	6
Project Description Edward	7
Project Overview	8
The Purpose of the Project	8
The User Business or Background of the Project Effort	8
Goals of the Project	8
Measurement	8
The Scope of the Work	9
The Current Situation	9
The Context of the Work	9
Work Partitioning	10
Competing Products	10
The Scope of the Product	10
Scenario Diagram(s)	11
Product Scenario List	12
Individual Product Scenarios	12
Stakeholders	12
The Client	12
The Customer	12
Hands-On Users of the Product	13
Maintenance Users and Service Technicians	13
Other Stakeholders	13
User Participation	13
Priorities Assigned to Users	13
Mandated Constraints	14
Solution Constraints	14
Implementation Environment of the Current System	14
Partner or Collaborative Applications	14
Off-the-Shelf Software	14
Anticipated Workplace Environment	14
Schedule Constraints	14
Budget Constraints	15
Naming Conventions and Definitions	15
Definitions of Key Terms	15
UML and Other Notation Used in This Document	15
Data Dictionary for Any Included Models	15

Relevant Facts and Assumptions	15
Facts	15
Assumptions	15
Requirements	<b>16</b>
9 Product Use Cases	16
9a Use Case Diagrams	16
9b Product Use Case List	17
9c Individual Product Use Cases	17
10 Functional Requirements	19
11 Data Requirements	20
12 Performance Requirements	21
12a Speed and Latency Requirements	21
12b Precision or Accuracy Requirements	21
12c Capacity Requirements	21
13 Dependability Requirements	22
13a Reliability Requirements	22
13b Availability Requirements	22
13c Robustness or Fault-Tolerance Requirements	22
13d Safety-Critical Requirements	23
14 Maintainability and Supportability Requirements	23
14a Maintenance Requirements	23
14b Supportability Requirements	23
14c Adaptability Requirements	24
14d Scalability or Extensibility Requirements	24
14e Longevity Requirements	24
15 Security Requirements	25
15a Access Requirements	25
15b Integrity Requirements	25
15c Privacy Requirements	26
15e Immunity Requirements	26
16 Usability and Humanity Requirements	26
16a Ease of Use Requirements	26
16b Personalization and Internationalization Requirements	27
16c Learning Requirements	27
16d Understandability and Politeness Requirements	27
16e Accessibility Requirements	27
16f User Documentation Requirements	28
16g Training Requirements	28

17 Look and Feel Requirements	29
17a Appearance Requirements	29
17b Style Requirements	29
18 Operational and Environmental Requirements	29
18a Expected Physical Environment	29
18b Requirements for Interfacing with Adjacent Systems	30
18c Productization Requirements	30
18d Release Requirements	30
19 Cultural and Political Requirements	30
19a Cultural Requirements	30
19b Political Requirements	31
20 Legal Requirements	31
20a Compliance Requirements	31
20b Standards Requirements	31
21 Requirements Acceptance Tests	32
21a Requirements – Test Correspondence Summary	32
21b Acceptance Test Descriptions	32
<b>Design</b>	<b>33</b>
22 Design Goals	33
23 Current System Design	33
24 Proposed System Design	34
24a Initial System Analysis and Class Identification	34
24b Dynamic Modelling of Use-Cases	34
24c Proposed System Architecture	35
24d Initial Subsystem Decomposition	35
25 Additional Design Considerations	36
25a Hardware / Software Mapping	36
25b Persistent Data Management	36
25c Access Control and Security	36
25d Global Software Control	37
25e Boundary Conditions	37
25f User Interface	37
25g Application of Design Patterns	37
26 Final System Design	38
27 Object Design	38
27a Packages	38
27b Subsystem I: Site Module	39
27c Subsystem II: Game Module	39

27d Game/Site Adapter	40
Project Issues	<b>40</b>
28 Open Issues	40
29 Off-the-Shelf Solutions	40
29a Ready-Made Products	40
29b Reusable Components	41
29c Products That Can Be Copied	41
30 New Problems	41
30a Effects on the Current Environment	41
30b Effects on the Installed Systems	41
30c Potential User Problems	41
30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	42
30e Follow-Up Problems	42
31 Migration to the New Product	42
32 Risks	42
33 Costs	42
34 Waiting Room	43
35 Ideas for Solutions	43
36 Project Retrospective	43
Glossary	<b>44</b>
References/Bibliography	<b>44</b>
Index	<b>44</b>

## **List of Figures**

Figure 2 - Sample Use Case Diagram from Bruegge & DuToit ( modified )	16
Figure 3 - Sample Use Case Diagram from Robertson and Robertson	17

## List of Tables

*Table 2 - Requirements - Acceptance Tests Correspondence*

31

# **I Project Description**

## **1 Project Overview**

Pac-Rats is a multiplayer implementation of the arcade classic, Pac-Man. The purpose of Pac-Rats is to offer a unique twist on a classic game and, in doing so, reinvigorate the interest of players who are familiar with the original, but are looking for something fresh. In order to achieve our goal of revitalizing Pac-Man, we will introduce a number of new features. Features such as new game modes (co-op, free-for-all), an extensive leaderboard system, and community based features like map creation/sharing. Pac-Rats will offer an accessible experience that will be enjoyable to players of all skill levels.

Users can easily link up with friends and do not require a high end pc to play. Once signed-up they will be registered into the database and can login anytime as long they have a computer and internet. This game is intended for all ages and since it is inspired by the classic pac-man game, the mechanics/controls are easy to learn. Even a child will be able to pick it quickly and play with each other.

## **2 The Purpose of the Project**

The project aims to create a game that combines traditional Pac-Man style gameplay with in order to create a game that will be accessible to a wide range of players whilst still maintaining the attention of the serious “hardcore” gaming audience. The hope is that the project will be a profitable venture for the publisher whilst being an enjoyable experience for a variety of players.

### **2a The User Business or Background of the Project Effort**

The business that the client is a part of is the gaming industry. Specifically the section of the games industry which is in charge of remaking/remastering classic games. This is relevant to the project since the developers need to understand the wants and needs of the average gamer and balance that with the requirements given by the stakeholders.

### **2b Goals of the Project**

The goal of this project is to take a classic game and breathe new life into it by applying modern gaming practices. Furthermore, the project aims to make a classic game profitable again by retrofitting modern monetization schemes into a game that didn't have them originally. Essentially the project aims to take the addicting retro gameplay of Pac-Man and combine it with innovations in multiplayer gaming that have come since.

### **2c Measurement**

Pac-Rats will keep detailed usage statistics about users in order to monitor audience retention. Things like the number of new users every day/month/week that log on, the amount of time the average user spends on the app, etc. The success of our product will be measured against the user retention of the average Steam game in order to see if our product is able to compete with comparable dedicated gaming products. If our



game is able to match the retention rate of the average Steam game (or better yet, exceed it) we will have achieved our goal.

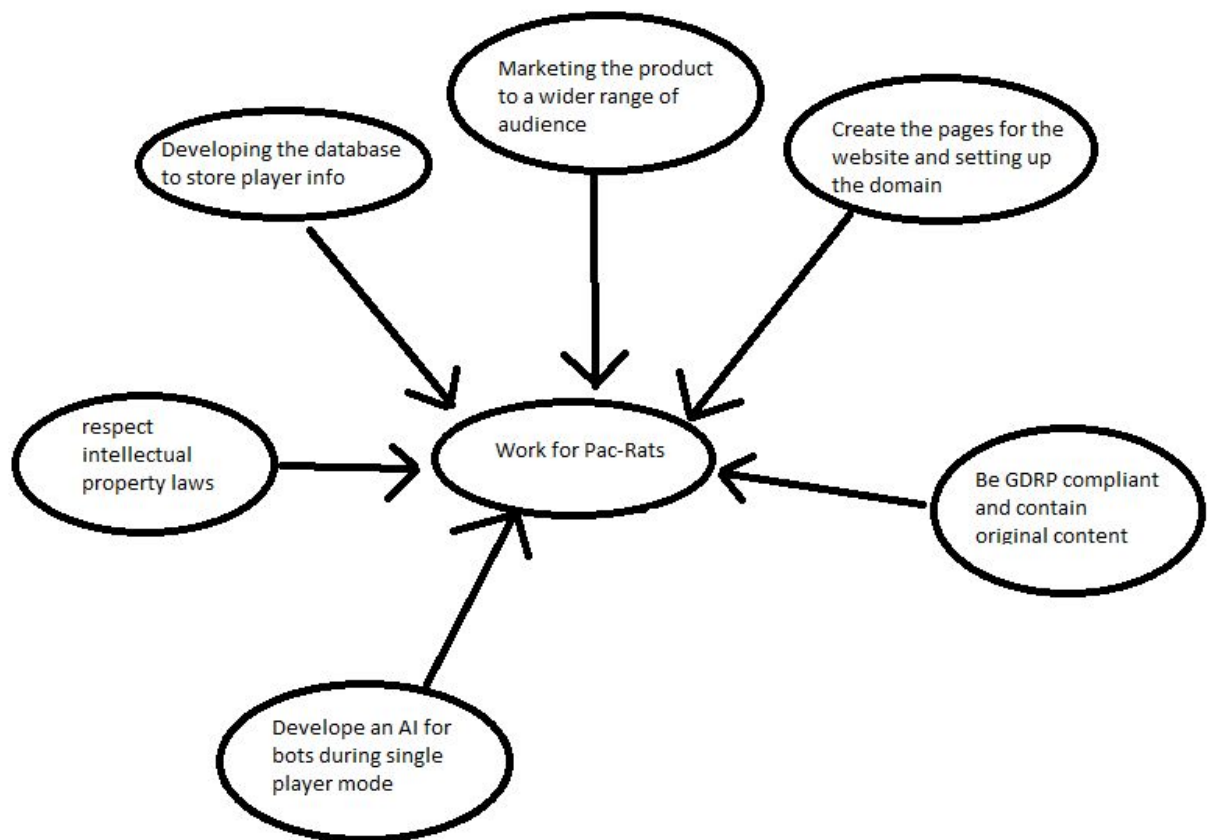
### 3 The Scope of the Work

The business is that of the games industry. Specifically, recreating old classical arcade games that everybody knows growing up and featuring new and original content. The work can be addressed with this project by recreating a traditional pac-man game from the arcade with pac-rats.

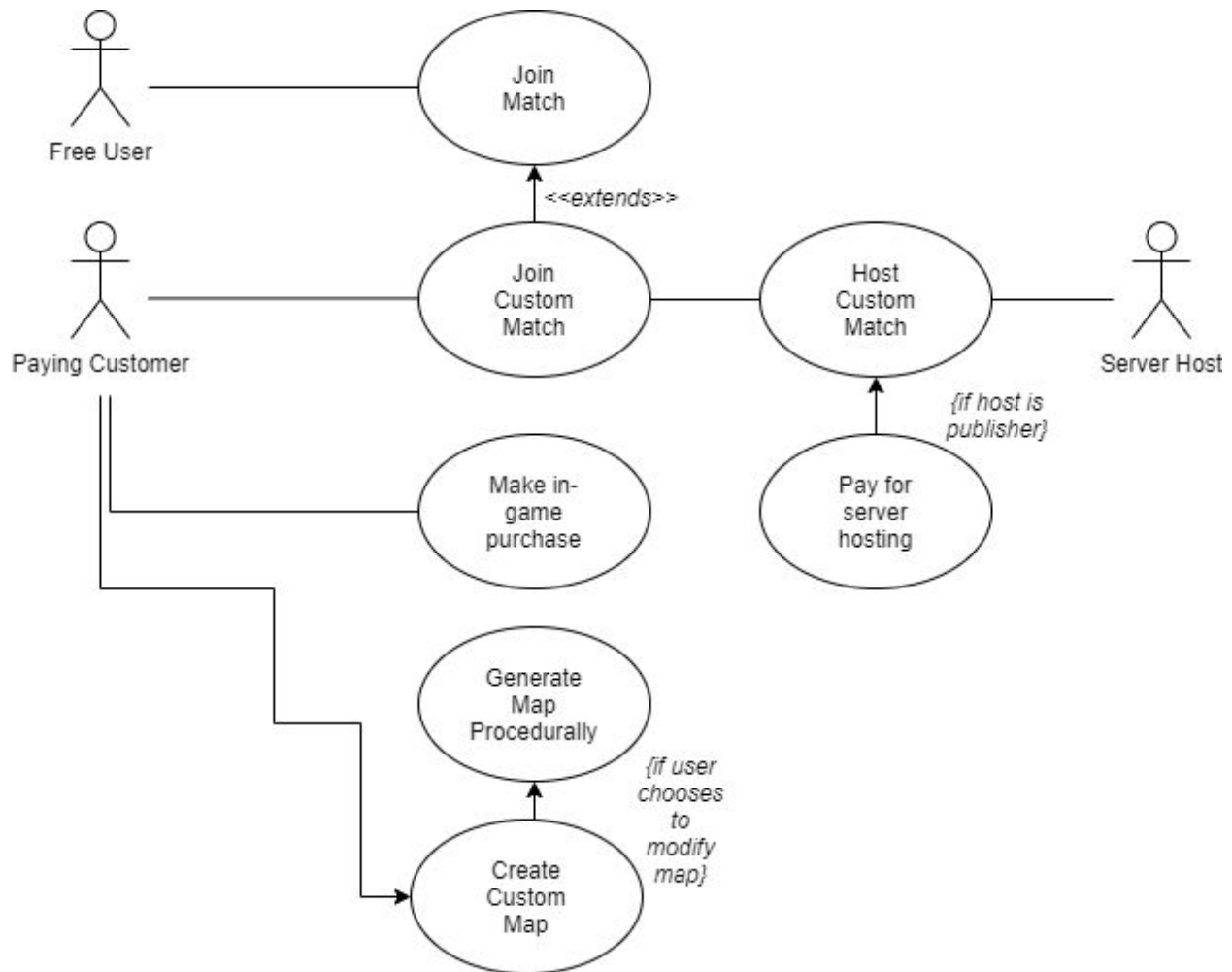
#### 3a The Current Situation

Presently, the client has ported a standard version of Pac-Man to a wide variety of dedicated gaming consoles and mobile devices. All of these releases contain monetization schemes which either involve upfront payment for usage of the software, or are monetized via in game advertisements.

#### 3b The Context of the Work



### 3c Work Partitioning



### 3d Competing Products

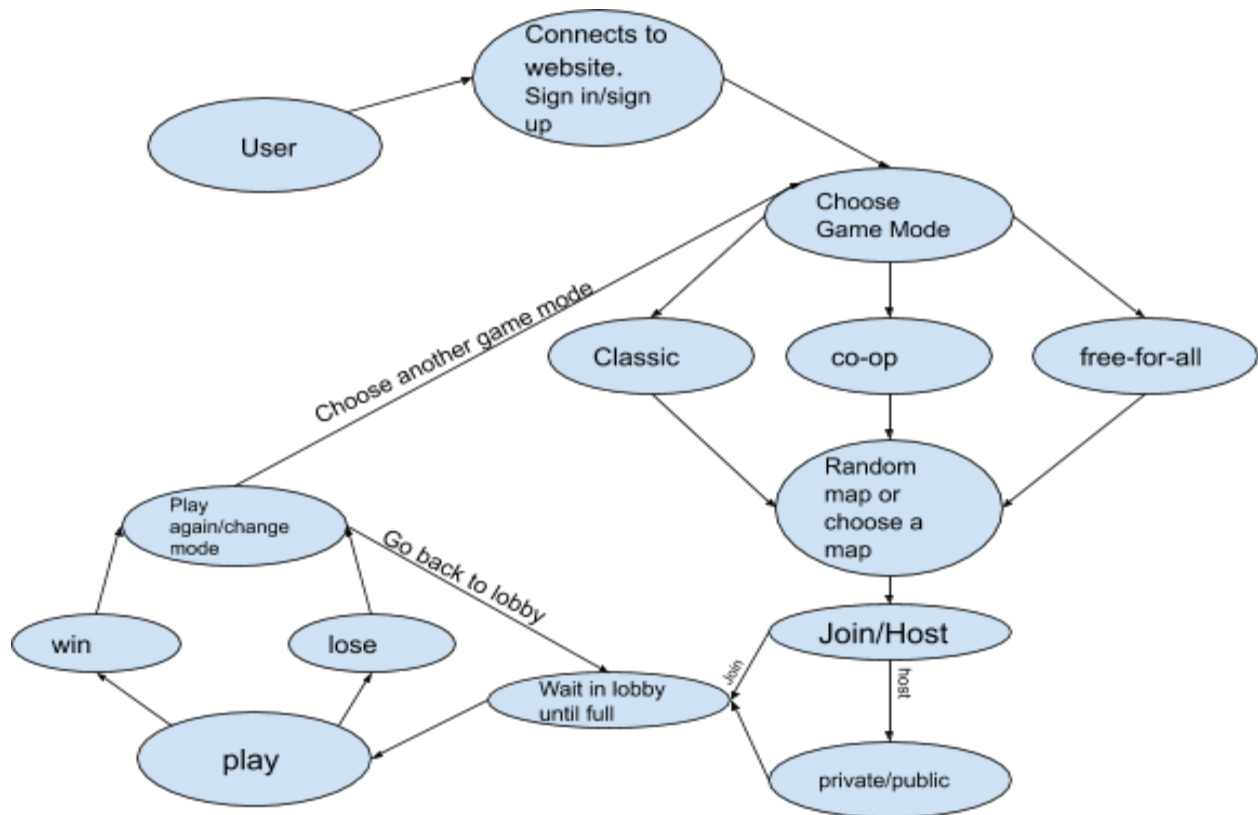
The game will be competing in the casual, free-to-play, market. Direct competition includes other Pac-Man style games, whereas indirect competition includes free to play mobile games in general. Our product is still useful as there is still room to innovate within the free-to-play market. By revilitazing a classic game we would be able to capture a market of players who would not otherwise be interested in a more serious/novel game.

## 4 The Scope of the Product

The product must handle all aspects pertaining to the product's gameplay. This means that the product must allow for the user to join a server, host their own custom server, browse community made maps, create their own maps, browse the leaderboards and share their creations with the community. The product will not cover the creation of

things outside of the scope of the game. This includes the creation of a payment platform, the infrastructure needed for server hosting, etc.

#### 4a Scenario Diagram(s)



#### 4b Product Scenario List

Scenario name	External Actors	Relevant information
Account creation	users, customers	user creation will be the same regardless of whether the user is a customer
Gameplay mode selection	users, customers	
Hosting/Joining game	host, publisher	The user can opt to host the game themselves, or pay for hosting from the publisher
Waiting in Lobby	users, customers	
Game Starts	users, customers	
Returning to lobby or start new game	users, customers	

#### 4c Individual Product Scenarios

1. New users create signup through the website and returning players sign in.
2. Choose one of the many game modes to play (co-op, free for all)
3. Enter a seed to randomly generate a map for the match or choose from one of the many community made maps or one of your self made maps from map creator.
4. Choose to host or join a game. If hosting, choose to make it private for friends to join or public for random players. If joining a game, wait in the lobby.
5. Wait for the lobby to fill up to the required number of players and host to start.
6. Play game
7. Choose to remain in the current lobby or return to the main menu to try another game mode.

## 5 Stakeholders

### 5a The Client

The client would be a game company/publisher such as Bandai Namco, Koei Tecmo, Capcom, etc. Any company interested in revitalizing one of their existing IPs and applying modern gameplay mechanics and monetization schemes in order to capitalize on a growing market would be interested in our product.

### 5b The Customer

The customer of the product would be a user who is invested in the game to a point where they would be willing to spend money upgrading their experience. Upgrades

such as private servers hosted by the publisher, cosmetic upgrades like player skins, and the removal of in game advertisements would allow customers to differentiate themselves and enhance their experience.

### **5c Hands-On Users of the Product**

Hands on users of the product would be both customers and people who play the game free of charge. In contrast to our paying customers, “free” users are less involved in the game and are more likely to be a casual gamer who plays in order to kill some time -- rather than to be involved competitively. This is relevant as customers interests may conflict with the interests of the average user.

### **5d Maintenance Users and Service Technicians**

Since the game will run entirely in the browser, the game will require the user to do absolutely no installations/updates/etc whatsoever. If the player wishes, they will have the option to host their own private server or pay a subscription for hosting from the publisher.

### **5e Other Stakeholders**

Beta-testers: This group will test out the game before it is fully released for the general public. They test out the game for still remaining glitches and bugs that still need to be addressed.

Marketing Manager: They are responsible for advertising the product. This game can only reach so many players by itself and from friends/family. Marketing Managers help spread it to other players from various backgrounds and create a diverse community.

### **5f User Participation**

User participation is not required for development of the project, however, the developers may find it useful to provide polls for the community to determine which direction the game should head in in order to improve player retention and increase the quality of the product.

### **5g Priorities Assigned to Users**

Paying customers are given much higher priority than the average user as they are the ones who are most invested in the game. Furthermore, they are the ones who are most likely to continue to spend money on the game . Whilst a free user may install the game and play it for a week, a premium user is likely to continue playing the game for a far longer period of time.

## **6 Mandated Constraints**

### **6a Solution Constraints**

The game must be color blind friendly in order to be accessible for color blind users. The game must not rely on auditory cues in order to accommodate users who are hard of hearing or are in an environment where they cannot hear audio cues (public transit)

### **6b Implementation Environment of the Current System**

This game operates exclusively within the browser, meaning that it is inherently compatible with any device that is capable of running a web browser. Because of the web based nature of our product, developers are afforded the ability to easily create *native* versions of the product through the use of software like React Native, Electron, etc. In regards to hardware requirements, since the game will retain a retro aesthetic it won't require a powerful machine to run. All the user needs is a stable internet connection and a way to interface with the game.

### **6c Partner or Collaborative Applications**

The product must be compatible with all major browsers. Specifically, it should be compatible with browsers that account for at least 80% of the world's internet users. This means no use of outdated software like [Flash](#), and no cutting edge software like [WebAssembly](#). The game should be written using a game engine that utilizes JavaScript and HTML5 -- to maximize compatibility.

### **6d Off-the-Shelf Software**

No commercial off the shelf software is required for our product. The product will instead rely entirely on free, open source, software for auxiliary code outside of what is needed for the game itself. Relying on free software will allow development costs to stay down whilst still utilizing high quality libraries.

### **6e Anticipated Workplace Environment**

The product is intended for a wide variety of environments, in accordance with the games philosophy of accessibility. Because of this, information must be conveyed to the user in a clear and concise way that doesn't necessarily require any sort of auditory feedback.

### **6f Schedule Constraints**

The developers must focus on the most fundamental elements of the application before moving onto smaller issues. This means getting the foundations of the program functional before focusing on niceties. Graphical fidelity comes second to a functioning game and in game purchases come second to graphical fidelity. If the game is too broken, players will not be interested in the product. If the product is too frustrating to look at/use, players will not be interested in spending money on the

product. Because of this, our priorities must be gameplay, then aesthetics, then monetization.

## **6g Budget Constraints**

The budget given will be relatively small due to several factors. Firstly, since the developers will not be required to write any auxiliary structural code, the development team will not need as many members. Secondly, the aim of this project is not to push graphical fidelity or story but instead create a fun game that appeals to a wide audience.

## **7 Naming Conventions and Definitions**

### **7a Definitions of Key Terms**

Pac-Rats: Equivalent to Pac-Man in a typical game of Pac-Man, only in *Pac-Rats* there will be multiple player controlled Pac-Men on a single board.

Foxes: Natural predators of Pac-Rats. They are the equivalent to ghosts in Pacman and work collectively to hunt down the Pac-Rats.

Free-for-all: A game mode in which every rat is pitted against one another. There are no foxes on the board, instead, Pac-Rats compete with one another.

Co-op: A game mode in which multiple Pac-Rats play alongside one another to achieve the highest score possible without any opponents getting in their way.

### **7b UML and Other Notation Used in This Document**

The database(s) must be blueprinted thoroughly through the use of an ER-diagram. Use case diagrams must be employed to model both casual player interaction as well as dedicated user interaction.

### **7c Data Dictionary for Any Included Models**

The project will use a relational database to store all of the data required for gameplay. The specific flavor of the database is down to whichever implementation the developers are most comfortable with, however, they may not use a non-relational database. Map recommendations will not be on a star system, but instead be either “liked” by the user or “disliked”. This is because [both Youtube and Netflix](#) have discovered that almost every user rates their experience either 5 stars or 1 star.

## **8 Relevant Facts and Assumptions**

### **8a Facts**

- The game must comply with the privacy laws of every market the game is intended for. Most notably, the game must be GDPR compliant.
- The game must not infringe on any other publishers intellectual property. This means no stolen assets, copyrighted terms, gameplay systems, software, etc.

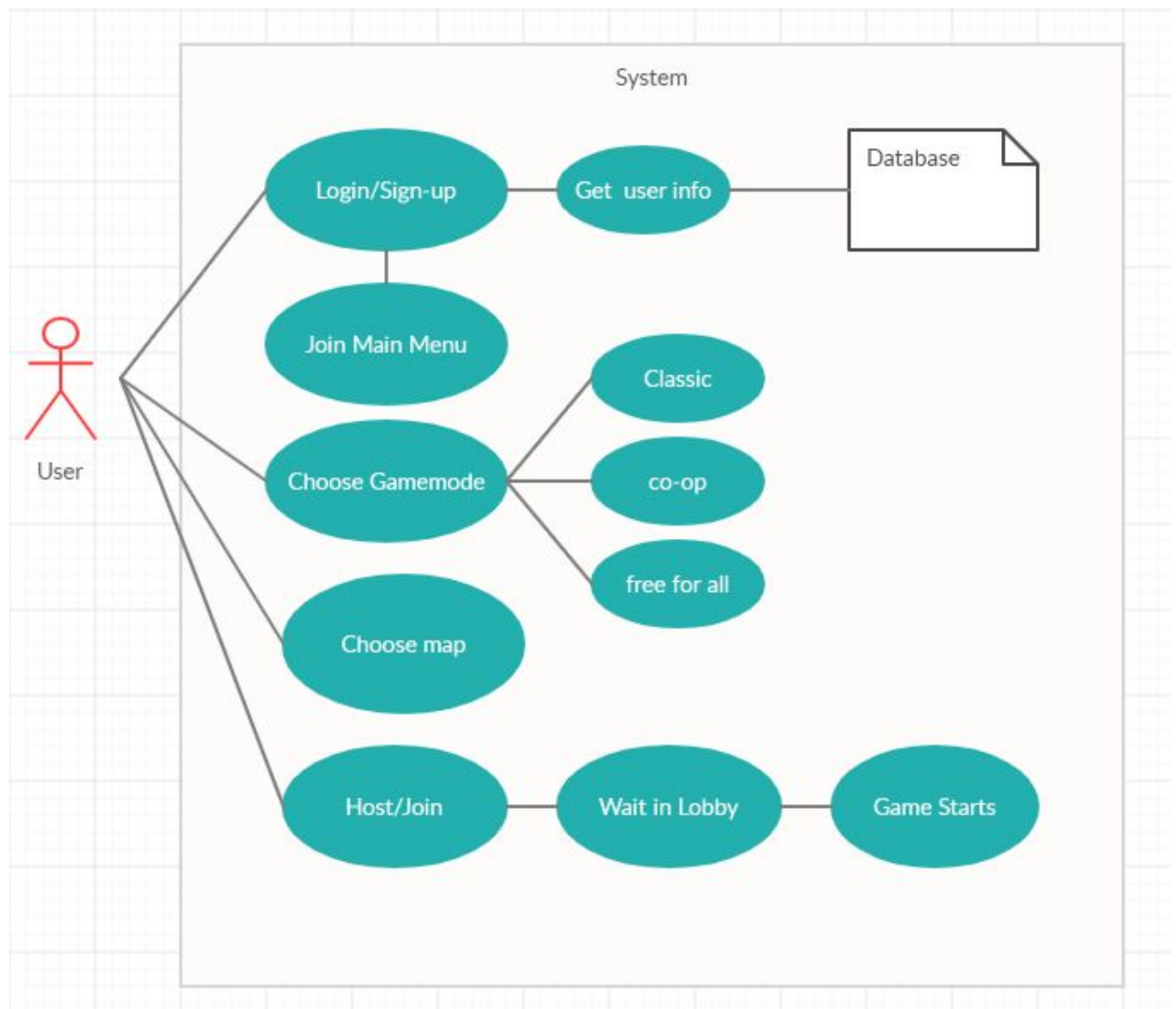
### 8b Assumptions

- 90+% of the players will not spend any money on in game purchases of any variety
- The player is familiar with how a standard game of Pac-Man
- The player has a stable internet connection

## II Requirements

### 9 Product Use Cases

#### 9a Use Case Diagrams





## 9b Product Use Case List

1. Signing in
2. Gameplay loop

## 9c Individual Product Use Cases

Use case ID: 1                      Name: Signing in

pre-conditions: User has internet access

post-conditions: User is able to play Pac-Rats

Initiated by: User (either F2P or P2W)

Triggering Event: Visiting the Pac-Rats homepage

Additional Actors: None

Sequence of Events:

1. User visits the domain that hosts Pac-Rats
2. User reaches Pac-Rats landing page
3. User clicks on the “login” option so that they can log in to their previously existing account
4. Server responds with a login
5. User signs into their account using their login credentials
6. User is redirected to the gameplay scene of Pac-Rats and is able to join a game

Alternatives: The user does not have an account and instead would like to create one

Exceptions: There is an internal server error when accessing the user’s credentials/creating the user’s session information

Use case ID: 2

Name: Joining a game

pre-conditions: User has internet access

post-conditions: User is playing a game of Pac-Rats

Initiated by: User (either F2P or P2W)

Triggering Event: Signing to an account

Additional Actors: There has to be at least one other user who is also in the lobby

Sequence of Events:

1. After the user has successfully logged into a Pac-Rats account, they will be greeted with a list of gameplay modes
2. The user will select the gameplay mode that they want to play
3. Once the user has selected a game mode, they will be shown several additional options regarding how they would like to play said game mode.
4. If the user has selected the default/standard options, they will immediately be put in a lobby
5. If the user has specific options for how they would like to play a game (they would like to play with specific people/have a privately hosted game) they can proceed to enter those options
6. The player will wait a certain period of time in the gameplay lobby and will be able to chat with other players.
7. Once the countdown has finished, the players will begin their match

Alternatives: The user decides they would rather play a different game mode

Exceptions: The user is unable to join any game because there are no other players currently available to play with.

## 10 Functional Requirements

### F0# - Account Creation

**Description:** The player should be able to sign up for an account.

**Rationale:** In order to keep track of important information regarding many aspects of the game, it is important to have a functional account creation/maintenance system.

**Fit Criterion:** Users should be able to create an account and their data should persist across sessions (be stored permanently within a database).

**Acceptance Tests:** Account handling

### F1# - Account Sign in

**Description:** Similar to there being an account creation requirement, the system will also need to have .

**Rationale:** This is important for both obvious and non obvious reasons. In contrast to a game like slither.io that has you sign up for an account every session, our game will require that there is a sign in system so that users can save their progress.

**Fit Criterion:** Users should be able to sign in to a previously created account and access all of their saved information.

**Acceptance Tests:** Account handling

### F2# - Joining a game

**Description:** The player should be able to join a game with options matching their gameplay preferences.

**Rationale:** In order to have a functioning game, we need to have a system in place for users to select the type of game they are interested in playing and to have them be placed into a lobby that matches their gameplay preferences.

**Fit Criterion:** Users should be able to join a game with all options reflective of their choices.

**Acceptance Tests:** 2

### **F3# - Gameplay loop**

**Description:** The user should be able to play a game of Pac-Rats with other users

**Rationale:** In order to have a properly functioning game, it is paramount that the gameplay loop works correct and all player actions are perfectly accounted for and coordinated cross connections. This means having a system of real time bidirectional communication between clients (users)

**Fit Criterion:** The user experience during gameplay should be equivalent to as if they were playing a multiplayer game locally rather than over the internet.

**Acceptance Tests:** gameplay loop

### **F4# - Completing a game**

**Description:** Information regarding a completed game should be stored and reflected in other aspects of the game's design.

**Rationale:** In order to have an enjoyable user experience the results of a game should persist between sessions and be stored for review. This includes keeping track of player score, the amount of games played, and the ability for users to share their gameplay snippets with other players so that they can be viewed.

**Fit Criterion:** The user experience during gameplay should be equivalent to as if they were playing a multiplayer game locally rather than over the internet.

**Acceptance Tests:** gameplay loop

## **11 Data Requirements**

### **U1# - Account Creation**

**Description:** The system must provide a mean for all users of this product to create an account with a unique name/id and email

**Rationale:** Having unique id/email helps the system to track the users easier and create reports.

**Fit Criterion:** The name/email of every user must be unique and contain no duplicate in the database.

**Acceptance Tests:** Account handling, data persistence

## 12 Performance Requirements

### 12a Speed and Latency Requirements

#### U2# - Gameplay Latency

**Description:** The system must limit lag between stimulus and provide fast response time.

**Rationale:** We do not want latency in the system to produce a horrible UI experience for the user.

**Fit Criterion:** The login process to the server should not take more than 300 milliseconds.

**Acceptance Tests:** Gameplay loop

### 12b Precision or Accuracy Requirements

#### U3# - Server client communication

**Description:** The system must provide the means for the user to be able to precisely retrieve and render information.

**Rationale:** Handing out misinformation may result in bugs or crashes in our system

**Fit Criterion:** The system must provide means for the user to play the game in real time with other players..

**Acceptance Tests:** Gameplay loop, data persistence

### 12c Capacity Requirements

#### U4# - Concurrent Players

**Description:** The system must be able to support many user connections at the same time.

**Rationale:** This is an online multiplayer game and it would not function as one without multiple connections to the server.

**Fit Criterion:** The game, per match, holds upward of 10 connections concurrently.

**Acceptance Tests:** Gameplay loop

## 13 Dependability Requirements

### 13a Reliability Requirements

#### U5# - Data integrity

**Description:** The system must provide the means to safely store user information from system failures

**Rationale:** Unreliable systems might result in unnecessary data lost in the server.

**Fit Criterion:** If a system failure occurs, the user can safely login back to the system with all their previous info once the system is back operating.

**Acceptance Tests:** Data persistence

### 13b Availability Requirements

#### U6# - Game availability

**Description:** The System must be on active for every day of the year unless due to mandatory system maintenance.

**Rationale:** If the system is kept active only up to a certain time zone, then other players at a different time zone will not be able to play the game.

**Fit Criterion:** Users in the United States will be able to play with users from Britain any time of the day as long there is stable internet ID# and/or names here . . . ternet.

**Acceptance Tests:** Gameplay loop

### 13c Robustness or Fault-Tolerance Requirements

#### U7# - Maintenance notifications

**Description:** The system must deliver messages to all users in game when maintenance is required.

**Rationale:** This is to keep some functionality intact and notify players with a warning before the systems go on maintenance.

**Fit Criterion:** Upon detecting a fault the system will go on a 1 hour maintenance break.

**Acceptance Tests:** Data persistence

## 13d Safety-Critical Requirements

### U8# - Safety compliance

**Description:** The system must provide messages for the user to view the number of hours they have been in game.

**Rationale:** This is to prevent any user from abusing the game and playing more than over the recommended amount.

**Fit Criterion:** For users playing this game for more than 3 hours a message will appear to remind them the total hours played today.

**Acceptance Tests:** Gameplay loop, user experience

## 14 Maintainability and Supportability Requirements

### 14a Maintenance Requirements

#### U9# - Game maintenance

**Description:** The game will be maintained by its administrators, in charge of the build of the administration's technical aspects will be the Sysadmin.

**Rationale:** Having a dedicated team of administrators will alleviate other stakeholders the need to maintain the site. Having a dedicated Sysadmin will allow for the administration team to handle technical issues without the need to contact the development team.

**Fit Criterion:** Have a team of at least 5 administrators and one system administrator.

**Acceptance Tests:** Data integrity, user experience

### 14b Supportability Requirements

#### U10# - Customer Support

**Description:** The system must provide the means for the user to be able to write an inquiry. There must be a team of people with experience in customer support which will read and answer customer inquiries.

**Rationale:** We want a product to have reliable customer service and be able to gain the trust of our customers. In order to do so we must ensure that there is a reliable team of people who will be able to answer as soon as possible.

**Fit Criterion:** There should be a team of people available 24 hours a day on weekdays if possible all seven days of the week.

**Acceptance Tests:** User experience

## 14c Adaptability Requirements

### U11# - Devices

**Description:** The system must provide the means for the game to be able to run on devices that support browsing.

**Rationale:** Since the game will be run on browsers such as Chrome, Firefox, Edge, etc. Devices like tablets and computers should be able to run the browser game.

**Fit Criterion:** A library computer is sufficient for entering the game with the proper url.

**Acceptance Tests:** User experience, gameplay loop

## 14d Scalability or Extensibility Requirements

### U12# - Concurrent users

**Description:** The game must be able to host thousands of users simultaneously without the server getting overloaded

**Rationale:** In order to allow for the potential of the game growing in user base drastically, we must create a scalable system that allows for thousands of users to play the game concurrently.

**Fit Criterion:** Write unit tests to test strain on the system with 10,000 concurrent users.

**Acceptance Tests:** Data persistence

## 14e Longevity Requirements

### U13# - Lifetime of Product

**Description:** The system must run as long as there are customers playing and supporting the game

**Rationale:** The browser game will be developed and further supported as long as there are people playing and supporting the game. Since we need to have enough revenue to keep the game up and running.

**Fit Criterion:** Every year we will have to review if the game is still being played or people are purchasing in game products to continue supporting the game.

**Acceptance Tests:** User experience



## 15 Security Requirements

### 15a Access Requirements

#### U13# - Accounts

**Description:** P2W (pay to win) users will have access to more advanced features than F2P (free to play) users.

**Rationale:** This will incentivize users to pay for premium P2W accounts.

**Fit Criterion:** Separate features into those available to P2W and F2P users.

**Acceptance Tests:** Account handling

### 15b Integrity Requirements

#### U14# - Database access

**Description:** Database routes should only be given write access to the database if absolutely necessary, otherwise, they should be read only.

**Rationale:** This greatly lowers the potential of data loss due to an SQL injection attack.

**Fit Criterion:** Routes should be split into two categories, read only, and read/write.

**Acceptance Tests:** Data integrity

#### U15# - Database deletions

**Description:** Database deletions should be exclusively soft deletes unless regional laws conflict with.

**Rationale:** Soft deletes will increase database integrity by keeping all information stored within the database, even if a mistaken deletion occurs.

**Fit Criterion:** Deletions from the database should only occur after a certain amount of time has passed from the soft delete.

**Acceptance Tests:** Data integrity

## 15c Privacy Requirements

### U16# - Privacy of the Costumer

**Description:** The system must provide the means to safely store confidential information for the user.

**Rationale:** This is to protect the safety of all the users and integrity of the business.

**Fit Criterion:** The system must safely store all surveys conducted and provide an option to be anonymous for any of our surveys.

**Acceptance Tests:** User experience, data integrity

## 15e Immunity Requirements

### U17# - Password protection

**Description:** The system must provide the means for all user sensitive information.

**Rationale:** This is to reduce hackers from possibly going into the database from stealing sensitive information.

**Fit Criterion:** The password entered must fit the length requirement and certain symbols and characters need to be added before creating an account. The system must prompt the user to enter their credit card information every time before making an in game purchase.

**Acceptance Tests:** The system must provide a safety procedure whenever a user has to deal with something confidential (account creation, in game purchases).

## 16 Usability and Humanity Requirements

### 16a Ease of Use Requirements

#### U18# - Easy usage of UI

**Description:** The system must provide a user interface for the users such that it is easy to use properly and hard to use improperly.

**Rationale:** Players can easily pick up the game and start experiencing the game without much trouble.

**Fit Criterion:** All symbols and icons in the game must be easily distinguishable.

**Acceptance Tests:** The game will conduct a survey to check for ease of use in the product.

## **16b Personalization and Internationalization Requirements**

### **U19# - Personalization**

**Description:** The systems must provide the means for user to modify some of the controllers for gameplay.

**Rationale:** This is to improve the ease of use in the product when the defaults in the system are not compatible with the user.

**Fit Criterion:** Users can edit controls made to the keyboard to move the avatar.

**Acceptance Tests:** All the letters, numbers, and symbols in the keyboard must be configurable.

## **16c Learning Requirements**

### **U20# - Easy Learning**

**Description:** The system must provide the means for the user to easily pick the game without previous experiences or spending much time at a tutorial.

**Rationale:** If a game is too difficult to learn then it will reduce any incentives for users to try the it out

**Fit Criterion:** The tutorial supplied will not take more than 3 minutes to go through

**Acceptance Tests:** The game will track the number of times each user will go through the tutorial.

## **16d Understandability and Politeness Requirements**

### **U21# - Family friendliness**

**Description:** The system must provide the means to censor and block out profanity displayed in messages.

**Rationale:** This is to protect younger audiences. The purpose of this product is to not promote profanity/racism/discrimination

**Fit Criterion:** All inappropriate words are censored out and displayed with asterisk

**Acceptance Tests:** A list of illegal words will be taken from a dictionary and applied to the algorithm to check for possible errors.

## **16e Accessibility Requirements**

### **U22# - Screen reader compatibility**

**Description:** The game must be fully compatible with screen readers, with labeled buttons and all images having alt text.

**Rationale:** This will make the game more accessible to visually impaired players, giving it a wider audience and potentially more customers.

**Fit Criterion:** Game should be accessible to users who use a screen reader

**Acceptance Tests:** User experience

#### **U23# - Color blind friendly color schemes**

**Description:** The game must have color schemes that use high contrast to allow even colorblind users to be able to distinguish elements.

**Rationale:** This will give the game a wider audience and thus more customers.

**Fit Criterion:** Game should be able to be played by users with all types of color blindness

**Acceptance Tests:** User experience

### **16f User Documentation Requirements**

#### **U24# - Usage Documentation**

**Description:** Although the game will focus on being accessible without any documentation, we will nonetheless provide documentation to any users who find it necessary.

**Rationale:** In order to keep our game accessible, we must provide thorough documentation regarding game operation. This includes not only how to play the game, but also how to join a specific server, submit gameplay snippets to the highlight browser etc.

**Fit Criterion:** Documentation should be accessible to any user

**Acceptance Tests:** User experience

### **16g Training Requirements**

#### **U25# - Moderator training**

**Description:** What specific users need to know in order to moderate the game

**Rationale:** Moderation is key when operating any sort of multiplayer game. In order to have a good community, the moderators must understand how to use moderation tools to their full potential.

**Fit Criterion:** Moderators must be quizzed on their ability to use the moderation tools before they can become moderators.

**Acceptance Tests:** Gameplay loop

## **17 Look and Feel Requirements**

### **17a Appearance Requirements**

#### **U26# - Friendly Branding**

**Description:** Website branding should focus on friendliness and accessibility over professionalism

**Rationale:** In order to convey the accessible, friendly, and open nature of the game, the branding must reflect that. These design goals can be reflected in the branding by employing the use of colorful colors, rounded sans-serif fonts, and a cute mascot.

**Fit Criterion:** Use a colorful, yet consistent colors for UI, have a cute mascot

**Acceptance Tests:** User experience

### **17b Style Requirements**

#### **U27# - Friendly Styling**

**Description:** Website style should focus on friendliness and accessibility over professionalism

**Rationale:** In accordance with the game's design philosophy of accessibility, Pac-Rats must present a forgiving, friendly, and intuitive UX. Making the UX friendly will increase accessibility to users less familiar with online games.

**Fit Criterion:** Avoid using serious technical language, allow for undoing/redoin in regards to state management.

**Acceptance Tests:** User experience

## **18 Operational and Environmental Requirements**

### **18a Expected Physical Environment**

#### **U28# - Devices and WiFi**

**Description:** User must have a device which can support browser games, such as a computer, phone, or tablet and have strong wifi connection.

**Rationale:** We want audiences from many different backgrounds and including the ones that do not have access to a computer.

**Fit Criterion:** The game will be cross platformed from pc to android and apple phones.

**Acceptance Tests:** Accessibility

## **18b Requirements for Interfacing with Adjacent Systems**

The product has no particular partner applications it is required to interface with.

## **18c Productization Requirements**

### **U29# - Internet Connection**

**Description:** The system must provide the means for the user to connect from the browser through a stable internet connection

**Rationale:** This to minimize the requirement for users to be able to play the game and maximize the number of users that will be able to join.

**Fit Criterion:** Even people with low end devices such as chrome will be able to join and play.

**Acceptance Tests:** Gameplay loop

## **18d Release Requirements**

### **U30# - Continuous release cycle**

**Description:** The game will take the approach of many modern online games and have a continuous release cycle with “seasons” separating each major release.

**Rationale:** Giving the game perpetual updates and distinguished by unique “seasons” will keep the user base interested in the game and allow for the product to be continuously improved.

**Fit Criterion:** Gameplay loop

**Acceptance Tests:** There should be a new gameplay season for each season of the year. Although these seasons don’t need to align with actual seasons, there should be at least four of them.

## **19 Cultural and Political Requirements**

### **19a Cultural Requirements**

#### **U31# - Careful symbol usage**

**Description:** Symbols need to be carefully chosen to avoid confusion

**Rationale:** When publishing the game for an international audience, we need to choose symbols wisely so that the message conveyed by them makes sense regardless of culture.

**Fit Criterion:** Symbols must have a generally agreed upon understanding. We can find which symbols work by looking at which symbols are used by other major websites.

**Acceptance Tests:** User experience

## **19b Political Requirements**

### **ID# - Subscription reminders**

**Description:** In order to increase revenue, we will include payment reminders in order to remind F2P (free to play) users of the advantages of a P2W account.

**Rationale:** Interaction reminders like these have been shown to increase revenue, and will make management happy.

**Fit Criterion:** The landing page must contain at least one P2W reminder

**Acceptance Tests:** List ID# and/or names here . . .

## **20 Legal Requirements**

### **20a Compliance Requirements**

#### **#1 - GDPR Compliance**

**Description:** Our game must be compliant with the GDPR (General Data Protection Regulation) for users within the European union.

**Rationale:** This will open up the EU as a market, increasing the pool of potential players greatly.

**Fit Criterion:** To fit this requirement, we must follow all guidelines specified within the [GDPR guidelines](#) document.

**Acceptance Tests:** Accessibility

### **20b Standards Requirements**

There are no particular standards the product needs to adhere to.

## 21 Requirements Acceptance Tests

### 21a Requirements – Test Correspondence Summary

Tests	Requirements														
	F0	F1	F2	F3	F4	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
Account handling	X	X				X									
Gameplay loop			X	X	X		X	X	X		X		X		X
Data persistence	X	X						X		X		X		X	X
User experience												X	X	X	X
Accessibility															
	U15	U17	U18	U19	U20	U21	U22	U23	U24	U25	U26	U27	U28	U29	U30
Account handling	X														
Gameplay loop		X		X		X		X			X		X		
Data persistence			X		X		X		X			X		X	X
User experience								X	X	X	X				
Accessibility					X	X								X	

### 21b Acceptance Test Descriptions

#### T1 # - Account Handling

**Description:** This test handles all of the functionality surrounding account creation and maintenance. This includes creating an account, updating account information, and moderation duties like the banning of accounts.

#### T2 # - Gameplay loop

**Description:** This test handles all of the functionality surrounding the core gameplay loop of Pac-Rats. This includes setting up a game, playing a game, and completing a game.

#### T3 # - Data persistence

**Description:** This test handles all of the functionality surrounding data persistence. This includes data persistence between sessions, database integrity, and database maintenance.

#### T4 # - User experience

**Description:** This test regards the user experience of the game. This includes what users think of the game's branding/styling, whether they are able to find what they are looking for within the GUI, how intuitive the game is, etc.



### **T5 # - Accessibility**

**Description:** This test handles all of the accessibility features/requirements of our game. This includes testing the color blindness options, the hardware requirements, environmental requirements, and so on.

## **III Design**

### **22 Design Goals**

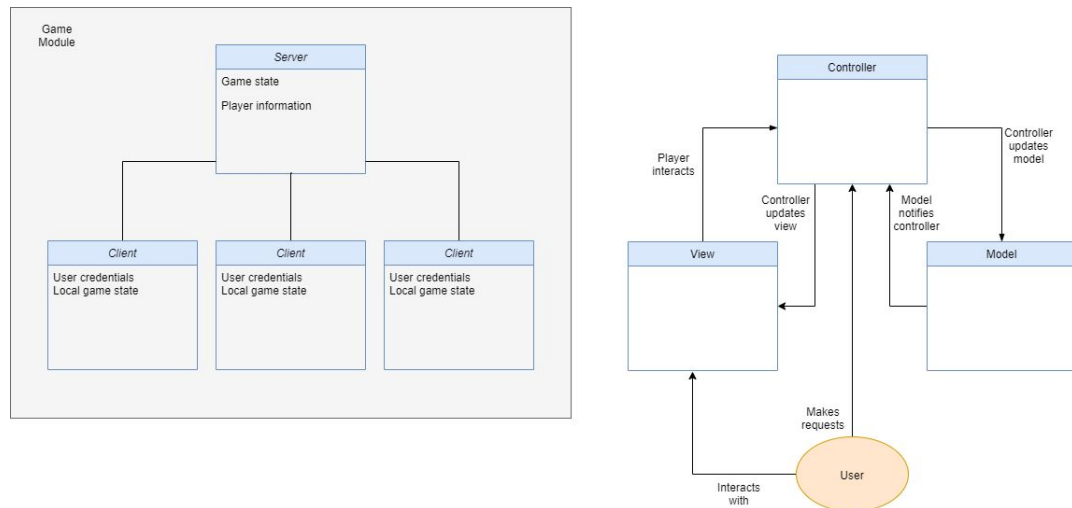
Core to *Pac-Rats*' design philosophy is to create a game that is accessible, fast, and fun. In regards to accessibility, this philosophy is extended to several different concerns. Firstly, the game is designed to run on as many devices as possible. This means supporting a range of devices, whether it be mobile phones, chromebooks, or powerful gaming computers. Secondly, *Pac-Rats* is designed to be a browser based game. We choose to implement it using web technologies because it makes it easy for players to get into a game -- without needing to install any external software.

### **23 Current System Design**

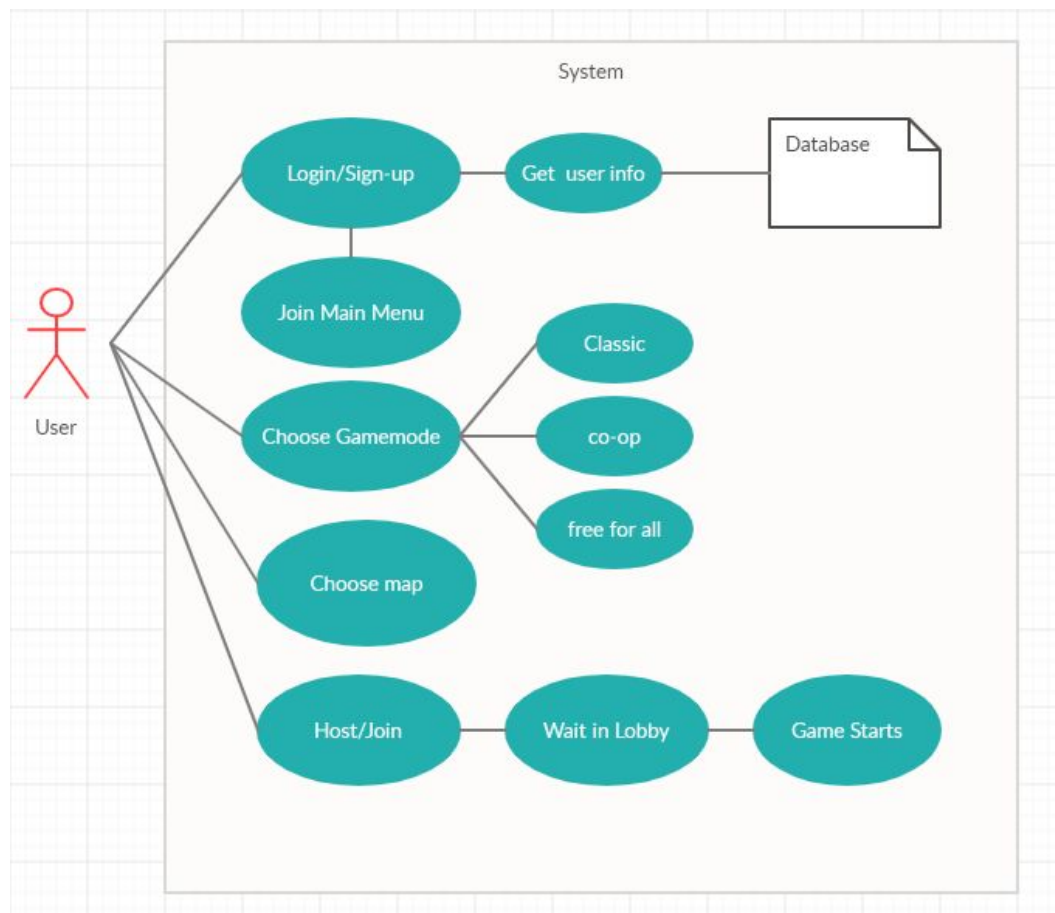
N/A -- There is no pre-existing system at the time of writing this document

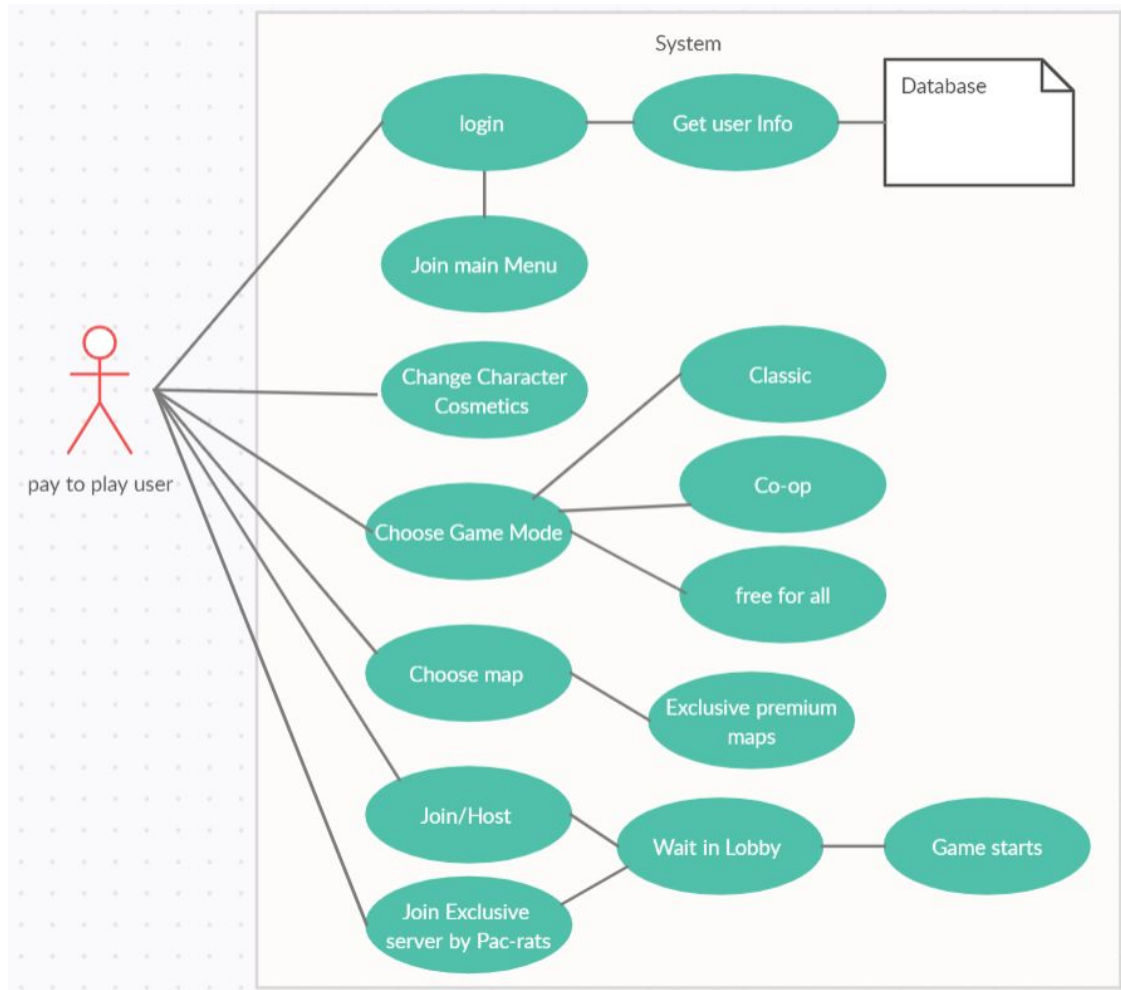
## 24 Proposed System Design

### 24a Initial System Analysis and Class Identification



### 24b Dynamic Modelling of Use-Cases





## 24c Proposed System Architecture

We will use the Model-View-Controller pattern for the broader site module, and then use the client server architecture for the game module. We choose the model view controller pattern for the site since it is a battle tested choice for building full-stack applications like the one the site utilizes. Using the MVC architecture will allow us to separate concerns and . For the game module, we once again opted for a commonly used pattern for the scenario we are going for.

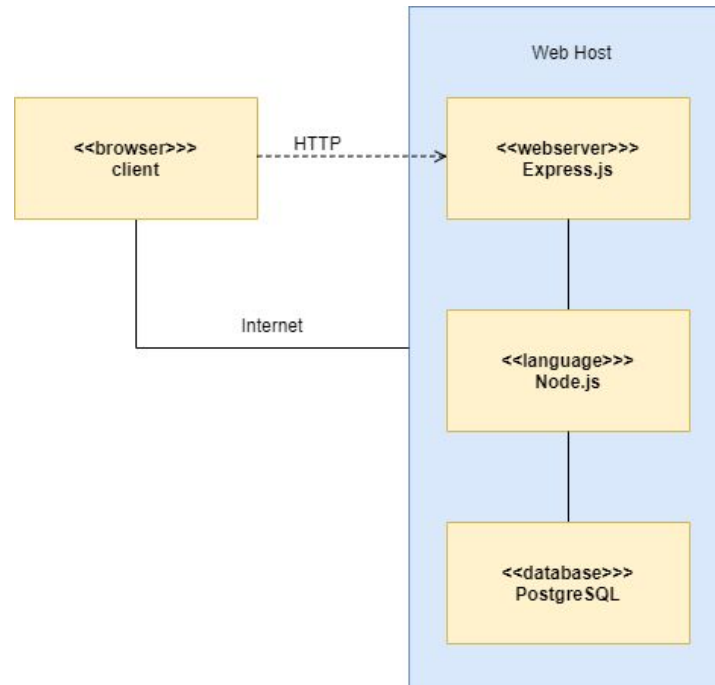
## 24d Initial Subsystem Decomposition

The application will be decomposed into primarily two subsystems. A “game” module, and a “site” module. The game module contains all of the site’s architecture and will implement the model view controller, as that is a standard pattern for full stack web applications. The “game” module will be a subset of the site module and contain all of the logic pertaining to the game itself. The “game” module will implement a client server architecture in order to host games. Furthermore, the game

module will utilize the observer pattern in order to synchronize views between players during gameplay.

## 25 Additional Design Considerations

### 25a Hardware / Software Mapping



### 25b Persistent Data Management

To store persistent data we will be using a relational database, specifically a PostgreSQL database. In the database, we will store user information such as usernames, passwords, emails, and so on. Furthermore, aspects of this database will be accessible and modifiable by game admins through an intuitive GUI. Other persistent information stored might include reports generated, and other possible user confidentials.

### 25c Access Control and Security

Access to premium features of the site should have their access restricted to users who meet the user class requirements necessary to access them. For example, server hosting features should only be accessible to P2W users. Furthermore, P2W users should not see any sort of ads within the game. Access to moderation tools should also be restricted to moderators and above.

In order to restrict access to certain features, we would need to implement a security proxy. We would do this by implementing a relatively standard “proxy” design pattern.

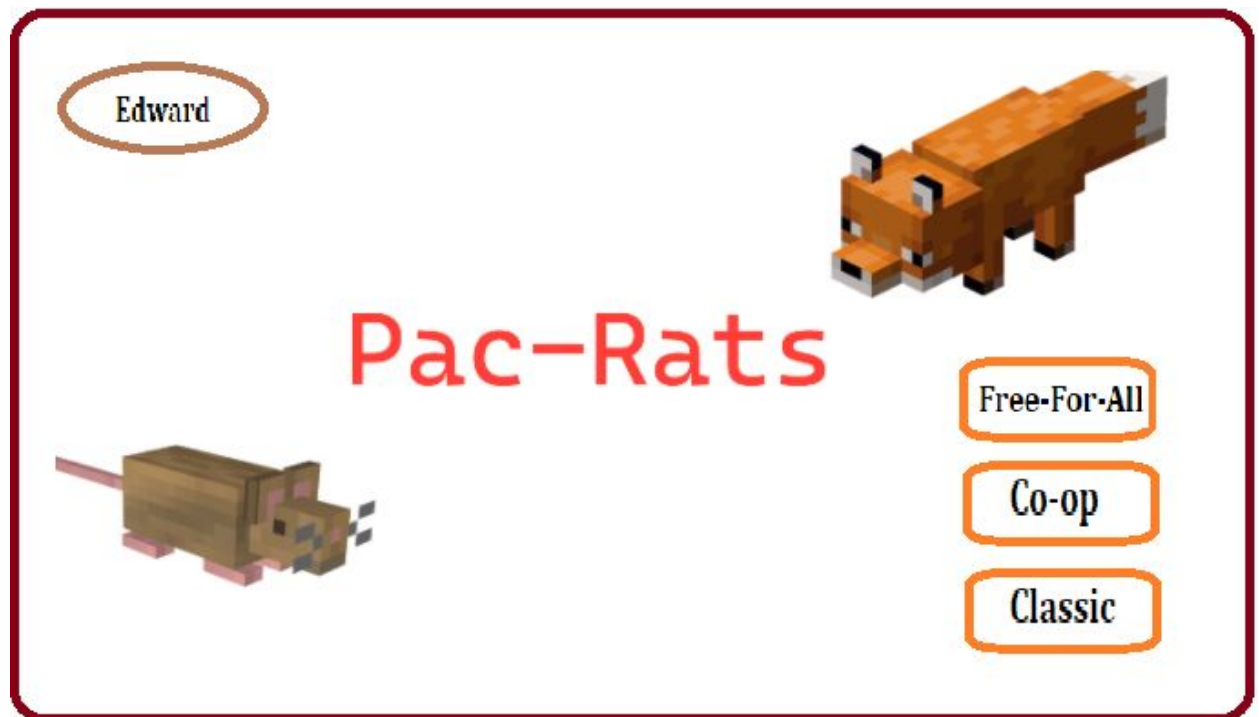
## 25d Global Software Control

We have found no global software control concerns with our system.

## 25e Boundary Conditions

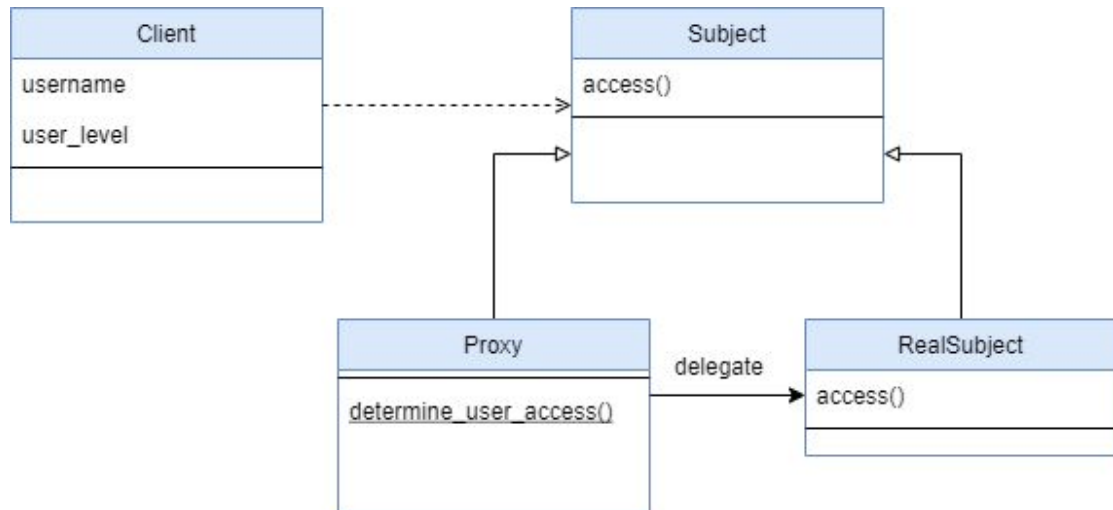
Boundary cases may occur when updating the user information of a user who has recently upgraded their account from a F2P account to a P2W one. For example, if there were a database lock error, it is possible that the user may have upgraded their account yet not be able to see those changes reflected in the view.

## 25f User Interface

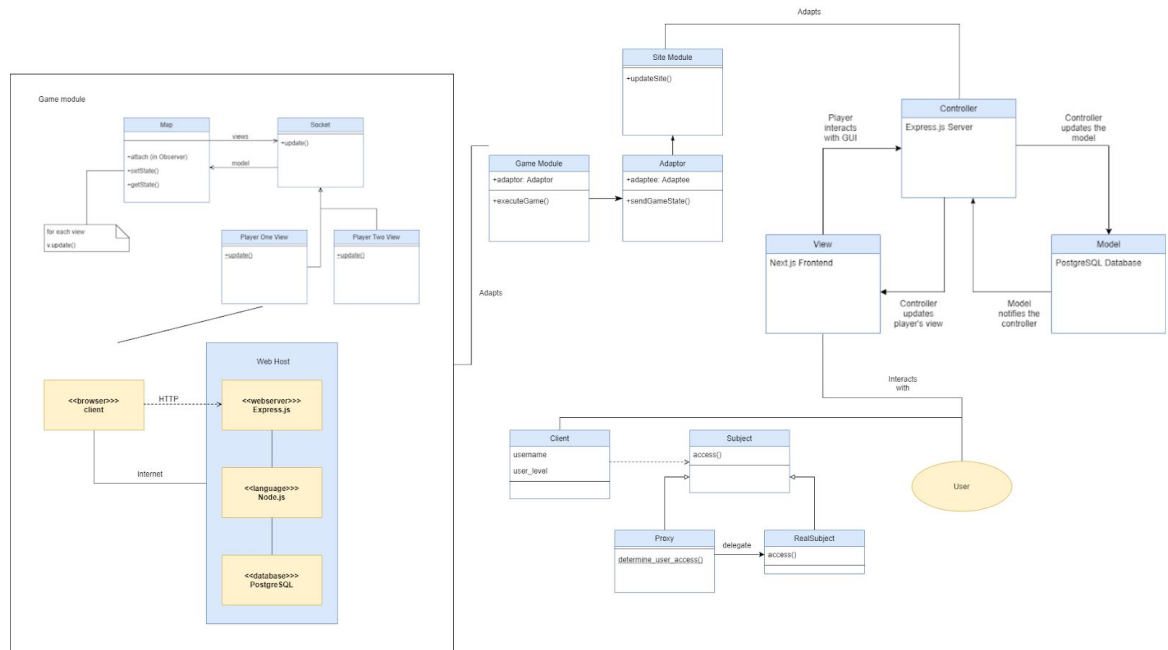


## 25g Application of Design Patterns

The proxy design pattern



## 26 Final System Design

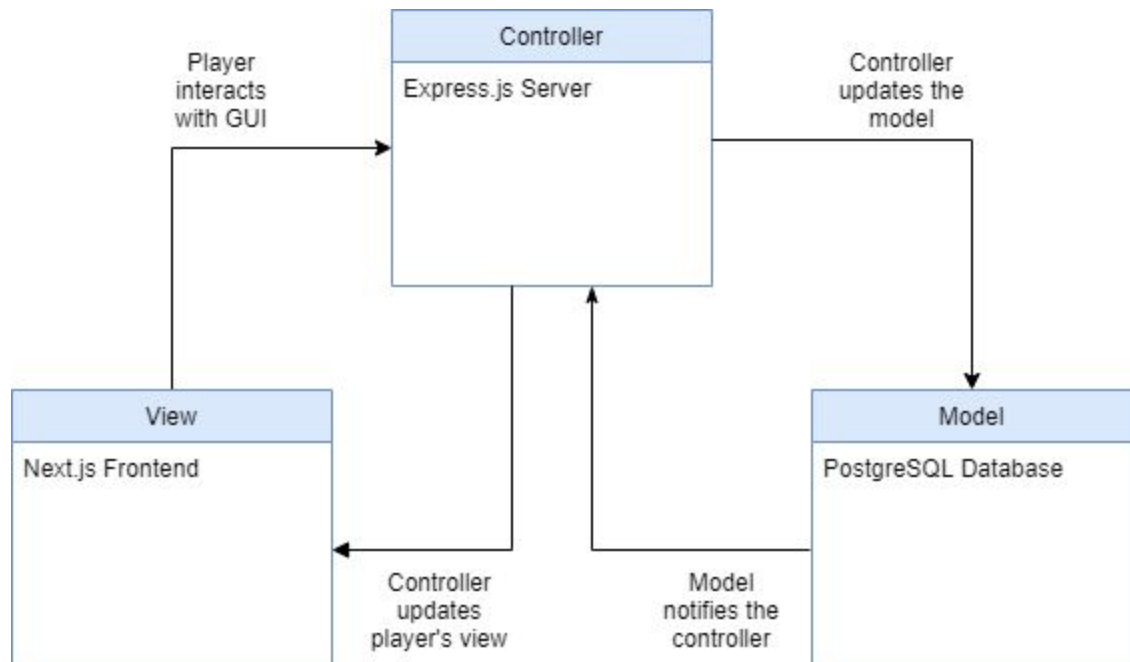


## 27 Object Design

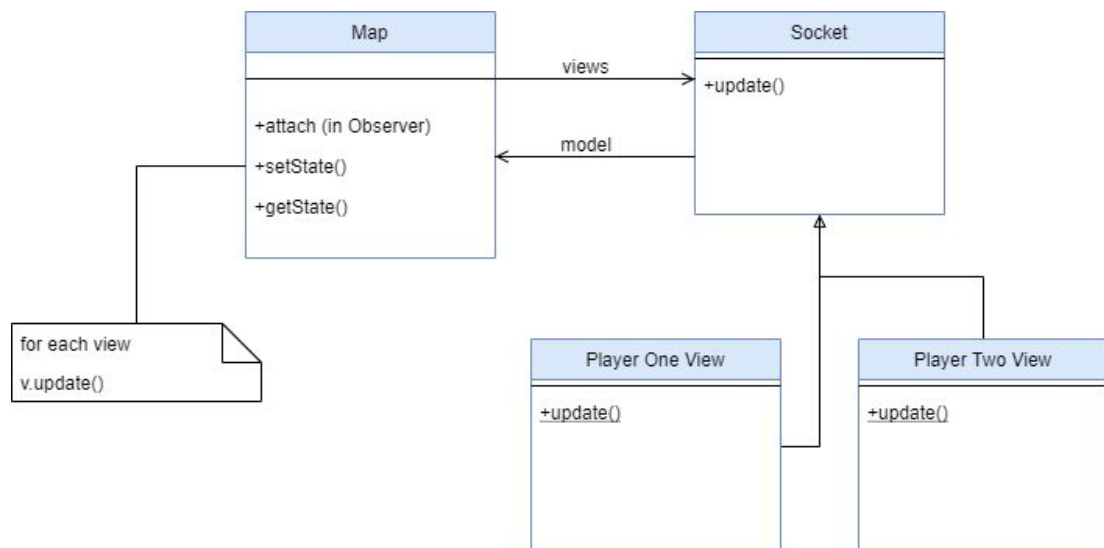
### 27a Packages

N/A -- Since our project will be written in JavaScript, we will not utilize a traditional package system like Java.

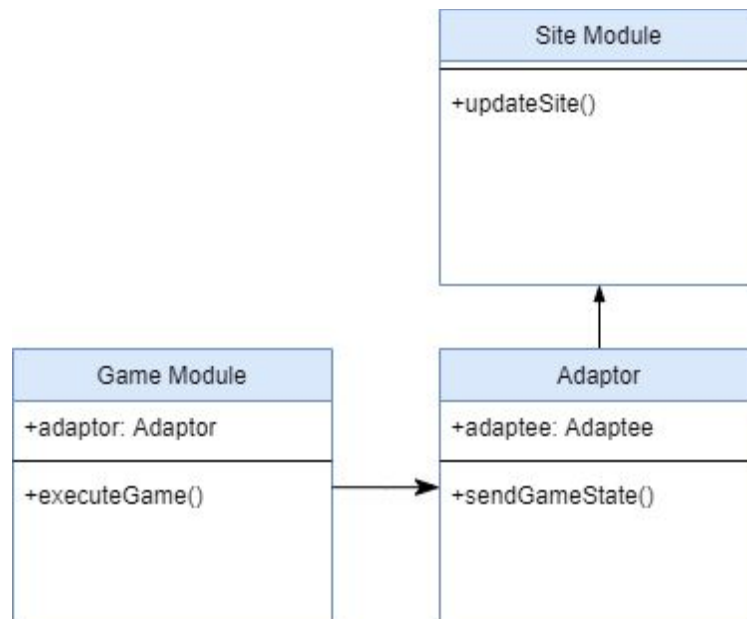
## 27b Subsystem I: Site Module



## 27c Subsystem II: Game Module



## 27d Game/Site Adapter



## IV Project Issues

### 28 Open Issues

Since this game is currently browser based there won't be an offline mode. Users with no no stable internet connection will not be able to play the game properly. Furthermore, the game currently does not have a proper plan for making the game properly cross platform -- ie, we do not have a control scheme for mobile yet. This game currently works for all the popular browsers (i.e microsoft edge, chrome, safari, firefox, etc) but it is still not released for mobile devices (ios, android).

### 29 Off-the-Shelf Solutions

In order to focus on developing the game, Pac-Rats would employ many open source, off the shelf components for specific subsystems within the game. For persistent data storage, we would use PostgreSQL. Furthermore, for the model view controller pattern, we would utilize Next.js for the "view" aspect of our design in order to focus on the UX/UI design rather than reinventing the wheel.

#### 29a Ready-Made Products

For the design of our avatars (rats and foxes) we will be using the animation software blender. Adobe XD could be used for UI design mockups.



## **29b Reusable Components**

Since this is a web based game, we will be using many Node.js libraries to aid in the development of the product such as...

1. Socket.io for real-time, bidirectional and event-based communication, specifically, in regards to the game module
2. Express.js will be utilized as our web framework in order to handle routing and so on.
3. Passport.js and Express-Session in order to securely save user session information

## **29c Products That Can Be Copied**

This game is a variation of *Pac-Man* but with a twist. The pixelated art style and the digital sounds can be legally used. The other aspect of the game such as the new game modes, character design, ingame powerups is original.

There are no particular assets we can reuse, but the simple style in general will allow us to avoid reinventing the wheel.

## **30 New Problems**

### **30a Effects on the Current Environment**

This system does not have any adverse effects on the environment. This product is designed so that every user can easily log in or log out wherever or whenever as long as there is stable internet connection. Since the product is a casual free to play game, there should be no resistance from lobbies, interest groups, unions, etc.

### **30b Effects on the Installed Systems**

This system does not have any adverse effects on other hardware or software since one of the design principles of the application is that it is to be lightweight and accessible for anyone. Furthermore, since the game runs in the browser, the application does not even require any software to be installed aside from a modern web browser. For these reasons, the requirements are minimal and should not have any adverse effect.

### **30c Potential User Problems**

We work to create a system where it thrives in an environment that does not create any adverse effects. However, since the game is monetized and the game is accessible to anyone, there is the possibility that a child may end up spending lots of money on the game without their parents permission. For this reason, it is important that a quality customer support system is established before launching the app.

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

As one of the core functional requirements in this game is having a stable internet, there might be environmental factors that could inhibit the proposed product. Such can include blackouts due to thunderstorms, electrical interferences, etc. All which can hinder from having a stable internet connection to the web server.

### **30e Follow-Up Problems**

It is possible that the game is marketed improperly and the game ends up being a commercial failure. This is outside the scope of the development team, however, it is important that the development team communicates with the advertising department so that it is crystal clear what the product has to offer.

## **31 Migration to the New Product**

“Not Applicable”

## **32 Risks**

One potential risk is not attracting the right audience and generating enough revenue to keep the game running on the server. The game requires a customer support and developer team for which all are very expensive. The server also has to be on everyday and that can prove very costly.

Making the game free can also prove fatal. We can charge users for a fixed price and be able to predict revenue using statistics but we might lose out on more than half of the audience. Due to the spike of many free indie games many users do not want to spend any money on games. Making it free and relying solely on ingame purchases/ads however might not generate enough revenue to power the system.

## **33 Costs**

- Following the scrum methodology the game will be ready for the general public in about a year and beta testing in half a year.
  - This includes the time for getting sound effects for the avatars
  - The animation for the avatars
  - the website UI
- Typically the cost from a low end indie game to a high end game ranges from (\$25,000-\$250,000)
- For and like all businesses, this is a gamble. There might be many lost opportunities
  - quitting a job to work on this wb game.
  - investing in other products due to the competitive nature of indie game development.

### **34 Waiting Room**

As stated previously this game is currently all browser based. This means that the game is not *fully* cross-platform. In order to improve on this, a future plan might be to build a dedicated native application for iOS and Android. This would provide a uniformly enjoyable for all users regardless of their platform of choice. In addition, we may choose to implement an “offline” mode that would allow you to access some features of the game without an internet connection.

### **35 Ideas for Solutions**

For developing a native app, we believe that utilizing a web to native framework like React native would be ideal. This would allow the developers to reuse a majority of the codebase without the need to rewrite everything in three separate languages. Furthermore, since we are already using react, this transition would be even easier. Although there might be some compromises when compared with a fully native app, we believe that the gains outweigh the costs.

### **36 Project Retrospective**

In retrospect, the guiding principles we used for this project were important to its development and aided in keeping the guiding vision of the project on point. Specifically, the aims of accessibility which we laid out in the initial design documents helped to guide our application design in developing ideas for the project and for influencing the final application design.

We believe that the project can be implemented properly using agile methodologies, specifically SCRUM. By having proper sprints and sprint review, a developing team could improve productivity and be able to upload a functional but incomplete product for the client to use.

## V Glossary

**F2P:** “Free to play”, free to play account. All players start out this user class when they start the game. This type of account has access to all fundamental features of the site, but does not have access to premium features.

**P2W:** “Pay to win”, premium account. Players who have spent money upgrading their account will have access to premium features of the site that are unavailable to F2P users.

**Nextjs:** An open source front end web development framework for react

**Nodejs:** An event driven single threaded runtime environment that executes JavaScript code outside the web browser. Allows us to use Javascript on both the front and backend.

## 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

Design	61, 63
Requirements	35, 51, 58
Test	64, 65