National Parks Builder

A gaming application based on National Parks

Prepared by
Maaz Ahmed
Bryan Gutierrez
Ammar Idrees
Zaid Patel
for use in CS 440
at the
University of Illinois Chicago

Spring 2020

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

National Parks Builder is a gaming application that allows users to build and manage a national park from scratch. Users first get an option of picking an undeveloped land from a current National Park's location. So for example, if the user wanted to rebuild Sequoia National Park, the location would be California. Then the user would be in charge of rebuilding the park and deciding where features of the park would go. This can include features such as benches, campsites, trees, a welcome center, parking lot, etc. Once the user has built their initial design, then they will be in charge of the management aspects of running a National Park.

The management portion would contain duties such as fixing issues within the park, setting ticket prices, business decisions within the park, keeping employee morale high, etc. If you don't meet certain revenue and visitor population goals then there is a chance your park could get shut down. This application would allow you to play and compete with your friends to see who can build and manage the better National Park. Also this game will have microtransactions and the funds from that would be donated to the National Parks Foundation.

Users can put tons of hours into this application because every time they log on there would be new tasks they would have to deal with that pertain to the National Park. There would also be a leaderboard within the application that would show which users have gained the most visitors and made the most revenue for their park. This entices competition within users and can help ensure they are enjoying the game.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Our application is most useful for the National Parks foundation especially those parks with lower attendance. The idea is that we feel National Parks are becoming less relevant in today's society. This is due to the fact that the usage of electronics and gaming is increasing significantly. So we felt that a gaming application like this would give exposure to National Parks and allow the user to appreciate its beauty and the effort that goes into managing one. We hope that this will encourage users to go experience these parks in person.

At the same time, we want this application to be equally entertaining as it is informative. We want to make sure that users get excited when logging into this application everyday. Furthermore, we want to convey the hidden message that users should be respectful to the parks and their management. We want them to become aware of the fact of how much time,effort and manpower goes into running a National Park. We have had experiences where we have seen people mess the trees and plants in National Parks because they think it'll just grow back easily.

2b Goals of the Project

We want to make sure that today's society is aware of all the National Parks that exist and the beauty within them.The goal of this project is to increase the number of people who visit National Parks. If the number of visitors increases then the hope is profit should increase as well. We hope to accomplish all this while making sure the user is being entertained as well.

2c Measurement

The simplest way to measure this is to track visitors monthly for a year before the application is released. Once the application is released, then the stats would be taken again monthly to see if visitors numbers increased monthly. To make sure that this application is the reason why, we could implement an idea that when visitors purchase their tickets they fill out a questionnaire. One of the questions would be Why have you decided to visit this National Park? Then we can have a drop down menu with options and one of the options would be because of the National Parks Builder application.

Furthemore, we can see if profit increased due to more people visiting the parks. We can count the revenue that came from people who bought tickets and put National Parks builders as the reason why. It would be difficult to track in park purchases such as the gift shop but it would be possible to track the sales of microtransactions. This can be tracked in monthly revenue. We would have to take a deeper dive into the revenue streams and keep track of the exact amount coming from microtransactions. We don't expect revenue to increase by a large margin from microtransactions, but something is better than nothing.

Additionally, we would want to try to see if management/employee morale increased since the user of National Parks Builder is becoming their own manager. We hope that they treat National Park managers with more respect and better manners when they visit themselves. This can be measured by an employee survey asking how their morale was this month. Also to see if less incidents/complaints are happening in National Parks due to this application.

3 The Scope of the Work

The business involved with this product is the National Park Service. Our product is a small part of this business that involves creating awareness for the struggles and every day tasks of taking care of a National park while also making it entertaining to play.

3a The Current Situation

This product does not necessarily replace or change how the business will be run directly. The National Park Builder is another form of awareness that is more interesting than the current options. Currently, there are only traditional ways of showing difficulties and requesting aid. There are many articles that list the many problems that National Parks must face that are mostly due to a lack of financial aid. Another form of awareness of the National Park Service is their website, it exhibits the beauties in National Parks and has an option for donations to improve the parks.

3b The Context of the Work

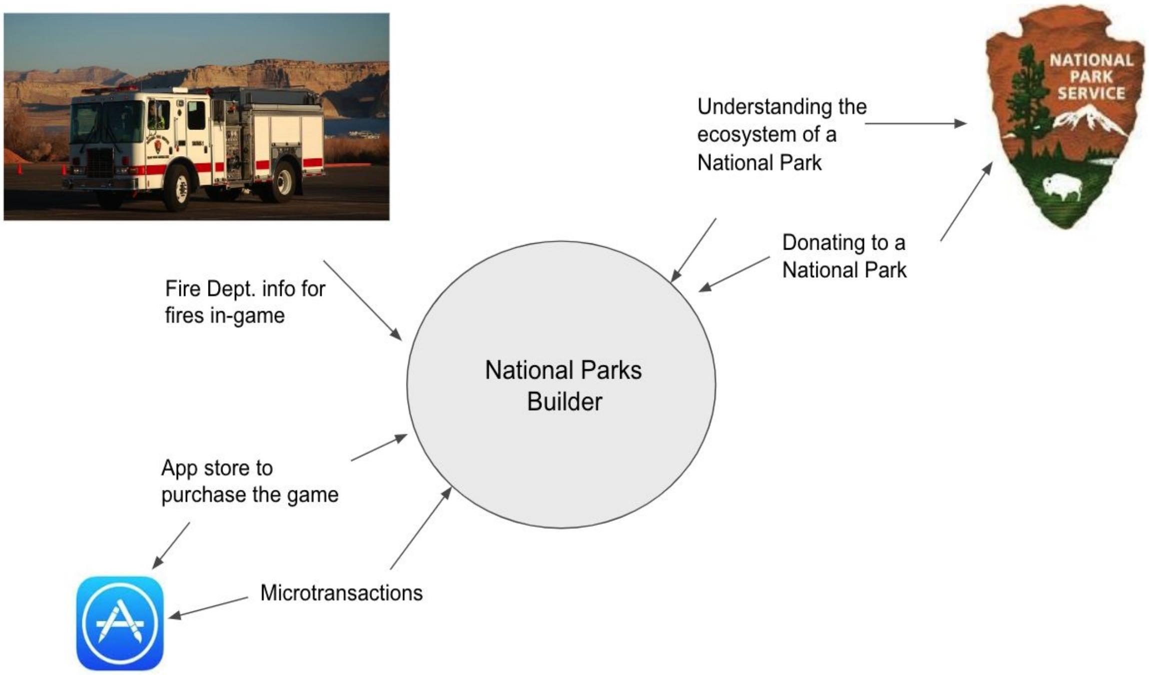
Understanding the ecosystem of a National Park
NATIONAL PARK SERVICE
Donating to a National Park
Fire Dept. info for fires in-game
National Parks Builder
App store to purchase the game
Microtransactions

Figure 1 - Context of Work Diagram

The National Park Builder will primarily take information from the National Park Service website to interpret the data and create a game out of. Phone app stores will be used to manage the purchase of the game and microtransactions. Natural Disasters such as fires will be in the game so it is important to make it realistic.

3c Work Partitioning

Table 1 - Work Partitioning

Event Name	Input and Output	Summary
User adds natural Units	Unit placed in terrain(Input), unit is displayed on map(out)	The user can add a new unit(, tree,animals,etc) to the terrain
Fire	Not enough rainfall, random, or possibly	There are a few ways for a fire to start within the park, the user

	accident by employee (in) Causes fire(out)	must stop the fire from spreading
Microtransactions	User purchases in-game item(In), Item is added to inventory(out)	When an item is purchased, the user must be able to see and use it
User adds Building Unit	Building unit placed on terrain(in), building is in place(out)	The user can decide between different buildings to add depending on what they need.
Cleaning area	Clicking dirty sections of land (in), land will clean up(out)	The user's map/land can get messy with visits from tourists so the user can clean it up
Donation	User donates money to NPS(In), App sends money to NPS	The game will come with an option to donate money to the National Park Service to be used for renovations/repairs. They can pick which park to donate to.

3d Competing Products

There are no other products similar to this one that relate to the same topic of national parks. There are similar building games such as Clash of Clans or The Sims but they do not involve national parks.

4 The Scope of the Product

The business involved with this product is national parks but the main focus of the product is to create more ways to raise awareness of the struggles that national park workers deal with. The few forms of awareness are mostly articles/readings but this product creates a way to entertain users while also teaching them.

4a Scenario Diagram(s)

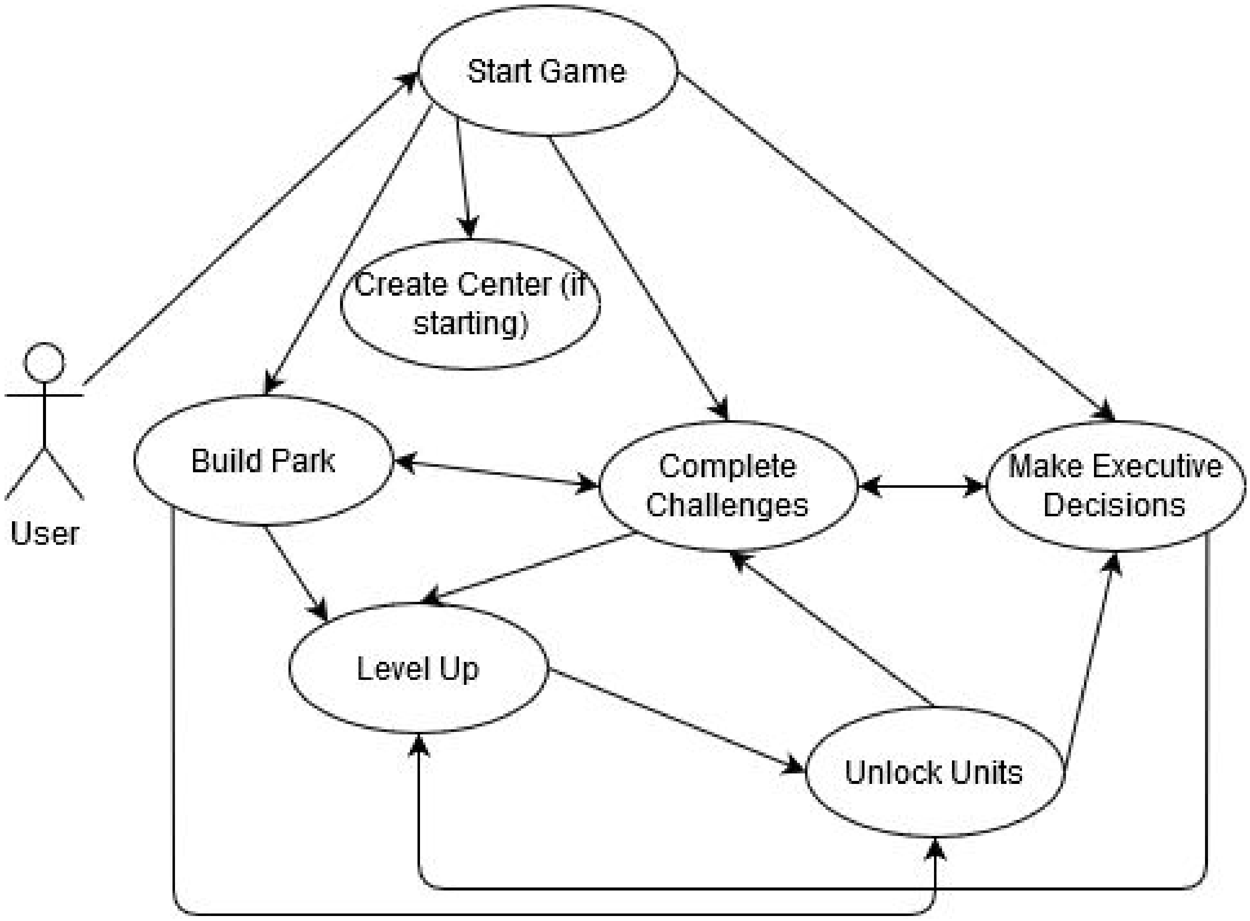

Figure 2 - Scenario Diagram

4b Product Scenario List

1.User adds natural units
2.User adds Building Unit
3.Fire
4.Cleaning area
5.Microtransactions

4c Individual Product Scenarios

A user will be able to select between “natural” units such as trees, bushes, a variety of animals, and many more units. The units will be purchased using currency earned from achievements and other forms of production. Similar to the natural units, a user will be to add Building units to their land that each serve their own purpose. When the game initially starts up, the user will need to create a “Town Hall” for the center of their national park. Fires can occur for many reasons that do not necessarily involve the user to be at fault. Fires can occur for lack of rain(random) or for putting units too closely together. Cleaning an area is a task that the user will have in order to maintain a clean park for guests/tourists. The user will simply have to click the dirty areas and select clean but this will take time and resources. Microtransactions will be used by the user in order to purchase items that they can use in-game to better their national park.

5 Stakeholders

The stakeholders that are interested in this product would be the private landowners who own a major portion of the park's land. Furthermore, in an indirect sense activists that are in support of the success and future of national parks would also have an interest in the product being developed.

5a The Client

The client in this case would be the land owners, and different investors who are interested in owning a stake within the parks. Alternatively, some investors who are avid in terms of keeping the longevity and sustainability of the parks could invest in our product. For those interested in the sustainability of the parks, they would want to spread awareness across society through the game trying to educate a mass amount of people about the efforts and financial needs that go into merely running the park.

5b The Customer

The customers will be those looking to gain experience on national parks, or whoever is interested in downloading it within the app store. There could possibly be a deluxe edition for those actually visiting National Parks, via a QR code given at the beginning entrance to customize the experience within the app during their visit. The app would be given on mobile devices for customers, and would be able to choose the maps between different national parks.

5c Hands-On Users of the Product

The hand on users for this product could be used by non profit organizations or as an educational tool within schools. Non profit organizations could use this product as a way to get people informed about the app, and could be a method to bring in donations for the parks to use. The directors of the parks could also be an effective user for this product, and they could possibly have enhanced features regarding the product, being able to view which playing customers are at their current parks, and if possible be able to alter or bring on some additions for their tour throughout the park.

5d Maintenance Users and Service Technicians

The app itself would be free for users looking to download it within the app store. Overall, the National Park members themselves who are enrolled as premium members may have to pay a beginning fee to have the app linked to their parks, and possibly monthly recurring payments to fix patches and have updates within the app. The app itself could possibly have in game purchases for the customers, depending on how they want to customize the experience, and monthly updates would be free of charge to them.

5e Other Stakeholders

Technology Experts: These members will be in charge of offering the best service for the app, adding the best services for the app. They could possibly translate the app into a CRM, trying to normalize the relationship between customer and the application. This team will also be in charge of gathering all the fixes needed monthly, from bug patches, to updates, to marketing. They are basically the focal point that gathers all the data from all the other teams and gets it into production.

Software Developers/Quality Testers: This team will be in charge of the weekly springs enabling the back end, database, gameplay, and overall mechanics of the whole application. All the heavy load programming is done here, and so obviously they have a big stake on whether the application is successful or not. Overall, development is constantly needed with applications such as these, so this team will be having to constantly update the app, think of new ideas to implement, and be in charge of providing the best experience for the customers. Furthermore, the quality testers are in charge of constantly making improvements and completing unit tests possible utilizing maven.

Strategic Analysts: This team will be implementing critical solutions through reviews and large amounts of testing. Furthermore, they will be trying to begin to increase profitability and growth for the application as a whole. Overall, this team will be trying to gain the analytics of the customers themselves, and have a large impact on the stakes if they want the app to be successful. They will have to analyze the current market for similar applications, and try and optimize the sales of the game.

5f User Participation

Users will have a large influence over the product, being able to share their thoughts within the reviews, or have a suggestion forum on the applications landing page. The customers basically have the ability to ask for new features, or different aspects of the app that they want changed and so forth. Overall, customers' feedback will be very important as it will also affect National Parks, thus by listening to their voice it will only strengthen the relationship between them and the parks themselves. Furthermore, after users are able to visit the Parks they will be able to meet with the office to share their experience, in which their recommendations will take higher priority.

5g Priorities Assigned to Users

The users should simply have access to the application, basically just able to access the game, in a programming realm just have the read option. The management of the specific National Parks that are related to the application could possibly also have a write option for certain aspects of the app, having permission to be able to change certain types of the product. The customers would also have the ability to voice their concerns to developers, who have access to write and can make subtle changes.

6 Mandated Constraints

6a Solution Constraints

Some solution constraints regarding this project is that the application follows the click and drag approach as that will be the majority of the functionality for the free version. Overall, there will be a pane regarding the application, imitating a full map of the National Park. There will be event handlers for each area of the map. For example, a player for example has a budget of a few million, and must allocate it in a way of keeping the park at a substantial level. There will be many conflicts within the game such as pollution in the water reservoirs, or high pollution levels affecting wildlife. The players will have to allocate resources in these areas to keep the park running. The game will have a real time health bar, possibly within javascript that will slowly decrease if the issues within the park are not being solved. The developers will need to have to link all the different aspects within the game such as wildlife, forestry, water reservoirs, tourists, income all within the health bar. If the player decides to visit a National Park with the app in hand, the developers will need to implement a version of currency that can be collected throughout the application with the camera. The user will give access to their camera to the app, and will walk around the Park until they find a token which could be converted to currency within the game. This will require some use of computer vision and a form of AI such as tensor flow.

6b Implementation Environment of the Current System

The product will need to be runned on mobile devices, and will need to follow a particular design pattern overall. The game will need to purchase its own server, to have all the different players be able to run on it. The game should be able to run with library graphics, and could be done optimizing with the application fusion. Furthermore, some data encryption will need to be implemented within the app, to keep a secure platform and save all information. There will need to be a form of SPIM which could provide an emulator if needed.

6c Partner or Collaborative Applications

Some collaborative applications that could be utilized would be obviously setting up some form of payment reaching out to a form of CTO commerce to obtain rights. Furthermore, security and hashing important information within the application would be important, so a possible partnership with a security solutions team would be essential in keeping the privacy of the customers who have given their information to the app.

6d Off-the-Shelf Software

There are quite a few off shelf softwares that would be needed to complete the application. Certain outside computer vision softwares such as YOLO could be utilized for the feature utilizing the camera for then the user decides to engage with the camera. Furthermore, the application needs to be able to run for both IOS and Android so a few options between Xamarin or Flutter could be utilized to ensure the application is properly published.

6e Anticipated Workplace Environment

The product will mainly be used on mobile devices, either indoor or outdoor depending on the user. The application should not take up too much storage, as it might scare customers away. Furthermore, depending on if the customer wants to try the premium version which is able to correspond with real National Parks, a live location will be needed, with possible access to the camera of the device.

6f Schedule Constraints

There are not many schedule constraints regarding the development and launch of the product. However, for the premium edition of the application, where users will have a personalized experience based on the National Park that they visit, could vary depending on the availability of the park. If some parks are closed seasonally, or have certain times when they are not open to admission, it could somewhat complicate that feature of the application.

6g Budget Constraints

There may be some budget constraints for this project, especially if we would want to implement a feature that would entail combining it with actually visiting National Parks. The National Parks are usually federally funded, and it is highly unlikely they have the means to be willing to invest in the application in itself. Furthermore, some non profit organizations may be apart, if they are able to conform with the overall interior plan for why the app is being created. Furthermore, being able to hire all the needed members, such as marketing, analysts, developers would be a process in creating such a large team.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Park - The area that the user will be building throughout the game.

Center - The main hub of the park. Will be the main source of organization for the entire map. Will be required to build at the start of the game.

Land - Unused space around the national park.

Unit - Some sort of object, building, or piece of nature that the user can place on the land.

7b UML and Other Notation Used in This Document

Any of the UMLs or notations that will be made in this project will be made based on the descriptions and diagrams throughout the project.

7c Data Dictionary for Any Included Models

User - Will require a server to work on the national park. Will need to be able to view currency, employee morale, and average visits.

Currency - Integer that will be a minimum of 0 (can't be negative).

Employee morale - Integer that will be displayed as some graphic.

Average visits - Will contain integers for age of park and total visits. Will calculate and display a double.

8 Relevant Facts and Assumptions

8a Facts

We want to compile facts and data about national parks so that they can be displayed to the user. They can be used as "fun facts" to give awareness. Furthermore, we can utilize data from actual national parks as a way to create challenges. Doing so would allow the challenges to mimic real life issues that national parks may face.

8b Assumptions

We would assume the user is able to perform basic tasks such as selecting certain options or navigate the menu and game. We would also assume the user has the necessary requirements to run the game.

II Requirements

9 Product Use Cases

9a Use Case Diagrams

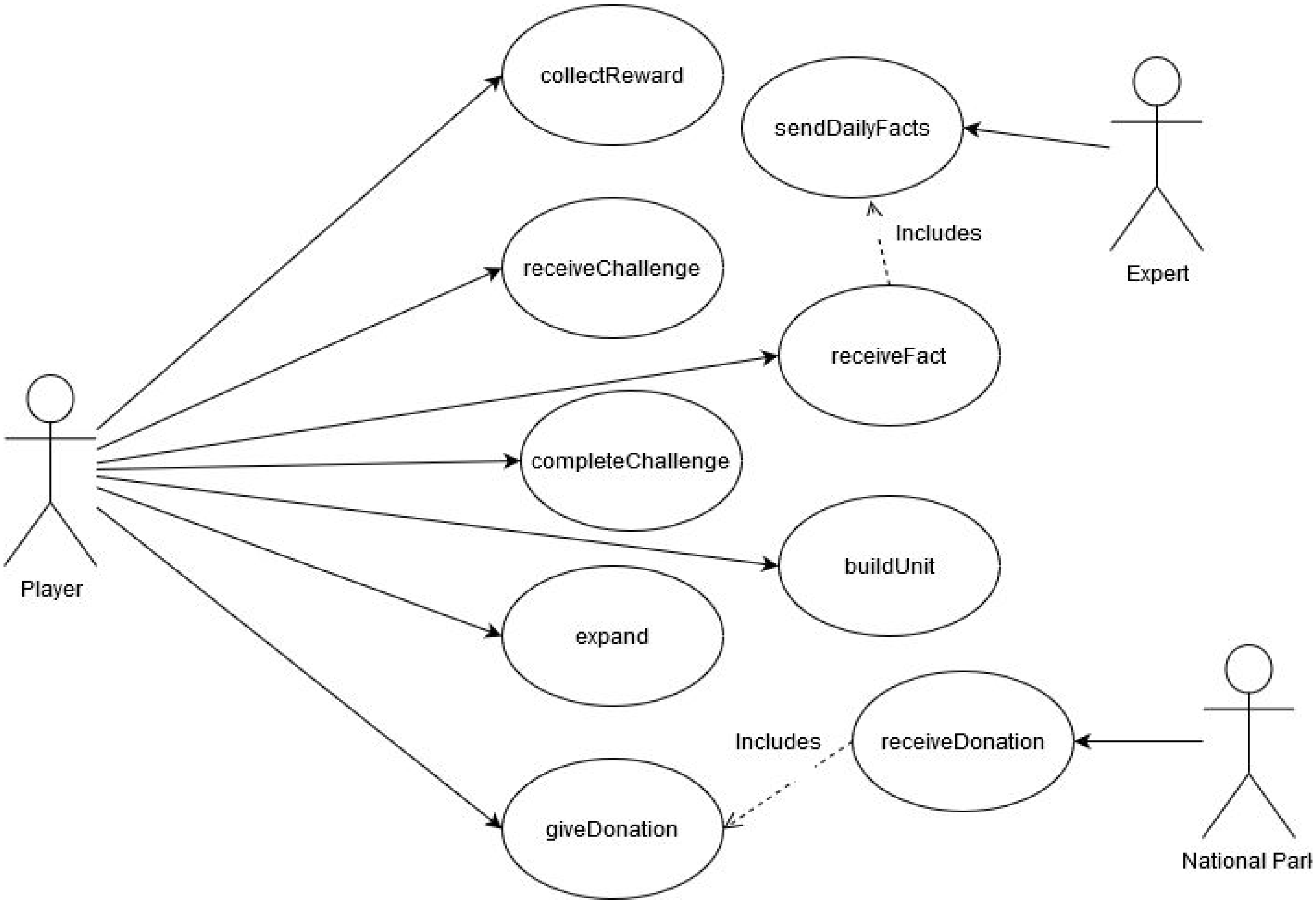

Figure 3 - Use Case Diagram

9b Product Use Case List

Table 2 - Product Use Case

ID	Name	Description
1	collectReward	Users can collect rewards on login.
2	receiveChallenge	Player receives challenges to complete to progress through the game.
3	completeChallenge	Game notifies user of progress during challenge
4	expand	Users can expand land to be protected by national reserve.
5	giveDonation	Users can donate to organizations dedicated to national parks.
6	sendDailyFact	Experts can send facts for users to be informed on national parks.
7	receiveFact	Players can receive facts on national parks.
8	buildUnit	Players can create essential buildings or create more nature.
9	receiveDonation	An organization can receive donations from users.

9c Individual Product Use Cases

Use case ID: 1 Name: collectReward

pre-conditions: Players must be logged into the game.

post-conditions: Players must have a reward given to them.

Initiated by: The Player logging in.

Triggering Event: Selecting the reward that is given.

Additional Actors: N/A

Sequence of Events:

1. Player launches the game and logs in.
2. The game sends messages to users to collect their daily reward.
3. Player selects the reward displayed.
4. The game moves the reward to the user's inventory.

Alternatives: N/A

Exceptions: N/A

Use case ID: 2 Name: receiveChallenge

pre-conditions: Player must be logged into the game.

post-conditions: A challenge will start for the player to complete.

Initiated by: The game sending the player a challenge.

Triggering Event: The player accepting the challenge.

Additional Actors: N/A

Sequence of Events:

1. Game sends the player a challenge.
   2. Game will display a message to the user about the challenge and will ask the player if they want to complete this challenge.
3. Player selects confirm option
   4. Game will generate challenges and it will start.

Alternatives: Users can deny the challenge.

Exceptions: N/A

Use case ID: 3 Name: completeChallenge

pre-conditions: Player must have a challenge already received.

post-conditions: Players will have their progress updated.

Initiated by: The game starting the challenge.

Triggering Event: The player accepting the challenge when displayed by game.

Additional Actors: N/A

Sequence of Events:

1. The game starts the challenge.
2. The player will complete the tasks as the game requires.
    3. The game will update the progress of the challenge.
4. Player completes challenge
    5. Game rewards players and progress is updated.

Alternatives: Players can fail challenges.

Exceptions: N/A

Use case ID: 4 Name: expand

pre-conditions: Land must be available to the player.

post-conditions: Size of park is updated.

Initiated by: Player requesting game to expand land.

Triggering Event: Player selecting area of land they wish to purchase.

Additional Actors: N/A

Sequence of Events:

1. Players select available land outside of national parks.
    2. Game sends a user message asking if they would like to purchase the land.
3. Player confirms the purchase.
    4. The game processes purchase.
    5. Game makes land available for player to build on

Alternatives: Player can change their mind on donation

Exceptions: Game will create errors if the player does not have enough currency and transaction will not go through.

Use case ID: 5 Name: giveDonation

pre-conditions: Players must have access to the donation screen.

post-conditions: Player is able to process gifts.

Initiated by: The Player using the donation screen.

Triggering Event: Player sending their donation

Additional Actors: National Parks or non-profit organization that works with reserves.

Sequence of Events:

1. Player selects option to make a donation
    2. The game displays different charities to the user.
3. Player selects selects a charity/fund
    4. Game displays the fund to the user and all the information related to it.
    5. Game prompts user to make a donation
6. Player enters amount they wish to donate
7. Player submits a donation.
    8. Game processes donations and sends it to the respective fund.

Alternatives: Players can deny donation if they change their mind.

Exceptions: N/A

Use case ID: 6 Name: sendDailyFact

pre-conditions: Expert must be able to access the feature to send facts.

post-conditions: Players should receive a daily fact to their client.

Initiated by: Expert accessing send facts feature.

Triggering Event: Expert enters fact for the day.

Additional Actors: Players that would receive these facts.

Sequence of Events:

1. Experts enter the fact they would like to send.
    2. Software confirms if they would like to send this fact.
3. Expert confirms.
    4. Software verifies the fact that is sent for accuracy.
    5. Game updates players that log into the game with the fact.

Alternatives: Experts can deny to send facts.

Exceptions: If fact is not true then the expert is notified by system.

Use case ID: 7 Name: receiveFact

pre-conditions: Player must be able to login to the game.

post-conditions: Player is informed.

Initiated by: The player logging in.

Triggering Event: Server is notified of player’s first login of the day.

Additional Actors: Expert who will send the fact.

Sequence of Events:

1. Player logs into the game.
    2. Game sends a message to the user a fact everyday in regards to national parks.

Alternatives: N/A

Exceptions: N/A

Use case ID: 8 Name: buildUnit

pre-conditions: Player must be able to access store to purchase units

post-conditions: Player’s park is updated.

Initiated by: Player in store.

Triggering Event: Player purchasing unit.

Additional Actors: N/A

Sequence of Events:

1. Player requests to open the store.
    2. Game displays the store to the user.
3. User selects the unit they wish to purchase.
    4. Game prompts the user if they would like to purchase the unit.
5. Player confirms the purchase.
    6. Game displays the unit in the player's park.

Alternatives: Players are given the option to deny purchase.

Exceptions: If not enough currency is available, the game will notify user and transaction won't go through.

Use case ID: 9 Name: receiveDonation

pre-conditions: Charity or fund must be able to receive donations

post-conditions: Park receives donation.

Initiated by: Player sending donation

Triggering Event: System notifies fund that a donation was sent.

Additional Actors: Player sends donation.

Sequence of Events:

1. Game sends notification to charity that a donation has been received.

2. Donation collects donation

Alternatives: N/A

Exceptions: N/A

10 Functional Requirements

FUNC - 1

Description: The game should consistently provide rewards for the players as they play.

Rationale: This will keep the player motivated to continue playing creating longevity.

Fit Criterion: The game should keep track of the player's progress to give appropriate rewards.

Acceptance Tests: FUNC - 1

FUNC - 2

Description: The game should display the next challenge a player must complete after they complete their current challenge.

Rationale: This will make sure a player will always have something to do.

Fit Criterion: The game must be able to generate a new challenge.

Acceptance Tests: FUNC - 2

FUNC - 3

Description: The player should be able to expand their park as they progress.

Rationale: The player will be able to further customize their park and can unlock new challenges, rewards, etc.

Fit Criterion: There should be a certain amount of land available for the user to expand upon.

Acceptance Tests: FUNC - 3

FUNC - 4

Description: The game should allow a player to donate to a charitable cause related to helping parks and their resources.

Rationale: This will give players the opportunity to help parks and give a sense of purpose to the player on why the game is important.

Fit Criterion: The game has the funds stored.

Acceptance Tests: FUNC - 4

11 Data Requirements

<u>**DATA - 1**</u>

Description: Game should contain a user's donation history.

Rationale: This should allow users to have receipts of their donations and can be good for the user themself.

Fit Criterion: The requirement is met if the transaction history is saved.

Acceptance Tests: DATA - 1

<u>**DATA - 2**</u>

Description: Game should contain resources to auto-verify facts

Rationale: It would be a bad idea to have an expert give misleading facts to uninformed players.

Fit Criterion: The requirement is met if the software can distinguish between a reliable and unreliable fact.

Acceptance Tests: DATA - 2

<u>**DATA - 3**</u>

Description: Game should be able to know the amount of currency a player possesses.

Rationale: A player should be able to know this data from the game so that he can buy units, items, etc.

Fit Criterion: If it can accurately count the currency overtime this requirement should be met.

Acceptance Tests: DATA - 3

12 Performance Requirements

12a Speed and Latency Requirements

Table 3 - Speed and Latency Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
SPLA-1	The application must quickly load the game environment and menu options.	A user shouldn’t have to wait long for a mobile application to open up and for them to be playing. A lot of users open mobile applications for a couple minutes when they have downtime such as waiting for a bus or on commute etc. By having it load quickly, their playtime is being maximized to the fullest extent,	This can be tested by having a specific amount of seconds it should take for the application to load and display the game menu. Unit tests could also be used here to test this task.	SPLA - 1
SPLA - 2	System should properly update the user's moves within 3 seconds.	The game should be played at a fast pace. There are a lot of things you can do in this application. We don’t want users to progress to slow down because of the system being slow.	This can be accomplished by having extensive unit tests that make sure the system properly responds to the user’s input within 3 seconds constantly.	SPLA-2

12b Precision or Accuracy Requirements

Table 4 - Precision or Accuracy Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
PRAC - 1	All in game words should be easily readable and in a clear font	Our goal of this application is to be entertained. We don't want users to struggle reading the words and font of the game. This game should be able to be played by users from various countries and various age groups. We will do our best to meet everyone's standard and we feel making sure everyone can clearly read the text is a good place to start.	We will implement a language option where you can choose your specific language and that's how the words will appear. Furthermore, we will give font size adjustment settings so that way users can customize based on their own preference.	PRAC - 1
PRAC - 2	The timezone for the game should be synchronized with a time server.	The reason for this is that way upgrades that are done happen in an accurate measure no matter what time zone you are in. This ensures it's fair for every user and you don't get an advantage based on time zone.	This can be tested by deployment and seeing if using the same account in a different timezone has an impact on how fast the user's upgrades are ready.	PRAC - 2

12c Capacity Requirements

Table 5 - Capacity Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
CAPA-1	The backend server should be able to host a huge amount of devices all connecting concurrently from anywhere in the world.	National Parks Builder is a mobile application and should be easily accessible by a user at any time of the day. This is why it's crucial that the backend servers are stable and secure to host a large number of devices.	Extensive testing will have to be done on the backend. Stress tests will need to be done to make sure multiple users can access the app. Also we can do a soft launch if needed and gradually build up so servers aren't overloaded right away.	CAPA - 1

13 Dependability Requirements

13a Reliability Requirements

Table 6 - Reliability Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
RELI-1	If the program/server crashes at any point, the user's data or latest progress will not be lost.	The user should be able to restart the application and start from their left over progress before the application crashed. As a gamer myself, I understand the frustration when a game crashes and you	We will do testing here by implementing unit tests. These unit tests will check to make sure that data isn't lost when a program isn't closed. Also	RELI - 1

		lose all the progress you were working on. That's why this will be a priority to better serve the user.	we'll update a random autosave feature about every 60 seconds to ensure that the user won't lose progress.	
RELI - 2	The server should be up and running at any point of the day.	Since this application is available for any user that meets the hardware requirements, we aren't going to restrict what time they can use the app. Everyone has busy schedules and we want them to be able to access the app at their convenience.	We will do extensive tests on the server to make sure that it's not prone to failures and can run 24/7.	RELI-2

13b Availability Requirements

Table 7 - Availability Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
AVAI-1	The application shall be available during all hours of the day.	We want the user to access the app based on their convenience. It shouldn't matter what time of the day it is. The servers should always be running.	We will ensure this by doing extensive tests on the servers and try to foreshadow any possible errors we might encounter. Unit tests and automation tests will be implemented to help make sure the servers can handle being on 24/7.	AVAI - 1

AVAI - 2	This product will have a 95% uptime.	We know that no matter how much testing we do, some bug or error might fall through and require the developer team to fix the issue. Crashes can happen during a mobile game especially when user population is high or an unforeseen issue arises. But ultimately, our goal is to have the application up and running all the time with no hiccups.	This will be handled by notifying the user population when an issue has been found. Then there might be a scheduled downtime that will be announced in advance so we can easily fix the issue and redoply the software back to the user. We will try to do quick automation tests before redeployment to make sure the issue won't reappear.	AVAI -2

13c Robustness or Fault-Tolerance Requirements

Table 8 - Robustness Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
ROBU-1	The application will continue to run in local mode whenever it loses connection with the main server.	This also allows the user to keep continuing their progress in offline mode. If the main server is down, then all onlines modes and features will be disabled.	If a server crashes, there will be an update message sent out through push notification in the game notifying the issue. We'll do our best to display the eta on when the issue will be fixed as well. Once the server is up and	ROBU- 1

			running again, offline progress will be saved on the main server and updated to the user account.	

13d Safety-Critical Requirements

Table 9 - Safety Critical Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
SACR-1	This application will not cause your phone to overheat or cause any damage to the users.	Our number one concern is the safety of our users. We want to make sure they are aware we are fully committed to this policy. Furthermore, we want to not cause the user of this product any additional charges.	The easy way to test this is to create unit tests that makes sure the device isn't taking up too much CPU. Furthermore, we will have gameplay testers who test the game for a long period of time and see if they notice any overheating. Safety testers will also analyze our product before it's released.	SACR - 1
SACR - 2	This product will only contain your account information. This will include things such as name and age. Email will be required as	There will be payment information that you could choose to save to your account as well. We will make it clear what information we need from you to ensure that you have a smooth time playing and everything else will be optional. If you choose to delete	Developers will be aware of the information given and be given training on how to handle data. We will make sure everyone is informed about data privacy. If anything gets leaked, we will inform the user	SACR - 2

	well to login and receive optional updates.	your account, then all your information will be gone from our database and be gone forever.	immediately.	

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

Table 10 - Maintenance Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
MASU - 1	Scheduled maintenance will come in the form of push notification through app and on the loading screen when starting the application.	Maintenance will be to improve servers and ensure all users are having a smooth experience. It will make sure to protect users progress and will try to take place where the user population tends to be at the lowest in the app. Sometimes it might be required for emergency maintenance in extreme situations.	Maintenance teams need to make sure they are aware of the fixes they are going to make and that they will fix the issue. The need to perform it with the window given. There will also be patch notes within the app clearly stating what was fixed.	MASU-1
MASU - 2	All forms of maintenance should take place outside of high usage hours. So most probably early morning hours central time.	We want to try to make maintenance be a little a hindrance as possible to the user. That's why unless it's an emergency we are going to perform maintenance at the time where it impacts the lowest amount of	Maintenance will be performed on off hours. The maintenance team will do extensive testing before the window and then release it during the window. Flexibility is needed if	MASU - 2

		users.	unforeseen circumstances arise.	

14b Supportability Requirements

Table 11 - Supportability Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
SURE - 1	We will implement a forum where users can share issues that arrive for the game.	We will have our maintenance team look over this forum and try their best to respond to feedback on things the user posted. If it's clear many users are having the same issue, then they will work on patching it.	Any user could post on this form 24/7. Maintenance team would only be required to look at it during normal business hours.	SURE - 1
SURE-2	We will also have a twitter page where dms can be sent to the account for more personal issues.	This will be looked at by our customer service team and then passed on to the appropriate member who can help assist you. Users can use twitter to get help on account issues, game issues etc.	The twitter account will be looked at through normal hours of operations. We will try to respond to users within a 3 day time period. For more general issues, we would recommend them to post it on the forum.	SURE - 2

14c Adaptability Requirements

Table 12 - Adaptability Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
ADRE - 1	This product must be able to run on all modern tablet devices and smartphones.	This is a mobile application and thus we expect all handheld devices to run this application. We want the newer devices because their software is more optimized and will help minimize issues with the software.	The application will be tested on every platform to make sure everything is operating smoothly. Extensive testing will be done to determine what version of the device is the cutoff.	ADRE - 1
ADRE - 2	This product will support both android and apple products.	These are the 2 most population products people use for their tablets and mobile devices. We want to cater to the user and make sure users of both platforms have easy ease of access.	The application will be tested on android and apple devices. We will also test it on apple store and google play store, to make sure that the installation goes fine regardless of platform.	ADRE - 2

14d Scalability or Extensibility Requirements

Table 13 - Scalability Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
SCAR - 1	This product should be able to handle a large user population size with no restriction on the amount of people that can join the server.	National Parks Builder should be able to handle users of all different platforms from various countries. This is to make sure everyone gets the experience of using the app with no restrictions.	Extensive testing will have to be done to see how much the server can handle and optimize it to handle a large number of clients.	SCAR - 1
SCAR - 2	We expect this software to handle an unrestricted amount of in game transactions at once.	We don't want to restrict users purchases and progress because of other users. We want to be able to handle every user's needs at once. Every user has a busy schedule so we want to make sure that when they log in they get the tasks done that they want to accomplish.	To make sure this is possible, we will perform testing on multiple transactions being done on various accounts and note any delays that may occur due to this. Extensive testing will be done to make sure that all users who decide to buy in game transactions at once can do so with no restrictions.	SCAR - 2

14e Longevity Requirements

Table 14 - Longevity Requirements

NAME-ID	Description	Rationale	Fit Criterion	Acceptance Tests
LONG - 1	We expect this app to continue to be available as long as the user population keeps growing and staying active.	We hope that we can keep coming out with new ideas that entices users to keep playing the game. We need an active user base to help justify putting in all on this work and continually improving the app.	It's important for our teams to continually take feedback of the users and do their best to implement it. As long as we keep the users entertained, they will keep putting hours into the game.	LONG - 1

15 Security Requirements

15a Access Requirements

Table 15 - Access Requirements

Name-ID	Description	Rationale	Fit Criterion	Acceptance Tests
ACRE-1	Access to users personal payment information. This will only be optimized if the user decides to opt in and purchase in game add ons.	Those employed within the company would only have access to this information. After payment has been received, encryption would occur	Users should have the ability to know their information is secure and that their payment method will stay encrypted within the system.	ACRE-1

		for the private user information given,		
ACRE - 2	Users will need to occasionally give the application access to their WIFI. This is needed so that the app can download updates and patches to fix bugs.	There will be updates regarding the application as time goes on to improve the user experience. To accomplish this, a reliable source of the internet will be required.	Users should know that the app will need to need to have access, and could possibly utilize cookies throughout their process. This could be needed to view certain content or ads	ACRE-2

15b Integrity Requirements

Table 16 - Integrity Requirements

Name-ID	Description	Rationale	Fit Criterion	Acceptance Tests
INRE-1	All data given to the application by the user will only be viewed by specific personnel in charge of the app. They are only gonna look to contribute positive cases through them.	Customers should be able to know that a certain ranking personnel will be the only ones viewing their information, only on a must needed basis. They should also be aware of the fact that their information will	Users should have the ability to know their information is secure and that their payment method will stay encrypted within the system.	INRE-1

		be kept secure, and not given to any third parties.		
INRE - 2	One of the utmost priority of the application is to protect databases filled with customer information, and if breached will notify customers immediately.	Customers should know immediately if their data has been leaked, so they could take precautions and possibly change other information and cancel credit cards if needed.	In an era of computer privacy, it's important that the consumer knows when their information has been leaked. This could affect other accounts and other system logins.	INRE-2

15c Privacy Requirements

Table 17 - Privacy Requirements

Name-ID	Description	Rationale	Fit Criterion	Acceptance Tests
PRRE - 1	Customers need to be notified if there are any changes regarding the data of a customer.	Customers should be notified if any changes are happening in regards to their application and how it will affect their experience with the application.	This application will not run until the user accepts that the new policy has changed. By implementing this measure, it will prevent any repercussions that could	PRRE-1

			happen if changes are made without customer consent.	

15d Audit Requirements

Table 18 - Audit Requirements

Name-ID	Description	Rationale	Fit Criterion	Acceptance Tests
AURE-1	Audits will be completed multiple times throughout the year regarding financial transactions.	There will be multiple audits throughout the year to keep the applications functionally working, and with up to date software for financial and gameplay features.	It's important to keep the application up to date with audits. This task will be done in a timely manner from the game developers, which will help provide a complete optimal performance year round.	AURE-1

15e Immunity Requirements

Table 19 - Immunity Requirements

Name-ID	Description	Rationale	Fit Criterion	Acceptance Tests
IMRE-1	The system should be hosted by third party	With constant feedback from security firms, the application	There will be multiple rounds of tests placed such as	IMRE - 1

	security softwares. These should be constantly backing up and keeping information encrypted.	will be free of any unauthorized packets being set, and any other malware from corrupting files.	OWASP. These tests will have multiple iterations to protect personal data.	

16 Usability and Humanity Requirements

16a Ease of Use Requirements

ID# - Name EAU1

Description: The application's usability should be fairly easy, as the user just has to drag and drop applications onto the national park. Furthermore, the rest of the gameplay relies on touching certain items for usability.

Rationale: The application is a game, which could be focused more on the younger generation, so we would want to keep the gameplay as simplistic as possible.

Fit Criterion: If we overcomplicate the gameplay, it might gear users away from the application, and with just dragging and touching it would be very easily accessible to play on a mobile device.

Acceptance Tests: EAU - 1

16b Personalization and Internationalization Requirements

ID# - Name PAI1

Description: The application will eventually be available in some of the more known languages such as spanish, french, hindi, etc. Furthermore, currency to buy additional features will accept a multitude of currencies.

Rationale: Through this customization it will give more players around the world the incentive to try out the application, knowing that they can add their own bit of personalization to enhance their experience..

Fit Criterion: Through many different iterations of testing, it will ensure that all the personalization is active and can be enabled by the users.

Acceptance Tests: PAI - 1

16c Learning Requirements

ID# - Name LR1

Description: The user will have to learn the intended reasoning of the game, such as building certain items for the park, and knowing which even parties they need to take care of such as forestry.

Rationale: It's important that there is a tutorial when the application first opens up, to ensure that the user knows the objective of the game, including what they need to drag onto the screen, and how to ensure that the park is at optimal health.

Fit Criterion: This important usability would be done to ensure a smooth process for the players as they will have the confidence in the instructions of the game, and will showcase the ease of the application.

Acceptance Tests: LR - 1

16d Understandability and Politeness Requirements

ID# - Name UPR1

Description: The application will have a constant reminder feature on what the current objective is, so the user will always know what they need to complete to have the best experience.

Rationale: It is important that the user is always knowing the program is meant to do, and what the current objective is, so they are able to complete certain tasks and progress through the game.

Fit Criterion: This will make sure it's clear through the tutorial when you first place the game. Furthermore, in the game we will have a clear section where it's labeled with what the user's current objectives are. This will be the developers job to test it and make sure that's it displaying clearly and in an understable manner for the user.

Acceptance Tests: UPR - 1

16e Accessibility Requirements

ID# - Name

Description: The product should be accessible for any individuals with a mobile device, and there will be options of audio and sensory, so different individuals will have the capability to play.

Rationale: All users should have the ability to participate in the application, and by making it more user friendly, it will give everyone the chance to try it out.

Fit Criterion: This should be implemented within the first release of the software, as all individuals will immediately have access to the application.

Acceptance Tests: UPR - 2

16f User Documentation Requirements

ID# - Name UDR1

Description: Every time the user drags an icon onto the screen after purchasing it, it should be recorded in a sort of manual within the application. Furthermore, it should document the level of different buildings, and somehow have a numeric value for how the parks areas are doing.

Rationale: It's important different features of the game are documented, so the player will be able to keep track of their progress, and understand what exactly they have done so far within the game.

Fit Criterion: All documentation will be approved by third party companies to ensure its compliance, and also make sure all important information is listed within documentation so there will be no confusion for the users.

Acceptance Tests: UDR - 1

16g Training Requirements

ID# - Name TR1

Description: Users will need to learn how to operate smart phones, and understand the objective of the game such as learning how to be resourceful, and how to maintain and operate a park.

Rationale: Users need to learn so that they will be able to play the application, and have the training to exceed within the game.

Fit Criterion: The training of the application is basically a tutorial, and furthermore if any additional training is needed the developers could always update the software, so the users would be able to read any new features that they need to train up on.

Acceptance Tests: TR - 1

17 Look and Feel Requirements

17a Appearance Requirements

ID# - Name LFR1

Description: The application should be able to switch around maps within the game, having the ability to switch over a multitude of different national parks.

Rationale: If there are a multitude of parks to choose from, it will bring in greater awareness for national parks, and will give players the incentive of playing the game longer considering there will be more content.

Fit Criterion: We will have multiple different tests through Maven to test all the different maps and ensure they all have the same functionality.

Acceptance Tests: LFR - 1

17b Style Requirements

ID# - Name SR1

Description: The product should have a current modern feel for teenagers and young adults, possibly incorporating current trends within the game to captivate that specific audience.

Rationale: It is important that the application has a trendy feeling to the clients as they will primarily be younger adults, and its important to market towards them.

Fit Criterion: There will be a lot of different testing and research done, to try and make the game seem more modern and have a feel for current games to enhance its image to the public.

Acceptance Tests: SR - 1

18 Operational and Environmental Requirements

18a Expected Physical Environment

ID# - Name OER1

Description: The product should be able to work in any environment, as all that is needed is a sufficient smartphone.

Rationale: The application does not require an internet connection, unless updates are needed, so it's important that no environmental or operational constraints are placed for the game.

Fit Criterion: There will be no constraints for the application to increase usability, and the user should be able to play anywhere in their home or where their device is functioning.

Acceptance Tests: OER - 1

18b Requirements for Interfacing with Adjacent Systems

ID# - Name RIA1

Description: The application should work with the operating system of different mobile devices, and be at an efficient ram and data storage.

Rationale: We don't want the app taking up too many gigabytes on a mobile device, so we want to keep it efficient and compact, not slowing down the performance of the device at all.

Fit Criterion: The product should work on all mobile devices, and have possible updates to download different maps and new features, all while testing them.

Acceptance Tests: RIA - 1

18c Productization Requirements

ID# - Name PR1

Description: The app will primarily live within the device, but possible cloud based saves could be possible to transfer game data within different devices.

Rationale: This will give users a sense of insurance knowing that their progress will still be saved within the cloud.

Fit Criterion: It is important that the application will be compatible with the cloud so saves of the game are recorded, and will require less memory usage for the hardware.

Acceptance Tests: PR - 1

18d Release Requirements

ID# - Name RR1

Description: There will be multiple releases throughout the year, and will try and add a new feature each time.

Rationale: This will prevent the game from getting stale, and will constantly try and improve the software to be able to obtain the best experience.

Fit Criterion: Each release will be constantly tested, and the developer team will be constantly working from their end to provide up to date changes.

Acceptance Tests: RR - 1

19 Cultural and Political Requirements

19a Cultural Requirements

<u>ID# - CR1</u>

Description: The product should ensure to avoid any relation to religion

Rationale: The reason for this is to relate to a larger audience and avoid any problems

Fit Criterion: Developers should make sure to avoid any talk of religion when creating the game

Acceptance Tests: CR - 1

19b Political Requirements

<u>ID# - PORE - 1</u>

Description: The product won't restrict anyone to log in because of the country they are from.

Rationale: We want users from all backgrounds to be able to play this game.

Fit Criterion: We will make sure that we are just and avoid anything related to politics or any controversial topics. This will help ensure users won't get offended and we won't have to deal with any governments.

Acceptance Tests: PORE-1

20 Legal Requirements

20a Compliance Requirements

<u>ID# - CMPR1</u>

Description: The only information that may be stored is a Users' login information

Rationale: No other personal information is necessary for the game to function per user

Fit Criterion: Ensure that a User class does not have any personal information data members besides a username and password

Acceptance Tests: CMPR - 1

ID# - CMPR2

Description: Because a Username and password will be stored for each user of the product, no developer should have access to this information

Rationale: This must be done in order to avoid stolen accounts and information

Fit Criterion: Check a User Class and ensure that developers do not have access to viewing this information once it is stored

Acceptance Tests: CMPR - 2

20b Standards Requirements

ID# - SR1

Description: The product must have some sort of protection against server DDos attacks

Rationale: This is to prevent the whole game from going down

Fit Criterion: The game should have some anti DDos software

Acceptance Tests: SR - 1

21 Requirements Acceptance Tests

21a Requirements – Test Correspondence Summary

FUNC - 1	X																			
FUNC - 2		X																		
FUNC - 3			X																	
FUNC - 4				X																
DATA - 1					X															
DATA - 2						X														
DATA - 3							X													
SPLA - 1								X												
SPLA - 2									X											
PRAC - 1										X										
PRAC - 2											X									
CAPA - 1												X								
RELI - 1													X							
AVAI - 1														X						
AVAI - 2															X					
ROBU - 1																X				
SACR - 1																	X			
SACR - 2																		X		
MASU - 1																			X	
MASU - 2																				X

SURE - 1			X																															
SURE - 2				X																														
ADRE - 1					X																													
ADRE - 2						X																												
SCAR - 1							X																											
SCAR - 2								X																										
LONG - 1									X																									
ACRE - 1										X																								
ACRE - 2											X																							
INRE - 1												X																						
INRE - 2													X																					
INRE - 3														X																				
PRRE - 1															X																			
AURE - 1																X																		
IMRE - 1																	X																	
EAU - 1																		X																
PAI - 1																			X															
LR - 1																				X														
UPR - 1																					X													
UPR - 2																						X												
UDR - 1																							X											
TR - 1																								X										
LFR - 1																									X									
SR - 1																										X								
OER - 1																											X							
RIA - 1																												X						
PR - 1																													X					
RR - 1																														X				
CR - 1																															X			
CMPR - 1																																X		
CMPR - 2																																	X	
SR - 1																																		X

21b Acceptance Test Descriptions

Every Requirement has a description that will explain the reason behind it and what will be done to test it if necessary.

III Design

22 Design Goal

The general design of the National Park Builder takes into consideration that the majority of the screen/user interface will be the terrain. The terrain is where the user will manage all of their buildings. The average user needs to be able to enter the game and have a clear understanding of what is where. All buildings/structures will be clearly highlighted and labeled to be able to distinguish them from other types.

Other key features of the game will be buttons around the screen of the user's device. There will be clearly identifiable buttons for different features of the game. These buttons include the shop, alerts, missions, settings, and more. The buttons will be designed to not blend into the rest of the environment. The game will be easy to manage and it will not be difficult for the user to find what they are looking for in any menu/button.

23 Current System Design

Because this is a brand new game, there is no real existing system implemented, though it will be similar to the UI of other mobile strategy games.

24 Proposed System Design

24a Initial System Analysis and Class Identification

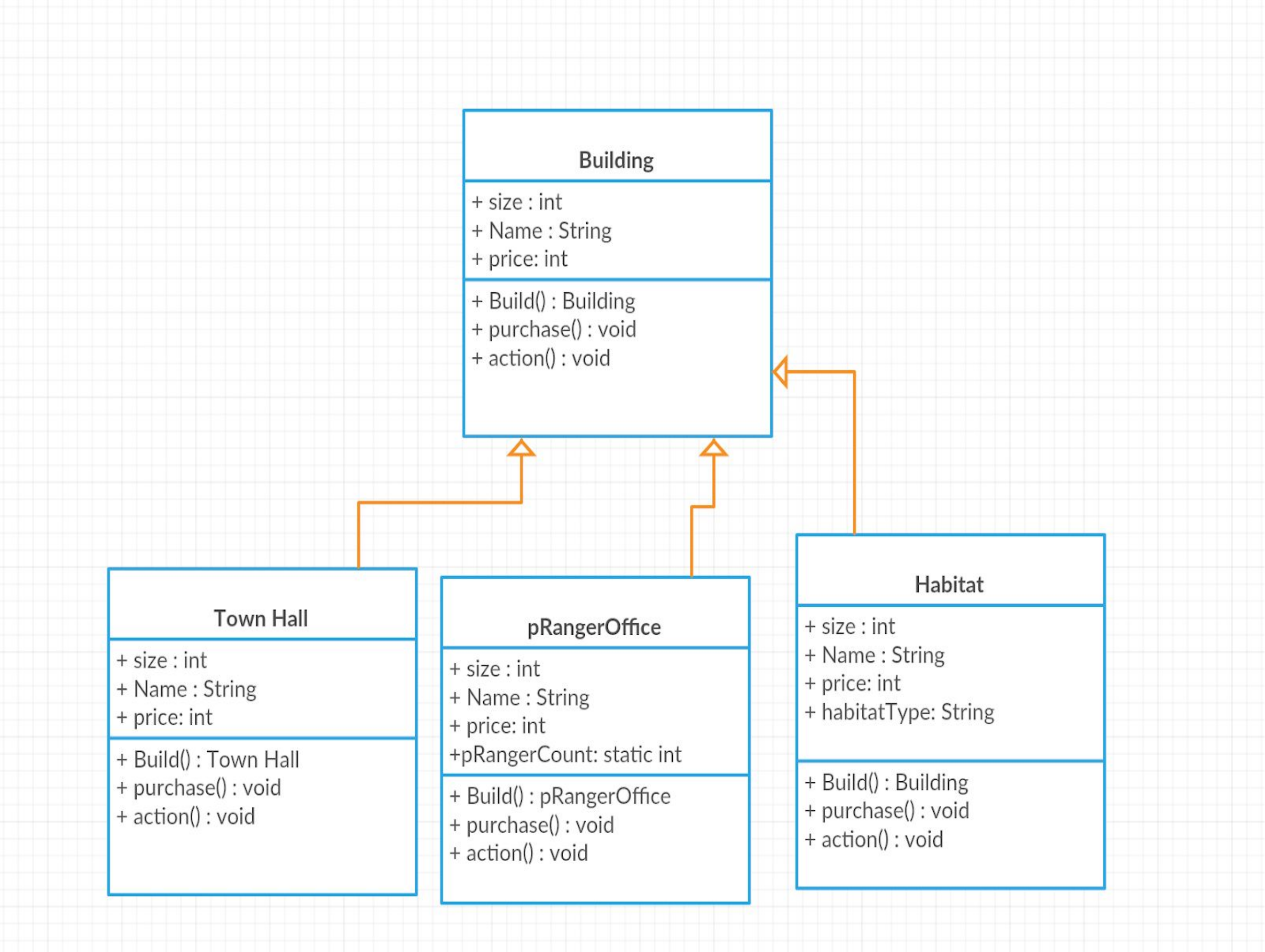

Figure 4 - Buildings UML Class Diagram

24b Dynamic Modelling of Use-Cases

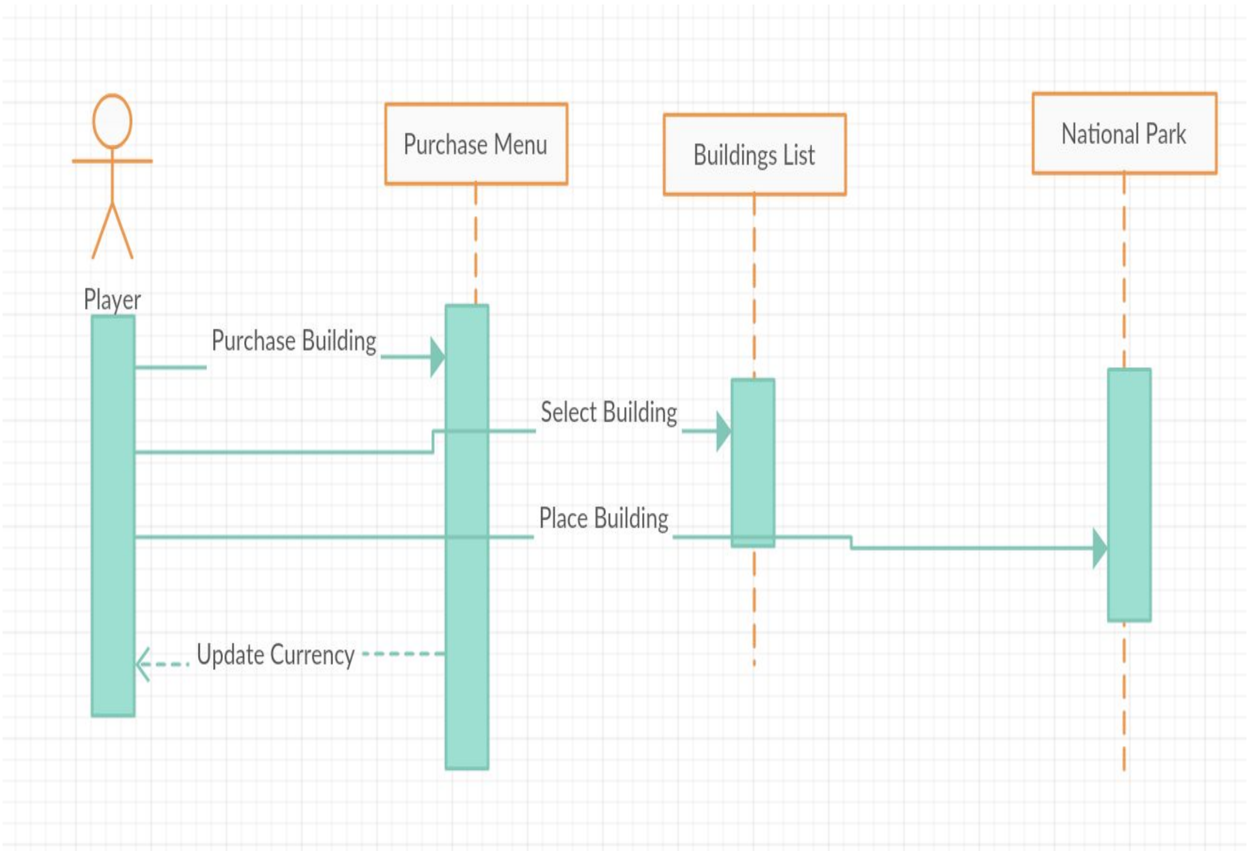

Figure 5 - Sequence Diagram for Purchasing and Placing a new Building Type

24c Proposed System Architecture

As for the Development Platform, Unity will be used to create the game. Unity is a great platform to be used for this game because it is incredibly compatible for mobile platforms IOS and Android. It is also great for incorporating 2D and 3D models of the different buildings into the game for an amazing design. For saving all the data of every user, mySQL can be used as a great database/server.

24d Initial Subsystem Decomposition

Building: This is the Interface and what all building types will be using for the general components. This includes the size of the building, the name of the building, and its price. The size is for the space it will take up in the user's terrain, the name is used to specify the different building types available in the shop, and the price is obviously for the price of the building(Each building will have its own respective price). It will also include a Build() function which will be used once the user has placed their building. Lastly, Action() will be some unique action that each building will have differently.

Shop: The shop is where the user will be able to buy any and all structures. This includes Buildings and any other type item that may be available in the shop.

National Park: The national park is the user's whole terrain available to them where they will place buildings and where all animals will live. They will be able to expand their park by paying for more space. A player can have a name for their own National Park.

Player: A player will have a userID to be identified by other players. They will have their own list of owned buildings and other useful data about their National park.

25 Additional Design Considerations

25a Hardware / Software Mapping

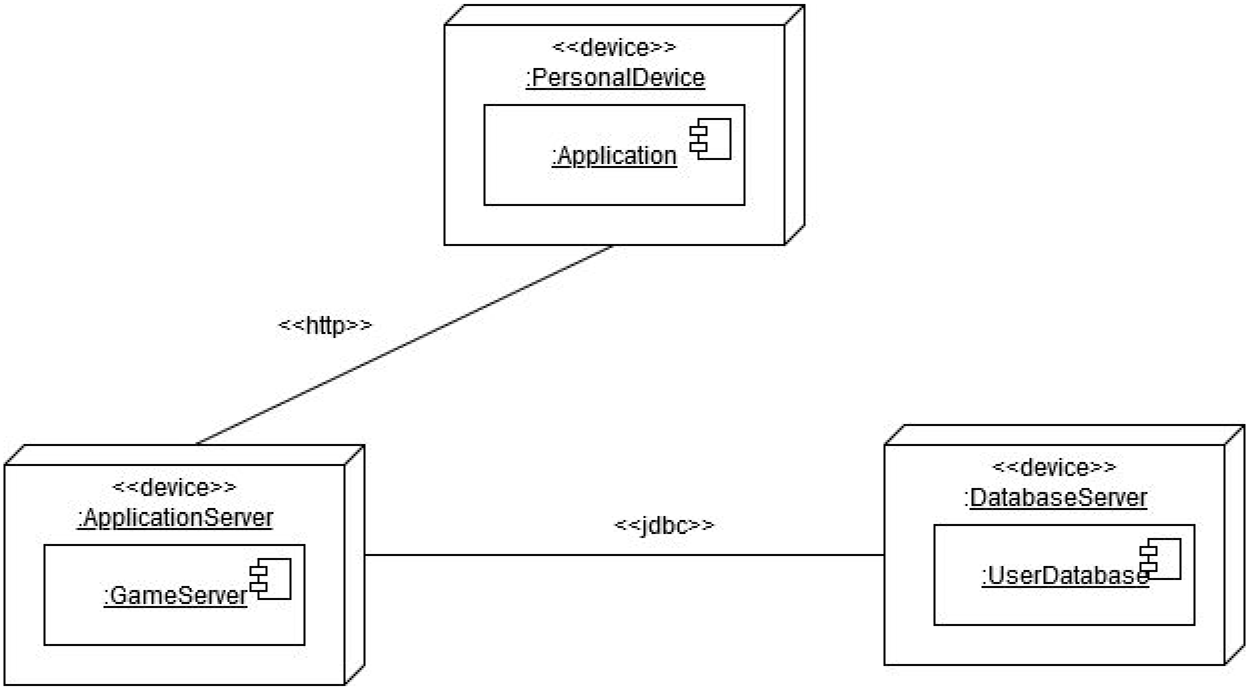

Figure 6 - Hardware/Software Mapping

25b Persistent Data Management

The database will store the user's data and will be communicated with by the Controller. This will allow restoration of any data in the result of a system failure. This user data would include things such as account information and game progression. This should avoid account information or game data being lost.

25c Access Control and Security

The game will require a decent amount of security. Although some of the security concerns can easily be handled by the standard security of the platform (e.g. firewalls), the game itself will require its own security measures as well. Since any person who has the game downloaded has access to the game, the application must be able to effectively limit the access that players can have to a large portion of the game. In other words, users that play the game should only have access to their client and nothing else. They should not be able to access other clients, server data, game data, etc.

Another security concern is the protection of player information and transaction data. The game requires the use of some sort of email to register (this can be done by creating some sort of verification class). Furthermore, the game will allow the user to buy microtransactions as well as make donations. Consequently, this will require some sort of encryption and other security measures that would ensure credit card data as well as emails and passwords are not exposed.

Lastly, another concern that must be addressed is the possibility of intentional server failure. This is usually done through a DDOS or DOS attack (Denial of Service). These sorts of attacks can cause interruptions and deny a user's authorization to access their network or connect to the server. These attacks are most commonly executed through "Smurf Attacks" or "SYN floods" (refer to glossary for definitions). These attacks are unavoidable so there's no clear solution. However, the game will require an advanced server and network that would allow developers to monitor activity to ensure that these attacks are as rare as possible.

25d Global Software Control

Requests that are made will be handled based on the situation. In most cases, the users will be the requests handled. However, some requests handled by other groups can be made as well.

Once users launch the game, a request will be sent to the server which will allow the client that the user is using to connect to the server. In addition to this, the controller will access the database for any necessary data, and update the model accordingly which would update the view for the user. Any feature or action that the user tries to implement will be used by the controller to update the model and then the view, as well as update data when necessary.

For national parks, a request can be made when collecting donations for them. This would require the use of a controller accessing a database that contains the transaction.

For experts that want to inform users on national parks, they can create a request for the system to send facts. This would again require the controller to update a database which can transfer the facts to the server and be distributed to the clients.

The main concerns for these requests would be security, and how it should be handled is highlighted in 25c.

25e Boundary Conditions

During startup, the system must ensure that the user's client is up to date. If not, the application will download the update before fully starting the game. After the startup is completed, the game should be able to display exactly where the player left off and possibly more if the player needs to be updated on certain things.

Normally, when the user shuts the application down, the game should be able to save all the user data. This would be done by ensuring the user's data is saved within the database. That way, the user would be able start the game up at a later time without any issues on lost data.

However, the game may shut down under abnormal circumstances (e.g. crashing). Generally, the user who experienced the crash should not have any major issues with data loss. This is because the game should constantly be loading and saving user progression. That way, if the game unexpectedly closes, the data that was saved should remain intact.

25f User Interface

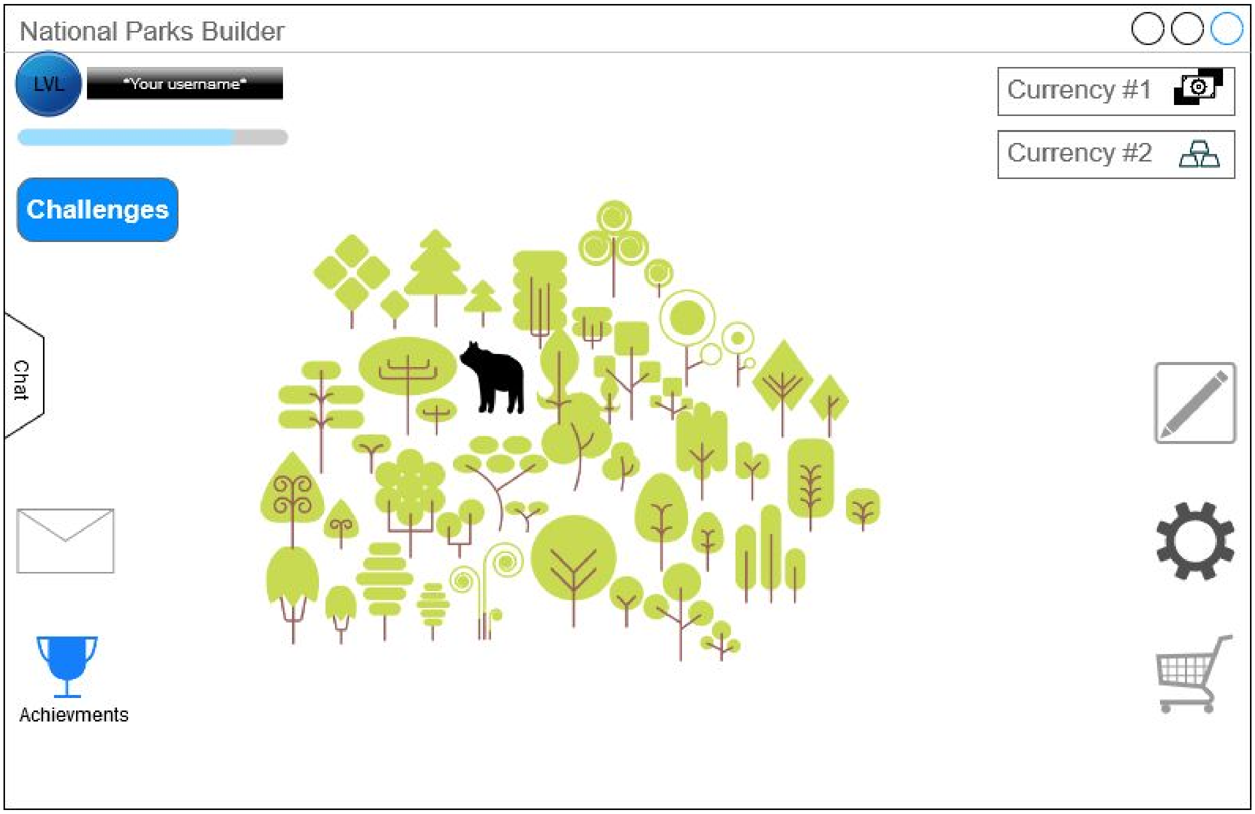

Figure 7 - User Interface

25g Application of Design Patterns

N/A

26 Final System Design

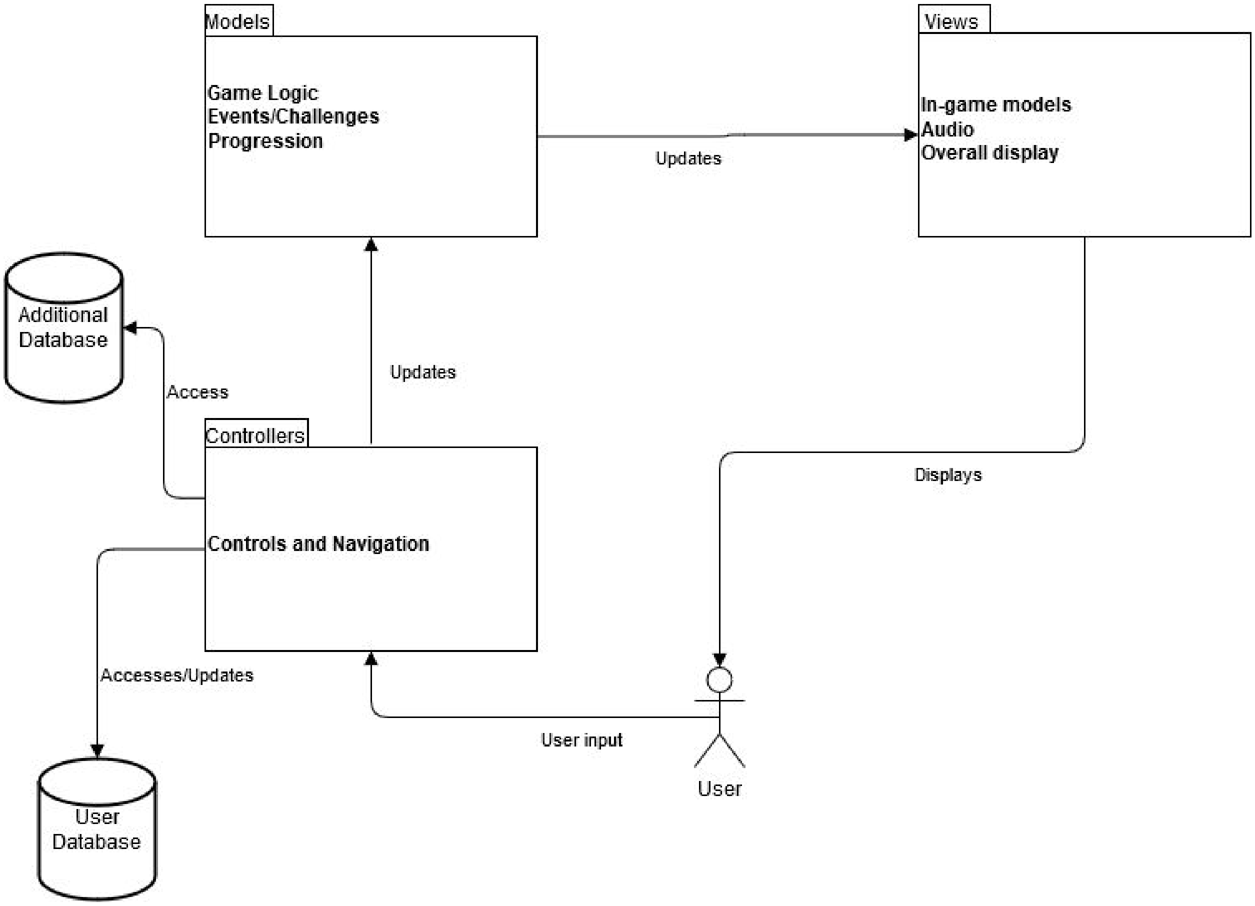

Figure 8 - Final System Design

The application will primarily implement a Model-View-Controller (MVC) pattern. The controller will be focused on listening for specific events, mainly input. From there, the controller will update the model. The model will control the game logic and the user progression. Next, the model will update the view that would contain the models and display. The view will display to the user based on certain inputs.

A database will be used to mainly store account information, but can also be used for user progress as well as transaction data. An additional database could possibly be implemented as well for things such as terrain data for different parks or facts for the user.

This design pattern is advantageous because it separates the application into parts. These parts can be developed individually which can streamline the process. Furthermore, it will make future updates to the application easier to carry out because a specific portion of the program can be worked on without relying on the rest of the application.

27 Object Design

27a Packages

The project as it stands currently does not contain any specific packages. They can be added to the project if necessary.

27b Subsystem I

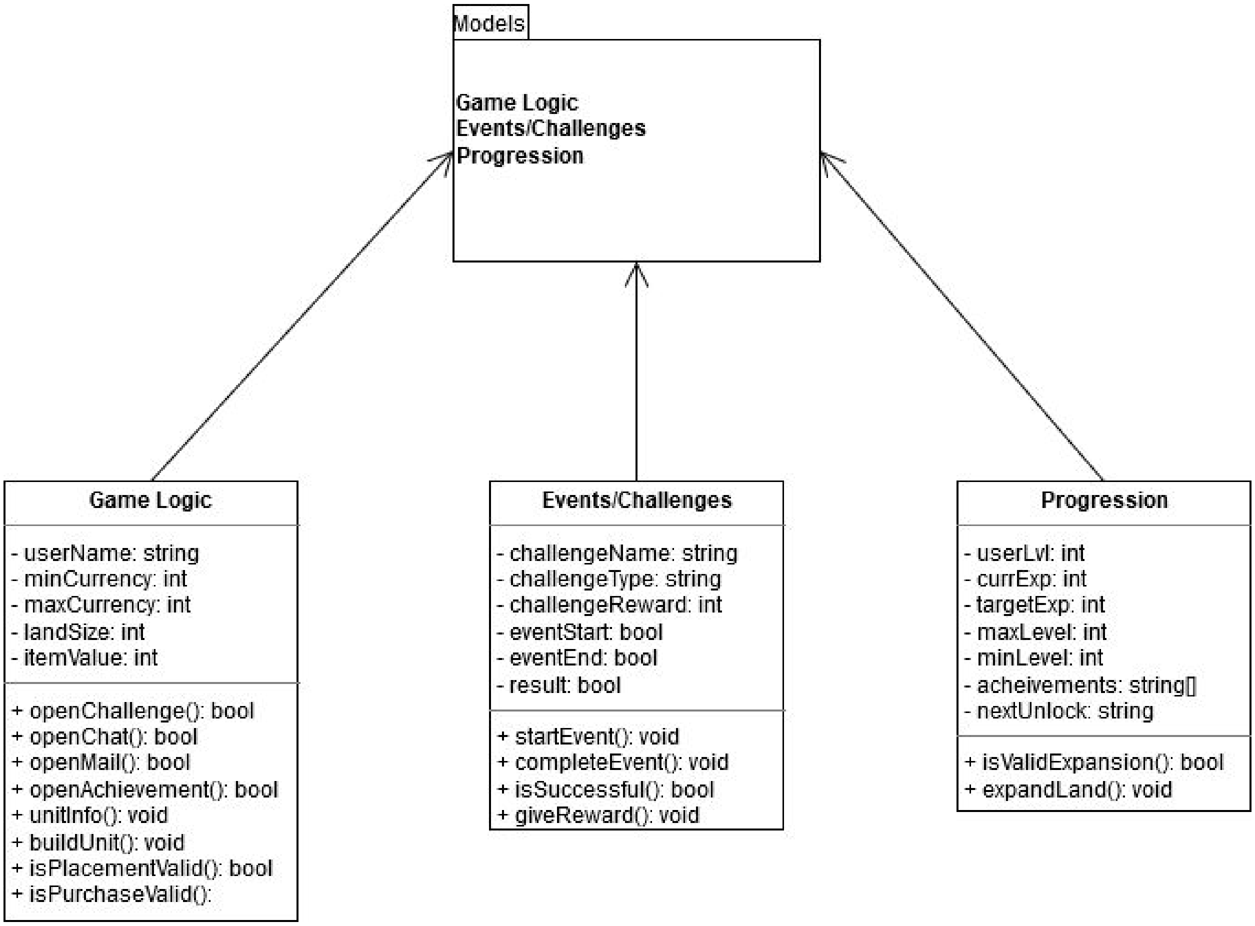

Figure 9 - Subsystem I

Subsystem I will primarily contain all the code that revolves around the Model. The Model will be focused on game logic, events/challenges, and user progression.

The game logic will focus on aspects of the game that revolve around “If I do input x, what do I need to create output y?”. This would include aspects such as knowing the user’s name, currency, the size of land, and value of items in the store. This would also include methods that would gather data together to give to the view for opening different menus as well as methods that check if an action is valid. In addition, there will be some methods that would have the code necessary to complete tasks such as building units.

27c Subsystem II

Subsystem II will primarily contain code revolving around the View. The View will focus on the design of the models in the game (trees, animals, etc), audio cues, and the user interface (this would include pop ups or error messages). The view would update and display to the user to see.

The in game models would include natural objects such as trees, hills, mountains, lakes, geysers, animals, and other aspects of nature. In addition, it would include buildings such as the center and other spots like rest spots, picnic areas, etc. These models would include dimensions, texture, and would require using some separate software to develop.

The audio cues would be pre recorded and would be used for certain actions. This would include sounds from button pressing or sounds for pop up messages. The fields that would be stored here would simply be the type of audio cue.

The user interface would contain the items of any sort of GUI designer (buttons, panels, windows, etc). It would also include the design of all menus as well.

No diagram is necessary for the View simply because it would mainly contain models and design. The actual code that would revolve around the functionality and what each part of the design would contain would be stored in the Model portion of the data.

27d Subsystem III

Subsystem III will primarily contain code revolving around the Controller. The Controller would contain the user controls that would be used for input (button presses, typed messages, etc).

The input would be used to navigate through the game. The fields would essentially revolve around what types of input can be applied. The methods would be focused on whether to access or update the Model or database. These methods would also work to know what features in the Model to use based on the input.

No diagram is necessary for the Controller mainly due to it varying depending on the platform. The way input would be handled on a mobile device would be different from a computer.

IV Project Issues

28 Open Issues

Some open issues that our product could face are the competition of games within the app store, and the ability to keep customers interested with the game as time goes on. These are very relevant issues that all applications face, as it is very difficult to create games that are successful within the long run. Since our game is based on National Parks, we hope to modernize and design it in a way that the current youth will be able to enjoy. Furthermore, in hopes of keeping people interested as time goes on, there will be constant updates and adding new features throughout the years. Another possible issue that there could be, is not straying away too much from the aspect of National Parks. The whole point of the application is to bring awareness, so as time goes on it is important to not sway too far from the original idea.

29 Off-the-Shelf Solutions

29a Ready-Made Products

The application could utilize possible Google API's for different National Parks to try and give a real life feeling within the game. For example, if there is precipitation at the time of a specific park, we could utilize that information and try to have a similar feeling within the game. Furthermore, another ready made product we could utilize is a possible drag and drop sort of application that could be the original prototype, and from there new features could be added afterward to the application.

29b Reusable Components

Some reusable products could possibly be sprites or certain objects within the game. This will let us keep on utilizing certain products whenever we would need to access them. Also, there could be certain areas within the application that could also be reusable such as certain UI designs, or functions regarding aspects of the game.

29c Products That Can Be Copied

There may be a few products out there in the market, that would be willing to give us the ability to reuse or try and implement the same style that they may have utilized. If a pre existing game has a possible implementation of a drag drop type of game, there are specific functionalities we could use so we would not have to create them ourselves.

30 New Problems

30a Effects on the Current Environment

Overall, the system does not seem to have any adverse effects on the working environment. It could possibly change the workflow on how software is developed, or could even help companies begin to change their perspective on games, and try to incorporate awareness to other issues throughout the world.

30b Effects on the Installed Systems

The effects on hardware or software is that it could change the software of games, making them more likely to adapt the drag and drop approach. Furthermore, with the multiple aspects of the game a larger hardware capability may be needed to support all the different functionalities and updates of the game.

30c Potential User Problems

Overall, we don’t think users would have a negative response with the new system. The system itself will give a new spin on a game, and should keep the users refreshed with how the system was designed and prepared. Furthermore, the drag and drop approach may be difficult for some users, so a tutorial will be recommended.

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

Some limitations that could possibly occur are lack of storage on the device, or the actual software on the device being outdated. Furthermore, any other problems that could affect the device could also have an effect on the performance of the game.

30e Follow-Up Problems

Some additional problems could be that the product is too difficult for certain people to use, or they are simply unable to get it working due to external issues. In that case, we would have a support team that would handle different cases like that.

31 Migration to the New Product

31a Requirements for Migration to the New Product

Not Applicable

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

Not applicable

32 Risks

With any new product, there are always potential risks that will arise. The first potential risk we could encounter is having trouble keeping the audience's attention and engagement within the application. A gaming application is a form of entertainment for the user and isn’t a necessity application that they need to open everyday. Thus, it’s critical to make them want to open this app constantly.

Another risk is the servers having trouble running smoothly with all the users using the application. This would cause a risk of users being unhappy with the product and thus look to spend their time in another way. As a gamer myself, it’s very frustrating when you want to play a game and the servers are down. This makes you frustrated with the company and makes you question if this game is worth playing. We don’t want our users to have experience like this and thus it’s important to consider this risk.

Additionally, a risk is the payment information of the users getting leaked. This would be a big privacy issue and give us a bad and untrusting reputation as a company. Privacy concerns are always a big concern for users. We realize that when asking for people's payment information there's always a big risk involved. This is why we would need to make sure we have a strong security and encryption team to handle this business..

These risks all are something we need to keep in mind when building this product. We need to prepare for the possibility of these risks happening and have a plan in place to ensure these possible risks don't cause our product to sink. Also it's important that we must not fall under the silver bullet syndrome.

We must make sure that we are looking at one feature at a time and not rushing to complete one task. There must be a process in place where we perfect each feature and test it thoroughly to make sure that it's going above and beyond expectations. It's important to make sure that there are no flaws within the feature. Then when we implement the next feature, we test the first feature and next one together to ensure they both work flawlessly together. Going along with that, it's important to make sure that the project definitions, documentation and control is clear so everyone is on the same page and mistakes are limited. This allows us to have a big team working on the product but ensuring everyone is on the same page. As long as we can keep these things in mind, it will make sure we stay focused on the project objective and ensure a strong initial product versatility.

33 Costs

An estimated cost of the entire project should be around at max of $500,000 for this initial project. This price includes every aspect of creating and maintaining this mobile gaming application. The main coverage of this price would go to the coding,graphic content and plot development that goes into making National Parks builder. This application has a lot of factors that need to be considered when developing it and making it so thus there is a high price tag associated with it. Within the funds, we also expect to have a strong security team that helps with securing payment information and any possible hacking. With all this in mind, we hope that this application can be accomplished in a 6-10 month time frame. We realize that this is a high initial cost but we feel that once this project is up and running we'll attract a high enough customer base that we would be able to make a product that gives higher gains than our original investment.

Game graphics is a critical piece we need to invest our funds into for National Parks Builder. We will mainly use 2D images. However, there's a lot to implement within a game like this. We would have thousands of objects, characters and scenes. We need to make sure every character is animated. We want to make the scenes visually appealing and high quality to illustrate the beauty of National Parks. We want to include vivid details for various weather to depict the seriousness of it. We feel this will have the biggest cost associated out of the total and will be approximately $275,000.

Another aspect that we need to make sure we cover is the payment integrations portion. We want to make sure we can implement multiple payment systems and that they are encrypted properly. Going along with this, we need to make sure we have a good security team that prevents any hacking or leaks of user's payment information from occurring. Therefore, the cost of this feature would be about $20,000.

Furthermore, a big part of National Parks Builder is the social and mobile features involved. We want to send push notifications from the game to the users. It would inform them about various things happening to their National Park when they aren't logged on. We also want to implement a friends list system where you can view your friends progress. Additionally, we would want a leaderboard system where users can see the top rated National Parks that users have built. Overall, we expect this to cost $15,000.

Furthermore, we need to make sure that account creation and user login functionality is there. We need to be able to store users' login information in a database. This is so it makes it easy for them to login and retrieve a missing password/username if they forget. We would also need a way to store the user's progress. We also need a capable server that includes all users to run this application at once without crashing. We also need to have a developer team that's ready to send constant updates to the game adding new items to obtain or goals to complete. These are all big and essential features that the application must have. We estimate these features will cost around $100,000 to accomplish.

Lastly, we need to take into account the marketing aspect of releasing a new game. We need to put funds to help get the word about the product. We were thinking about including ads on applications you use on your phone such as twitter, facebook, youtube and reddit. This way we'll be directly sendings advertisements to the main targeted audience for this application. We expect to put $50,000 into this. This puts at a total of $460,000. We left an extra $40,000 to account for any issues or bugs that arise with the product. We are aware that they'll be bumps on the road and we need to be prepared to have extra cash to be ready for it. This is the estimated cost we expect will be needed to make the best version of National Parks Builder possible.

34 Waiting Room

With a mobile gaming application, there's always more features and items you can add to the game. For initial release, we wanted to focus on the essential items that help ensure the users have a smooth and enjoyable experience out of the gate. We know after release, they'll be errors and bugs that users will find that we need to make sure we take care of. After this phrase, then we can get ready to add new features to our application. We want the user population to be a big part of new things we incorporate in the game. However, we do have features we want to add after the initial launch, Some features we have planned already are included in the table below.

Table 20 - Waiting Room

Requirement	Priority	Motivation	Intended Version Number
Create a chat messaging service where users could message their friends in game	High	This would help keep the users entertained while building their own island. They could message their friend on advice or how to approach something in regards to their personalized National Park.	Version 2
Offline Mode	High	Allow the users to play without having any internet connection. This would benefit people who have poor internet or not in range of an internet route. The downsides would be no interacting with friends or viewing the leaderboard.	Version 2
Add more challenges and objectives to complete	Medium	Give the users more challenges and objectives to complete. This is to keep player engagement high.	Version 3
Create an in game forum	Low	Give users the ability to access the forum in game. This allows them to share feedback and things they want	Version 3

		improved or changed within the game.	
Create an application on PC of the game	Low	If the success of the game on mobile is going well, then it would be a good idea to expand business. We were thinking of creating a version of the game on PC so it reaches a bigger audience.	Version 4

35 Ideas for Solutions

To implement National Parks Builder, we were thinking that Unity or Pygame can be used. We feel the best language to code this application would be java. This would be consistent throughout the entire system except for the database. As for the database aspect, MySql can be utilized. Lastly, Amazon Web Service (AWS) could be used to save game data. Libraries will vary based on individual features and will be left up to the developer's judgement on what libraries they choose to incorporate. Libraries can be helpful though as they can make the job easier and save time.

As for the new features we wanted to implement, the features are things that the team of software engineers will be able to develop. Before these features are deployed, they would still go through the same rigorous testing and inspection as the other features. Ideally, we hope that we can use the same IDEs and software that we already implemented in the previous features. If decided to make this application run on PC as well, more resources may need to be allocated and hardware requirements will need to be taken into consideration.

36 Project Retrospective

National Parks Builder is a fun mobile gaming application where you can build, manage and maintain your own unique National Park. Overall we are pleased with how our plan and design for National Park Builder turned out. We tried to be as specific and include as many details as possible to help future groups on coding this. We would love to see another group code a prototype for this product. Ultimately, our main goal is to help increase awareness about National Parks and encourage more people to check out the natural beauty included within the parks.

The methods that were effective when completing this document was our weekly meetings and following up weekly with the progress each person is making on their assigned task. This was effective because it ensured every group member was on the same page and also that we made weekly progress. Before assigning each section, we all had a brainstorming session on how we want to approach a section and what things to make sure we focus/clarify on in the documentation. When a member finished their assigned part, we would go over it in a group setting just to make sure it was clear and well written. Our priority was to make sure that this document is easily readable and understandable.

A process that didn't quite work was formatting issues. For formatting, we did use the template provided by Professor Bell. However, we each wanted to make some tweaks to how the format is based on our personal preference such as different bullet point style or making a table, etc. Yet, when this happened it caused the whole document to vary in terms of format and thus making it inconsistent with varying format styles. This required us to sit down and analyze the formatting at the end and to implement consistency. This was ineffective as it took more time in the end and could have been prevented at an earlier point of the documentation.

For future development, it would be best if we all agreed on a formatting style and made sure we stayed consistent with that throughout the whole document. This would allow us not to waste time fixing formatting but allow us to spend more time on rereading and fixing errors within the actual context of the document.

Overall, we worked well as a group. We helped each other out when someone had confusion with their section or needed advice on how to approach a section. Work was evenly divided up and everyone completed their tasks on time.

V Glossary

denial-of-service (DoS) attack: an interruption in an authorized user's access to a computer network, typically one caused with malicious intent.

Smurf Attack: A DoS attack where the attacker sends Internet Control Message Protocol broadcast packets to a number of hosts with a spoofed source Internet Protocol address that belongs to the target machine. The recipients of these spoofed packets will then respond, and the targeted host will be flooded with those responses

SYN flood: A DoS attack where the attacker sends a request to connect to the target server but does not complete the connection through what is known as a three-way handshake—a method used in a Transmission Control Protocol (TCP)/IP network to create a connection between a local host/client and server. The incomplete handshake leaves the connected port in an occupied status and unavailable for further requests. An attacker will continue to send requests, saturating all open ports, so that legitimate users cannot connect.

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.

[5] Jeremy, Damian, Fabian,Marcos, Waitless Product Description Spring 2019

[6] “Security Tip (ST04-015).” *Understanding Denial-of-Service Attacks | CISA*, Department of Homeland Security, 20 Nov. 2019

[7] “How Much Does It Cost to Develop a Successful City-Building Game App?” *Gamasutra*, www.gamasutra.com/blogs/AndreiKlubnikin/20170815/303674/How_Much_Does_it_Cost_to_Develop_a_Successful_CityBuilding_Game_App.php.

VII Index