

Conservatory Catch



Project Development Report for Conservatory Catch

***Prepared by
Jose Cuellar-
Aguirre,
Marcos
Echevarria,
Christian
Garcia, and***

Nick Tryba CS
440 at the
University of Illinois Chicago

Table of Contents

I	ProjectDescription	7
1	ProjectOverview	7
2	The Purpose of the Project	7
	2a The User Business or Background of the Project Effort	7
	2b Goals of the Project	7
	2c Measurement	8
3	The Scope of the Work	8
	3a The Current Situation	8
	3b The Context of the Work	9
	3c Work Partitioning.....	9
	3d Competing Products	11
4	The Scope of the Product	11
	4a Scenario Diagram(s)	11
	4b Product Scenario List	13
	4c Individual Product Scenarios	14
5	Stakeholders	14
	5a The Client.....	14
	5b The Customer	14
	5c Hands-On Users of the Product	14
	5d Maintenance Users and Service Technicians	15
	5e Other Stakeholders	15
	5f User Participation.....	16
	5g Priorities Assigned to Users	16
6	Mandated Constraints	16
	6a Solution Constraints	16
	6b Implementation Environment of the Current System	17

6c	Partner or Collaborative Applications	17
6d	Off-the-Shelf Software	17
6e	Anticipated Workplace Environment	17
6f	Schedule Constraints	18
6g	Budget Constraints	18
7	Naming Conventions and Definitions	18
7a	Definitions of Key Terms	18
7b	UML and Other Notation Used in This Document	19
7c	Data Dictionary for Any Included Models	19
8	Relevant Facts and Assumptions	19
8a	Facts	19
8b	Assumptions	19
II	Requirements.....	21
9	Product Use Cases.....	21
9a	Use Case Diagrams.....	21
9b	Product Use Case List.....	22
9c	Individual Product Use Case.....	22
10	Functional Requirements.....	28
11	Data Requirements.....	29
12	Performance Requirements.....	30
12a	Speed and Latency Requirements.....	30
12b	Precision or Accuracy Requirements.....	31
12c	Capacity Requirements.....	31
13	Dependability Requirements.....	31
13a	Reliability Requirements.....	31
13b	Availability Requirements.....	32
13c	Robustness or Fault-Tolerance Requirements.....	33

13d	Safety-Critical Requirements.....	33
14	Maintainability Requirements.....	33
14a	Maintenance Requirements.....	33
14b	Supportability Requirements.....	34
14c	Adaptability Requirements.....	34
14d	Scalability or Extensibility Requirements.....	35
14e	Longevity Requirements.....	35
15	Security Requirements.....	35
15a	Access Requirements.....	35
15b	Integrity Requirements	36
15c	Privacy Requirements.....	37
15d	Audit Requirements.....	38
15e	Immunity Requirements	38
16	Usability and Humanity Requirements.....	39
16a	Ease of Use Requirements.....	39
16b	Personalization and Internationalization Requirements	40
16c	Learning Requirements	40
16d	Understandability and Politeness Requirements	41
16e	Accessibility Requirements	41
16f	User Documentation Requirements.....	42
16g	Training Requirements	42
17	Look and Feel Requirements	43
17a	Appearance Requirements.....	43
17b	Style Requirements.....	43
18	Operational and Environmental Requirements.....	44
18a	Expected Physical Environment.....	44
18b	Requirements for Interfacing with Adjacent Systems.....	44
18c	Productization Requirements.....	44
18d	Release Requirements	44
19	Cultural and Political Requirements	45
19a	Cultural Requirements.....	45
19b	Political Requirements.....	45

20	Legal Requirements	46
20a	Compliance Requirements.....	46
20b	Standards Requirements	47
21	Requirements Acceptance Tests	47
21a	Requirements – Test Correspondence Summary	47
21b	Acceptance Test Descriptions	48
III	Design	56
22	Design Goals	56
23	Current System Design	56
24	Proposed System Design	56
24a	Initial System Analysis and Class Identification.....	56
24b	Dynamic Modelling of Use-Cases.....	57
24c	Proposed System Architecture	58
24d	Initial Subsystem Decomposition.....	58
25	Additional Design Considerations	59
25a	Hardware / Software Mapping	59
25b	Persistent Data Management	60
25c	Access Control and Security	60
25d	Global Software Control.....	60
25e	Boundary Conditions.....	60
25f	User Interface	61
25g	Application of Design Patterns.....	62
26	Final System Design	63
27	Object Design	64
27a	Packages	64
27b	Subsystem I	64
27c	Subsystem II	65
IV	Project Issues	66
28	Open Issues	66
29	Off-the-Shelf Solutions	66
29a	Ready-Made Products	66
29b	Reusable Components	66
29c	Products That Can Be Copied	66

30	New Problems.....	67
30a	Effects on the Current Environment.....	67
30b	Effects on the Installed Systems.....	67
30c	Potential User Problems	67
30d	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	67
30e	Follow-Up Problems	68
31	Migration to the New Product	68
31a	Requirements for Migration to the New Product	68
31b	Data That Has to Be Modified or Translated for the New System	68
32	Risks	68
33	Costs	68
34	Waiting Room.....	69
35	Ideas for Solutions	70
36	Project Retrospective	70
V	Glossary	70
VI	References / Bibliography	71
VII	Index	72

I Project Description

Conservatory Catch is a mobile application that uses augmented reality to create an educational game promoting interest in the Garfield Park Conservatory and its nonprofit organization, the Garfield Park Conservatory Alliance (GPCA). The game is played while at the conservatory, which promotes increased visits, new visitors, and more donations.

1 Project Overview

In Conservatory Catch, players will be able to walk around the conservatory in real time alongside their character and have random chances of different species appearing. When a species appears, the player has the ability to catch it by playing a minigame, thus building their collection. Players can view their collection to view information on all of their collected species. The end goal of the game is to complete your collection.

2 The Purpose of the Project

The Garfield Park Conservatory programming does not extend into the world of mobile applications. With Conservatory Catch, the GPCA would be able to offer a fresh new way of educating visitors on the different plant species found in the conservatory. The hope is that the game will lead to an increased number of visitors, and an increased number of donations. Donations to the conservatory will greatly contribute to the different programs that the GPCA runs, as well as accessibility (i.e., free programs) to those programs for the youth.

2a The User Business or Background of the Project Effort

The clients of this project are the GPCA, people who like to learn about plants, and gamers. Much of what goes on at the Garfield Park Conservatory runs through the GPCA. This includes education programming, events, and creating resources for visitors. For gamers and people who like to learn, the project serves an exciting new medium for learning about plant species.

2b Goals of the Project

This project serves as a way to enhance and broaden the operations of the GPCA through encouraging donations and visits. While the game is free to play, we hope many players enjoy the game enough to where they feel the conservatory is worthy of a donation. For gamers and those learning about plant species, we want to provide a fun and accessible way of combining augmented reality gaming with learning.

2c Measurement

We will be measuring the increase in donations on a particular cycle, such as week to week. Increase in donations means two different particular measurements. One being the total amount of money donated to the conservatory, and the other being the frequency of donations regardless of the amount. These two measurements will allow us to gauge changes in donations starting from before the app's launch, to well after.

Players will be able to measure their own success via their collection level. Each new species collected will increase their collection level by one point. We will also track feedback through user surveys, where we can also inquire about the game's accessibility

3 The Scope of the Work

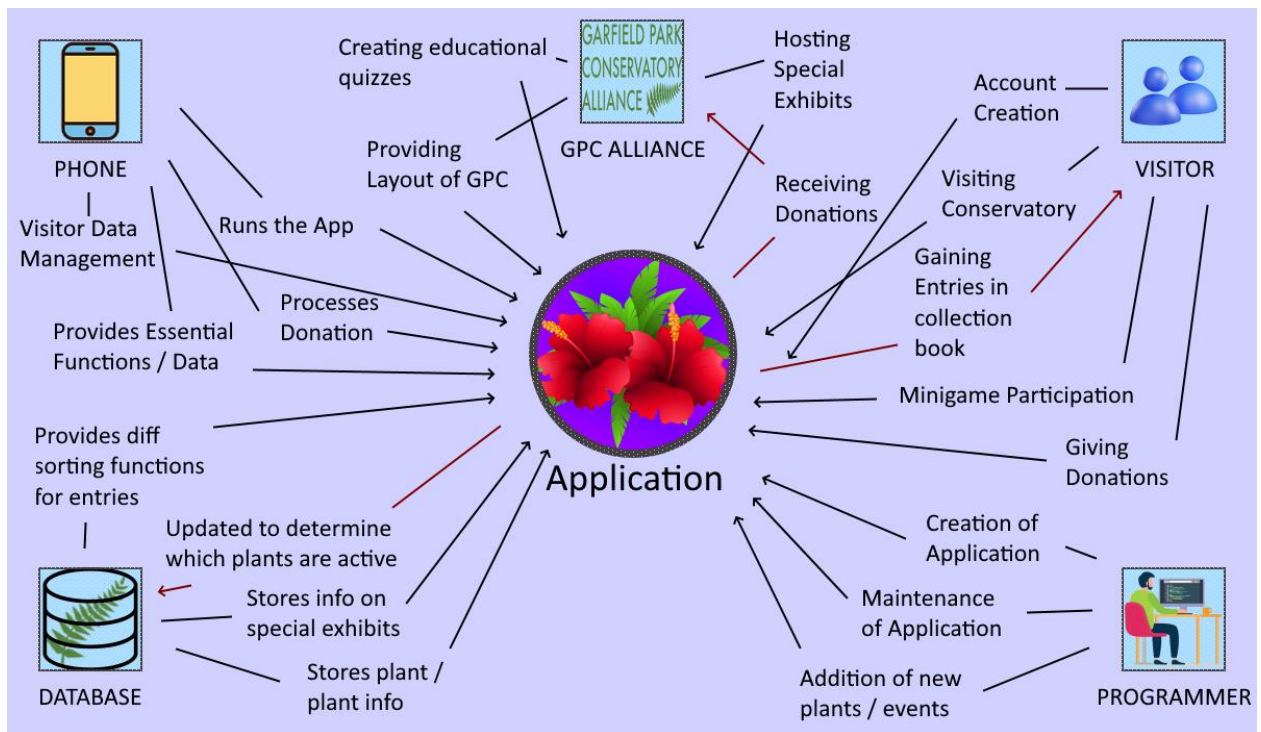
This product will work inside the work's business, that being the GPCA's vision and mission, that being wanting to change people's lives through the power of nature, and that they hope to achieve this by showing them the importance of nature. They use the GPC as their main work, and provide educational programming, events, and resources in order to further achieve their goal.

3a The Current Situation

Garfield Park Conservatory Alliance goes about achieving their goal by providing educational programming, events, and resources to GPC visitors. These programs can vary, taking place at different, specific times, as well as location, and some of them are available for free, while some of them require a fee on the participator's end. Schools are also able to take their class on field trips to the conservatory for free.

The application will serve as another way to further achieve their goal, providing a means to extend the education to every tourist coming to GPC in a fun, interactive manner for free, and at any time in the GPC. It will also help fund future events through donations.

3b The Context of the Work



3c Work Partitioning

Business Event List

Event Name	Input and Output	Summary
<u>GPC Alliance</u>		
1. GPC Alliance creates educational quizzes	Creating Education Quizzes (in)	Quizzes will be implemented by programmers in minigames
2. GPC Alliance gives the layout of the conservatory	Providing Layout of GPC (in)	Layout of conservatory used by programmers in creating app
3. GPC Alliance plans and hosts special exhibits	Hosting Special Exhibits (in)	Special exhibits tied to special events in the application
4. GPC Alliance receives donations from visitors	Receiving Donations (out)	Donations to GPCA received by visitors through the app
<u>Visitor</u>		
	Account Creation (in)	

1. Future visitor of GPC create an account on the application 2. Said visitor goes and visits the GPC with account created 3. Visitor participates in the minigames on application 4. Visitor gains entries in their collection book on application 5. Visitor gives a donation through the app to the GPCA	Visiting Conservatory (in) Minigame Participation (in) Gaining Entries in Collection Book (out) Giving Donations (in)	Creation of account by visitor allows access to app functions Visitor visits conservatory to use app and gain entries Minigames are how visitors gain entries in collection book Entries can show visitor more info on plant caught +fill book Donation given because either enjoy app or + cosmetic item
<u>Programmer</u> 1. Programmer works on creating the application 2. Programmer maintains the application after completion 3. Programmer adds new content to application	Creation of Application (in) Maintenance of Application (in) Addition of New Plants / Events (in)	Creation of application core to being able to use its functions Will need to handle visitor data, ties to the database Addition through new plants and special exhibits, new data
<u>Database</u> 1. Database stores info on GPC's plants 2. Database also stores info on GPC's special exhibits 3. Database gets updated for active plants in conservatory 4. Database provides different sorting functions for entries in collection	Stores Plant / Plant Info (in) Stores Info on Special Exhibits (in) Updated to Determine Which Plants are Active (out) Provides Different Sorting Functions for Entries (in)	Keeps info on GPC plants + can add future ones Info such as runtime, when they happen, special plants, et. Some plants may be seasonal, and some may be added Uses database functions and params to sort the plants, be it by time, if special, other info
<u>Phone</u> 1. Phone manages the visitor's data	Visitor Data Management (in)	Phone deals with data such as social profile and entries GPS and camera are vital for making the application work

2. Phone provides essential functions, such as GPS and camera 3. Phone has the necessary software to run the app 4. Phone uses the visitor's inputs to process and send donation	Provides Essential Functions / Data (in) Runs the App (in) Processes Donations (in)	Software required to run the app Donation given to GPCA if they like the game or just want an extra cosmetic for profile
--	---	---

3d Competing Products

Competing products do exist, such as the ones listed below, but they do not combine fun, interactions, and availability into one.

- Books on plants
- GPC's events
- Courses on plants

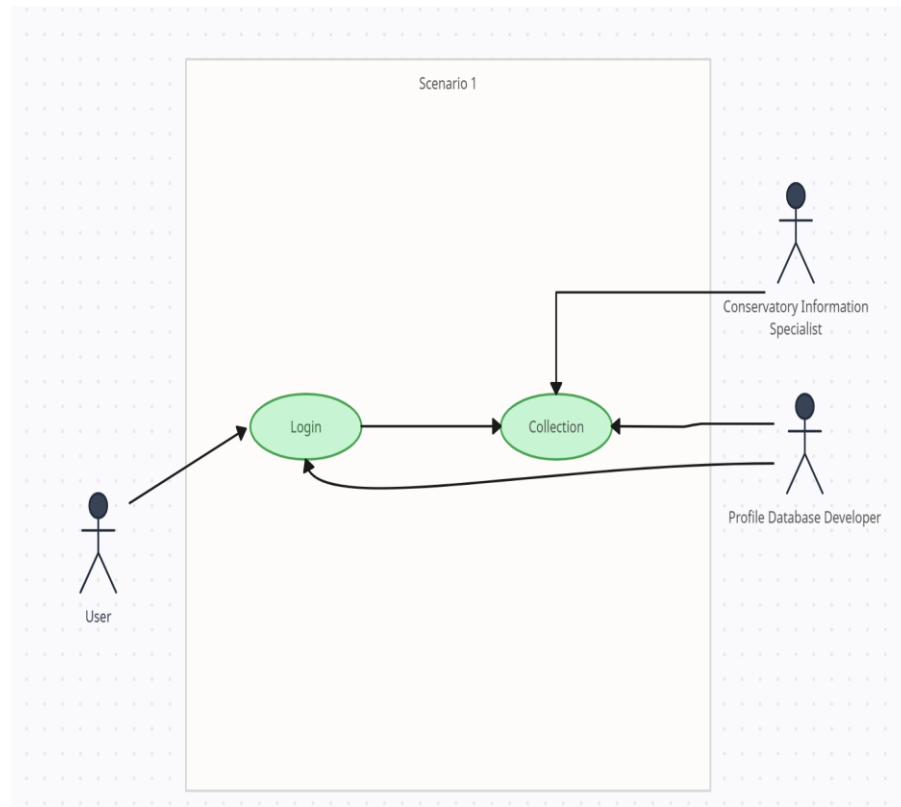
4 The Scope of the Product

This project will allow the following actions to be performed by users:

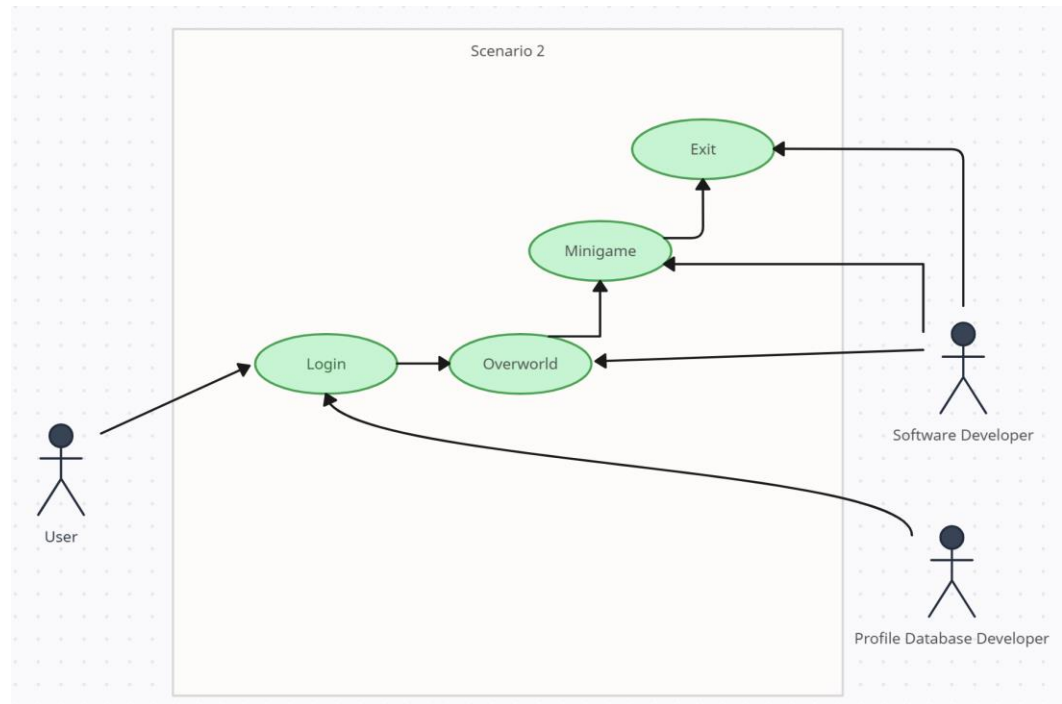
1. "Login" – allows new players to create accounts and returning players to login to their respective accounts in order to retain all previously earned rewards.
2. "Overworld" – allows players to physically walk through a one-to-one mapping of their environment, using their location data provided by their smart devices, and complete different reoccurring events and challenges in the form of minigames.
3. "Minigames" – allows users to perform specific tasks in order to complete a variety of events spawned in from the overworld, their reward being a new addition to their growing collection of rewards.
4. "Collection" – allows the user to access any rewards gained from the minigames completed in the overworld and retains all completions specific to the currently logged in user.
5. "Exit" – allows the user at anytime to exit the app without losing important data

4a Scenario Diagram(s)

Scenario 1 (User logs in and checks their current collection):



Scenario 2 (User logs in, finds an event in the overworld but exits before completing the game):



4b Product Scenario List

Scenario Names	Scenario Number	Next Scenario	External Actors
Overworld Scene	1	2	
Minigames Scene	2	3	Conservatory Specialist
Collection Scene	3	4	Conservatory Specialist
Login Scene	4	1	

4c Individual Product Scenarios

1. **Overworld Scene** – Our user, Ryan, enters the conservatory with the app up and running, his location is displayed to him in the form of his real life location casted over a digital map on his phone. He gets to walk around in real life and encounter random events that trigger minigames.
2. **Minigames Scene** – Ryan encounters a random event in the overworld, he taps it. A randomly generated minigame is displayed to Ryan for him to complete. Depending on the results, he may be rewarded with an item for his collection.
3. **Collection Scene** – Ryan, after successfully completing his minigame, enters his collection to view his newly acquired item.
4. **Login Scene** – Ryan, after seeing his newly acquired item, logs out for the day knowing that his data will be saved to his specific login so he may return and continue hunting for more rewards.

5 Stakeholders

5a The Client

- The Garfield Park Conservatory Alliance (GPCA) would be the main benefactor for this application, as this would allow for a different avenue for repeat visitors, leading to a higher potential for donations. A monetization shop can also be introduced down the line, leading to even more avenues for donations.

5b The Customer

Potential customers range from:

- Those who enjoy Augmented Reality style games
- Those who enjoy learning about exotic species
- Those who enjoy collectathon games

5c Hands-On Users of the Product

Hands-on Users of this product will most likely be:

- Conservatory visitors, ages range from children to elderly. It is necessary to have easy to understand UI elements and minigames that are widely accessible to all ages due to novice technology knowledge depending on the visitor.
- New and untrained Conservatory Employees. Assumption that this app can be used as training device as well, for employees to practice their trivia and learn new things they may not be learning on immediately. Assumption that if they

are working for the conservatory then their technology prowess and intellect is at the bare minimum, high school level

5d Maintenance Users and Service Technicians

Maintenance users:

1. Debugging Maintenance Users
 - i. These users will be tasked with debugging potential errors that may occur when updating the app with new content
2. New Content Users
 - i. These users will take client feedback and provide updates for the app in terms of new content (new minigames, different versions of displaying the overworld), and quality of life updates (easier to understand UI design)

5e Other Stakeholders

Potential Stakeholders may include:

- Conservatory Workers: the workers themselves may be impacted both negatively and positively. This app as specified above, could be used to help train new employees, and also help guide guests through the conservatory with information they may not find out via the guides placed there. Yet, if the app does such a good job of guiding users, workers could be phased out entirely.
- Testers: With new content, there will need to be extensive testing concerning each and every new update. Testers would need to understand how to push the game to its limits in terms of not only minigames but also the overworld aspect of physically walking around the map. This would require not only extensive knowledge on programming to determine bugs in the UI/Minigames themselves, but also physical ability to walk around the area and mimic the true user experience.
- Technology Experts: The use of technology cannot be understated in this project. Not every person has the same phone. Technology Experts must be tasked with ensuring that the app will be made available for all devices, regardless of being on IOS or Android.
- Usability Experts: For an app to be truly available for all, usability experts will need to work hand in hand with programmers to determine different how to approach a much broader user experience that has the ability to be played by anyone no matter the physical disability/impairment.
- Exotic Species Experts: This app is meant to help teach users about exotic plant species located in the conservatory, and around the world. If the data isn't

accurate, it would fail to be a benefit to the conservatory and its guests. Exotic Species Experts must be on hand when creating rewards, trivia, and minigames in order to legitimize the content we wish to reward players with. Exotic Species Experts must also maintain a high degree of knowledge so that we ensure this knowledge is passed on to the user.

5f User Participation

Users we be expected to attend weekly meetings where they will update the team on the completion of weekly objectives that pertain to content updates and bug fixes. If potential app breaking bugs pop up, it is expected that all teams will work to fix these bugs immediately before anything else.

5g Priorities Assigned to Users

Key Users: Conservatory workers and the conservatory itself will take highest priority in terms of what is required. It will essentially become their tool to help visitors navigate exotic species, so they must have the highest priority in how that tool takes shape.

Secondary Users: The visitors themselves, specifically repeat visitors, will be the 2nd highest on the food chain, but since users may want things that exit the realm of what the product is meant to do, they cannot come before the key users.

Unimportant Users: Non-visitors that download the app to try will have no priority in terms of what becomes requirements for the app. The whole point of the app is to stimulate visitation, so users that can't meet that requirement won't have a say in any requirement.

6 Mandated Constraints

6a Solution Constraints

Constraint 1:

- Description: This product will be available on mobile devices
- Rationale: The augmented reality style of game we are aiming for will require easy transportation methods in order to convey the overworld style to the user. Mobile devices will allow users to walk around the map physically.
- Fit Criterion: The product will be easily accessible via mobile devices, allowing all users to effectively participate as opposed to being constrained by bulkier technology such as desktop computers.

Constraint 2:

- Description: This product will be available on a mobile platforms (IOS, Android, etc)
- Rationale: If all mobile devices is a goal that must be achieved, then it is imperative that all mobile development platforms contain a version of the application
- Fit Criterion: The product will be available for all devices, not gating off certain financial brackets that may not be able to afford the newest hardware from the most well-known brands.

6b Implementation Environment of the Current System

Conservatory Catch will be developed for mobile platforms using Flutter Framework, this will ensure development for both operating systems used in mobile platforms. Phones will be used to gather location data and project users onto the digital space within the app.

6c Partner or Collaborative Applications

Conservatory Catch, as stated in 6b, will be developed using Flutter Framework as it's primary coding software. Weekly meetings as explained in section 5f, will use Microsoft Office products to present data, such as Word for word documents, and PowerPoint for presentations.

6d Off-the-Shelf Software

Flutter Framework by Google will be an External COTS software used to develop the app.

6e Anticipated Workplace Environment

The product will be developed as a mobile app so the main environment will be a phone app, used in the conservatory.

The workplace can be noisy at times, depending on how many visitors show up on a given day, so audible based minigames may not work as well depending on the atmosphere that day.

The user will be navigating a potentially congested environment so there must be steps implemented to maintain ensure events don't spawn in specifics areas that are dangerous to enter or causes potential health concerns by multiple users piling up to access said event.

6f Schedule Constraints

Product development should expect to take 3 years, with rigorous testing and client feedback, this deadline would allow for time to ensure maximum device reach, allowing the client to encourage the maximum amount of visitor retention. Monetization aspects should be expected to be implemented 6 months after launch, after a player base can be nurtured and donations for the client can begin to be collected digitally.

6g Budget Constraints

This product is expected to be rather low budget in terms of game development. Considering the fact that this is being made for a non-profit organization, it shouldn't be expected to have hundreds of millions of dollars at disposal. At most, the budget would be estimated to be around 4 million dollars, covering expenses of the development team, and required testing.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Game Mechanics Keywords

Overworld: This will be the digital representation of the users physical location in the real world. Here the user will be able to interact with other elements of the game

Events: randomly appearing events will populate the overworld, allowing users to initiate different minigames

Minigames: randomly generated sets of games, ranging from encounters to trivia

Minigame Keywords

Encounters: two types of encounters will be used within minigames, one being a battle encounter, the other being a capture encounter

Species: species will be the primary focus of all trivia and encounters. These will be exotic plant species that users can see/learn about at the conservatory.

Trivia: not just any trivia will be displayed in minigames. This trivia must pertain to the exotic species the player encounters in the overworld events.

7b UML and Other Notation Used in This Document

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in "UML Distilled, 3rd Edition"[4]. Any exceptions are noted where used.

7c Data Dictionary for Any Included Models

Species rewards from minigames will be stored in a class structure known as the collection class, which will have a primary data member in the form of a map structure, where the key will be the species ID number, and the value will be the trivia/image associated with said species. This collection object will be paired with the player accounts player ID that will then be combined into another map object that will use the player ID as the key, and the collection object as the value. This will be used to store the required data used to identify player logins and accounts.

8 Relevant Facts and Assumptions

8a Facts

The application will be made for Android phones.

The application will be coded in Flutter.

The application will be up and running and maintained for about 2-4 years, depending on GPCA's willingness.

GPCA and the programmers maintaining the application will be communicating with each other at least once a day after the program has been created

8b Assumptions

Any and every visitor who plans on using the app is assumed to be able to read English at a middle-school grade level.

Every visitor who plans on using the app should have a phone that has the necessary components, such as GPS tracking, a touchscreen, and a camera, in order for the application to use to carry out some of its functions. This will also require permission from said visitor access to these functions so the application can utilize them.

Every visitor who plans on using the app should have a phone that is recent enough to have the necessary software to be able to run the app.

Every visitor who plans on using the app should have enough available storage space to download the app.

Every visitor needs to have the app running while they are at the conservatory, and should be able to hold their phone and utilize their touchscreen efficiently enough to properly use the application.

The Garfield Park Conservatory Alliance will provide to us the layout of the conservatory, so that we may use it for GPS purposes, plan out possible spots for users to engage with the application's minigames.

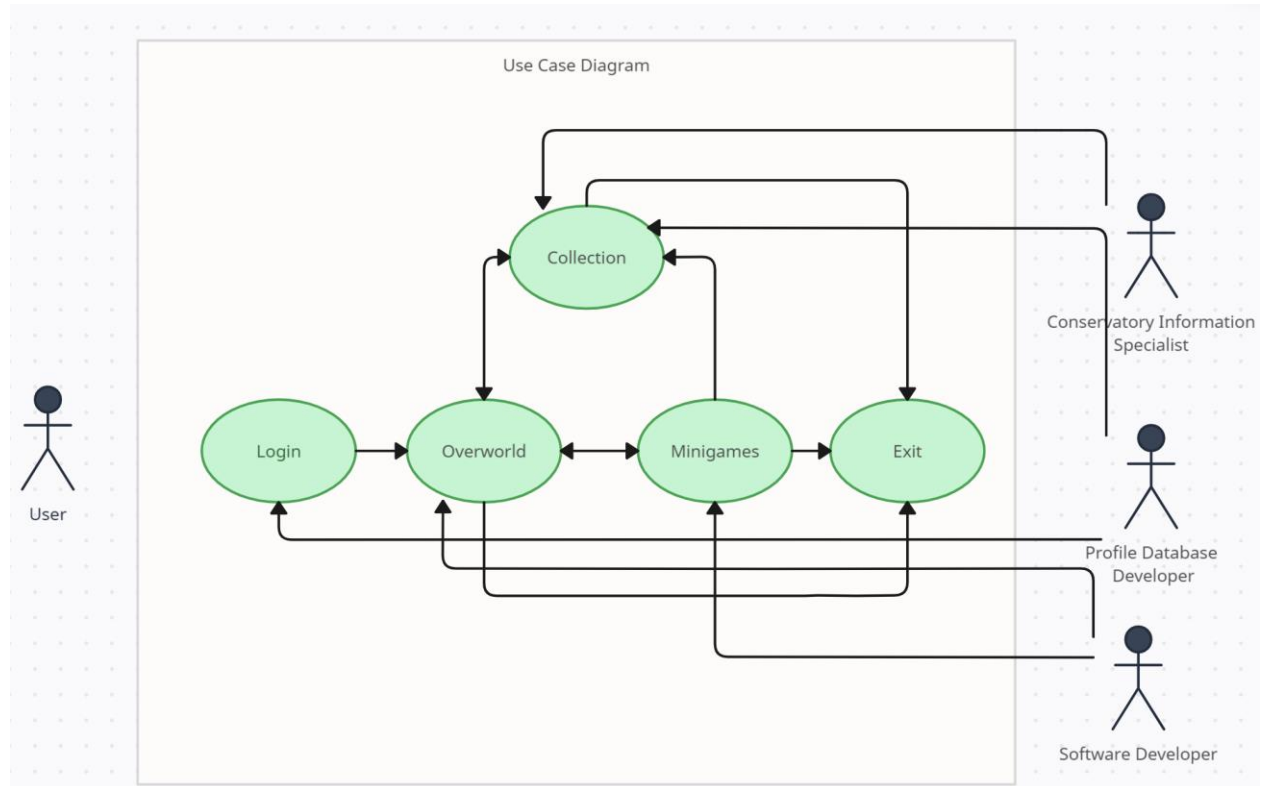
The Garfield Park Conservatory Alliance will provide to us educational quizzes for every plant that they want logged into the database, as well as information on special exhibits and information on special plants that will be logged into the database for that specific exhibit.

The database will be kept up and running, and it will be visitor's responsibility to remember their login information, such as their username and password.

II Requirements

9 Product Use Cases

9a Use Case Diagrams



9b Product Use Case List

<u>Use Cases</u>
Login
Overworld
Minigames
Collection

9c Individual Product Use Cases

<p>Use case ID: CCL Name: Login Case</p> <p>pre-conditions: (1) User must have an account created.</p> <p>post-conditions: (1) Once a user logs in, they will be allowed to access their collection and interact with the overworld.</p> <p>(2) If a user cannot log in, they are prompted to make an account or recover a lost password</p> <p>Initiated by: User initiating app startup.</p> <p>Triggering Event: User prompting app to login with user credentials</p> <p>Additional Actors: An mobile cellular service provider is required as the user will be expected to connect to app servers to verify credentials via the user profile database</p>
<p>Sequence of Events:</p> <p>1. As the player chooses to open the application.</p> <ul style="list-style-type: none">• Player is greeted with login screen requesting credentials tied to their account

2. The player will then enter their credentials

- The system will cross reference the entered credentials with the profile database holding all player credentials.
- If a user's entered credentials do not match any credentials the system will not allow the user to login until proper credentials are entered
- When correct credentials are supplied, the user will enter the game, starting on the overworld scene.

Alternatives: Users that don't have an account will be directed to how to create one. Users with accounts that cannot remember their credentials will be allowed to enter a recovery process.

Exceptions: Users without an internet connection will not be allowed to log in nor will they be allowed to create any account, since without internet it would be impossible to track user movement on the overworld. Users that have previously logged in will be allowed to enable a saved log in, and skip the login process entirely.

Use case ID: CCO

Name: Overworld Case

pre-conditions: (1) User has logged in with proper credentials.

(2) User has completed a minigame.

(3) User has returned from visiting their collection

post-conditions: (1) User has entered a minigame via an overworld event.

(2) User has entered their personal collection.

(3) User has exited the application.

Initiated by: User interacting with an augmented reality map displaying their IRL coordinates onto a map.

Triggering Event: User will initiate returning to overworld from other scenes like Login, Minigames, and Collection scenes.

Additional Actors: The user must maintain a stable internet connection, via a mobile service provider, in order to accurately update their location on the overworld map.

Sequence of Events:

1. The user has entered the world from the log-in screen.
 - The overworld receives the user's GPS coordinates and plots them on a 1 to 1 map of their location, much like Google would when showing your location on Google Maps.
 - Once the location is updated on the map, the overworld is populated with random events, that when interacted with will trigger a minigame.
 - A button will also appear in a convenient area of the UI, allowing the user to enter their personal collection of items received from minigames.
2. The user interacts with a randomly generated event displayed to them as they walk through the real world.
 - The user will be transported to another scene that will display the minigame that has been chosen for them to play, leaving the overworld behind until the user can finish the encounter.
3. The user returns from the minigame and wishes to enter their collection. They press the collection button while on the overworld screen.
 - The user will be transported to the collection scene, which will display all rewards earned from minigames/events/encounters.
 - The user may return to the overworld and have their location again updated, to accurately reflect any changes while in other scenes.
4. The user has finished playing for the day, the user exits the games.
 - Depending on how the user has set up their account log-in, the application will either exit and not store their saved credentials, leading to them to have to reenter them upon opening the application again, or the game will have their credentials saved. Skipping the log-in process.

Alternatives: Users may be allowed to interact with events on the overworld map, simply by clicking on said events as they generate, bypassing the need to physically walk through the map.

Exceptions: In the case of users that wish to experience the app but health problems won't allow them to do so, clearance will be given to allow location updating to be simulated via

button presses on the map, as opposed to having to physically walk to new locations that they cannot feasibly access in their condition.

Use case ID: CCC

Name: Collection Case

pre-conditions: (1) A user has a registered account.

(2) A user has completed at least 1 minigame

post-conditions: (1) User will be allowed to view their collection of rewards.

(2) User will be allowed to return to the Overworld in order to continue playing.

(3) User will be allowed to exit the application.

Initiated by: The user wishes to see the rewards they have collected through minigames/events from the Overworld.

Triggering Event: User clicks on the button labeled “Collection” that appears on the Overworld.

Additional Actors: Profile database developers will be in charge of maintaining the profile systems, allowing smooth interaction between log-in and entering their respective collections. They will also be tasked with allowing for user’s collections to accurately display their rewards, maintaining all progress made throughout their lifetimes using the application.

Sequence of Events:

1. The user has recently finished a minigame and wishes to interact with whatever reward they may have received, so they press the button labeled “Collection” on the Overworld map.

- Upon button press, the user’s rewards will be displayed back to them within a different widget, this widget will double check with the profile data to make sure the user is seeing only what he has unlocked so far.
- The rewards themselves will be able to be interacted with, examined, and trivia pertaining to said rewards will also be displayed to the user.

2. The user, after interacting with their rewards, chooses to either return to Overworld, or exit the application.

- Upon returning to the Overworld, the application will pull the user’s GPS location once again, in order to update the map with the location they are now at. Since the user may have changed location since entering the Collection.
- Exiting will work the same as exiting the Overworld Case/Scenario

Alternatives: Down the line, monetization aspects may be implemented in order to help boost donations. Some aspects of monetization that are being considered are allowing users to pay to gain rewards they may have missed out from time specific events.

Exceptions: N/A

Use case ID: CCM

Name: Minigames

pre-conditions: (1) User must have interacted with an event on the Overworld map

(2) User must have a reliable internet connection, as some minigames may require precise server/client-side actions

post-conditions: (1) Once a minigame is completed, the user may exit the application, or return to the Overworld with their rewards.

(2) If the user cannot complete the minigame, the user may exit or return to the Overworld but without the rewards from post-condition (1)

Initiated by: The user sees an event on their Overworld screen and wishes to interact with it.

Triggering Event: The user enters the event through on screen pressing, or physically walking through said event on the Overworld map.

Additional Actors: N/A

Sequence of Events:

1. The user sees an event on their Overworld screen and presses the event to enter the encounter/minigame.

- The Overworld scene changes to whatever randomly generated minigame for that area is pulled from the random scene generator.
- The minigame propagates itself to the user and allows for much deeper interaction between the user and the application, depending on the minigame that is chosen.

2. The user completes the encounter/minigame and a reward is displayed to the user.

- Upon completion, a reward is chosen from a list of potential awards, some locked behind specific minigames, and the rewards are then sent to the player's collection which is saved to the player's profile permanently.

Alternatives: As mentioned in the Alternatives section of the Collections Use Case, monetization aspects lean themselves to allowing for more retries, or even tools that will help conquer minigames that some users cannot complete normally.

Exceptions: N/A

10 Functional Requirements

ID#: F1 – **Name:** Smooth Scene Transition

Description: The application will have a smooth transition back and forth between different scenes

Rationale: Many scenes will be swapped out while playing the game. Scenes like the Overworld will need to seamlessly transition to random minigames, and even to the player's collections. Smooth transitions will be important so that the user experience is not diluted.

Fit Criterion: Transitions must have an obvious flow to them, they cannot have harsh fast swapping of scenes so as long as the presentation doesn't mimic a power point presentation while swapping, it should be fine.

Acceptance Tests: 1

ID#: F2 – **Name:** Player reward tracking

Description: Player rewards will be accurately tracked upon completion/failure of minigames.

Rationale: Users must trust that actions they perform will be backed by rewards or else player retention could fall through due to unreliable reward systems

Fit Criterion: Complete minigames and ensure that rewards are being accurately updated within the collections. Also ensure that players are not rewarded for failing to finish minigames.

Acceptance Tests: 2, 3

ID#: F3 – **Name:** User Location Accurately Updated

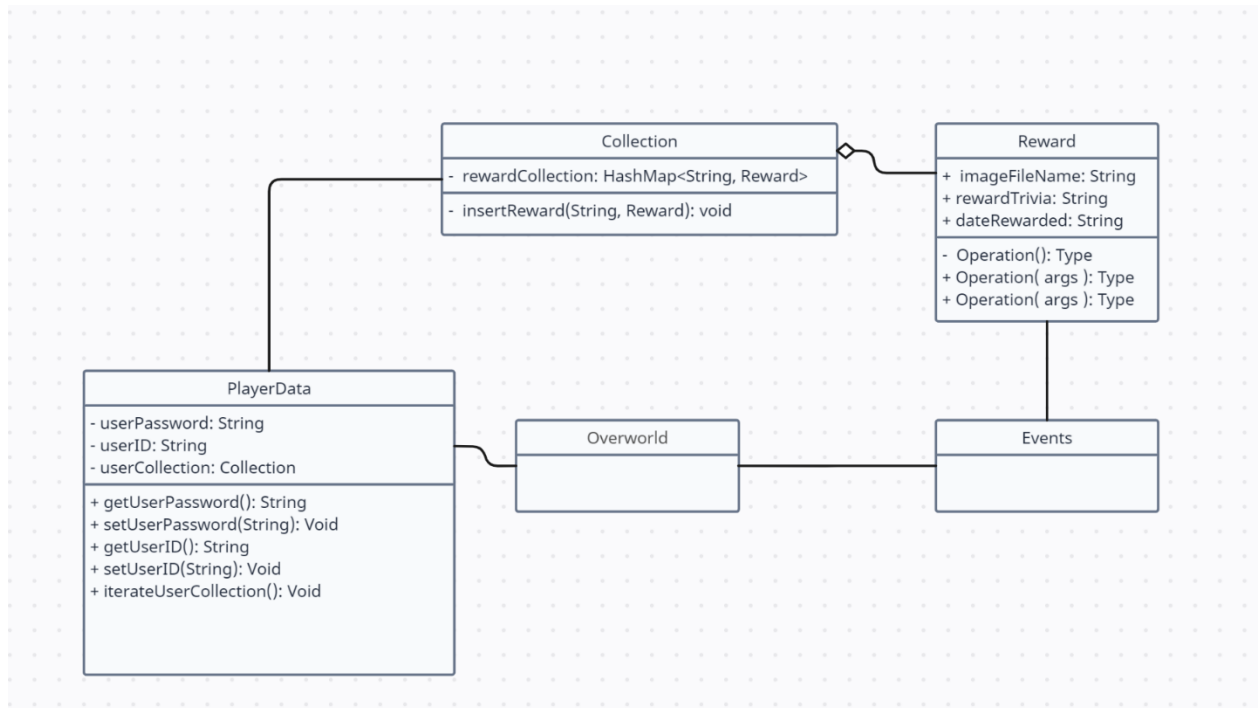
Description: User location on the Overworld is accurate to their real-world location

Rationale: One of the main hooks in this game is that it is an Augmented Reality style game, and the feature that takes advantage of that is the Overworld map. If this isn't working correctly then there wouldn't be anything to separate this game from other regular games.

Fit Criterion: Display the location data of a user for testing, if this matches their GPS coordinates then it should mean that the location tracking of the mobile devices is accurately being read into the application

Acceptance Tests: 4

11 Data Requirements



ID#: D1 – **Name:** Reward Naming Format

Description: Rewards are expected to follow a coherent and similar naming format

Rationale: For both Users and Programmers, having a naming format that is clear and concise will allow for a much easier ease of application when trying to differentiate between all the potential rewards, ensuring there aren't any potential duplicates within the HashMap for the Collection

Fit Criterion: Naming formats must use the name of the species being rewarded and include nothing else. If duplicate species exist, follow species name with a sub identifier.

Acceptance Tests: 5, 6

ID#: D2 – **Name:** Unique User IDs

Description: User IDs are expected to be unique to each user.

Rationale: ID log-ins are expected to differ from each other user in order to accurately track each user's data. Repeat names may allow for confusion when the systems checks for valid credentials.

Fit Criterion: 2 ways to accomplish this task, allow for all users to use any names, but specify their login with a unique identifier that trails their given username IE: nick#123. Or don't allow repeat usernames.

Acceptance Tests: 7

12 Performance Requirements

12a Speed and Latency Requirements

ID#: S1 – **Name:** Latency

Description: Latency must not exceed 100 ms

Rationale: In a game with potential to have minigames that rely on effective timing, it would be incredibly detrimental to the player if the latency allowed for unresponsiveness, leading to high levels of player frustration, which would lead to low levels of player retention.

Fit Criterion: Attempt different actions in specific minigames, if any actions don't provide a fast enough response, throw a warning and add to bug report.

Acceptance Tests: 8

12b Precision or Accuracy Requirements

ID#: A1 – **Name:** Actions Accurately Interpreted

Description: Actions performed by the user, primarily touch screen presses, are accurately interpreted by the application

Rationale: Since this game will primarily be played on mobile devices with touch screens, it is imperative that the users actions track correctly when pressing the device.

Fit Criterion: Test user actions via touch screen presses for all actions, thoroughly.

Acceptance Tests: Test 66

12c Capacity Requirements

ID#: C1 – **Name:** Number of Users

Description: Product must be able to handle multiple users playing at once

Rationale: Since this game will primarily be played at the Garfield Park Conservatory, it will be expected to handle many different users stressing out the servers and systems at the same time

Fit Criterion: Rush times would primarily exist between opening at 10 AM – 2 PM, allowing for up to 500 users to be playing at once, in case of school trips and other educational trips that visit the conservatory. Later implementation would include being able to play outside the conservatory, so a higher user limit would need to be achieved to allow for full day/night coverage.

Acceptance Tests: 9

13 Dependability Requirements

13a Reliability Requirements

ID#: R1 – **Name:** Crash Allowance Rate

Description: Application must not allow frequent crashes

Rationale: This game is meant to be a learning tool for users that visit the Garfield Park Conservatory, if it crashed all the time, it would severely hamper the user experience not just concerning the app, but also the experience within the park, as with constant unreliable crashing, visitors would disrupt others in an attempt to find a solution.

Fit Criterion: System failures should be limited to 1 crash an hour, if not 1 crash every few hours of being open. Ideally crashing should be limited to 1 time every 24 hours of game play (the application being opened and used)

Acceptance Tests: 10

ID#: R2 – **Name:** Player Reward Retention

Description: On crashing, any rewards gained from the user completing an event, must be retained on crashing during transitions from minigames to Overworld.

Rationale: Considering the higher likelihood of the application crashing during transitions between minigames and the Overworld, it should be expected that any rewards that were gained prior to crashing will still be retained post-crash.

Fit Criterion: The application should effectively store all player data, prior to scene transitions.

Acceptance Tests: 11

13b Availability Requirements

ID#: A2 – **Name:** 24/7 server availability

Description: Users should be free to visit the Overworld map, and interact with their collections, whether it is during visiting hours or at the comfort of their home.

Rationale: Users should feel a sense of pride in what they have, they may want to view some of their hardest earned rewards while they are at home, or even with friends. 24 hours server availability will allow users to openly access all the data they earned, at any time.

Fit Criterion: Check server availability at all times, alerting developers to any outages that may occur. Peak times will tend to lead to higher server congestion and potentially server failure.

Acceptance Tests: 12, 13

13c Robustness or Fault-Tolerance Requirements

ID#: RF1 – **Name:** Offline Collection Browsing

Description: Users that opt to save their log-in credentials to their specific device, will be allowed to view their collection while servers are offline.

Rationale: Collection browsing is a major selling point to user retention, allowing offline saves for users that opt-in, will allow users to browse their rewards even if the servers are down.

Fit Criterion: Add an offline save of a user's collection, but upon coming back online, only show the user their online collection, and update their offline save back to the online version. This allows for bypassing of potential user exploits, changing their offline collection values somehow.

Acceptance Tests: 14, 15

13d Safety-Critical Requirements

ID#: SC1 – **Name:** Epileptic-free Animations/Transitions

Description: Ensure that animations and transitions are not prone to triggering epileptic seizure

Rationale: Epilepsy is something that can be triggered by intense flashing in media, since this game should be open to everyone, it must consider the chance of provoking this condition.

Fit Criterion: Keep flashing at or below 3 seconds [6], as any prolonged flashing could lead to higher chances of seizures.

Acceptance Tests: 16

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

ID#: M1 – **Name:** Server Maintenance

Description: Ensure weekly if not daily server maintenance.

Rationale: Server stability is a major aspect to the application running smoothly, so weekly checkups should maintain any problems that may occur

Fit Criterion: Weekly server maintenance checkups to ensure any server congestion/slowdowns are dealt with.

Acceptance Tests: 17

ID#: M2 – **Name:** Broken Maintenance

Description: Ensure any game breaking bugs that crop up are dealt with at the highest priority

Rationale: Game breaking bugs occur all the time. This could lead to user retention falling off a cliff if those bugs aren't fixed fast enough.

Fit Criterion: Run user submitted bug reports, prioritize fixing the most game breaking bugs that appear during the applications lifecycle.

Acceptance Tests: 18, 68

14b Supportability Requirements

ID#: SUP1 – **Name:** Bug Reporting

Description: Enable user-based bug reporting

Rationale: Not everything can be found through Q&A. Allowing users to report specific bugs and the steps they took to reproduce them, allows developers to recreate bugs based on a much higher population of testers

Fit Criterion: Enable a button that submits user-based bug reports into a queue that the development team has access to.

Acceptance Tests: 19

14c Adaptability Requirements

ID#: Mob1 – **Name:** Mobile Device Porting

Description: The game will be made to work on mobile devices, including Android/Apple products.

Rationale: This is expected to be a game that users can easily walk around the conservatory and play. Mobile devices allow this and allowing both main platforms, Android/Apple, to house the application will allow a majority of users to play it.

Fit Criterion: This game is being made with the intention of being played on phones, build a working phone model first, then branch out to mobile devices like tablets.

Acceptance Tests: 20

14d Scalability or Extensibility Requirements

ID#: Pop1 – **Name:** External User Retention

Description: Allow for a broad range of population to access the application, even outside the expected Garfield Park Conservatory population.

Rationale: With the potential for this game to expand outward and help other animal/plant conservation efforts, it is important that we build the game with that in mind, so we can scale the servers and game to allow for a much higher population.

Fit Criterion: When stress testing servers and application aspects, test and build with the idea that one day thousands of users could be playing this. Ensure smooth and bug free multithreading throughout the process of development.

Acceptance Tests: 21

14e Longevity Requirements

ID#: L1 – **Name:** Lifetime

Description: This application is expected to last as long as the Garfield Park Conservatory exists.

Rationale: This is meant to be a specific tool to help the Garfield Park Conservatory, so this won't be something that dies off after 5 months as some sort of special promotion. This is being built with the intention to last.

Fit Criterion: Develop a fundamentally sound product, with the intention to last decades, and feasibility to scale to newer software. This means regularly updating and maintaining the code base.

Acceptance Tests: 22

15 Security Requirements

15a Access Requirements

ID# Sec-1 – Name: Code Security

Description: Users and conservatory staff should not be able to alter the game code in any way.

Rationale: Users should only be allowed to play the game, not change aspects of it through changing code. Staff should not need access to the game code.

Fit Criterion: Code should only be altered by the game developers.

Acceptance Tests: 23

ID# Sec-2 – Name: Location Data

Description: Users are generally only able to access their own location data, unless they are in a party, where they can view their party members' locations. Staff will have access to users' location data so long as they are in the park.

Rationale: Users can only view their party's locations, but under no circumstances need to be able to view the location of non-party users. Staff will have access to locations of users to maintain general safety.

Fit Criterion: Users can view their own location data, and only staff can view all user's location data,

Acceptance Tests: 24

ID# Sec-3 – Name: Personal Data

Description: Users can only view and edit their own personal data, such as name, email, and date of birth. Staff can only view the names and email addresses of users.

Rationale: Users have no need to see the personal information of other users. Staff may view email addresses and names for communication purposes.

Fit Criterion: Users can view and edit their own personal data, and staff can only view names and email addresses of users.

Acceptance Tests: 25, 26

15b Integrity Requirements

ID# Sec-4 – Name: Database Integrity Protection

Description: Plant data will only ever come from the database, to which only staff and developers have full access to. Users can only view data from their collections, no other permissions.

Rationale: Plant data needs to remain accurate and credible. Therefore, users can only view data. Staff and developers can make changes to the database as needed.

Fit Criterion: Only staff and developers are granted full access to the database.

Acceptance Tests: 27, 28

ID# Sec-5 – Name: Data Loss Prevention

Description: Account data and the database itself will be backed up on a separate private server in the event that current systems are unusable or corrupted. Only developers will have access to the backup.

Rationale: In the event that an update corrupts the servers, or incomplete changes are pushed, or servers are physically damaged, the backup will allow for quick and easy game restoration, with minimal loss.

Fit Criterion: Only developers have access to the backup.

Acceptance Tests: 29, 30

15c Privacy Requirements

ID# Sec-6 – Name: Data Collection Notification

Description: Users will be notified of what data will be collected prior to account creation. Users will be made aware that sensitive data is not shared with anyone, and only names and email addresses will be shared with staff.

Rationale: Users need to know what information will be collected and shared.

Fit Criterion: Sensitive personal data aside from location, will not be shared with anyone.

Acceptance Tests: 31

ID# Sec-7 – Name: In-Game Purchases Privacy

Description: Payment details of users will not be made available to anyone. Payment will be handled through the Google Play Store, and the Apple App Store, thus no credit card information will be stored. Only personal transaction dates and amounts will be viewable to the user, staff, and developers.

Rationale: Maintain security of users' financial data.

Fit Criterion: Only Apple and Google will have access to the credit card information of users.

Acceptance Tests: 32, 33, 34

15d Audit Requirements

ID# Sec-8 – Name: Transaction History

Description: Payment is handled externally, but logs of transaction dates and amounts will be kept for auditing purposes.

Rationale: Keep track of financial information.

Fit Criterion: Users will be able to view their own transaction history. Developers can view the transaction history of all users.

Acceptance Tests: 34

15e Immunity Requirements

ID# Sec-9 – Name: Manipulation Protection

Description: Only authorized developers will be able to alter game code. In the event that the system gets infected, the system will be reverted to the backup server.

Rationale: The program needs to be protected from potential threats.

Fit Criterion: Only authorized developers are able to alter game code. Devs can also enable the backup server in case of infection.

Acceptance Tests:

16 Usability and Humanity Requirements

16a Ease of Use Requirements

ID# Use -1- Name: Accessibility Options

Description: Developers must consider the needs of users with visual impairments, auditory impairments, motor impairments, and cognitive impairments. To do so, special accessibility options will be made available in the game settings.

Rationale: The game needs to be accessible so that all visitors of the conservatory can play without difficulty.

Fit Criterion:

General Criterion:

- Anonymous survey shall that 98 percent of users feel as though they are adequately able to enjoy the game given the accessibility options.

Visual Impairment Accommodations:

- Adjust game colors for different types of color blindness.
- Text-to-speech option
- Font size option
- Adjust brightness and contrast.

Auditory Impairment Accommodations:

- Audio description option for sound effects

Motor Impairment Accommodations:

- Ability to turn off specific minigames that require swiping and tapping.
- Enable voice command to play trivia minigames.
- Enable voice navigation.

Cognitive Impairment Accommodations:

- Disable AR, instead use 2d background to collect plant species.
- Ability to enable shorter and less flashier animations.
- Easy menu navigation
- Music and sound slider

Acceptance Tests: 35, 36, 37, 38, 39, 40, 41, 42, 43, 44

16b Personalization and Internationalization Requirements

ID# Use -2 – Name: Language Option

Description: Users will have the ability to change the language of the game text.

Rationale: Keep the game accessible to as many people as possible.

Fit Criterion: User will be able to select a language from a list of supported languages.

Acceptance Tests: 67

ID# Use -3 – Name: Measurement Option

Description: Users will have the ability to choose either imperial or metric as the in-game measurement system.

Rationale: Allow users further customization to ensure better experience.

Fit Criterion: User will be able to select imperial or metric system.

Acceptance Tests: 45

16c Learning Requirements

ID# Use -4- Name: Education

Description: All players shall be given the opportunity to learn about different plant species.

Rationale: One of the main goals of the program is to teach visitors about plant species. We want the learning experience to be simple and effective.

Fit Criterion:

- Include pictures and text to convey information.
- Aim for around middle school level reading.
- Keep descriptions concise.
- Provide short summaries where necessary.

Acceptance Tests: 46

16d Understandability and Politeness Requirements

ID# Use -5- Name: Dictionary

Description: An in-game dictionary shall be available for basic plant definitions.

Rationale: Allow users with zero prior knowledge to still participate in the game.

Fit Criterion: A list of words in bold text will be accompanied by their respective definitions.

Acceptance Tests: 47

16e Accessibility Requirements

ID# Use -1- Name: Accessibility Options

Description: Developers must consider the needs of users with visual impairments, auditory impairments, motor impairments, and cognitive impairments. To do so, special accessibility options will be made available in the game settings.

Rationale: The game needs to be accessible so that all visitors of the conservatory can play without difficulty.

Fit Criterion:

General Criterion:

- Anonymous survey shall that 98 percent of users feel as though they are adequately able to enjoy the game given the accessibility options.

Visual Impairment Accommodations:

- Adjust game colors for different types of color blindness.
- Text-to-speech option
- Font size option
- Adjust brightness and contrast.

Auditory Impairment Accommodations:

- Audio description option for sound effects

Motor Impairment Accommodations:

- Ability to turn off specific minigames that require swiping and tapping.
- Enable voice command to play trivia minigames.
- Enable voice navigation.

Cognitive Impairment Accommodations:

- Disable AR, instead use 2d background to collect plant species.
- Ability to enable shorter and less flashier animations.
- Easy menu navigation
- Music and sound slider

Acceptance Tests: 35, 36, 37, 38, 39, 40, 41, 42, 43, 44

16f User Documentation Requirements

ID# Use -6- Name: Game Manual

Description: An in-game manual will be provided to teach users how to play the game.

Rationale: Users should have a reference as to how the game is played, how the game mechanics work, and how to navigate the game.

Fit Criterion: The manual shall be created by the developers. Upkeep on the document will be maintained by the developers.

Acceptance Tests: 48

16g Training Requirements

ID# Use -7- Name: Tutorial

Description: After account creation, users will be put into a tutorial sequence which will teach them the basic mechanics of the game.

Rationale: To teach players the basic mechanics of the game.

Fit Criterion: The tutorial will be created by the game developers. Users will automatically experience the tutorial after completing account creation. The tutorial can be revisited at any time.

Acceptance Tests: 49, 50

17 Look and Feel Requirements

17a Appearance Requirements

ID# LF -1- Name: Color Palette

Description: The game shall utilize a nature inspired color palette.

Rationale: A nature inspired color palette fits in with the setting that the game is played in.

Fit Criterion: The client shall approve of the color palette, or potential changes.

Acceptance Tests: 51

ID# LF -2- Name: Fonts and logos

Description: The game shall utilize fonts and logos that are simple yet engaging.

Rationale: Create a welcoming and fun look and feel for the game.

Fit Criterion: The client shall approve of the all fonts and logos used.

Acceptance Tests: 52

17b Style Requirements

ID# LF -3- Name: Style

Description: The presentation of the game shall look fun, welcoming, and engaging.

Rationale: Create a welcoming environment where anyone can play. Maintain users by simultaneously providing a fun environment.

Fit Criterion: After the user's first time playing, at least 60 percent shall have played for at least 75 percent of their trip.

Acceptance Tests: 53

18 Operational and Environmental Requirements

18a Expected Physical Environment

ID# OE-1- Name: Environment

Description: The game shall only be playable at the conservatory.

Rationale: The product is meant to promote visits to the conservatory; therefore, it is only playable at the conservatory.

Fit Criterion: The product can only be used as intended at the conservatory. Outside of the conservatory, no plant species can be caught.

Acceptance Tests: 54

18b Requirements for Interfacing with Adjacent Systems

The product runs independently without the need for partner applications.

18c Productization Requirements

ID# OE -2- Name: Installation

Description: The product shall be distributed on the Google Play and Apple App Store. These will be the only supported methods of installation.

Rationale: Keep the installation easy. The process will be identical to any other mobile app.

Fit Criterion: Nearly all users are able to install the app without issue.

Acceptance Tests: 55

18d Release Requirements

ID# OE -3- Name: Game Updates

Description: The product shall receive bug fixes post-launch, as well as seasonal updates that correspond to new exhibits at the conservatory.

Rationale: Keep players engaged by providing a bug-free experience, and more content.

Fit Criterion: Fewer bugs are reported. Player counts stay consistent.

Acceptance Tests: 22

19 Cultural and Political Requirements

19a Cultural Requirements

ID# CR -1- Name: Content Sensitivity

Description: The product content and imagery shall not include elements that could be offensive, derogatory, culturally insensitive to any religious, ethnic, or social groups.

Rationale: This requirement ensures that the content of the product remains inclusive and respectful of diverse cultural backgrounds, promoting a positive user experience for everyone.

Fit Criterion: The product shall be reviewed by a diverse group of people to ensure that nothing offensive is contained in the game.

Acceptance Tests: 56, 57

ID# CR -2- Name: Cultural Awareness

Description: The product shall incorporate features or information that demonstrate an awareness and consideration of various cultural norms, practices, and holidays.

Rationale: This requirement aims to enhance user engagement by recognizing and respecting the diversity of cultural practices among potential users.

Fit Criterion: The product shall provide features such as the option to view content in multiple languages, display relevant cultural holidays relevant to the Conservatory, and avoid scheduling notifications or events on significant cultural observance days.

Acceptance Tests: 58

19b Political Requirements

ID# PR -1- Name: Compliance with Organizational Politics

Description: The product shall comply with any internal political directives or preferences set forth by key stakeholders within the client company.

Rationale: This requirement recognizes that internal political dynamics within the client organization may impact product development decisions.

Fit Criterion: Documented evidence of compliance with internal political directives shall be provided, demonstrating adherence to specific organizational preferences.

Acceptance Tests: 59, 60

20 Legal Requirements

20a Compliance Requirements

ID# CR -1- Name: Data Protection Compliance

Description: Personal information processing within the product shall adhere to the regulations outlined in the Data Protection Act, ensuring the privacy and security of individuals' data.

Rationale: This requirement addresses legal obligations regarding the handling of personal information, safeguarding against potential legal repercussions and protecting individuals' privacy rights.

Fit Criterion: Legal opinion provided by the company's legal department confirming compliance with the Data Protection Act.

Acceptance Tests: 61

ID# CR -2- Name: Intellectual Property Protection

Description: The product shall incorporate measures to protect copyrights and other intellectual property rights, preventing infringement and safeguarding the company's proprietary assets.

Rationale: This requirement ensures the protection of intellectual property, reducing the risk of legal disputes and safeguarding the company's creative and innovative work.

Fit Criterion: Legal opinion from the company's legal department confirming the implementation of appropriate measures for intellectual property protection.

Acceptance Tests: 62, 63

20b Standards Requirements

ID# SR -1- Name: Insurance Industry Standards

Description: The product shall adhere to the standards set forth by the insurance industry, ensuring compatibility and compliance with industry-specific requirements.

Rationale: This requirement ensures that the product meets the criteria established by the insurance industry, reducing the risk of compatibility issues in regard to mapping the Conservatory and ensuring acceptance within the sector.

Fit Criterion: Certification from the designated authority affirming adherence to insurance industry standards.

Acceptance Tests: 64, 65

21 Requirements Acceptance Tests

21 a Requirements – Test Correspondence Summary

Test	Requirements																			
	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10	Req 11	Req 12	Req 13	Req 14	Req 15	Req 16	Req 17	Req 18	Req 19	Req 20
Test 1	X																			
Test 2		X				X														
Test 3			X	X																
Test 4					X	X														
Test 5																				
Test 6																				
Test 7																				
Test 8																				
Test 9																				
Test 10																				
Test 11																				
Test 12																				
Test 13																				
Test 14																				
Test 15																				

Table 1 - Requirements - Acceptance Tests Correspondence

21b Acceptance Test Descriptions

Test #1

Description: Check for all scene transitions, make sure they're at least 0.5 seconds long or greater, accompanied by an animation that leads to the next screen.

Test #2

Description: minigameReward function checked to only reward players once minigamePassed is true, and not at other points of the minigame.

Test #3

Description: minigameReward function checked to give reward to players after minigame has been completed, no matter what happens afterwards, such as game closing, sudden server crash, etc.

Test #4

Description: compare location of user to GPS coordinates, done for multiple user + GPS coordination points, checked to see they match within a certain threshold.

Test #5

Description: Pass all existing names into a function, if any duplicates are found in said function, throw a duplicateName error.

Test #6

Description: Pass all existing names and species into a function. If a name is found that doesn't use a species name, throw a noSpeciesName error.

Test #7

Description: accountCreation function tested with a list of usernames. accountCreation function should never allow users to create an account with a username that already exists in the list.

Test #8

Description: Perform actions in the minigames (or force the game to if not possible) that would cause it to exceed 100 ms. Make sure warning goes off 100% of the time, as well as the bug report.

Test #9

Description: Have 500+ instances of the app running at once, with the servers being able to handle requests from each of those instances, ranging from minigame events, checking social profile, as well as payments.

Test #10

Description: Have 500+ instances of the app running at once, with the servers being able to handle requests from each of those instances, with the amount of crash reports being below 1 per hour per instance

Test #11

Description: Instance of reward created upon the start of the minigame, considered valid upon completion of minigame, and checked to see it given to the player upon booting up the game and connecting to the servers once more. Multiple tests with (ideally) 0% error rate.

Test #12

Description: serverDownNotifier should notify whenever server is down, checked with 1000 instances, ranging from when server is down and expected to notify vs when server is not down and expected to not notify.

Test #13

Description: App checked to see if able to pull user data from anywhere with a connection.

Test #14

Description: Check instances of user saves stored on phone and save on servers (except in instances of crash on minigame completion). Should always be the same save, throw error if not and set save to one stored in server.

Test #15

Description: Check instances of app upon booting up with no access to server, should still be able to access their user data, specifically collection.

Test #16

Description: Check for all animations and transitions longer than 3 seconds and scan them to see if they contain any sudden shifts in color or brightness (determined by a function). Throw an error if found and replace or rework on animation / transition that threw the error.

Test #17

Description: Check serverMaintenance function, set it to a weekly time and ensure it goes off in order to perform the routine checkups and shut off the servers if needed.

Test #18

Description: Pass all user reports through a filter that enables them to be listed via amount of reports as well as severity (checked via keywords). If needed, shut down the server in order to quickly patch these bugs.

Test #19

Description: Check bugReport function to make sure users are able to submit a bug report, as well as the button related to attached function is able to lead users to the bug report page.

Test #20

Description: Create an APK and an IPA file to be tested on an Android phone and a iPhone, respectively. Ensure the application is able to run and be interacted with on both phones.

Test #21

Description: Checks that app supports multithreading throughout the process of development as well as the release version of the product to ensure it supports it afterwards.

Test #22

Description: Checks that app is able to support updates and patches down the line.

Test #23

Description: Checks that code is unable to be directly accessed by the user, barring the ability to edit them.

Test #24

Description: Users will only be able to access their own data, and only able to view the public profile of other users. They will not be able to access private information of other users, such as their location and date of birth.

Test #25

Description: userInfo function checked to see if it allows the accessing it to edit their own information, and not someone else's information. It will also check to see if the user can only view their own information.

Test #26

Description: Check to see if staff are allowed to view the information of every user in the database.

Test #27

Description: Check to see if staff and developers can edit and pass changes to the database.

Test #28

Description: Check to see if users are unable to gain access to the database.

Test #29

Description: backupData function checked to see if it backs up the data from the main servers to the separate server, about once a week.

Test #30

Description: Check to see that developers and no one else are able to access the backup server.

Test #31

Description: privacyNotification function checked to see if it runs on every instance of an account being created. Happens every time someone signs up for a new account.

Test #32

Description: Pass the sensitive information through the stores and make sure none of the sensitive information gets stored in our servers.

Test #33

Description: transactionSave function checked to see if it works, which only takes the amount paid and the date the payment was made and saves it to the server.

Test #34

Description: transactionView function checked to see if it works, allowing users to only view their own transactions while developers are allowed to view everyone's.

Test #35

Description: check colorBlindness function is able to work and complies with the standards.

Test #36

Description: implement subtitles and check to see if the subtitles function works (closed captions appended to it if needed).

Test #37

Description: ability to change font size in settings works.

Test #38

Description: ability to change contrast and brightness in settings works.

Test #39

Description: ability to navigate scenes, do some minigames with them (trivia), combination of them via voice in settings works.

Test #40

Description: ability to turn off minigames that require swiping and tapping in settings works.

Test #41

Description: Swapping animations function activated in settings, which changes the animations to be shorter and less flashier, works.

Test #42

Description: Ability to change volume for specific things, such as music and sound effects in settings, works.

Test #43

Description: Ability to make the background in AR 2D via settings works.

Test #44

Description: Ability to swap the menus to one more easily navigable in settings works.

Test #45

Description: Function that changes measurements from imperial to metric and vice-versa, check to see if it works (changes the value as well as the accompanying text).

Test #46

Description: Check all information given via the collection and make sure it is simple and easy to digest.

Test #47

Description: wordDefinition function checked to see if words in bold, when clicked on, provide a little popup that gives a definition of said word. Test with different words.

Test #48

Description: manual function checked to see if it works (leads the user to a manual that teaches them how to play), as well as the button attached to said manual function leads them to the manual.

Test #49

Description: tutorial checked to see if it launches every time a new user creates their account, as well as saving their point in the tutorial in case the user leaves in the middle of it.

Test #50

Description: tutorialButton checked to see if it leads back to the tutorial, being able to be done any amount of times.

Test #51

Description: all images and colors used in the application will be passed through a function, checking to see if at least 90% of each one match a nature-inspired color palette, otherwise ask the client for approval of said palette.

Test #52

Description: Fonts must be approved by the client.

Test #53

Description: Checks the amount of time users had the app open and checks the amount of time users spent in the Garfield Park Conservatory. If about 60% of them have a ratio of time in GPC and time in app of 0.75 or higher, pass.

Test #54

Description: minigame function checked to see if it works only at the conservatory and nowhere else. Alternatively, checks to see if it works at the conservatory to begin with, then we can see about not making it work elsewhere.

Test #55

Description: Check to see if at least 95% of users are able to install the app without issue from the Google Play and Apple App stores.

Test #56

Description: Have a list of words that cannot be added to the game and pass all the text of the game through a function. Check to see if any of those words are in, and if so, remove them.

Test #57

Description: Conduct a demo of the product to a diverse group and take feedback on anything they see problematic in regard to sensitive content.

Test #58

Description: culturalHolidays function checked to see if it works on specific days of the year where it coincides with special events at the Conservatory.

Test #59

Description: Provide documentation of adherence to internal political directives.

Test #60

Description: Conduct stakeholder interviews to confirm alignment with organizational politics.

Test #61

Description: Validate that personal information handling processes align with the requirements of the Data Protection Act and document a legal opinion affirming compliance with relevant data protection regulations.

Test #62

Description: Verify the presence and effectiveness of measures implemented to protect copyrights and intellectual property.

Test #63

Description: Obtain and document a legal opinion affirming compliance with intellectual property protection requirements.

Test #64

Description: Verify that the product aligns with the standards specified by the insurance industry.

Test #65

Description: Obtain and document certification confirming compliance with insurance industry standards.

Test #66

Description: Function that handles touch screen presses coupled with current scene, area where screen is expected to be touched, and expected next scene or event. Multiple trials conducted with a 0% expected error rate.

Test #67

Description: Function that allows users to changes settings via setting, as well as upon initial bootup of the application

Test #68

Description: check patchRelease function, ensure that it's able to detect whenever a new patch is out and prevent the user from being able to interact with the application until it is downloaded.

III Design

22 Design Goals

Our main design goals align with 3 main quality of life features for ongoing and future content in the game.

1. Seamless integration of new minigame types
2. Fast and accurate storage of newly collected rewards into a Player's collection
3. Reliable/Consistent representation of players on Overworld map

23 Current Design System

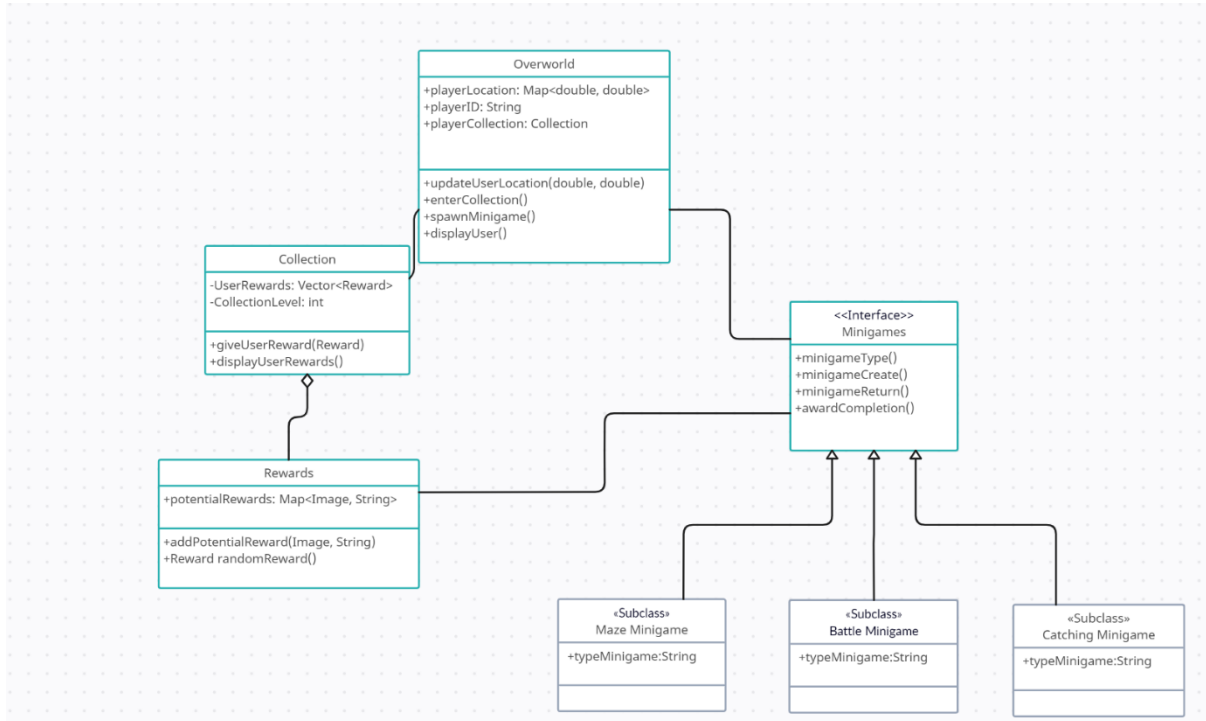
There is no pre-existing system.

24 Proposed Design System

24a Initial System Analysis and Class Identification

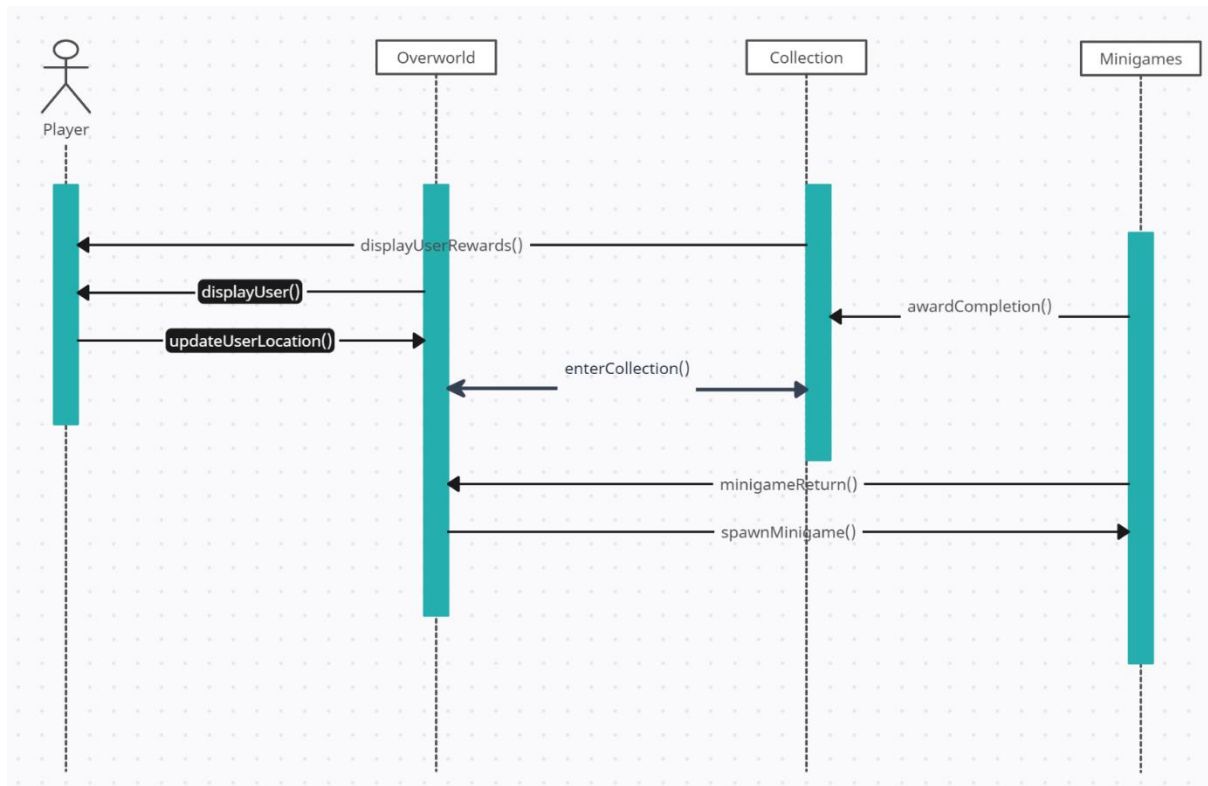
The following class diagrams is an example of what to expect the main Overworld system to look like. The player will be able to traverse the overworld map, and interact with minigame events that randomly pop up, the given minigame being produced from the Minigames superclass, and being returned to the Overworld class. A reward class will exist, with its main priority being that it will produce rewards for the player to obtain via the minigames played. These rewards will be passed into the Collection

class, which will hold all collection data for each player. The Collection can be accessed via the Overworld map.



24b Dynamic Modeling of Use-Cases

Below is a sequence diagram of the player playing Conservatory Catch use-case, which is essentially all the potential use cases described in section 9. Essentially, the player is logged into their account and the player's info is given to the Overworld class via their phone's location tracking system. This information is passed into the `updateUserLocation()`, and the user icon is displayed back to the user using the `displayUser()` function that takes the newly acquired user location. Once the user is displayed via the Overworld map, they can enter and exit their collection, when the user's rewards are displayed to them via the `Collection` class function, `displayUserRewards()`. Now another potential path from the Overworld map, is the potential minigames that are spawned and enter via the user's screen presses and the `spawnMinigame()` function. Once the user completes the minigame, on successful completion, the `awardCompletion()` function will award the user with a reward, storing it instantly in the user's `Collection`. The user is then returned to the Overworld via the `minigameReturn()` function, which will call `updateUserLocation()` from within in order to accurately track any movements made while in their minigame instance.



24c Proposed System Architecture

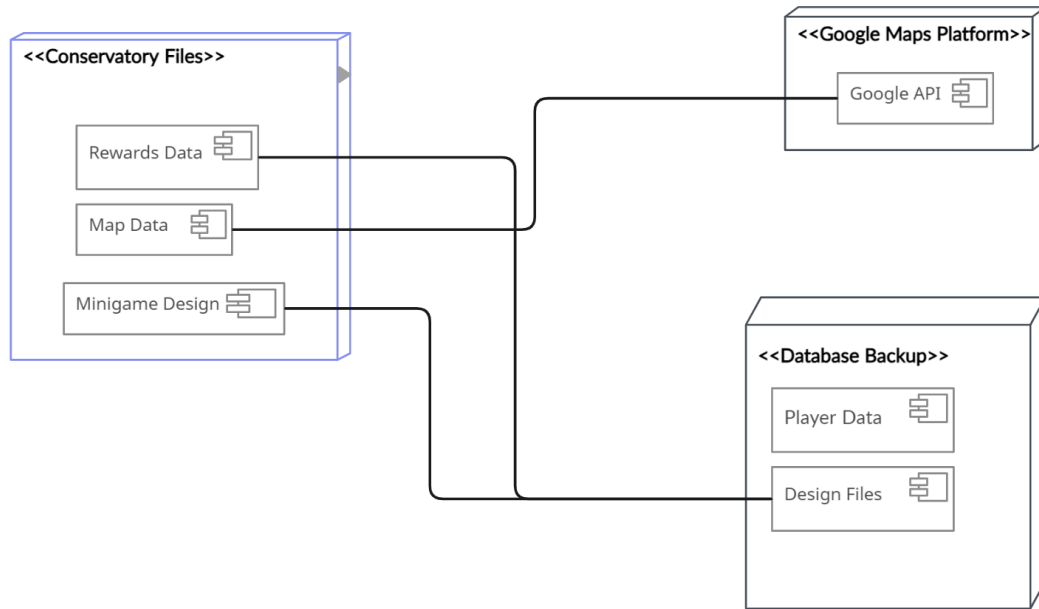
The system architecture for Conservatory Catch will be the Client-Server architecture, since this project will constantly need to display constantly changing data between users and the server-side Conservatory map. With potential for displaying users to all other users also playing at the same time, everyone must be able to reliably track all client instances on the main server.

24d Initial Subsystem Decomposition

Subsystems will be split into the main 3 subsystems seen in 24b's sequence diagram. Overworld will be in charge of displaying current user location on the map, while allowing for instances of minigames to propagate the map and allowing a user's collection to be accessed. The Collection subsystem will be in charge of all player rewards storage. Minigames subsystem will only be in charge of creating minigame instances to pass into the Overworld and passing back new rewards to the Collection subsystem.

25 Additional Design Considerations

25a Hardware / Software Mapping



This Hardware/Software architecture diagram displays the different systems in play when holding all the possible data for the program. Rewards, map data, and Minigames will be stored in the main Conservatory File system, while rewards and minigames will also be backed up in the database backup system in case the Conservatory files become corrupted. Map data will constantly be retrieved using the Google Maps Platform [5] which uses the Google API to access user location data for the map display. Player data will be all usernames and passwords attached to said usernames, this will be stored in the database backup server.

25b Persistent Data Management

Player data management will be very important in terms of persistent data management. To achieve this, player data will need to be backed up by multiple Database instances, to ensure that any potential problems that occur won't result in permanent data loss for players. In terms of map data, that won't be able to be kept current, since if Google API goes down, there won't be any potential for the user's location to be updated.

25c Access Control and Security

Player data cannot, at any time, be compromised in terms of privacy and integrity. Player data like passwords must be protected at all times, so a player password's must be private data objects at all times, and only be retrieved by specific classes, nothing more than the required calls. A class can be created that creates an encryption method, that passes these passwords into an encrypting method, that hides the user's password to anyone that doesn't know the encryption method.

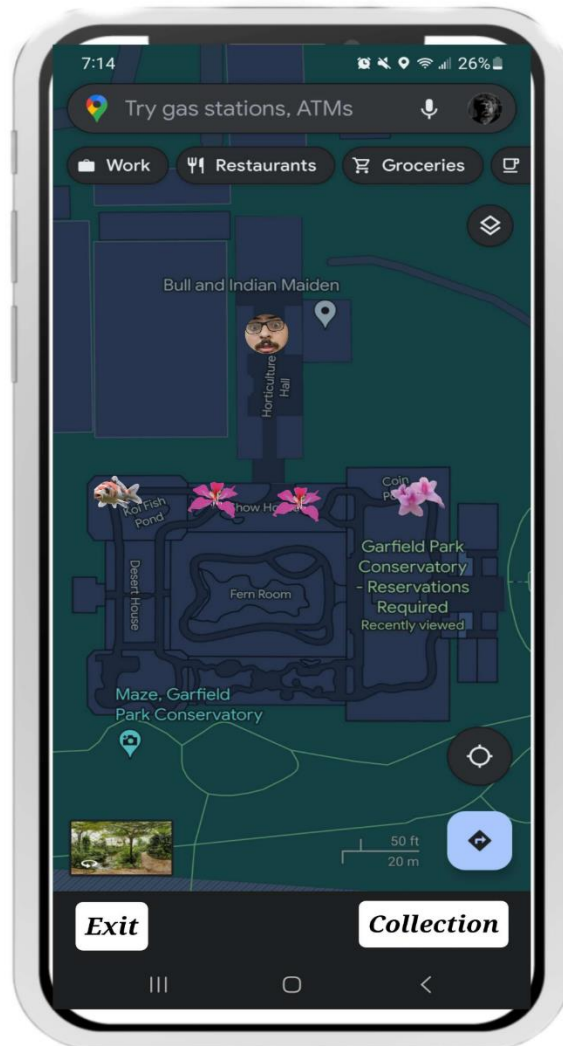
25d Global Software Control

This product will be controlled by user inputs on their phone screens, ensuring all appropriate elements that can be tapped for actions work will be the only important control aspect.

25e Boundary Conditions

One main concern for an edge-case is ensuring rewards are correctly distributed, and users are not given repeat rewards, unless rewards in the future include incentivization of multiples of the same rewards being obtained. In the meantime, a class or method will need to be created to ensure that the given rewards don't exist yet in the given user's collection. Reward creation will also need to ensure that rewards aren't accidentally created using the same strings/images for multiple rewards. Each reward should be unique so that the user knows their time is valued when trying to collect all rewards.

25f User Interface

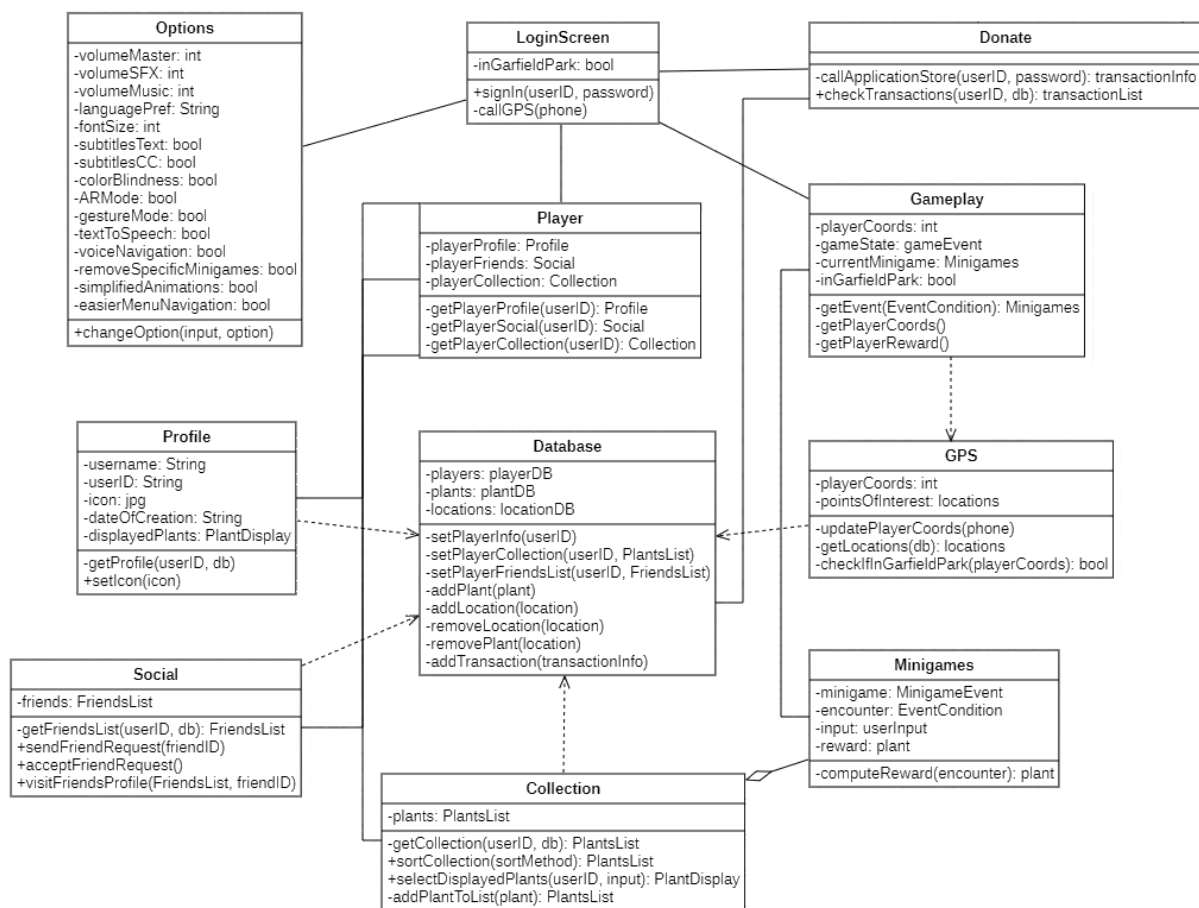


Here is a mock design of the game in action, specifically the overworld section. Prior to displaying multiple players, this version will only display the user which is denoted by the icon under the Indian Maiden location. This would be updated as the user moves around the conservatory. The different icons that exist along the path are all minigame events that the player can click on and trigger the minigame. The Collection is accessed on the bottom right button.

25g Application of Design Patterns

Factory Method: Minigames, and potentially rewards, will be created using the factory method design pattern. This is important so we can have different types of minigames, and different types of rewards that can generally follow the format shown in the main factory method interfaces/abstract super classes.

26 Final System Design



The figure above showcases the final design of our project, getting the important components of it into one UML diagram. The user is able to access one of four screens after logging in, those being their Player scene, the Options scene, the Donate scene, and finally the Gameplay scene. The two most important ones here

are the Player and Gameplay scene, as that's what users are most likely going to boot up the application for, each featuring additional scenes in of themselves. The Options scene is for fine-tuning their experience to fit their needs, and the Donate scene is for if the user feels generous and would like to help support the application and the Garfield Park Conservatory.

27 Object Design

27a Packages

For this project, we will not be using any packages.

27b Subsystem I

Main Class: Player

Subclasses:

- **Profile-** This subclass is where information specific to the user can be found, such as their userID, their username, the icon they have set for their profile, date of creation, among other things, along with a display for plants of their choosing (so long as they have them in their Collection).
 - **DisplayedPlants-** The subclass is used to set up the display of plants for the user, with the user selecting up to a certain amount of plants to display on their profile for whenever they visit their profile or someone else visits. They may select a plant for a myriad of reasons, being for its exoticness, a specific theme, or they just like it in general.
- **Social-** This subclass deals with the user's friends that they added through the application, be it directly via their respective userID or through other means.
 - **FriendsList-** A subclass that contains a list of the user's friends that they can click on in order to go to a particular friend's profile. The exact details of the FriendsList is unknown, but a possible structure for displaying each friend would be their profile picture, their username, and the DisplayedPlants they have set for themselves.
- **Collection-** The subclass deals with the collection of plants the user has gathered through their usage of the Gameplay class while visiting the Garfield Park Conservatory.

- **PlantsList-** The subclass that contains the info about each plant the user has gathered, interesting tidbits and trivia about each of them as well as the more technical information, such as lifespan, their natural habitat, seasons where they flourish, and more. There may also be different sorting orders that can be applied to the list if the user desires.

27c Subsystem II

Main Class: Gameplay

Subclasses:

- **GPS-** This subclass deals with all things having to deal with coordinates, checking the user's coordinates to determine whether or not they fall within the range. If so, they pull up locations throughout Garfield Park Conservatory.
 - **Locations-** These would be the points of interest throughout the Garfield Park Conservatory, which would tie in to special events down the line as the average encounters are randomly generated.
- **Minigames-** The intended main draw, this subclass works with the encounters the user will deal with whenever they're walking around the conservatory with the application open. These encounters will then trigger a minigame depending upon the EventCondition, and should the user be successful in the minigame, they will be rewarded.
 - **EventCondition-** This subclass deals with the conditions around the encounter, such as the location in the conservatory, a random choice of a plant from a pool determined by the location, as well as a random choice of a minigame from a pool depending on the plant itself.
 - **MinigameEvent-** This subclass works hand-in-hand with the EventCondition and determines the type of minigame to be played by the user, and rewards the player should they complete the minigame.

IV Project Issues

28 Open Issues

A few issues that arose from Conservation Catch are those related to latency, security, as well as accuracy. Latency issues stem both from in-game performance as well as stability from Wi-Fi connection strength. To ensure that our game is as smooth as possible we have to focus on creating quick and fast allocation of new memory for late game players in the game. In terms of Wi-Fi stability, we have to ensure that the user's connection is stable at all times to reduce input lag for a smooth experience. Outside of game performance issues, is the issue of user's data privacy as Conservatory Catch uses the players GPS coordinates to spawn events near them. The app must ensure compliance with any privacy regulations as well as securing the user from data breaches related to their data. The final note we have is related to accuracy, this is more focused on mapping accuracy so that if a route was changed in the conservation, the app would reflect those actions in be modified to remove that section as well.

29 Off-the-Shelf Solutions

Some of our issues stated in open issues could already be solved with Off-the-shelf software or methods that are used to solve these. Such solutions could be the use Ready-Made Products such as the content delivery network (CDN), which is used in heavy traffic applications that would ensure that Conservatory Catch has a stable connection regardless of where the user is located in the Conservatory. These would be servers that work together to split up the players request to different servers to effectively serve in a timely manner. The CDN would be used to solve the issue if ensuring a stable and quick connection between the user and the Conservatory.

For Reusable Components, the app could use memory management tools and libraries to efficiently reuse memory blocks which reduce dynamic allocations. These tools and libraries would help the app's memory usage to ensure that the user has a clear and lag free system regardless of the state of the game they are in. Whether late or early game, the user would be ensured a smooth experience as they play using those tools and libraries.

As for Products that can be copied, the app can copy and follow industry-standard regarding privacy regulations and requirements to ensure the user that their data is secured and not sold to anyone for profit. Following the regulations and compliance allows the app to use the players GPS coordinates to generate events and then removes previous data from a set time ago. This creates a system that periodically clears some memory to save for future events and complies with privacy regulations.

30 New Problems

The effects on the current environment that Conservatory Catch introduces could be security related. Conservatory Catch is meant to be played in bounds of the Conservatory to help promote plants for the user to enjoy. But like all games, it only takes a few to create issues. If a player is dedicated enough and the app creates an out of bounds reward in a restricted area, the user could trespass into said area. This issue should be taken seriously as it could be a recurring offense created by the app. A solution could be to create clear spawn boundaries in the Conservatory as well as informing the user of restricted areas. These issues could be created if Conservatory Catch was released to the public and should be accounted for.

The effect on installed systems would be the introduction of a heavily traffic Wi-Fi connection on the servers introduced by Conservatory Catch. Without the app, the servers wouldn't need to hold as much data as it once needed to as the game would introduce player data to allocate and generate servers which create lots of packet requests that create traffic. The servers should be well prepared for slow downs and advised to improve servers for it to hold smooth connection between the user and the servers. In special events, the level of traffic would also increase, which would place more strain on the servers which could cause potential problems in connection and security.

The potential user problems that Conservatory Catch could create would be if Conservatory Catch is disconnected while the user is obtaining new data from plants, a system must be in place to restore back their last save as well as potential rewards they were meant to receive. If the user is playing a mini-game and completed the task but at the same time was disconnected, the last save in the servers should backup their completion status and reward the user once they reconnect. This will ensure that the user doesn't feel cheated when they accomplish a task and their connection with Conservatory Catch is disconnected, but they will still receive their rewards.

The limitations in the anticipated implementation environment that may inhibit the Conservatory could be environmental factors as well as sustainability of the servers. For environmental factors, any electrical interference that affects Wi-Fi or servers would hinder the application from working as it needs to be online to update and track the user to generate events. This would create a problem with the product as it would not be functional under those circumstances and unreliable. To solve this issue, the Conservatory would need a backup generator to sustain the server connection, but if the user's hardware faces those issues, then the app wouldn't be able to help solve them. Another issue comes from the new load on the servers that have to be maintained to keep Conservatory Catch up and running, which might hinder any other connection the

Conservatory has. The extra clutter and heavy traffic could slow Wi-Fi connections for other users as more traffic occurs. To solve this problem, the Conservatory has to maintain servers to keep a quick and stable connection.

No follow-up problems.

31 Migration to the new Product

Not Applicable

32 Risks

Potential Issues:

- Latency
 - Probability: High
- Performance
 - Probability: High
- Data Privacy
 - Probability: Moderate
- Maintainability and Supportability
 - Probability: High
- Environmental Limitations
 - Probability: Low
- Plant Data Accuracy
 - Probability: Low
- Server Scalability
 - Probability: Moderate
- Map Accuracy
 - Probability: Low
- Operating System Compatibility
 - Probability: Moderate

33 Costs

The budget for Conservatory Catch shall be around \$200,000. This number is based on the budget for similar mobile apps, accounting for scaled down operations. This shall cover the complete development of the game.

In order to ensure a stable release, the game shall have a release window of 2 years, with a maximum of 3 years. This should be enough time to handle preliminary tasks

such as creating the database and mapping the conservatory, as well as development of the game.

The number of people needed to create this project falls between 10 to 15. Developers will be needed to create the game, as well as UI/UX designers, testers, management, and sound. Anywhere in the 10 to 15 range shall provide enough support for smooth development.

34 Waiting Room

The waiting room for this project includes many additional features, perhaps fit for a 2.0 update. In order of important, these items include adding in-game weather effects for real life weather, a leaderboard, in-game themes, additional languages, and additional profile customizations.

35 Ideas for Solutions

Group 9 suggests the usage of Unity to develop the application. Unity is a popular game engine with vast amounts of documentation and support. It also features built-in assets which may assist the team with prototyping and scripting. It is also suggested that some sort of prototyping software be used to prototype the application. Group 9 suggests InVision, which is a popular prototyping software used by different game studios.

36 Project Retrospective

Many things went well for the group when designing this project. Communication was done mainly through weekly meetings and Discord. Weekly meetings provided a great opportunity to share ideas for the project, as well as divide work for the report. Discord was useful for checking in on each other, and for reminders on what needed to be done or when meetings were being held.

Something that did not work well was time management. There were a few times where sections for the report were not written up until hours before the deadline. This is something we struggled with as a group, though deadlines were still met. We tried to alleviate this by reminding each other on Discord, though this proved to not be as effective as we had hoped. Dividing the work also proved to be a difficulty early on. Some members of the group needed to take on more work to pick up after others. Luckily, this was only the case for the first release.

As a whole, the project went mostly smoothly for the group. All deadlines were met and the report has enough detail to be executed.

V Glossary

Overworld: This will be the digital representation of the users physical location in the real world. Here the user will be able to interact with other elements of the game.

Events: randomly appearing events will populate the overworld, allowing users to initiate different minigames

Minigames: randomly generated sets of games, ranging from encounters to trivia

Encounters: two types of encounters will be used within minigames, one being a battle encounter, the other being a capture encounter.

Species: species will be the primary focus of all trivia and encounters. These will be exotic plant species that users can see/learn about at the conservatory.

Trivia: not just any trivia will be displayed in minigames. This trivia must pertain to the exotic species the player encounters in the overworld events.

GPCA: Garfield Park Conservatory Alliance, group behind the programming at the Garfield Park Conservatory.

Collection: A list of all collected plant species.

Party: Group of players playing together in a session.

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] *Google Maps Platform FAQ | Google for Developers*, Google, developers.google.com/maps/faq. Accessed 2 Dec. 2023.
- [6] "Photosensitive Epilepsy and Online Content." Epilepsy Action, 31 May 2023, www.epilepsy.org.uk/press/photosensitive-epilepsy-and-online-content.

VII Index

Project Description.....7-21

Requirements.....21-56

Design.....56-66

Project Issues.....66-70

Glossary.....70-71

Reference / Bibliography.....71

Index.....72