

Teddy's Brochure Report



A document for creating the brochure of the future

**Prepared by
Cecilia Avila
Eugenio Perez
Kaveesha Weerasiri
Adrian Zavala
for use in CS 440
at the
University of Illinois Chicago
February 2020**

Table of Contents

List of Figures	8
List of Tables	9
I Project Description	11
1 Project Overview	11
2 The Purpose of the Project	11
2a The User Business or Background of the Project Effort	11
2b Goals of the Project	11
2c Measurement	11
3 The Scope of the Work	11
3a The Current Situation	12
3b The Context of the Work	12
3c Work Partitioning	13
3d Competing Products	14
4 The Scope of the Product	14
4a Scenario Diagram(s)	14
4b Product Scenario List	14
4c Individual Product Scenarios	14
5 Stakeholders	14
5a The Client	14
5b The Customer	15
5c Hands-On Users of the Product	15
5d Maintenance Users and Service Technicians	15
5e Other Stakeholders	15
5f User Participation	15
5g Priorities Assigned to Users	15
6 Mandated Constraints	16
6a Solution Constraints	16
6b Implementation Environment of the Current System	16
6c Partner or Collaborative Applications	16
6d Off-the-Shelf Software	16
6e Anticipated Workplace Environment	16
6f Schedule Constraints	16
6g Budget Constraints	17
7 Naming Conventions and Definitions	17
7a Definitions of Key Terms	17

7b	UML and Other Notation Used in This Document	17
7c	Data Dictionary for Any Included Models	17
8	Relevant Facts and Assumptions	17
8a	Facts	17
8b	Assumptions	17
II	Requirements	18
1	Product Use Cases	18
1a	Use Case Diagrams	18
1b	Product Use Case List	19
1c	Individual Product Use Cases	20
2	Functional Requirements	20
3	Data Requirements	21
4	Performance Requirements	23
4a	Speed and Latency Requirements	23
4b	Precision or Accuracy Requirements	24
4c	Capacity Requirements	24
5	Dependability Requirements	25
5a	Reliability Requirements	25
5b	Availability Requirements	25
5c	Robustness or Fault-Tolerance Requirements	26
5d	Safety-Critical Requirements	27
6	Maintainability and Supportability Requirements	28
6a	Maintenance Requirements	28
6b	Supportability Requirements	28
6c	Adaptability Requirements	29
6d	Scalability or Extensibility Requirements	29
6e	Longevity Requirements	30
7	Security Requirements	30
7a	Access Requirements	30
7b	Integrity Requirements	31
7c	Privacy Requirements	32
7d	Audit Requirements	33
7e	Immunity Requirements	33
8	Usability and Humanity Requirements	33
8a	Ease of Use Requirements	33

8b	Personalization and Internationalization Requirements	35
8c	Learning Requirements	36
8d	Understandability and Politeness Requirements	37
8e	Accessibility Requirements	37
8f	User Documentation Requirements	38
8g	Training Requirements	39
9	Look and Feel Requirements	39
9a	Appearance Requirements	39
9b	Style Requirements	40
10	Operational and Environmental Requirements	41
10a	Expected Physical Environment	41
10b	Requirements for Interfacing with Adjacent Systems	41
10c	Productization Requirements	42
10d	Release Requirements	43
11	Cultural and Political Requirements	
43		
11a	Cultural Requirements	43
11b	Political Requirements	44
12	Legal Requirements	45
12a	Compliance Requirements	45
12b	Standards Requirements	46
III	Design	46
21	System Design	46
21a	Design goals	46
22	Current Software Architecture	47
23	Proposed Software Architecture	47
23a	Overview	47
23b	Class Diagrams	47
23c	Dynamic Model	47
23d	Subsystem Decomposition	47
23e	Hardware / software mapping	48
23f	Data Dictionary	48
23g	Persistent Data management	48
23h	Access control and security	48
23i	Global software control	48
23j	Boundary conditions	48

24	Subsystem services	48
25	User Interface	48
26	Object Design	49
26a	Object Design trade-offs	49
26b	Interface Documentation guidelines	49
26c	Packages	49
26d	Class Interfaces	49
IV	Test Plans	49
27	Features to be tested / not to be tested	49
28	Pass/Fail Criteria	49
29	Approach	49
30	Suspension and resumption	49
31	Testing materials (hardware / software requirements)	50
32	Test cases	50
33	Testing schedule	50
V	Project Issues	50
34	Open Issues	50
35	Off-the-Shelf Solutions	50
35a	Ready-Made Products	50
35b	Reusable Components	50
35c	Products That Can Be Copied	51
36	New Problems	51
36a	Effects on the Current Environment	51
36b	Effects on the Installed Systems	51
36c	Potential User Problems	51
36d	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	51
36e	Follow-Up Problems	51
37	Migration to the New Product	52
37a	Requirements for Migration to the New Product	52
37b	Data That Has to Be Modified or Translated for the New System	52

38	Risks	52
39	Costs	52
40	Waiting Room	52
41	Ideas for Solutions	52
42	Project Retrospective	53
VI	Glossary	53
VII	References / Bibliography	53
VIII	Index	53

I Project Description

1 Project Overview

Teddy's Park Pamphlet aims to provide a consistent, versatile, and reliable pamphlet for all national parks and/or forests in the United States. Simply enter the national park you wish to visit, and the app will download a package to your phone containing an interactive map, park specific rules and/or information, park specific survival tips should you get lost, a record of the weather forecast, and a list of volunteer groups working for the park.

While connected to the internet, you will be able to leave reviews for volunteer groups, post pictures for the park's gallery, and receive perpetually updating weather forecasts, and the most up to date info on parks.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

There are currently 154 national parks in the US. Each with their own management systems and websites. Some are informative and clean, while others seem like they haven't been updated since the founding of the internet. The client of this app will be whoever manages said national park, for the benefit of its visitors.

2b Goals of the Project

The goal of this service is to provide a centralized database of information for each and every national resource. For a country as vast and geographically diverse as the United States, each location has different weather conditions, topology, wildlife, and volunteer groups. This app will act as a "one stop" for park visitors, displaying all the information one could possibly need before visiting. National Parks aren't known for their cellular connection and wifi reliability, which is why at the discretion of the user, the app can download a digital copy of the park information onto their phone, so that they can have a copy of the information ready to go wherever they are in the park, regardless of mobile service.

2c Measurement

We will know the goals of the app are met when we have at least 85% of the country's national parks in our database. On the client side, the app must be able to connect to a server that contains our database. While there is an internet connection, the app should be able to load a national park, and return the following data specific to that park: Weather, logistics, links to volunteer opportunities, safety tips, and a gallery.

Should the client want access to this information offline, they can download a lightweight copy of the information onto their mobile devices. This lightweight copy of the information would

only have a “snapshot” of the weather forecast, along with park rules and logistics, maps, and safety tips for use inside the park in the event there is no service.

3 The Scope of the Work

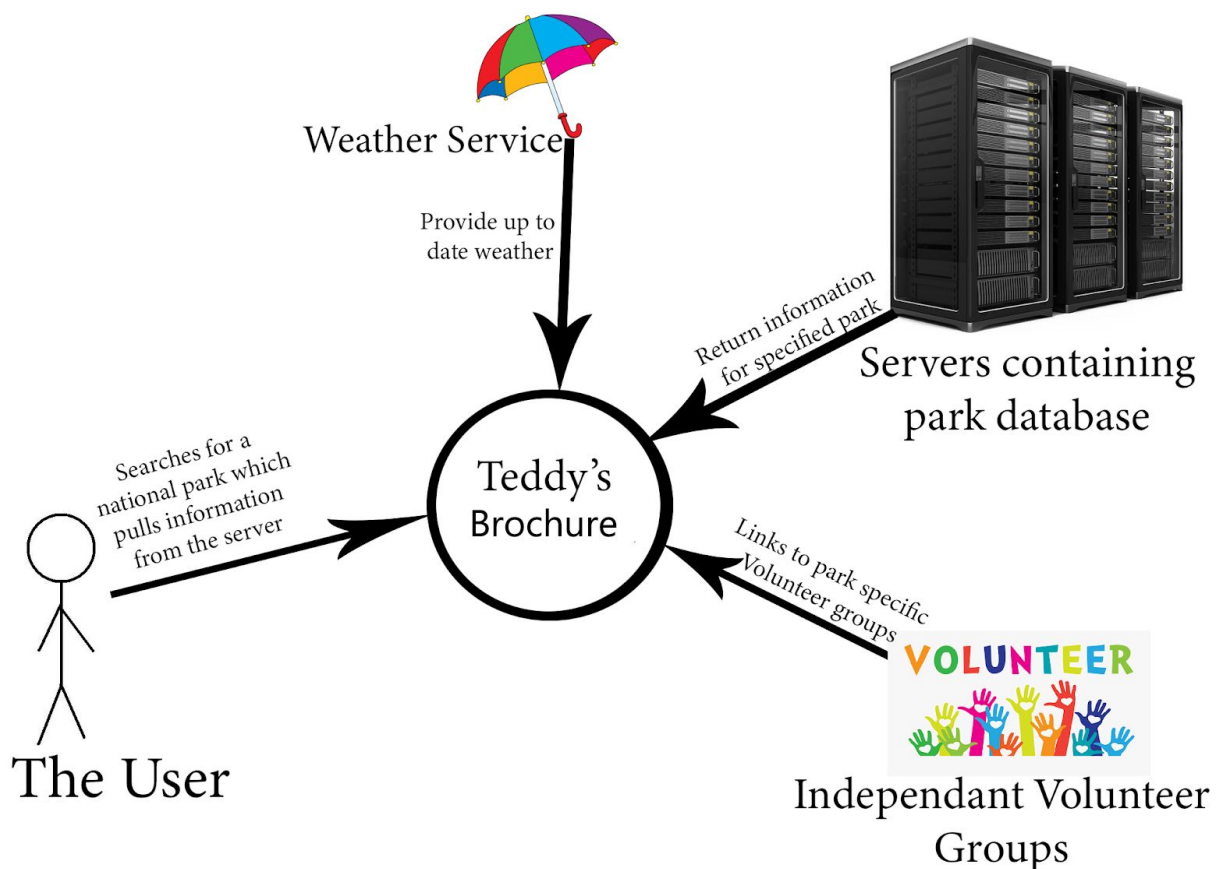
The work will be the improvement of accessibility for park information

3a The Current Situation

Currently, if the client wishes to visit a park, they can visit the park website, where information and presentation varies from park to park. If they want access to this information while they’re in the national park, they’ll have to roll the dice with shoddy phone service, or make sure they have a paper copy of the pamphlet.

If you are lost somewhere in the national park, and without a map or cellular service, then you are on your own.

3b The Context of the Work



The programmer will manually populate the database containing all the info about all the National Parks. From here, the server will contain and store all this information. When the user submits a request to see the information for any given park, the database will also pull

information for the weather in the area from a participating weather station, and a list of volunteer groups.

3c Work Partitioning

Event Name	Input and Output	Summary
Park Database	Park listings (in)	Return information for the specifies park
Weather Service	Update weather forecast for park (in)	Google API will assist in retrieving weather for chosen park
Independent Volunteer Groups	Volunteer listings (in)	Links and information will be provided for specific volunteering opportunities
User search	General information (in)	Search for a specified National Park and display its information.

3d Competing Products

The National Park System already has a database of National Parks in place. Many of which have websites that are filled with bounties of valuable and pertinent information similar to what this app offers. Unlike the database offered by the NPS, Teddy's pamphlet aims to reduce the reliance on an internet connection, by providing a downloadable copy of National Park information should you find yourself in an area without service. We also provide information for what to do when you are lost or stranded. Which is information that is not easily accessible when you do not have cellular service.

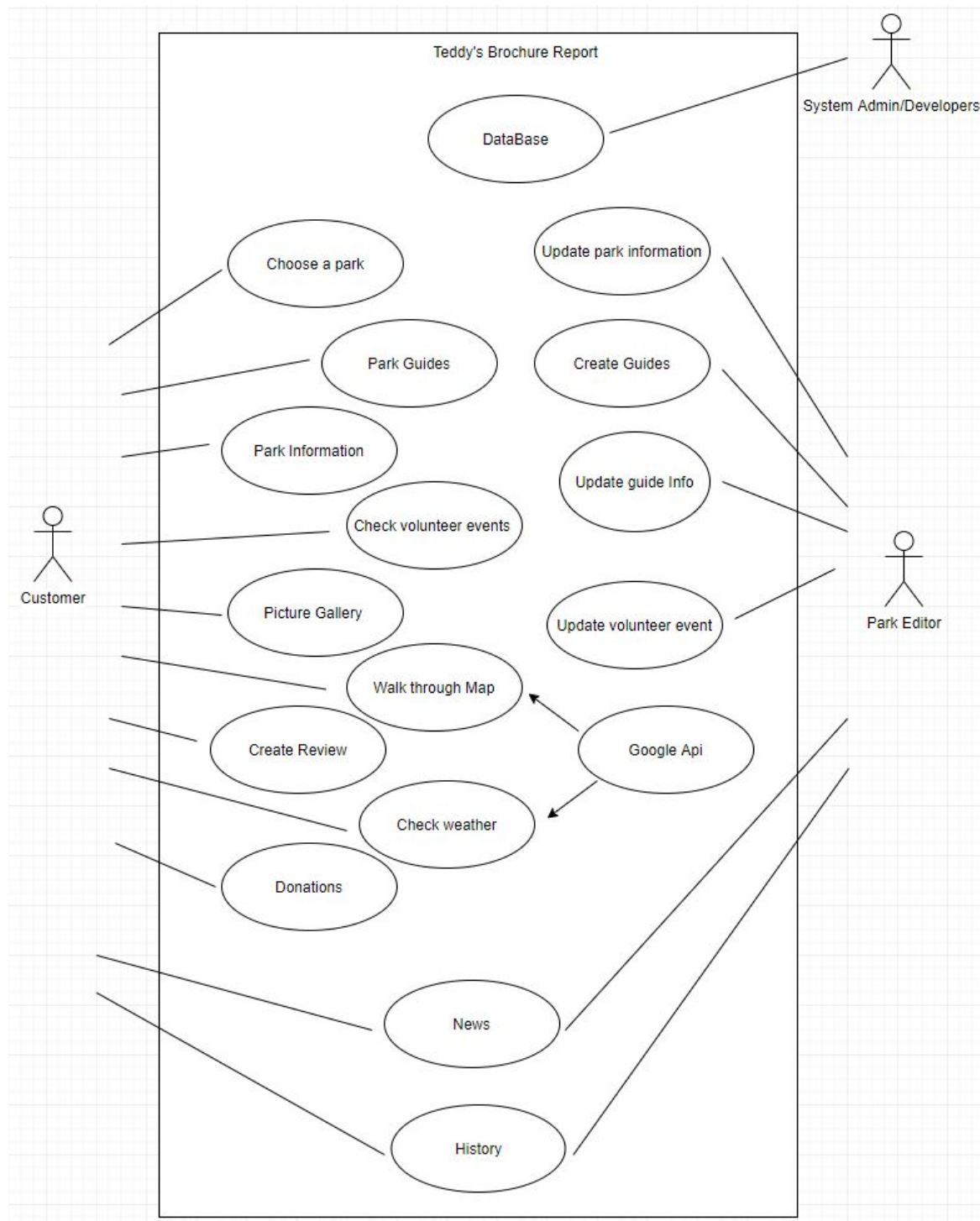
4 The Scope of the Product

Upon starting up the app, the user will be able to search for a national park, or select one from a drop down menu based off of location. the system will then pull data from a database for said national park. Information includes: Park hours, rules, maps, topological information, wildlife, volunteer groups, weather, and survival/safety tips.

The programmer will populate the server with park information. For things that are constantly updating, like the weather, the system will simply pull the most up to date information at the time the user requests for that information.

Independent Volunteer Groups will have their own websites that the app will link to. The User can leave reviews for any particular volunteer group, or visit their websites linked by our system.

4a Scenario Diagram(s)



4b Product Scenario List

1. Choose a park
2. Park Guides
3. Park Information
4. Check Volunteer Events
5. Picture Gallery
6. Walk through Map
7. Create Review
8. Check weather
9. Donations
10. News
11. History
12. Update park information
13. Create guides
14. Update guide information
15. Update volunteer event

4c Individual Product Scenarios

1. Choose a park: There are multiple national parks and the user can have the option of which park to access.
2. Park Guides: Each park is different and there are different guides to each park. What to do if lost, plan a visit, how to properly hike, what equipment to bring. All informational guides for people that have never been to a National Park.
3. Park Information: The general park location and contact information
4. Check volunteer events: The user can check each park for volunteer events. The user can check whenever they wish to volunteer. Events can be named, categorized by event, date and time, duration, and location.
5. Picture Gallery: Since each park is unique the park can showcase the highlight areas, and post photos that people have submitted from their visits.
6. Walk through Map: If a person is going to visit the national parks, there is not much cellular connection. The user can download a map for the park that uses google's street view, which can be used offline. This can make user's more confident exploring the parks.
7. Create Review: The user can create a review for the park or the volunteer events that have attended.
8. Check weather: A user can check the weather with google api. When planning a visit the user can see the park's available dates and weather forecast.
9. Donations: If a user wants to donate to the park(s), they can donate securely through the application. Maybe the application was very useful and was very thankful for the application they can donate to the parks.

10. News: The user can check news updates to each park on weather forecast, events, and history, animal endangerment, etc..
11. History: Each park is beautiful and unique in its own way. Each park can showcase its history to show its identity.
12. Update park information: Park information changes and editors can update the information accordingly.
13. Create guides: Each park can have its own specific guides to best deal with the environment.
14. Update guides: Guides can eventually be outdated, and having the best information available for people to plan their visits is key.
15. Update volunteer events: Park editors can add upcoming events for people to visit. They can also remove events that are canceled or past date.

5 Stakeholders

Teddy's Brochure could interest the National Park Service U.S. Departmental of the Interior as they focus on caring for the national parks with the help of volunteers and partners. Teddy's Brochure provides a lot of similar features they already have on their website, increasing their interest for the proposed product. Volunteers and park employees would also be interested in the development as they could potentially use the brochure as well.

5a The Client

The client is the National Park System.

5b The Customer

The customer is the National Park System, and each individual park that wishes to adopt the applications to include their park.

5c Hands-On Users of the Product

The hands-on users of the product are the general public and park employees that visit the parks. The general public would use the product to enhance their experience at the park. Their subject and technological experience could be novice. The employees would need to be masters in the subject and journeyman in the technology experience in the case they are questioned by the general public. For those maintaining the brochure, they would need to be masters.

5d Maintenance Users and Service Technicians

Maintaining the application will be handled by the development team. The app will be in constant update with multiple releases. After the last stages of the project, there can be a small team that deals with security updates and bug fixes.

Park managers and editors will be able to use the application to update events and information.

5e Other Stakeholders

Park employees might be replaced if their job is to assist or guide visitors around the park. Their job duties would change as there would be a system to care for and update.

Testers to test efficiency and functionality of the brochure.

System designers to ensure functionality matches system specifications specified per sprint.

Usability Experts will attempt to download the package and report difficulties with the brochure.

5f User Participation

The client will participate through inspecting the app at different phases of development such as when the developers decide on what the app UI will look like. The client will also comment on functionalities that the app is missing. At the end, the client would test the application to make sure the user experience is up to par with the park standards.

5g Priorities Assigned to Users

As indicated in the use case diagram, the customer would be the most important user, the park editor would be the second most important user, and the system administrator would have the least amount of priority. This is mainly because the customer has the most use from the application, the park editor has less use cases than that, and the system administrator only has one use case.

6 Mandated Constraints

Mandated constraints include the app showing park information to the user. It must be compatible with at least one park. It must have a downloadable map of that one park that the user can interact with. It needs to show some volunteering activities within the specific park.

6a Solution Constraints

General constraints include the app being built within 3 months. Another constraint would be that each national park being worked on will have its own database. It should handle 1000 users at a time by the time that it is built and gradually scale up. It should be built with the client in mind as the client will greatly contribute to the development.

6b Implementation Environment of the Current System

The physical environment that the product will operate in is in each national park. Communication issues would be based on the availability of the wifi. It will be used for all operating systems including, but not limited to, Linux, Mac OS, and Windows. The hardware would include a server that is able to connect to the internet.

6c Partner or Collaborative Applications

The app must be compatible with google maps and must be able to obtain and parse information from APIs such as the one used to get weather. It must be compatible with storing jpegs and pdfs.

6d Off-the-Shelf Software

The product will come with all resources it needs. No additional software is needed for the application to function.

6e Anticipated Workplace Environment

The anticipated workplace environment for the application is the national parks where there is limited connection. Key features such as guides and maps should be accessible without internet connection. The rest of the application can be used anywhere with internet connection. There will be many first time users on the application/website, so the interface should be clean and simple to use.

6f Schedule Constraints

The pamphlet would be beneficial to be done 10 weeks after the start date. Allow 3 more weeks for testing to fix bugs and add functional enhancements. It would be beneficial to have it done by Spring into Summer when visitors begin to visit parks more due to the nice weather.

6g Budget Constraints

The application can be limited in budget by what the National Park Systems have allocated in their annual budget.

7 Naming Conventions and Definitions

Gallery: a server where people post pictures of the national park

Package: The application that is enhancing the national park service to the customers

Tips: hints on how to camp and hike at the national parks

Master: has complete knowledge of a domain

Maps: 2D maps

7a Definitions of Key Terms

Walk through map: Maps of the specific parks that people can interact with (google maps)

7b UML and Other Notation Used in This Document

This document generally follows the Version 2.5 OMG UML standard, as described by Fowler in [4]. Any exceptions are noted where used.

7c Data Dictionary for Any Included Models

Data structures that the app will use include hashamps that will store the developers, and editors information. These two users will have a name property that is a string. Editors will have an image property that accepts jpgs. The walk through maps would be of type roadmap. The listing of volunteering opportunities would be stored in an arraylist. Rules and how to prepare would be stored in a pdf.

8 Relevant Facts and Assumptions

8a Facts

Users will be able to use the brochure offline. Application is created to bring more exposure and visitors to the park.

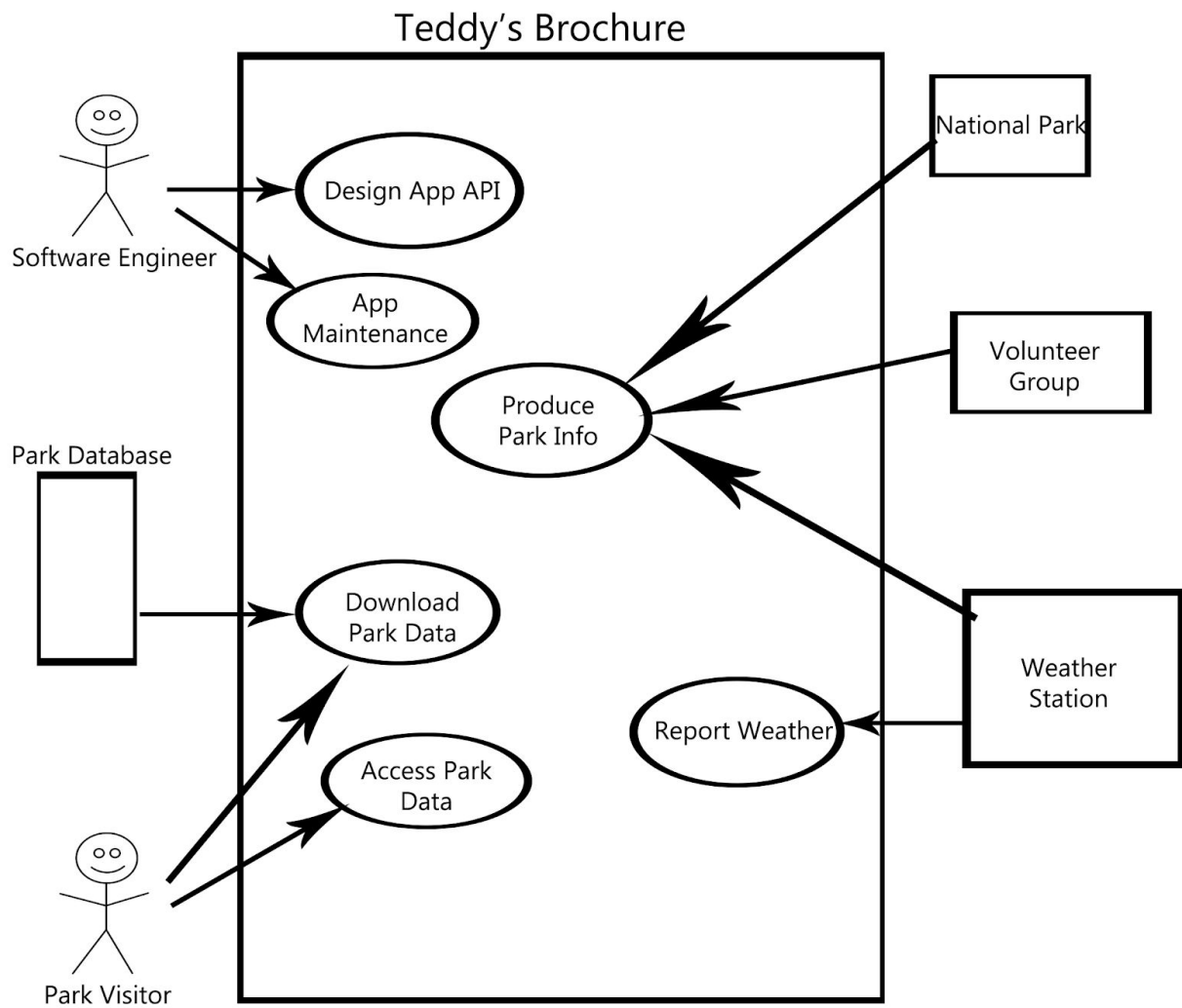
8b Assumptions

Assumptions include that the weather information is available from google's api. Other assumptions include having volunteering organizations already established for the app to use. Any api's that we use, either for volunteering or weather, must be working and updated frequently in order to display accurate information. The user must have a device such as a phone or tablet that they can look at and that can download information when the person has service.

II Requirements

1 Product Use Case

1a Use Case Diagrams



1b Product Use Case List

Use Case #1	Access Park Data
Use Case #2	Download Park Data
Use Case #3	Report Weather
Use Case #4	Access Volunteer Group Information

1c Individual Product Use Cases

<i>Use Case - #1</i>	<i>Access Park Data</i>
<i>Participating Actors</i>	<i>User</i>
<i>Flow of Events</i>	<ol style="list-style-type: none">1. <i>User selects a park from a list of National Parks in the app by tapping it</i>2. <i>The System will then pull information from a database</i>3. <i>System will display the park information to the user</i>
<i>Entry Condition</i>	<i>User selects “choose park” in menu</i>
<i>Exit Condition</i>	<i>Park info is fully displayed on user’s device</i>
<i>Quality Requirements</i>	<i>Loading of park info should ideally take less than 5 seconds to load, but absolutely no more than 10 seconds</i>

<i>Use Case - #2</i>	<i>Download Park Data</i>
<i>Participating Actors</i>	<i>User</i>
<i>Flow of Events</i>	<ol style="list-style-type: none">1. <i>User selects a park from a list of National Parks in the app by tapping it</i>2. <i>The System will then pull information from a database</i>3. <i>While park information is being pulled from database, a button will appear prompting the user to “download to phone”.</i>4. <i>[OPTIONAL User can also wait until information is fully loaded on cellular device, to then press a button that says “download to phone”]</i>5. <i>System will save a copy of the currently loaded information, and download it to user’s device</i>

<i>Entry Condition</i>	<i>User presses “download to phone”</i>
<i>Exit Condition</i>	<i>Park info is fully downloaded to a user’s device</i>
<i>Quality Requirements</i>	<i>downloading of park info should ideally take less than 30 seconds, but absolutely no more than 1 minute.</i>

<i>Use Case - #3</i>	<i>Report Weather</i>
<i>Participating Actors</i>	<i>Weather Service Park Database</i>
<i>Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User selects a park from a list of national parks</i> 2. <i>System will prompt database to send a GET request to a weather website</i> 3. <i>Database will update to have the most up to date weather information</i>
<i>Entry Condition</i>	<i>User selects “choose park” in menu</i>
<i>Exit Condition</i>	<i>System sends GET request to a weather website</i>
<i>Quality Requirements</i>	<i>The updating of the weather information should take no more than 2 seconds,</i>

<i>Use Case - #4</i>	<i>Access Volunteer Group information</i>
<i>Participating Actors</i>	<i>User Volunteer Group Park Database</i>
<i>Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Selects National Park</i> 2. <i>System Displays page showing general information for national park</i> 3. <i>User presses “View Volunteer Groups”</i> 4. <i>System displays a list of Volunteer Groups</i> 5. <i>User presses the name of a volunteer group</i> 6. <i>System displays information of chosen volunteer group</i>

<i>Entry Condition</i>	<i>User selects “View Volunteer Groups” in menu</i>
<i>Exit Condition</i>	<i>Volunteer Group information is displayed on user’s device</i>
<i>Quality Requirements</i>	<i>List of Volunteer Groups should not take more than 5 seconds to load. Same for when a user selects a specific park group. Group specific information should not take more than 5 seconds to load.</i>

<i>Use Case - #5</i>	<i>Leave Review for Volunteer Group</i>
<i>Participating Actors</i>	<i>User Volunteer Group Park Database</i>
<i>Flow of Events</i>	<ol style="list-style-type: none"> 1. User selects “Leave Review” for Volunteer Group 2. [OPTIONAL] System pulls up a form asking user to describe their experience with Volunteer Group 3. User enters a rating 4. [OPTIONAL] User presses cancel 5. [OPTIONAL] If User presses cancel, System exits out of prompt, and frees any memory. 6. User presses submit review 7. System checks if mandatory fields have been entered 8. [OPTIONAL] If System detects a mandatory field is empty, user is notified that it must be filled before submitting a review 9. System accepts the review scores, and pushes it to park database. 10. Database aggregates Volunteer group review score 11. System notes user’s cellular device has left a review for this volunteer group as to prevent duplicate or fraudulent reviews
<i>Entry Condition</i>	<i>User selects “View Volunteer Groups” in menu</i>
<i>Exit Condition</i>	<i>User submits review for Volunteer Group or presses cancel</i>
<i>Quality Requirements</i>	<ol style="list-style-type: none"> 1. Volunteer Group rating is updating within 2 minutes after user submitting a review

2 Functional Requirements

Requirement #: 1000

Requirement Type: n/a

Event / BUC / PUC #:

Originator: Eugenio Perez

Origination Date: 3/3/20

Latest Update: 3/3/20

Description

The System should be able to display a list of National Parks

Rationale:

To show the breadth of the app, as well as provide an alternative for the user should they be unable to use a keyboard to search for their national park

Fit Criterion:

The system has a scrollable list of National Parks in our system, or something to that effect

Customer Satisfaction: 8

Customer Dissatisfaction: 7

Priority: High

Conflicts: n/a

Supporting Materials: n/a

History: n/a

Requirement #: 1001	Requirement Type: n/a	Event / BUC / PUC #:
Originator: Eugenio Perez	Origination Date: 3/3/20	Latest Update: 3/3/20

Description

The System must download a copy of the park data to the user's phone when prompted, as an "offline mode" for the app

Rationale: Large national parks are infamous for having large "deadzones" where there is no coverage. By giving the user the option to download a cached copy of the park data, they will have access to a majority of our app's functionality regardless of internet access

Fit Criterion:

The system will download a cached copy of the park data to the user's phone, and this information will be available even while the phone is on airplane mode.

Customer Satisfaction: 7	Customer Dissatisfaction: 6	Priority: High
Conflicts: n/a	Supporting Materials: n/a	
History: n/a		

Requirement #: 1002	Requirement Type: n/a	Event / BUC / PUC #:
Originator: Eugenio Perez	Origination Date: 3/3/20	Latest Update: 3/3/20

Description

The System should accurately display weather information for any given National Park

Rationale:

In order to succeed on the promise of being a one stop for all national parks, the system should have a means of displaying any and all information that could be pertinent to a user's park experience.

Fit Criterion:

System has a means of displaying the weather for any given National Park

Customer Satisfaction: 7	Customer Dissatisfaction: 5	Priority: High
Conflicts: n/a	Supporting Materials: n/a	
History: n/a		

Requirement #: 1003

Requirement Type: n/a

Event / BUC / PUC #:

Originator: Adrian Zavala

Origination Date: 3/3/20

Latest Update: 3/3/20

Description

The system must have a page for each park volunteer events.

Rationale:

National park's have many volunteers. Having a dedicated page for events, can bring more attention to events. It ideally would make volunering events easier to manage for each park. Each event will have its own description and contanct information.

Fit Criterion:

Each park has a page for volunteer events.

Customer Satisfaction: 8

Customer Dissatisfaction: 7

Priority: High

Conflicts: n/a

Supporting Materials: n/a

History: n/a

Requirement #: 1004

Requirement Type: n/a

Event / BUC / PUC #:

Originator: Cecilia Avila

Origination Date: 3/3/20

Latest Update: 3/3/20

Description

The system must have a page for user experience reviews from the park and from using the application.

Rationale:

Feedback from users can be beneficial to the national parks as it can bring more attraction as well as solve potential problems. Application feedback would allow improvement for the application as we can see what may or may not be working and what users would like to see.

Fit Criterion:

Each park will have a review page available for users.

Customer Satisfaction: 7

Customer Dissatisfaction: 8

Priority: Medium

Conflicts: n/a

Supporting Materials: n/a

History: n/a

Requirement #: 1005	Requirement Type: n/a	Event / BUC / PUC #:
Originator: Kaveesha Weerasiri	Origination Date: 3/3/20	Latest Update: 3/3/20
Description		
The system must have accurate user location stored.		
Rationale:		
The walk through map needs to be accurate		
Fit Criterion:		
App must ask user to give their phone location so that it gives an accurate walkthrough map		
Customer Satisfaction: 7	Customer Dissatisfaction: 6	Priority: Medium
Conflicts: n/a	Supporting Materials: n/a	
History: n/a		

3 Data Requirements

Content

Class **Weather** contains a collection of the class forecast. Class **Forecast** contains all information needed pertaining to the weather, such as the temperature, and the climate for the day.

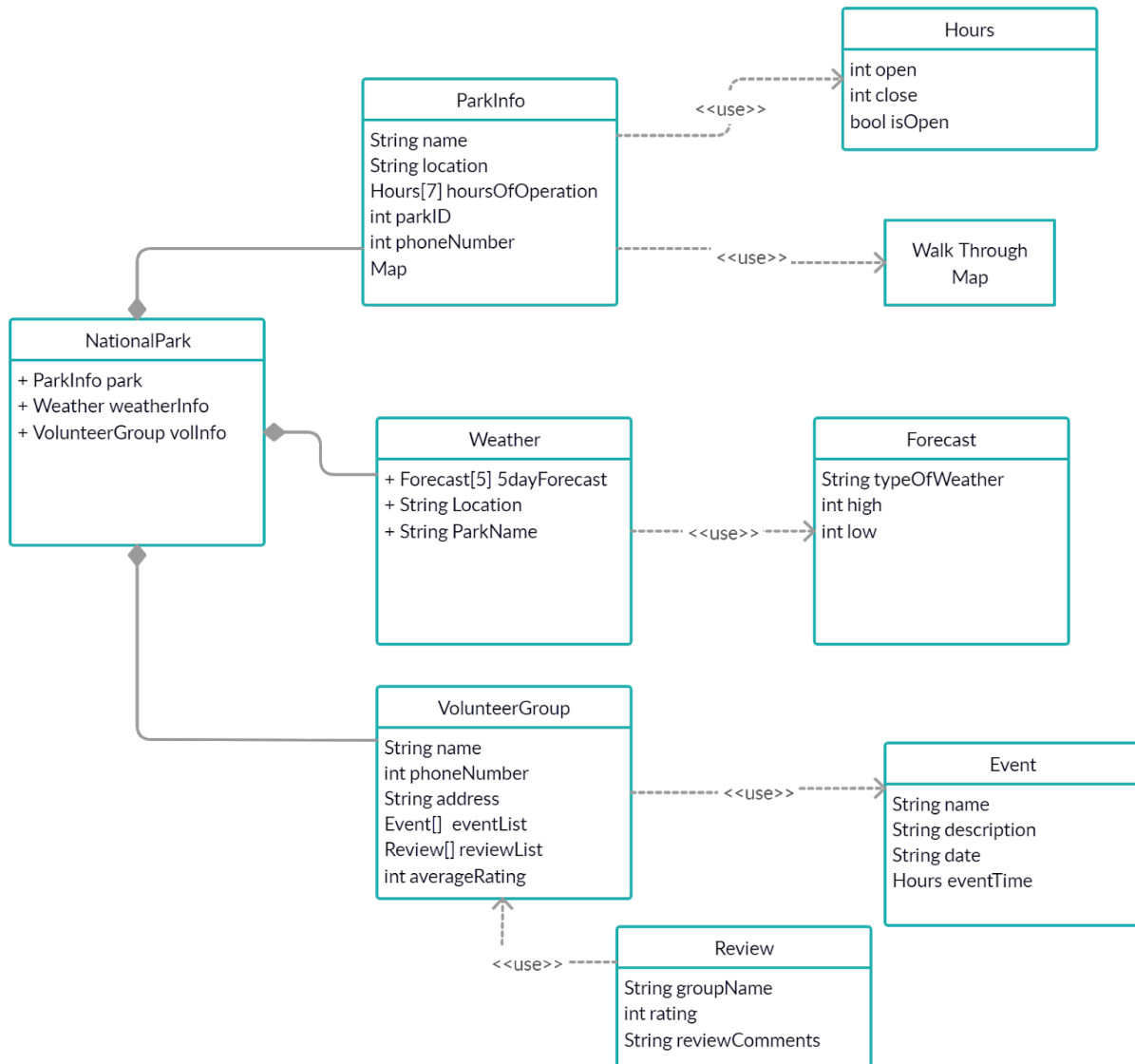
Class **ParkInfo** is a class that contains all the information pertaining to the National park itself such as its name, location, phone number, ID, and hours of operation. The **Hours** object contains all this information, and it is stored in a collection inside the NationalPark class. The **map** class contains an interactive map of the national park in question. This map will utilize the Google Earth API to provide a high quality map for the user to interact with.

Class **VolunteerGroup** contains all information pertaining to the volunteer groups that run events in any national Park in particular. it contains references to **Review** and **Event** objects.

Class **NationalPark** contains references to parkInfo, Weather, and Volunteer Groups. This class acts as the central “hub” containing objects which also contain information, and is the primary object the application will be referring to whenever populating the screen.

Motivation

Teddy's Brochure is at its core, a database for all national parks in the United States. It's only fitting then that the majority of the entities are for storing information



4 Performance Requirements

4a Speed and Latency Requirements

Requirement #4a1 - Speed and Latency Requirements

Description

Response times should be within a day, especially for people who need help with finding something. The product should operate with as low latency as possible, but there is room for moderate latency since not many people will be using the product on an internet connection.

Rationale

The motivation is so that users can update the gallery of the park in a reasonable time frame and can download guides and maps of the park in a reasonable time as well.

Originator: Kaveesha Weerasiri

Fit Criterion

The product shall respond in less than 1 second for 90 percent of the interrogations. No response shall take longer than 3.5 seconds. Moderate latency would be around 25 milliseconds.

Customer Satisfaction: 7

Customer Dissatisfaction: 3

Priority: Medium

Created: April 2, 2020

Examples

Any interface between a user and the automated system shall have a maximum response time of 4 seconds.

The product should take a user suggestion within 5 seconds.

The response shall not be slow enough to disrupt users thought process and cause them to leave

Considerations

The response time does not need to be really quick as most of it is not crucial to the user's survival. The control report will be run every 3 months to make sure that it continues to respond okay to user requests.

4b Precision or Accuracy Requirements

Requirement #4b1 -Precision and Accuracy Requirements

Description

The accuracy of the product in terms of giving users access to google maps, google weather, and google earth should be 97 percent of the time because a large part of the application is to allow users to get information offline.

Rationale

The motivation is that the client and the user can expect accurate representation of what the current state of the specific park is.

Originator: Kaveesha Weerasiri

Fit Criterion

The product should be accurate 97 percent of the time in terms of maps, but may be higher depending on the general accuracy of Google's products.

Customer Satisfaction: 7

Customer Dissatisfaction: 3

Priority: High

Created: April 2, 2020

Examples

Any donation that users give will be accurate to two decimal places since cents can be given as well.

Accuracy of the location of the user in respect to different maps will be within 5 meters of the user.

Considerations

The units will use imperial data just because it is used in the United States. If it transfers to other continents, then that data will also be converted to metric data. Timestamps will be reported within minutes for news updates and posts to the gallery. The app will change the time displayed in the application based on the time zone. Thus it will display accurate time. The currency will be in US dollars, but if the application is moved to different continents, the currency may change.

4c Capacity Requirements

Requirement #4c1 -Capacity Requirements

Description

The product must be able to deal with not a lot of volume because most of the application will be used offline.

Rationale

To ensure that the product is able to process the expected volumes of data that will be in the application

Originator: Kaveesha Weerasiri

Fit Criterion

The amount of data stored in the product will be close to 43 MB for iOS and 20 MB for Android. This accounts for a small portion of google maps, google weather, some pdfs of a park guide and park maps.

The developers will test to see whether the examples listed above are met or not. This is important because people may get lost during peak times and need help. The test will have multiple users try to use the application at the same time

Customer Satisfaction: 7

Customer Dissatisfaction: 3

Priority: High

Created: April 2, 2020

Examples

The product shall cater for 300 simultaneous users within the period 7:00 am to 10:00 am and 20 users from 10:00pm to 6 am. Maximum loading at other periods of time will be 100 simultaneous users.

5 Dependability Requirements

5a Reliability Requirements

5a.1: Consistent UX while connected to internet

Content

While an internet connection is present, The system will have access to all utilities, displaying up to date information for each park.

Motivation

To provide a consistent user experience, when the user is connected to the internet, no functionality should be hindered.

Examples

Before entering a national park of the user's choice, they can check the information and plan their trip according to things like the weather, where they park, and what's available in the park.

Fit Criterion

While an internet connection is present, check to see that general information is displaying correctly, and if some of the dynamic information such as the weather and volunteer group events is properly updated.

5b Availability Requirements

We would like to give the impression that the app is available for use 24/7/365. However, There will be times when we will need to do maintenance. Maintenance will be scheduled for time and days where we do not expect many people to be using the app, such as in the early morning, or weekdays that aren't holidays. But for the most part, the product shall be available 24 hours a day, 365 days per year. Everyone is different, and will explore parks on their own time, so it is important we offer a product that is usable by everyone.

That being said, accounting for maintenance, we aim for the app to achieve 97% uptime

5c Robustness or Fault-Tolerance Requirements

The product expects the user to go into places where there might be no internet, so it is important that the application work in the absence of wifi. So as long as the user downloads the "brochure" of the specific park they wish to visit before visiting it, there should be no degradation of utility in the app in the event an internet connection is lost. The user should still have access to general park information, maps, and weather information.

5d Safety-Critical Requirements

Requirement #1 Consistent battery life

It is important that the application be lightweight, as to minimize strain on the phone's battery. In the event that the user has the application open for long periods of time, it should not be any detriment to their device, in the event they must use it for some other emergency.

Fit Criterion

We will have the application run idle on an Oreo android device. We will check the battery usage using its built in battery tracker, and see how much power it is consuming. Acceptable ranges are within 1.5% to 6%.

Requirement #2 Park specific Survival Guides.

We aim for the app to be accessible to anyone regardless of experience. Which is why each park will come loaded with a guide that is specific to it. It is crucial that this information is up to date, as to ensure the safety of the user.

Fit Criterion

Starting off, we will implement a lot of "common sense" tips, but as the app grows, we will show drafts of our guides to the American Hiking Society, and then offer a survey that will aggregate their responses on a scale of 1 - 10. If this number is greater than or equal to 8, then it meets the standards of the app.

6 Maintainability and Supportability Requirements

6a Maintenance Requirements

Requirement #6a1-Maintenance Requirement

Description

The amount of time needed to make necessary changes to the product should not exceed more than 7 days.

Rationale

This non urgent maintenance requirement is because this application does not depend on it being continually updated. There is no real risk of not updating constantly, but it should still be updated periodically as new findings about different parks or suggestions are made. Parks do not change that rapidly, so there is no need to have constant updates.

Originator: Kaveesha Weerasiri

Fit Criterion: When needed, the application should update so that new information about a park is found.

Customer Satisfaction: 7

Customer Dissatisfaction: 3

Priority: Medium

Created: April 2, 2020

Examples

New weather should be updated every day to account for changes in the forecast.

Guide and park information should be updated when there are big changes to the park.

Considerations

The product will be open sourced so that developers outside of the application can contribute to it as well and maintain the code. There will be moderators of the code that will look at pull requests and resolve conflicts before merging the requested code into the master branch. Documentation is needed whenever a developer wants to add to the project. This will help for readability and help other people who are new to the project get up to speed on the code base. Testing should also be developed for any code that is to be merged into the final product.

6b Supportability Requirements

Requirement #6b1 -Supportability Requirement

Description

The application is periodically checked to see if there is new information that could replace the old information. This could be done weekly by one or two IT staff members. People could post complaints and suggestions in a dedicated tab on the app that would inform the people staffed to make changes to the application.

Rationale

These measures will ensure that customers will be able to be supported in adequate time.

Originator: Kaveesha Weerasiri

Fit Criterion: Support may be supplied through the use of other staff monitoring and organizing the pictures posted in the gallery.

Customer Satisfaction: 6

Customer Dissatisfaction: 3

Priority: Low

Created: April 2, 2020

Considerations

The level of support may increase, but not by much. That is because the pages rendered in the application will mostly be static. One goal is to eliminate the amount of paper maps, so a constraint is that there will be no printed manuals. The support may be online help, or potentially people staffed at parks answering questions about the application.

6c Adaptability Requirements

Requirement #6c1 - Adaptability Requirements

Description

The platform should be able to operate on both iOS and Android for mobile development.

Rationale

We want to enhance the accessibility of our app, so the more platforms that it supports, the better for the developers and the parks. It should be able to run on phones first, so iOS and Android would be more of a priority.

Originator: Kaveesha Weerasiri

Fit Criterion

The product must operate on at least version 8.0 on Android and iOS 13.

Future environments in which the product is expected to operate include Windows, Samsung, and Linux

The time allowed to make the transition could be up to 4 years because getting an operating system popular on a phone takes around 4 years.

Customer Satisfaction: 7

Customer Dissatisfaction: 2

Priority: High

Created: April 2, 2020

Examples

The product is expected to run on iOS and Android.

The product will be used in the United States at first, and then may be expanded to different countries and continents if the popularity of it rises.

The product is designed to be used on the go, but will not take away the experience for desktop users if they want to look at volunteering, history, news, or park information.

Considerations

We would work with any other application that has something similar to what we have so that we are not creating a redundant application. We would work with things like REI, Yellowstone, and the Great Smoky Mountains.

6d Scalability or Extensibility Requirements

Requirement #6d1 - Scalability or Extensibility Requirements

Description

The product must be able to handle an increase of 20 users per day.

Rationale

To allow for future capacities, the designers must design with a relational database in mind, or just something that can store lots of data. User interaction and posting will not be the core of the experience, a lot of the stuff will be static in delivery.

Originator: Kaveesha Weerasiri

Fit Criterion: Keep updating file formats as time goes on so that the gallery can continue to work. Keep doing system tests so that app response times will not fall drastically as time goes on.

Customer Satisfaction: 7

Customer Dissatisfaction: 2

Priority: Low

Created: April 2, 2020

Examples

The product should be able to download data from google maps, google weather, and google earth update information periodically. It should be able to store large maps and pdfs. It should be able to handle about 1000 customers using it within a year.

6e Longevity Requirements

Requirement #6e1 - Longevity Requirements

Description

The expected lifetime of the product is for as long as national parks exist. It should be able to last for as long as internet reliability in national parks is an issue.

Rationale

To ensure that national parks continue to give users the best experience possible. We want to make an application that can evolve as resources are needed.

Originator: Kaveesha Weerasiri

Fit Criterion: Stay up to date with national parks. Keep up to date with internet standards.

Customer Satisfaction: 8

Customer Dissatisfaction: 2

Priority: Medium

Created: April 2, 2020

Examples

The product should be able to give the experience promised for at least five years so that the parks and other entities will be able to use it without worrying about what they might change to in the near future

7 Security Requirements

7a Access Requirements

N/A

Reasoning: This application is a knowledge base that doesn't contain any information that is classified in any way. Everyone who downloads the app has equal access to it, thus, no login info is needed.

7b Integrity Requirements

Requirement #7b1

Description: Database information must not be writable by users.

Rationale: To protect park goers by preventing any misinformation that could be posted by malicious users

Originator: Eugenio Perez

Fit Criterion: App has no way to edit on screen information

Customer Satisfaction: 3

Customer Dissatisfaction: 3

Priority: Medium

Created: March 22, 2020

Requirement #7b2

Description: Database information must be exclusively writable from the server side

Rationale: To reduce the number of possible error sources, only developers should have access to writing and updating the database.

Originator: Eugenio Perez

Fit Criterion: While a direct line to the servers is present, developers are able to post update queries to the database.

Customer Satisfaction: 5

Customer Dissatisfaction: 4

Priority: High

Created: March 22, 2020

7c Privacy Requirements

Requirement #7c1

Description: Inform user application will need access to their location

Rationale: a part of the application's functionality is that it will use the phone's GPS to triangulate the user's location in our virtual map. Many people have hangups about allowing permissions for any part of their phones, but in this case it is necessary

Originator: Eugenio Perez

Fit Criterion: Upon installing the app, the device will ask the user permission to use their location

Customer Satisfaction: 7

Customer Dissatisfaction: 2

Priority: High

Created: March 22, 2020

Requirement #7c2

Description: Application will NOT store user's personal information

Rationale: There is no "account creation" when it comes to using this app, all the information that is present in our database is public knowledge. There is no reason we need to track or store user's information

Originator: Eugenio Perez

Fit Criterion: App has no account creation

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Priority: High

Created: March 22, 2020

7d Audit Requirements

Requirement #7d1

Description: Work closely with the National Park System to ensure that we have permission to use maps of their parks in our app

Rationale: Using names or images from national parks falls under copyright, and thus requires permission from the government to use in our app

Originator: Eugenio Perez

Fit Criterion: Permission is granted to use the names and images of National Parks in our app

Customer Satisfaction: 10

Customer Dissatisfaction: 10

Priority: Very High

Created: March 25, 2020

7e Immunity Requirements

See Requirements 7b1 and 7b2

8 Usability and Humanity Requirements

8a Ease of Use Requirements

Requirement #8a1 Efficiency of Use Requirement

Description

It should be easy for intended users of the product to operate it. The abilities of the expected users is minimal and the functionality should be usable enough so that the complexity is minimal.

The product will allow the user to quickly use the product with any use under 2 minutes.

Rationale

We need these ease of use requirements so that we can eventually meet the expectations of the users.

Originator: Kaveesha Weerasiri

Fit Criterion: 85 percent of a test panel of people 10 years or older shall be able to successfully complete posting to a gallery and looking at a map within 15 seconds.

Customer Satisfaction: 7

Customer Dissatisfaction: 4

Priority: Medium

Created: April 1, 2020

Requirement #8a2 Efficiency of Remembering Requirement

Description

It should be easy for intended users of the product to operate it. The abilities of the expected users is minimal and the functionality should be usable enough so that the complexity is minimal.

There should be little to no remembering needed to use the product.

Rationale

We need these ease of use requirements so that we can eventually meet the expectations of the users.

Originator: Kaveesha Weerasiri

Fit Criterion: Users should be able to pick up the app and not need to remember how to perform certain actions

Customer Satisfaction: 7

Customer Dissatisfaction: 4

Priority: Medium

Created: April 1, 2020

Requirement #8a3 Error Rates Requirement

Description

It should be easy for intended users of the product to operate it. The abilities of the expected users is minimal and the functionality should be usable enough so that the complexity is minimal.

Error rates should be low, under 5 percent, but it is okay for the most part for users to make occasional errors as long as it does not affect the APIs, the database, or the donations.

Rationale

We need these ease of use requirements so that we can eventually meet the expectations of the users.

Originator: Kaveesha Weerasiri

Fit Criterion: One month's use of the product will result in a total error rate of less than 5 percent.

Customer Satisfaction: 7

Customer Dissatisfaction: 4

Priority: Medium

Created: April 1, 2020

Requirement #8a4 Overall Satisfaction Requirement

Description

It should be easy for intended users of the product to operate it. The abilities of the expected users is minimal and the functionality should be usable enough so that the complexity is minimal.

Customers should be satisfied with the product, retaining user's most of the time (90 percent)

Rationale

We need these ease of use requirements so that we can eventually meet the expectations of the users.

Originator: Kaveesha Weerasiri

Fit Criterion: An anonymous survey shows that 85 percent of the intended users are periodically using the product after a one week familiarization period.

Customer Satisfaction: 7

Customer Dissatisfaction: 4

Priority: Medium

Created: April 1, 2020

Requirement #8a5 Feedback Requirement

Description

It should be easy for intended users of the product to operate it. The abilities of the expected users is minimal and the functionality should be usable enough so that the complexity is minimal.

There should be a moderate amount of feedback when the user requests his/her walk through map, when donating, and when updating the gallery. This feedback would include confirmation that things have been sent and received.

Rationale

We need these ease of use requirements so that we can eventually meet the expectations of the users.

Originator: Kaveesha Weerasiri

Fit Criterion: Feedback is given through the use of notifications, emails, and updates to the users screen

Customer Satisfaction: 7

Customer Dissatisfaction: 4

Priority: Medium

Created: April 1, 2020

Examples

The product shall be used by people who have no experience or training in park policies and possibly by people who do not understand English

The product shall be easy for anyone above the age of 10 to understand and use

The product shall help the user to avoid mistakes when donating, updating the gallery, or getting map and guide data of the park

The product shall be made in such a way that people will want to come back to it later

Considerations

The park editor, the client, and the system administrator will be contacted separately to gain knowledge on which considerations should be built into the application.

8b Personalization and Internationalization Requirements

Requirement #8b1- Language Requirement

Description

Have at least English present in the application because for the first release the application will deal with parks in the United States

Rationale

People inside of the United States will use the application first

Originator: Kaveesha Weerasiri

Fit Criterion: Users should be able to use the app in English

Sharing location is optional

Different currencies are supported

English is an option, but other languages choices are given as well

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

Requirement #8b2- Currency Requirement

Description

Any currency outside of the United States can be converted to US dollars so that they can donate to the parks

Originator: Kaveesha Weerasiri

Fit Criterion: Different currencies are supported

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

Requirement #8b3- Volunteering Requirement

Description

Configuration of maps and showing only volunteering organizations that one is interested in will be possible

Rationale

The user should not have volunteer listings that do not interest them because it will cause them to lose interest in the app

Originator: Kaveesha Weerasiri

Fit Criterion: Volunteering organizations are shown based off user needs and can be chosen through a survey

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

Requirement #8b4- Location Requirement

Description

Users will be able to choose whether to share location information

Rationale

Users should not need to give location if they do not want to

Originator: Kaveesha Weerasiri

Fit Criterion: Sharing location is optional

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

In general:

Personalization is important so that people will want to continue to use the product in the future. If customers are pleased with what they can do with the application, the stakeholders will want to continue to support the application, which is what we want. The product's users will not need to accept the builder's cultural conventions.

Examples

The product shall allow the user to select a chosen language.

The product shall let the user enter a different currency and still let them donate to the different parks.

Considerations

People outside of the United States will be able to change the keyboard inside the application to fit the users native language so as to not discriminate against them.

In terms of configurability, the general public will have access to maps, guides, and volunteering groups. Park employees can update park information, create and update guide information, and update volunteer events.

8c Learning Requirements

Requirement #8c1 - Learning Requirements

Description

There should be little to no learning curve to use this product. Thus the learning curve has zero time.

Rationale

The amount of time that is needed before a user can successfully use the product will be around 15 minutes. This includes download time and understanding how to use the application's UI. The product should have its functionality apparent upon first encountering it.

Originator: Kaveesha Weerasiri

Fit Criterion

After receiving 20 minutes of training a person staffed at a park shall be able to help 3 people in using the app in 7 minutes.

The engineers shall achieve 99% pass rate from the final examination of the training.

An engineer shall produce a map retrieved from Google api within 10 minutes of beginning to use the product, without needing to use the manual.

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

Examples

The product should be easy for an engineer to learn.

The people potentially staffed at different parks shall be productive within a short period of time.

The product shall be able to be used by members of the public who will receive no training before using it.

Considerations

The visitors to the park and the park employees are both considered in the ease of learning requirements.

8d Understandability and Politeness Requirements

Requirement #8d1 - Understandability and Politeness Requirements

Description

The understandability of the application will be given to them through the advertisements that are given throughout the parks that the application is being used in. It will fit their view of the world through giving them varying information about the park that they are in.

Rationale

The users will not need to learn terms and concepts that are part of the product's internal construction which would not be relevant to the user. It is to make going to a park easier and thus it will be adopted by users.

Originator: Kaveesha Weerasiri

Fit Criterion: Needs to be understandable to 100 percent of the users if possible.

Have accessible maps that anyone can understand

No use of internal information should be displayed in the app

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

Requirement #8d2 - Politeness Requirements

It will be polite in the sense that it will give them an interactive map, park specific rules and information, survival tips, a weather forecast, and a list of volunteering groups. It is trying to help the user. The user will not need to learn anything not related to the business problem.

Rationale

The users will not need to learn terms and concepts that are part of the product's internal construction which would not be relevant to the user. It is to make going to a park easier and thus it will be adopted by users.

Originator: Kaveesha Weerasiri

Fit Criterion: Needs to be polite to 100 percent of the users if possible.

Have accessible maps that anyone can understand

No use of internal information should be displayed in the app

Customer Satisfaction: 8

Customer Dissatisfaction: 4

Priority: Medium

Created: April 2, 2020

Example for understandability

The symbols on the walk through map and 2D map will be understandable because it is from google, and part of their goal is to make their products accessible and understandable to as many people as possible.

Example for politeness

The product will not show redundant or not useful things to the user such as things that only park rangers or system administrators should know.

Consideration for both requirements

The visitors to the park, the park employees, and the system administrators are all considered in the ease of learning requirements.

8e Accessibility Requirements

Requirement #8e1

Description

The application must have a zoom feature available.

Rationale

Users that are partially sighted might need to increase the size of the application for legibility.

Examples

The product shall be usable by partially sighted users.

Considerations

A user can lose their glasses or contacts. The ability to zoom on the app can give the opportunity to make the application more legible to partially sighted users.

Requirement #8e2

Description

The application must have a color blind feature available.

Rationale

Users that are color blind might have a hard time viewing the different images or colors on the website

Examples

The product shall conform to the Americans with Disabilities Act.

Considerations

A user can be color blind to certain colors. This gives the user the ability to view the application more comfortable at their leisure.

8f User Documentation Requirements

NA

Reasoning: We intend to design the app in such a way that it is accessible, and intuitive. As such, there is no need for any user manual or documentation for our app.

8g Training Requirements

Requirement #8g1 - Training Requirements

Description

The users will not need training to use the product.

Rationale

Since there is no training needed, there are little to no expectations for the training.

Originator: Kaveesha Weerasiri

Fit Criterion: Users should be able to use the product without any prior training

Customer Satisfaction: 7

Customer Dissatisfaction: 4

Priority: Low

Created: April 2, 2020

Considerations

If any training is necessary, that would be in part by the people who create the application. They need to be trained how to create and maintain the application.

9 Look and Feel Requirements

9a Appearance Requirements

Description

The application must be attractive and inviting to young adults through an older audience. Must conform with the companies logos and appropriate sponsors page.

Rationale

The average age of travelers are early 20s to mid 40s. Application should aim to target that age group.

Fit Criterion

The product's appearance should be accepted as simple and inviting by at least than 75% of testers.

9b Style Requirements

Description

The application should set a mood for users to feel welcomed and invited to visit national parks. However, parks have rules and policies. There should be a style of authority to enforce these rules of safety.

Rationale

There can be users that might not be informed of the dangers of ruining natural habitats. Nature can also be very dangerous, and safety should not be taken lightly.

Fit Criterion

After the first release, 70% of the users that have interacted with the product can feel that all national parks should be respected.

10 Operational and Environmental Requirements

10aExpected Physical Environment

Requirement #10a1 - Product must have a “bright” mode and a “dark” mode

Description

Application will come with the option to use a “bright” mode which has more vibrant backgrounds, or a “dark” mode which is easier on the eyes, especially at night.

Rationale

If the user uses the app at night, they might not want their phone to be sending a bat signal to the sky. I have personally been in situations where I would go camping at night, and when checking

my phone, I'd unintentionally blind myself with some app with a very bright background. We must assume users will use this app no matter the time of day.

Originator: Eugenio Perez

Fit Criterion: User is able to switch between two themes

Customer Satisfaction: 7

Customer Dissatisfaction: 5

Priority: Medium

Created: March 25th, 2020

Requirement #10a2 - Product must work without internet

Description

In the event that there is no internet, the application will seamlessly transition to cached data that was previously downloaded.

Rationale

This app is intended for use inside National Parks, if it doesn't work in dead zones, the user would be less inclined to have the app installed.

Originator: Eugenio Perez

Fit Criterion: User is still able to pull information such as park hours, maps, weather, and volunteer group events even without internet

Customer Satisfaction: 7

Customer Dissatisfaction: 10

Priority: High

Created: March 25th, 2020

10bRequirements for Interfacing with Adjacent Systems

Requirement #10b1 - Database must be able to pull information from weather sites

Description

Park database must be able to make a GET request to a weather website to pull forecast information when prompted by the user.

Rationale

Resources for weather forecasting already exist, it's redundant to code our own.

Originator: Eugenio Perez

Fit Criterion: System is able to accurately display weather information:

- a 5 day forecast, with each day containing a high temperature, a low temperature, and the climate for that particular day.
- This information will be retrieved via a GET request. It will then be stored in a data object, which will then be pushed to our database where the information will be updated.
- This will be done automatically whenever the user checks the weather for the national park they have chosen.
- The information retrieved from GET requests will be lightweight, and stored in strings.

Customer Satisfaction: 7

Customer Dissatisfaction: 10

Priority: High

Created: March 25th, 2020

10cProductization Requirements

Requirement #10c1

Description: The product must be available for download on google play store or the apple store, free of purchase. There we have to meet their requirement terms and regulations.

Rationale: It is important that everyone has access to the application from their device regardless of their device's operating system.

Originator: Cecilia Avila

Fit Criterion: User has successfully downloaded the application from their device and can begin using it.

Priority: High

Requirement #10c2

Description: The product should be able to easily be installed and usable for the user.

Rationale: The application should be user friendly and should not require an instructions manual or a trained expert as it is not intended to be complex.

Originator: Cecilia Avila

Fit Criterion: User can use application functionalities without difficulty. Testers and users will review their experience to see how efficient and user friendly their experience was.

Priority: Medium

10d Release Requirements

Requirement #10d

Description: The product should be released every 3 months to continually update features and enhance user experience.

Originator: Cecilia Avila

Rationale: Parks information must be updated and application functionalities must work as intended. It is important that an application aspect gets reported to be inspected and fixed.

Fit Criterion: The product will be inspected after releases and search for bugs. Complication reports will be recorded to keep track of what needs to be updated and fixed.

Priority: Medium

11 Cultural and Political Requirements

11a Cultural Requirements

Description

The product shall not be offensive to any religious or ethnic groups.

Rationale

The parks are intended for anyone to be able to visit. Any controversial pictures, posts, reviews, or statements will be reported and removed from the product. Any informational tool, rules, or

policies that the product shall provide should not offend any religious or ethic groups. Some users might be international with different customs. Rules, policies, and content must not offend anyone.

Fit Criterion

Surveys of the product will have a section about offensiveness to ensure the product has minimal to no offensiveness.

11b Political Requirements

Description

The product shall not take any political stance.

Rationale

There might be donations and organizations that support the wild parks, but there can not be an influence. This can potentially bring other factors that have an influence on the product's appearance.

12 Legal Requirements

12a Compliance Requirements

Description

The product shall have a statement regarding the compliance in each state that it will be used in.

Rationale

To comply with the law so as to avoid later delays, lawsuits, and legal fees.

Fit Criterion

Law firm's revisal and opinion that all legal requirements are met, and ensure any part of the product that requires a copyright law.

12b Standards Requirements

Description

Product shall have the appropriate rights to display each park's information publicly.

Rationale

Copyright and permissions of all data usage is approved.

Fit Criterion

All pending copyright approval should be received before first release.

III Design

1 System Design

1a Design goals

- We want Teddy's Brochure to be easy to use, and intuitive for the average individual
- Teddy's Brochure must retain some functionality even if an internet connection is not present. Thus reliability will be an important aspect of our application.
- We want this app to be a trove of pertinent information for all 62 National Parks.
- We want reliable connections and to optimize those connections to the database so that each request is met.
- We want to adapt to the changing environmental situation due to covid-19 and will update the application accordingly

2 Current Software Architecture

The application is going to give the separate entities a common ground for them to operate, increasing customer satisfaction.

Model-View-Controller (MVC)

- Model: This will be the database that stores all of the different park information so that when the user requests a park the application will give the right information
- View: This will be the different screens of the application
- Controller: This will be the different menus of the app which would include the volunteering, map, gallery, and history tabs

3 Proposed Software Architecture

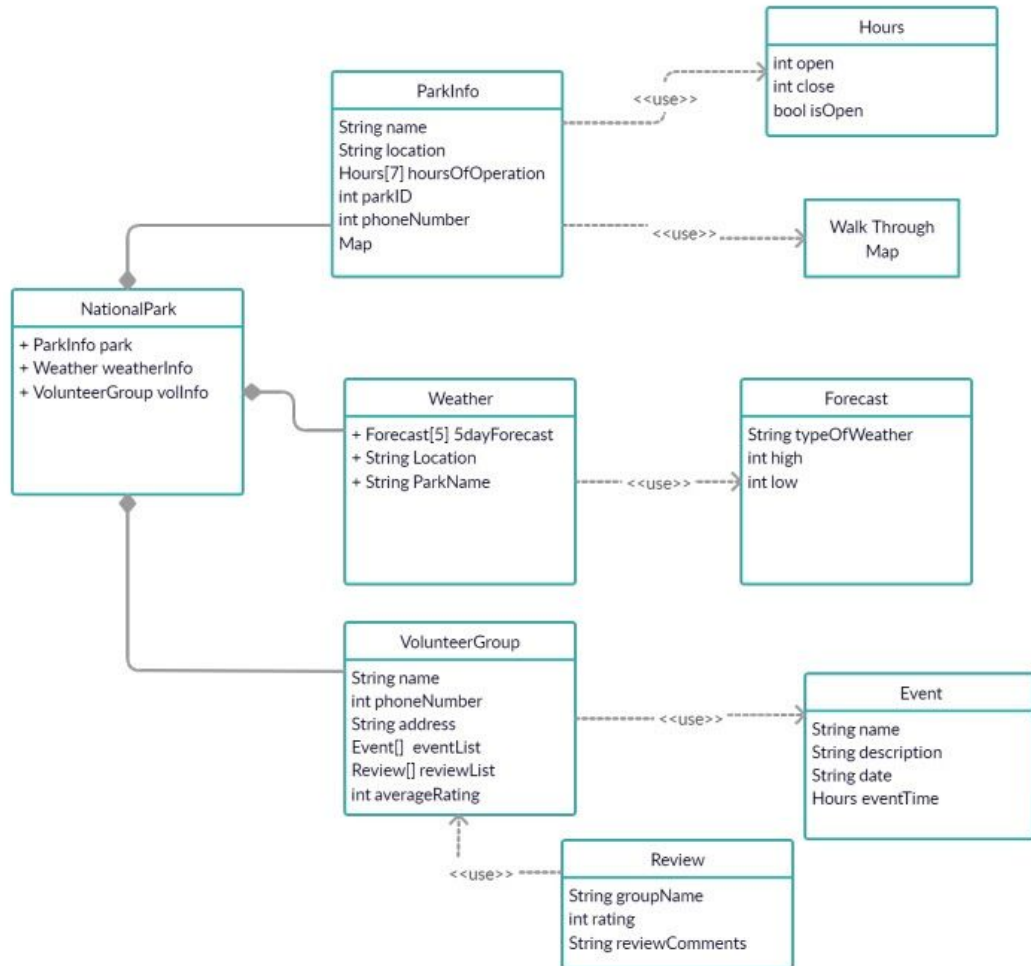
3a Overview

This application will live and die by the database. The database will serve to store all the information while an object oriented language will be used to pull and parse that information into views.

1. Database
2. Park List
3. National Park

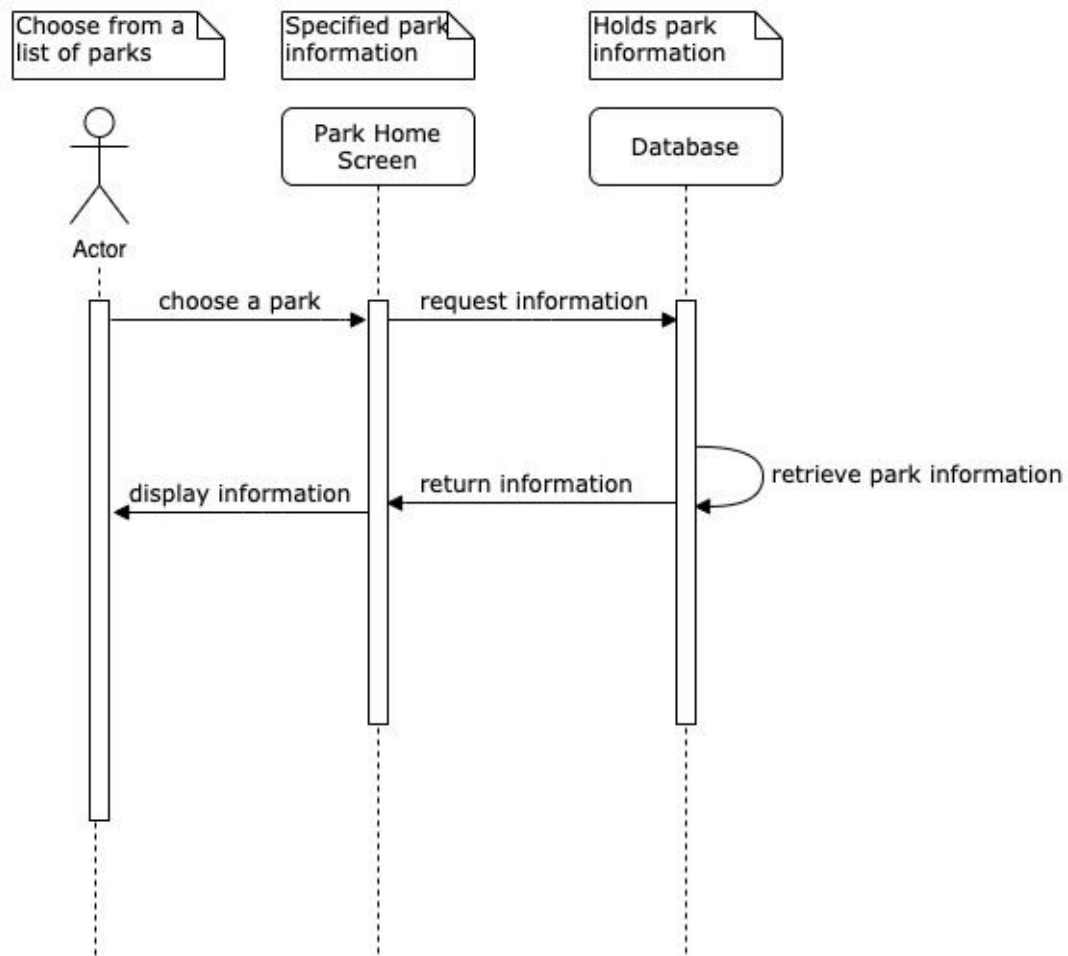
4. User interface
5. Review Forms

3b Class Diagrams

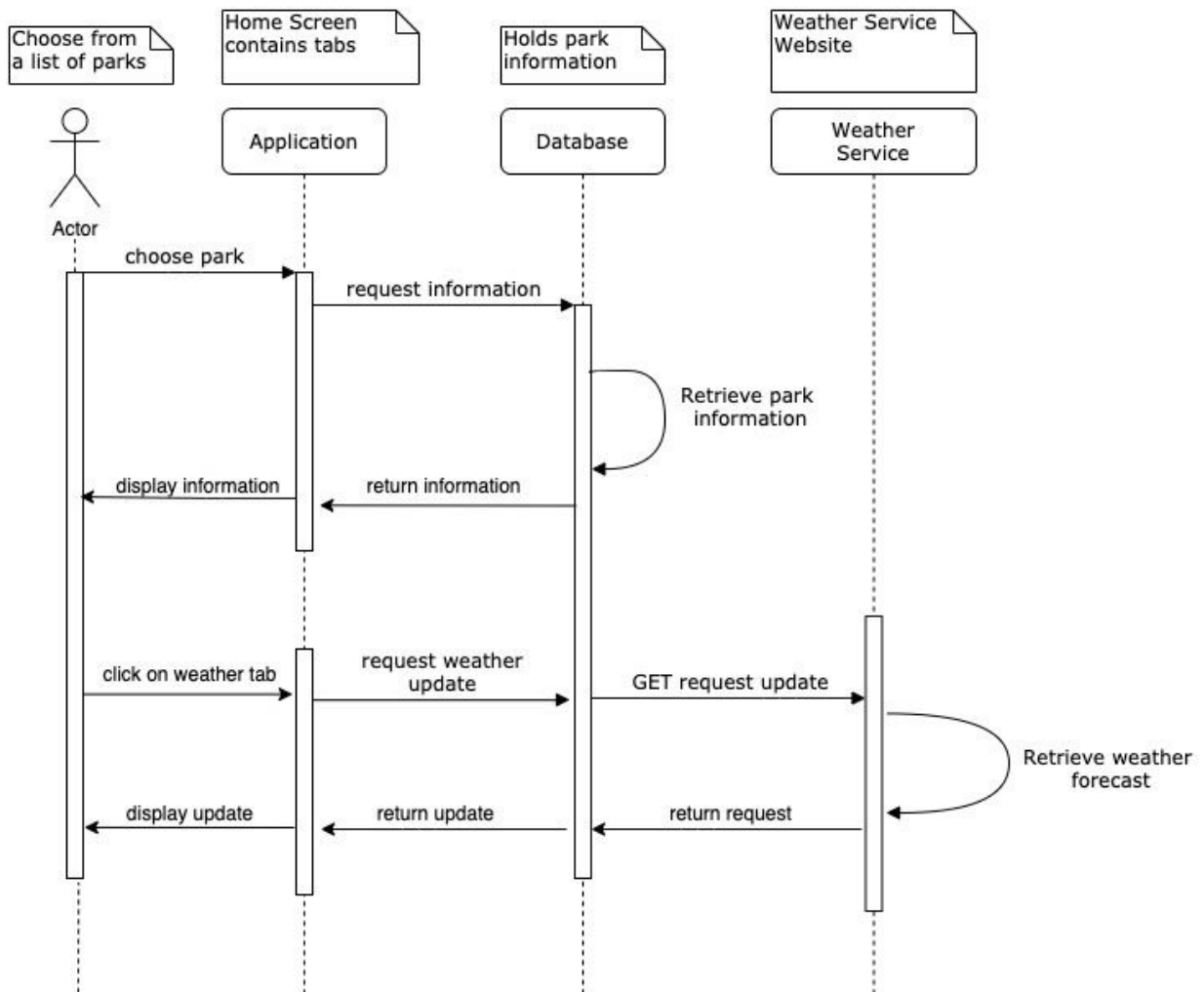


3c Dynamic Model

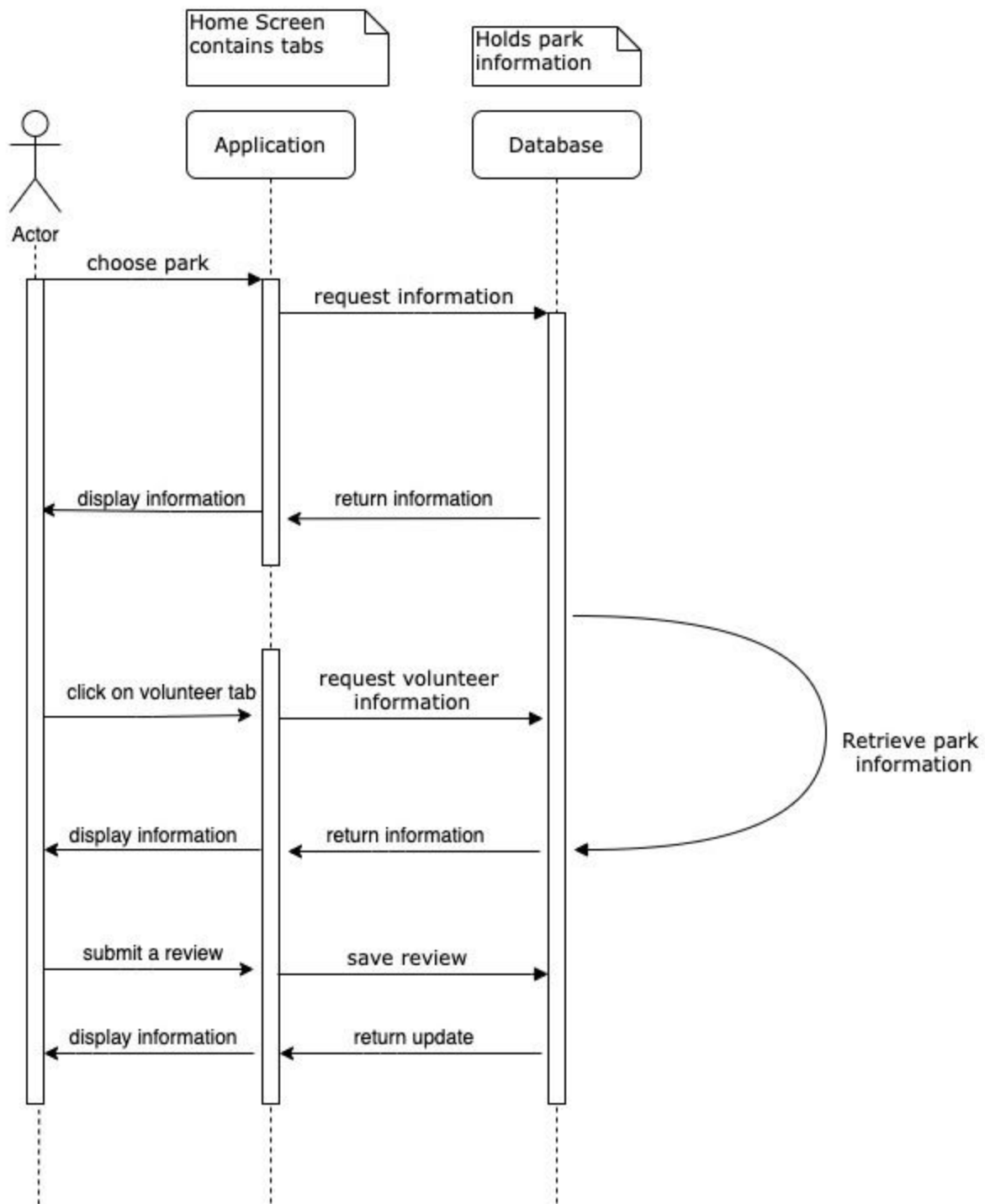
Access Park Data



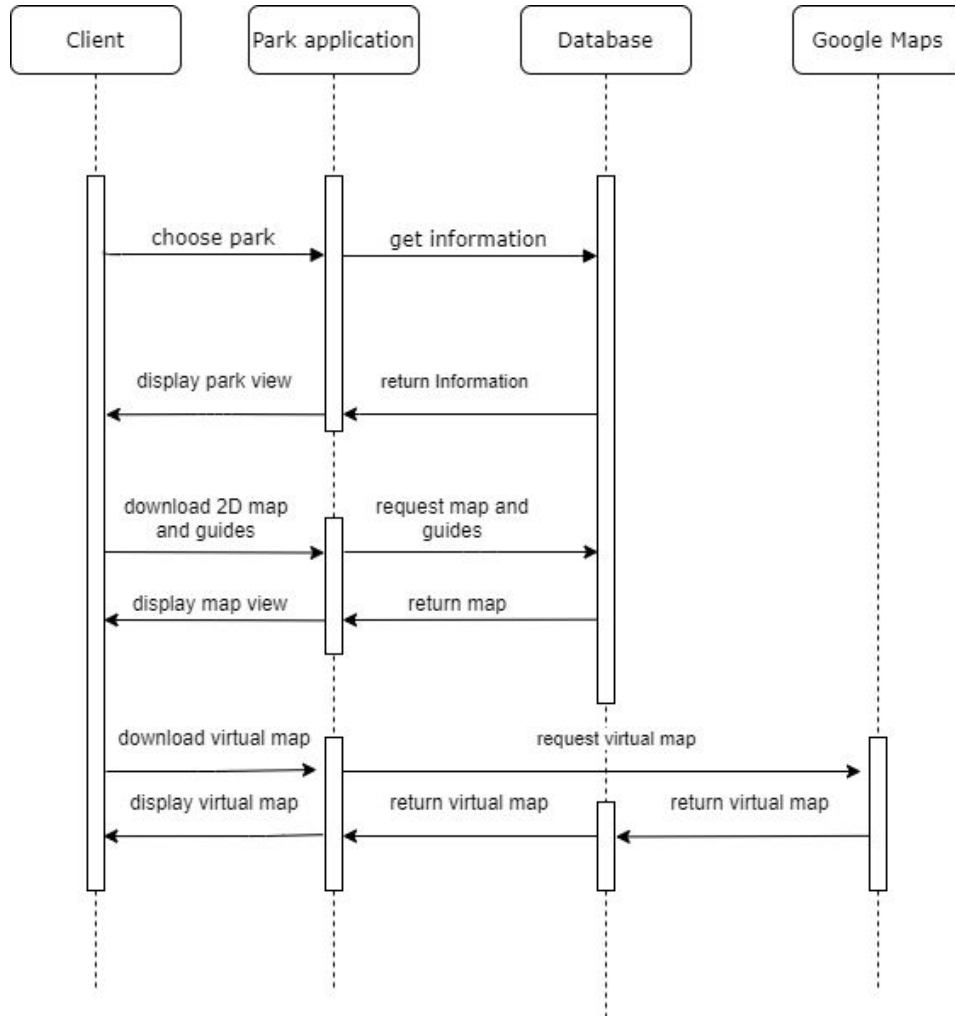
Report Weather



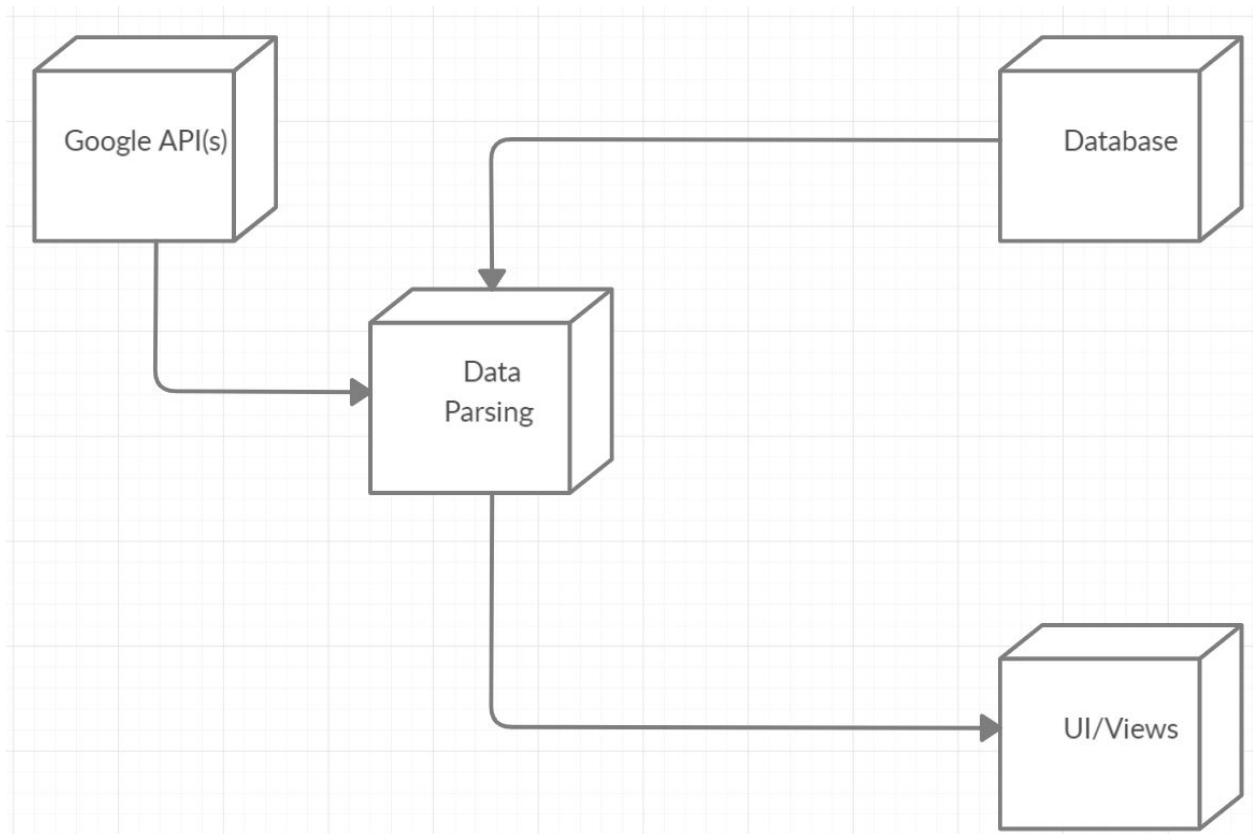
Volunteer Information



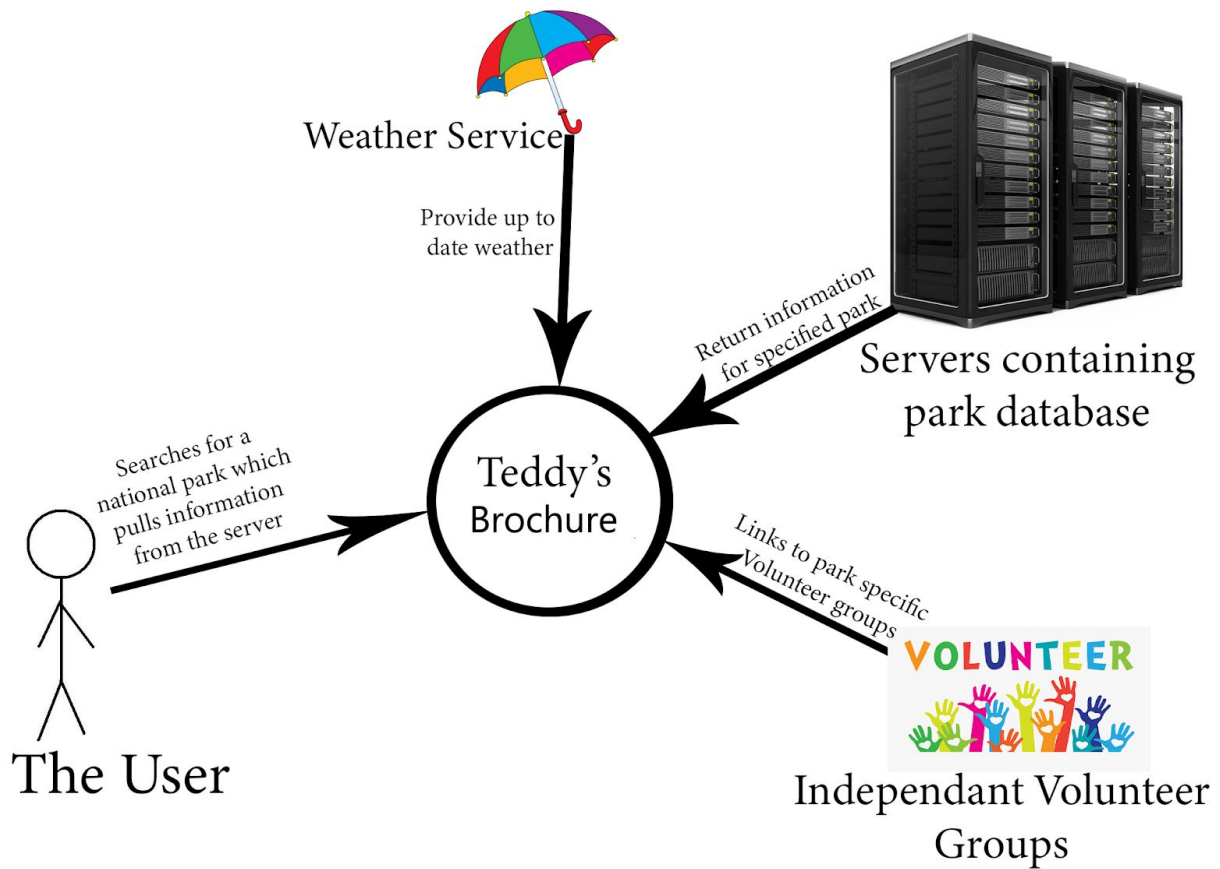
Download Map Data



3d Subsystem Decomposition



3e Hardware / software mapping



3f Data Dictionary

N/A

3g Persistent Data management

The app itself will cache the data so that if the system fails it will still be there for the user when the system starts back again. The app would use technologies like UserDefaults, KeyChain, and CoreData for iOS and shared preferences or SQLite for android. Data like maps and guides that the user wants to download will be stored locally on the device. This should stay on the device until removed by the user. This will ensure that the reliability for the user to look at the data without an internet connection.

3h Access control and security

Sensitive data of users' location may be exposed if not carefully considered. People may use the gallery picture locations if enabled to find out where users are at that time. This can be avoided by making our app transfer data via https and discarding sensitive data if not needed. Broken access control may occur as well if the administrators do not verify roles properly. This can be solved through the use of regression tests where the user is brought back to the start page if they are trying to access a page they are not authorized to.

3i Global software control

The Database will be independent from the app itself for the most part. However, the app should have full access to the information that is stored in the database, and should not have any issues with permissions.

3j Boundary conditions

The application should be responsive no matter where the user is in the park. Walk through maps should work when there is inclement weather.

Reviews should not be able to go through if the user tries submitting an empty review form.

4 Subsystem services

The Java code will manage the views, and parse the information in the database, while the database will store all the park information.

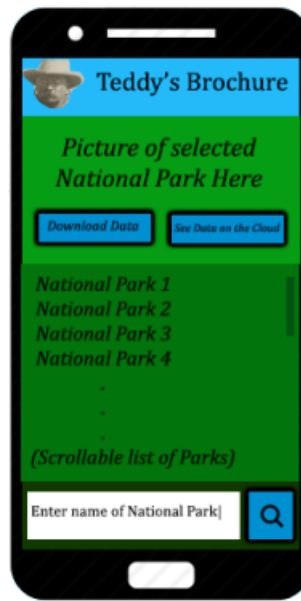
The database will be mostly populated by manual input, but for data that constantly changes, such as the weather, that will be updated on an hourly basis autonomously. But by and large, the database subsystem will mostly be used for distributing information to the application.

For database management, the programmer will be able to update the database from java when needed, but it will obviously be limited. This is best done on Azure, or some other SQL workbench.

The review subsystem will take user input in the way of a form. To discourage multiple reviews, we will tie each review to the user's unique device, and only allow one review per park or Volunteer Group.

5 User Interface

The splash screen should have a scrollable list of national parks, as well as a text field that will allow the user to search for a specific park.



When the user selects a national park, it should take them to a screen that contains all required information in a format that is easily readable. Functionality should come before aesthetic, so it is crucial that the programmer creates a system that allows for information to be easily parsed and distributed to all views used in the application.



6 Object Design

6a Object Design trade-offs

Performance might be a trade off for reliability of the application. This will be a mobile application. Therefore, if many users are trying to download the park maps or guides. Reliability will be prioritized over best possible performance.

Cost for the reusability of the code and system. There are going to be several parks listed on the website with a lot of information for each one. The cost of the project can be more expensive due to the high amount of reusability of the system. Very careful planning and organization is required to efficiently drive the project. More time might be taken in the actual creation of the project so the cost of the project might increase.

6b Interface Documentation guidelines

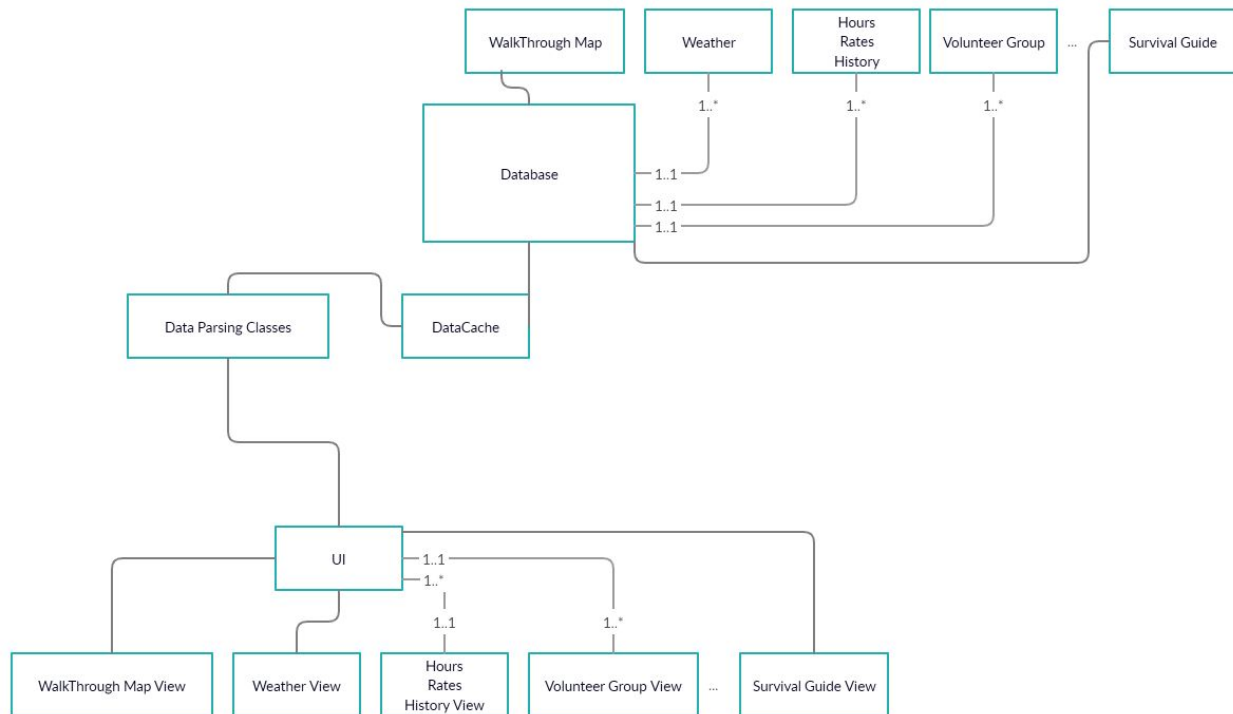
Classes are always uppercased

Note there is a distinction between database entities and java information objects, although total number of them should be 1:1

6c Packages

The packages would include Java, Kotlin, and SQL packages since this will be a mobile app.

6d Class Interfaces



IV Test Plans

1 Features to be tested / not to be tested

Features to be tested

1. Test whether donations go to the right organization.
2. Test whether google maps shows the correct map based on location.
3. Test whether users can place reviews on different volunteering organizations.
4. Test whether users can post to the gallery.
5. Test whether different parks can be chosen and the application formats based on that.
6. Each park will be tested for correct information displayed
 - a. information
 - b. history
 - c. maps
 - d. park guides
 - e. reviews
 - f. donations

Features not to be tested:

These features will not be tested for correctness of information in detail. The map might have updated maps in the future. This information will not be the same.

1. guide information
2. park information
3. history of the park

4 Pass/Fail Criteria

1. Donations to a specific organization should be correctly issued
2. When google maps api is called ensure that the location and map are correct
3. Any user is able to place a review on a event
4. Any user is able to upload a picture 90% of the time to the gallery page
5. Every park that is available at time of testing should be displayed. Each park should display the correct information and location.
6. A-B Each park information should be correct. The general file should correspond with the national park. No other park's information should be displayed in another park.

5 Approach

All tests will be automated with scripts. Then check corresponding files from the server that have been sent to the user. This will ensure that the application is receiving and sending appropriate files.

6 Suspension and resumption

If the application is not supplying many customers, then suspension may be used to save unnecessary consumption of resources. The testing may resume when user interaction with the application is high because the software needs to function during peak demand.

7 Testing materials (hardware / software requirements)

A mobile device with at least version 8.0 on Android and iOS 11 and internet connection. A Windows, Apple, or Linux based machine with an internet connection.

8 Test cases

- Test reliability for connections to the database
- Test retrieving information for each park
- Test download packages
- Test retrieving maps from googles API
- Test retrieving weather update from a weather service website
- Test database for functionality, secure safety, and ensure user reviews are saved and reflected on the database

9 Testing schedule

We will test according to which functionality gets implemented first. This will be done through the use of regression testing so that the code that is already there will not be disrupted. These tests will be automated and will be run every weekend to ensure maximum customer satisfaction.

V Project Issues

1 Open Issues

We do not know if the project will be able to get the funding it needs so that it is able to be completed. We do not know if the Trump administration will cancel all funding provided to the parks, so this application may not be viable in that instance. Of course, there is also the issue of whether parks will open back up and be used the same way as before the coronavirus. Budget cuts to the parks do not help the situation

2 Off-the-Shelf Solutions

The National Park website is a solution that can be improved upon and already has a couple of functions that will be implemented.

2a Ready-Made Products

The National Park website is a solution with several functions implemented, <https://www.nps.gov/index.htm>.

2b Reusable Components

All parts of the data can be reused into different display models. The general format of a park is very important because there are many parks. The information should have the ability to be applied to a model.

We can use components of the national park service website so that we do not have to

2c Products That Can Be Copied

N/A

3 New Problems

Security will be a concern for donations. Very highly tested and secured code for donations will be mandatory. If users create an account with payment information then the information will have to secure

3a Effects on the Current Environment

This system should not have any real adverse effects on the working environment. However, the application should not try to replace volunteering organizations but rather put everything into one application making it easier for the user to find what they need to find. People who may still be handing out park information or guides may be out of a job and have to find other ways to help.

3b Effects on the Installed Systems

The only effects that the application will have a negative impact is memory storage. The maps and guides for the user will have to be stored on the user's device for offline use.

3c Potential User Problems

The purpose of the application for offline users is to have a map and guides in case of emergencies. So downloading these maps and guides before going into the National Parks with a stable internet connection is very important. The National Parks may not have cellular connections at all. In this case, if the user needs any documents then they will not be able to.

3d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The lack of power may inhibit the product from working because the servers may be down if the application is not computed from the cloud. Severe weather may impact the application indirectly because volunteer events may not be possible, affecting which volunteering events are listed in the app. The server may not be powerful enough to cope with the projected growth pattern.

3e Follow-Up Problems

There is no internet connection close to the user so the user has to travel long distances to download the app. Some users may not have access to a smart device that allows them to download the application.

4 Migration to the New Product

N/A

4a Requirements for Migration to the New Product

N/A

4b Data That Has to Be Modified or Translated for the New System

N/A

5 Risks

Some potential risks include a pandemic occurring or war that forces people to stay inside. Another risk is that the people may want to experience virtual reality parks instead of actual parks. Other risks include low productivity, low quality, excessive schedule pressure from the people who are running the project.

6 Costs

Costs of the project will be minimal because most of the information in the app will already be accessible by the user via the national parks. The amount of money used to create the application would be \$47,000. This is because it will be a basic app without too many ground breaking ideas to implement via code. The amount of effort will be a moderate amount. The amount of time to build the app will be half a year. There will be around 3 business events.

7 Waiting Room

Application will be compatible with phone VR headsets so that people can truly experience being inside of the park. App lets users talk to customer support at the park.

8 Ideas for Solutions

Ideally, the user is going to want to use the flutter SDK because this app should work on Android and iOS devices. However, if the programmer is comfortable using Android Studio, that is sufficient for the Android side of development. But they should make an effort to include an iOS implementation as well.

The programmer should be able to make sure that when the user downloads maps for a given national park, that it doesn't take up too much of their device's memory.

9 Project Retrospective

It has been an enriching experience understanding how to create a solution from a prompt and create a system designed to tackle that prompt. One thing that did not work was that the project is something that would take about 6 months and not a couple of years. However, it is a good solution to the problem and would solve the issue of not having information offline while visiting national parks.

We tried to divide up the work evenly, and for the most part that is what happened. The process could be improved in the future by checking in on people's work more frequently to really make sure that everyone is on the same page, but that occurred anyway.

If covid-19 were happening when the project started, we may have implemented something to the effect of watching animals and track their vitals to make sure none of them are neglected.

VI Glossary

Gallery: a server where people post pictures of the national park

Package: The application that is enhancing the national park service to the customers

Tips: hints on how to camp and hike at the national parks

Master: has complete knowledge of a domain

Maps: 2D maps

Walk through map: Maps of the specific parks that people can interact with (google maps)

Brochure: a one stop shop for all of your park needs

VR: a simulated experience similar to the real world

API: allows access to a set of data or functionality that can be used in the application

Database: an organized collection of national park data in the United States

App: short for the mobile application we are developing in this document

VII 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.

VIII Index

Database	8, 9, 10, 14, 18, 19, 20, 25, 33, 36, 37, 40, 51	
Volunteer	44, 46, 54, 55, 63, 66	National Park 1-5, 6-10, 11-15, 33-40,
Brochure	13, 14, 25, 29	55, 58, 61

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