6-10-2019

GalápaGuide

Neel Sampemane
Stephen Pacwa
Mason Bruce
Paul Ahrens

Follow this and additional works at: https://scholarcommons.scu.edu/cseng_senior

Part of the Computer Engineering Commons

Recommended Citation
Sampemane, Neel; Pacwa, Stephen; Bruce, Mason; and Ahrens, Paul, "GalápaGuide" (2019). Computer Engineering Senior Theses. 139.
https://scholarcommons.scu.edu/cseng_senior/139

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Computer Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact rscroggin@scu.edu.
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Neel Sampemane
Stephen Pacwa
Mason Bruce
Paul Ahrens

ENTITLED

GalápaGuide

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING

Thesis Advisor

Department Chair
GalápaGuide

Neel Sampemane
Stephen Pacwa
Mason Bruce
Paul Ahrens

Department of Computer Engineering
Santa Clara University
June 10, 2019

ABSTRACT

In collaboration with the Ministry of Tourism of Ecuador, the Association of Guides on the Galápagos, and Santa Clara University’s Frugal Innovation Hub, we developed a mobile application and administrative website connected through a REST API to a backend database as tools to assist naturalist guides on the Galápagos archipelago in conservation efforts. The goal of this system is to enable the naturalist guides and administrators to report issues with greater efficiency and view the data in a more useful way. This report outlines our design for this project and the reasoning behind our design decisions.
Acknowledgements

Throughout the course of the this project we received outstanding support from many people and organizations, without them this wouldn’t have been possible, so we would like to thank them here.

Ministerio de Turismo de la República de Ecuador
Juan Carlos Izurieta

Asociación de guías naturalistas del Parque Nacional Galápagos
Sofía Darquea

Frugal Innovation Hub, Santa Clara University
Allan Baez Morales
Dr. Silvia Figueira
# Table of Contents

1 Introduction ........................................ 1
   1.1 Background & Motivation .......................... 1
   1.2 Solution ............................................ 2

2 Requirements ....................................... 3

3 Use Cases .......................................... 5

4 Activity Diagrams .................................. 8

5 Final Design ....................................... 10
   5.1 Mobile Application ............................... 11
   5.2 Administrative Website ......................... 17

6 Technologies Used ................................ 24
   6.1 Mobile Application ............................... 24
   6.2 Administrative Webpage ......................... 24

7 Architectural Diagram ........................... 25

8 Design Rationale .................................. 26
   8.1 Web .................................................. 26
   8.2 Mobile .............................................. 26
   8.3 Architecture ....................................... 27

9 Testing ............................................. 28
   9.1 Verification ........................................ 28
   9.2 Validation ......................................... 28
   9.3 User Testing ....................................... 28

10 Risk Analysis .................................... 29

11 Societal Issues .................................. 30
   11.1 Ethical ............................................ 30
   11.2 Social .............................................. 30
   11.3 Political ........................................... 30
   11.4 Economic ......................................... 31
   11.5 Health and Safety ............................... 31
   11.6 Manufacturability ................................ 31
   11.7 Sustainability ..................................... 31
   11.8 Environmental Impact ......................... 31
   11.9 Usability .......................................... 31
   11.10 Lifelong learning ............................... 32
   11.11 Compassion ...................................... 32

v
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Use Case Diagram</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Guide Activity Diagram</td>
<td>8</td>
</tr>
<tr>
<td>4.2</td>
<td>Administrator Activity Diagram</td>
<td>9</td>
</tr>
<tr>
<td>5.1</td>
<td>Mobile application: Login Page</td>
<td>11</td>
</tr>
<tr>
<td>5.2</td>
<td>Mobile application: Map</td>
<td>12</td>
</tr>
<tr>
<td>5.3</td>
<td>Mobile application: Report form</td>
<td>13</td>
</tr>
<tr>
<td>5.4</td>
<td>Mobile application: Report detail</td>
<td>14</td>
</tr>
<tr>
<td>5.5</td>
<td>Mobile application: Outbox</td>
<td>15</td>
</tr>
<tr>
<td>5.6</td>
<td>Mobile application: Settings</td>
<td>16</td>
</tr>
<tr>
<td>5.7</td>
<td>Administrative Website: Map</td>
<td>17</td>
</tr>
<tr>
<td>5.8</td>
<td>Administrative Website: Modal</td>
<td>18</td>
</tr>
<tr>
<td>5.9</td>
<td>Administrative Website: Analytics</td>
<td>19</td>
</tr>
<tr>
<td>5.10</td>
<td>Administrative Website: Tags</td>
<td>20</td>
</tr>
<tr>
<td>5.11</td>
<td>Administrative Website: Import Reports</td>
<td>21</td>
</tr>
<tr>
<td>5.12</td>
<td>Administrative Website: Import One User</td>
<td>22</td>
</tr>
<tr>
<td>5.13</td>
<td>Administrative Website: Import Many Users</td>
<td>23</td>
</tr>
<tr>
<td>7.1</td>
<td>Architectural Diagram</td>
<td>25</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Functional Requirements</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Non-Functional Requirements</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>Design Constraints</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>Submit Reports Use Case</td>
<td>5</td>
</tr>
<tr>
<td>3.2</td>
<td>Search Reports Use Case</td>
<td>5</td>
</tr>
<tr>
<td>3.3</td>
<td>View Reports Use Case</td>
<td>5</td>
</tr>
<tr>
<td>3.4</td>
<td>Edit Reports Use Case</td>
<td>6</td>
</tr>
<tr>
<td>3.5</td>
<td>Comment on Reports Use Case</td>
<td>6</td>
</tr>
<tr>
<td>3.6</td>
<td>View Analytics Use Case</td>
<td>6</td>
</tr>
<tr>
<td>10.1</td>
<td>Risk Analysis</td>
<td>29</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background & Motivation

The Galápagos Islands are home to over twenty endangered species of animals, and over 260 threatened species of plant life, making the Galápagos Archipelago not only beautiful, but a unique habitat unlike anywhere else in the world. Because of the exotic nature of the islands, they host over 250,000 visitors every year, but the protected habitat that attracts all those people restricts their access to only 3% of the archipelago. Around 800 naturalist guides give tours of this limited area throughout the year, all of them well-trained to identify threats to the environment in the form of invasive species and to notice certain hazards that could result in trail closures or maintenance. Easy access to this information is vital for the Ministry of Tourism, National Park, and also for the guides to have in order to conserve the sensitive ecosystem of the islands, but with the limited technological infrastructure on the island this kind of connected reporting is difficult to achieve.

The current solution to this problem is insufficient and prone to error. When a naturalist notices an issue on patrol, such as an invasive species or a blocked trail, they must wait until they are back in town to report it. This information is submitted through an online form that collects their ID number, the date, a description of the problem, and any photos that they may have taken. If guides choose to report the violations at all, they are likely to be forgotten or mis-remembered because of the long delay between being discovered and reported. However, many guides simply don’t report violations due to the inefficiencies in the current solution.

Furthermore, by using an open-ended description of the problem, it is difficult to classify issues based on type or location and prevents the use of analysis. Finally, with the lack of GPS data, the location of the issue may be vague or misunderstood, leading to a slower response time or a waste of limited resources.

The Ministry of Tourism of Ecuador and Galápagos National Park reached out to us through The Frugal Innovation Hub at Santa Clara University School of Engineering, and over the course of this project we worked closely with them—and even traveled to the archipelago—in order to provide them with the best possible solution to this problem.
1.2 Solution

In order to better serve the naturalists on the archipelago, and provide a better overall experience for all parties, we developed a mobile application, a custom backend database, REST API, and administrative website which solve many of the problems outlined above. Naturalists will be able to use our app to take a photo of an invasive species or other abnormality and describe their observation as soon as they see a violation, that is then cached locally on their phone and uploaded once they reach a networked location and choose to upload their reports. Additionally, through the REST API, the application will download and cache details of all submitted reports so they can view reports from other guides while off-network. The administrative website provides administrative users with the ability to create user accounts, dynamically modify certain report fields, and view reports and trends through an analytics page.

Our solution solves the problem outlined above by providing a unified platform for naturalists to identify issues, collect data, and automate the process of passing this information along. Data in our solution will be more structured than in previous solutions, allowing for clearer insights into the issues reported around the island. The addition of GPS data to the reports will also help link issues to specific locations on the island, enabling faster response times. Lastly, when the reporting can be done right away, naturalists are less likely to forget the details of the situation. This will help improve the accuracy of reports and ensure that the correct people are sent to fix the problems. Ultimately, naturalists on the Galápagos Islands are inhibited from performing at their full capacity by limited and disconnected technological resources. Our solution will provide a holistic solution to a few of the problems they encounter. Our primary challenge has been overcoming the lack of wireless internet on the island, but with a few sacrifices this issue has been surmountable.
Chapter 2

Requirements

Establishing critical and recommended requirements helped streamline our development process and ensure that we were meeting the needs of our client in the Galápagos Islands. Our functional requirements are in Table 2.1, our non-functional requirements can be seen in Table 2.2, and Table 2.3 shows the constraints which limited our design.
### Table 2.1: Functional Requirements

<table>
<thead>
<tr>
<th>Priority</th>
<th>Requirements</th>
</tr>
</thead>
</table>
| **Critical** | - Allow users to submit reports  
- Allow users to search for reports  
- Allow users to view reports  
- Allow users to edit reports  
- Track the timestamp in the report  
- Include descriptions of the issues in the report  
- Provide the location of the incident or sighting in the report  
- Give aggregated information for administrative decision makers  
- Store the reports in a centralized place  
- Update the reporting data on user’s mobile device when they have access to the internet |
| **Recommended** | - Classify the reports based on tags  
- Provide an interface for other guides to comment on reports  
- Allow for parts of the report to be hidden |

### Table 2.2: Non-Functional Requirements

<table>
<thead>
<tr>
<th>Priority</th>
<th>Requirements</th>
</tr>
</thead>
</table>
| **Critical** | - Efficient to submit reports  
- Quick for users to search for reports  
- Reliable even when limited or no internet is available |
| **Recommended** | - Easy to view analytics related to the reports |

### Table 2.3: Design Constraints

<table>
<thead>
<tr>
<th>Constraints</th>
</tr>
</thead>
</table>
| - Reporting feature must operate as a mobile application on iOS and Android devices  
- Administrative portal must be accessible through an internet browser |
Chapter 3

Use Cases

We have two primary actors using our system. The naturalist guides will use the mobile application to report issues and incidents around the Galápagos Islands. Meanwhile, the administrators will help manage the reports and address the issues brought up by the guides. Tables 3.1-3.6 show our use cases and Figure 3.1 is our use case diagram.

Table 3.1: Submit Reports Use Case

<table>
<thead>
<tr>
<th>Goal</th>
<th>Be able to submit reports to the application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor(s)</td>
<td>Naturalist Guide</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>Actor has collected data related to the issue and filled out the form on the application</td>
</tr>
<tr>
<td>Postcondition(s)</td>
<td>Report is added to the database and is viewable on the administrative interface or the mobile application</td>
</tr>
<tr>
<td>Exception(s)</td>
<td>Actor does not have the mobile application on their device or they have not collected sufficient data to complete the form</td>
</tr>
</tbody>
</table>

Table 3.2: Search Reports Use Case

<table>
<thead>
<tr>
<th>Goal</th>
<th>Allow actors to search for reports in the application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor(s)</td>
<td>Naturalist Guide, Administrator</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>Actor has access to database holding the reports and has criteria for what reports they wish to find</td>
</tr>
<tr>
<td>Postcondition(s)</td>
<td>Actor sees a list of reports matching the criteria they input</td>
</tr>
<tr>
<td>Exception(s)</td>
<td>No access to the database, no reports matching the criteria, or no reports in the database</td>
</tr>
</tbody>
</table>

Table 3.3: View Reports Use Case

<table>
<thead>
<tr>
<th>Goal</th>
<th>Be able to view reports in the application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor(s)</td>
<td>Naturalist Guide, Administrator</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>Actor has access to database holding the reports, has searched for reports, and has found one they wish to view</td>
</tr>
<tr>
<td>Postcondition(s)</td>
<td>Actor sees the content from the report they want to view</td>
</tr>
<tr>
<td>Exception(s)</td>
<td>No access to the database, no reports in the database, or actor has not searched for reports</td>
</tr>
</tbody>
</table>
Table 3.4: Edit Reports Use Case

<table>
<thead>
<tr>
<th>Goal</th>
<th>Enable editing of reports in the application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor(s)</td>
<td>Naturalist Guide</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>Actor has access to database holding the reports, has searched through reports they have submitted, and is viewing one they wish to edit</td>
</tr>
<tr>
<td>Postcondition(s)</td>
<td>Actor has edited the values of the report on the interface, and those changes are reflected in the database</td>
</tr>
<tr>
<td>Exception(s)</td>
<td>No access to the database, no reports in the database, actor has not searched for reports they have submitted, or actor is not viewing a report</td>
</tr>
</tbody>
</table>

Table 3.5: Comment on Reports Use Case

<table>
<thead>
<tr>
<th>Goal</th>
<th>Be able to comment on reports in the application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor(s)</td>
<td>Naturalist Guide, Administrator</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>Actor has access to database holding the reports, has searched for reports, and is viewing a report that they wish to comment on</td>
</tr>
<tr>
<td>Postcondition(s)</td>
<td>One or more new comments is added to the report. These comments can be viewed when another actor views the report.</td>
</tr>
<tr>
<td>Exception(s)</td>
<td>No access to the database, no reports in the database, actor has not searched for reports, or actor is not viewing a report</td>
</tr>
</tbody>
</table>

Table 3.6: View Analytics Use Case

<table>
<thead>
<tr>
<th>Goal</th>
<th>Be able to view system analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor(s)</td>
<td>Administrator</td>
</tr>
<tr>
<td>Precondition(s)</td>
<td>One or more objects present in the database</td>
</tr>
<tr>
<td>Postcondition(s)</td>
<td>A map is populated with pins marking the GPS location of the reports and a separate panel shows analytics about data found in the system.</td>
</tr>
<tr>
<td>Exception(s)</td>
<td>No objects in the database</td>
</tr>
</tbody>
</table>
Figure 3.1: Use Case Diagram

- Naturalist Guide
- Administrator

- System
  - Submit reports
  - Search for reports
  - View reports
  - Edit reports
  - Comment on reports
  - View analytics
Chapter 4

Activity Diagrams

Our project will have two principle users. The guides who will use the mobile application and the administrators who access the website.

![Figure 4.1: Guide Activity Diagram](image)

Guides will be able to view reports that have been already made. Here they can make comments, read descriptions, and view attachments. They are also capable of submitting new reports that can have images attached as well.
Administrators will be able to manage user accounts, export or import report data, view application analytics, and view reports. When viewing a report they will be able to view the information and make comments.
Chapter 5

Final Design

Users will interact with GalapaGuide in 2 ways: naturalist guides will submit and view reports through a mobile application, and administrators will view reports through an administrative web portal. The figures in Section 5.1 represent the final user interface of the mobile application, and those in Section 5.2 represent the final interface of the administrative web portal.
5.1 Mobile Application

Figure 5.1: Mobile application: Login Page

The first page a user is presented with is a Login screen where users can log in to the app using the account they have been assigned.
Figure 5.2: Mobile application: Map

Having logged in, guides will be presented with a map where they can view reports around them and also submit reports.
Guides will be able to add reports directly within the app. They can add a detailed description, notes, or images of the issue they are reporting.
Guides can view the reports of other guides by selecting the corresponding pin on the map.
Figure 5.5: Mobile application: Outbox

Having submitted reports, guides will be able to view unsent reports in their “Outbox,” and then can choose to upload those reports when they get to wifi or have cell service.
Figure 5.6: Mobile application: Settings

From "Settings," guides can choose how long in the past they want to view reports from, view reports they have submitted, and also logout of the application.
5.2 Administrative Website

The main page of the website provides a view of the Galápagos Islands and marks the map based on the GPS locations of the reports in the system. A list view on the left side makes it easy to search for reports.

Figure 5.7: Administrative Website: Map
### Trail Damage

<table>
<thead>
<tr>
<th>Site one</th>
<th>Site two</th>
<th>Site three</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Username</td>
<td>Timestamp</td>
<td></td>
</tr>
<tr>
<td>mbruce</td>
<td>June 3, 2019</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13:28</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>End</td>
<td></td>
</tr>
<tr>
<td>June 3, 2019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13:29</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Private boat</td>
<td>Registered boat</td>
<td>None</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>Number of passengers</td>
<td></td>
</tr>
<tr>
<td>12345</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Latitude</td>
<td>Longitude</td>
<td></td>
</tr>
<tr>
<td>37.349076</td>
<td>-121.939579</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.8: Administrative Website: Modal

The viewing interface of the website is a modal which appears over the main page when an administrator clicks on a pin on the map or a list item on the left sidebar. It allows the user to view the available fields. Additionally, the administrator can view and add comments, allowing for conversation on the report itself.
<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comments</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Files</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reports</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Users</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The analytics page displays information related to the comments, files, reports, and users of the system.
The tags page allows for administrators to add, edit, and remove tags. Tags are used to label reports into general categories, enabling for easier search and identification of issues around the islands.
The import reports page allows for administrators to upload a CSV file containing reports from the previous reporting system.
The import users page allows for administrators to give users access to the system. In this case, the administrator can add a single user. Once submitted, an email message is sent to the user to begin the password reset process.
If the administrator would like to import more than one user into the system, a CSV file can also be uploaded containing the relevant information. Once submitted, an email message is sent to the users to begin the password reset process.
Chapter 6

Technologies Used

6.1 Mobile Application

The mobile application was made natively for iOS and Android with the help of React Native, a hybrid mobile application development framework. While developing the mobile application, we also relied on Expo to speed up compilation of the application.

6.2 Administrative Webpage

The webpage was constructed with the help of several useful technologies:

- Framework: Django (Python)
- Database: MySQL
- Web Server: Nginx and uWSGI
- Host: Amazon Web Services (AWS)
- Media Storage: Firebase
- Other:
  - HTML
  - CSS
  - JavaScript
  - jQuery
  - Google Maps API
  - Bootstrap
Chapter 7

Architectural Diagram

Our architecture is based on a data-centric architecture. We have an AWS EC2 Instance which hosts our administrative website as well as the REST API. Firebase is used to store images and videos, while a MySQL database holds data for our system. The mobile application has a layered architecture, with the Data Manager designed to send and receive data while mitigating issues with the unreliable internet on the Galápagos Islands.
Chapter 8

Design Rationale

8.1 Web

Our administrative webpage utilizes the latest web development technologies. The Python web framework Django is the basis of our back-end because it is a scalable framework and has many packages available to help with certain functionality. Django also has an integrated testing package and enables efficient templating of HTML. We also utilize MySQL as our database technology because of its reputation as a powerful yet simple solution for storing large volumes of data. Additionally, Nginx and uWSGI are used to serve our Django project to clients who visit our administrative page and use the REST API.

For our front-end, we use the standard web development technologies of HTML, CSS, and JavaScript. jQuery also handles certain tasks that would be more difficult to develop with JavaScript alone, such as AJAX requests. Meanwhile, Google Maps API displays the reports on a map of the Galápagos Islands based on the GPS locations, providing a visual way of viewing the issues. Google Maps API is the best solution for mapping because it has many useful features and provides accurate location information. Lastly, Bootstrap is used to handle the structure and style of the web pages. This is because it is a library that makes it very easy to build visually appealing websites that work across a variety of devices.

8.2 Mobile

Throughout the course of development for this project, the strategy behind development of the mobile application changed multiple times. Knowing that we would need a cross-platform solution for this project, we knew we would need a cross-platform development method such as Ionic or React Native.

We began with Ionic, and quickly discovered that the lack of community support and substandard performance made it unideal for our specific use case–on an island with limited technical resources performance is critical. Therefore, we opted to use React Native, which provided a much cleaner, higher-performance solution for our needs.

We opted, additionally, to use Expo. This was, in hindsight, a larger headache than it was worth, due to poor
documentation with respect to creating a standalone application, but it provided easy access to several key libraries (accessing camera roll, user permissions, etc.) and a powerful solution for rapid development which we were able to use during our trip to the Galápagos Islands.

With respect to Cloud Storage solutions, we opted for Google’s Firebase service due to its simplicity and excellent documentation. It was simple to integrate with our project, and provided an inexpensive and reliable solution for storing assets associated with submitted reports.

8.3 Architecture

We created a hybridized data-centric and layered architecture to satisfy the need to store and manipulate vast amounts of data related to reports. Using a data-centric architecture helps ensure the data is stored and easily accessible in a central place. Meanwhile, our mobile application is layered so it can handle caching of the data when the internet is unreliable.
Chapter 9

Testing

To ensure that our system works properly and conforms to our requirements, we have applied standard industry testing practices to the project. Once we began completing the implementation, we tested at several levels: unit, system, acceptance, and user. This had allowed us to verify and validate our code as well as give us real world testing and user feedback.

9.1 Verification

Verification testing occurred throughout the development process to make sure that the system worked as planned. The system was tested at the unit, functional, component, and system levels. At the unit level, checks were made to ensure that all functions worked as expected. Django’s testing system was utilized to automate the unit testing on the administrative web page. Functional tests included testing the combined functionality of all methods that are related to each other. Component testing applied to the web page and mobile application separately. System testing culminated in testing of the entire combined project.

9.2 Validation

Our validation testing was relatively simple, as the number of requirements for this project is relatively low. We devised acceptance tests to ensure that all requirements have properly been implemented in the final product.

9.3 User Testing

Our final stage of testing will be a beta release to a restricted subset of the users in the Galápagos islands. We will administer this test after the completion of our senior year before we pass it off to the Frugal Innovation Hub for continued development. Our goals will be to ensure that all expected functionality has been incorporated and to allow the Frugal Innovation Hub to gain feedback from users.
Chapter 10

Risk Analysis

Table 10.1 depicts an analysis of potential problems we might run into during development, including consequences, relative probability of occurrence, severity, and ways we might avoid them. These probability numbers are not specifically calculated, but are generic representations of how likely we believe these problems are to occur. The impact column values are calculated by a simple multiplication of the probability and the severity values, to give a normalized impact rating on a scale from 1 to 10.

Table 10.1: Risk Analysis

<table>
<thead>
<tr>
<th>Risk</th>
<th>Consequences</th>
<th>Probability</th>
<th>Severity</th>
<th>Impact</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insufficient Time</td>
<td>Having insufficient time to complete our system could mean that some features are not implemented correctly or at all.</td>
<td>0.7</td>
<td>9</td>
<td>6.3</td>
<td>Stick to our schedule</td>
</tr>
<tr>
<td>Unforeseen bugs</td>
<td>Having unforeseen bugs at the time of implementation on the island could lead to improper testing</td>
<td>0.9</td>
<td>2</td>
<td>1.8</td>
<td>Test thoroughly in advance</td>
</tr>
<tr>
<td>Poor understanding of requirements</td>
<td>Misunderstanding the requirements of our client could lead to costly redesigns and reimplementation of core pieces of our project</td>
<td>0.6</td>
<td>7</td>
<td>4.2</td>
<td>Communicate clearly with our clients prior to implementation</td>
</tr>
<tr>
<td>Insufficient Knowledge</td>
<td>Having to learn new skills and technologies will impact our development schedule, and may have adverse effects on the quality of our final product</td>
<td>0.7</td>
<td>2</td>
<td>1.4</td>
<td>Create a reasonable development schedule with this in mind</td>
</tr>
</tbody>
</table>
Chapter 11

Societal Issues

11.1 Ethical

Other than the social and political considerations mentioned below, we faced few ethical dilemmas when building GalápaGuide. The most important to the users of our application was their ability to access the data as well as their ability to hide certain fields to preserve their privacy and prevent retaliation in the case of reporting other guides activity. Our project is a work that will actively aid the conservation of the Galápagos with no foreseeable repercussions. It is undeniably an improvement over the prior solution used by our clients.

11.2 Social

A potential concern of our project was that by building an application for the Ecuadorian organizations, we were taking an opportunity from entrepreneurs in Ecuador and the Galápagos to build their own solution. However the stagnation of the prior solution and inefficiencies we encountered with bureaucracy showed us that such a solution was unlikely to be built and even less so in a timely manner. The destruction of the Galápagos Islands is a very current threat, and there is serious time pressure to build a working solution. Ultimately the positive impacts towards conservation made possible by our project vastly outweigh the small possibility we took an opportunity away from young people in Ecuador.

11.3 Political

We encountered a conflict between some of the interest groups in the Galápagos over issues such as who would handle and own the information stored within the database. For example, the guides did not want either of the administrative organizations to be able to edit their reports. It was not our job as software engineers to make these kinds of policy decisions. Instead, we posed many of the potential conflicts and ownership issues to the stakeholders and let them internally discuss the issues and present us with a final verdict.
11.4 Economic

The first long term costs of maintaining our solution is extended code-base maintenance which will be provided by an SCU masters student (as mentioned in the sustainability section). The other cost will be upkeep on our cloud services/databases which we have verified will be more than affordable for our clients.

11.5 Health and Safety

As a mobile application, our product has no direct health and safety concerns.

11.6 Manufacturability

Our software has already been manufactured and is ready for a beta level deployment among guides in a real world application.

11.7 Sustainability

An SCU masters student will be maintaining our code-base and adding additional features requested by our clients. We have ensured the development transition will go smoothly by providing thorough documentation into the workings of our code and software stack. We have also provided documentation for mobile and administrative users for usage/setup.

11.8 Environmental Impact

One of the things we are most proud of is how our project will help the long term conservation efforts on the Galápagos Islands. By focusing on the practicality of our solution, we hope that guides will report problems at increased frequency preventing the extinction of endemic species in the Galápagos. Human activities like spreading invasive species and illegal fishing have already done significant damage to the ecosystem of the Galápagos. The Charles Darwin Research Center is working on an initiative known as Galápagos Verde 2050 which is working to restore the native species of the islands and to restore native fauna and flora to islands to counteract the damage done by humans.

11.9 Usability

Working with real clients has made us put usability front and center as a goal. While our initial prototype was made with end-user in mind, potential users of the app gave us quite a bit of feedback into improving usability. For example, our stakeholders requested that the database model for reports be refactored as they felt that it did not accurately represent what was currently being reported by guides.
11.10  Lifelong learning

The product we built required us to learn many new technologies across a full software stack. We were pushed well out of our comfort zones and overcame many obstacles that initially felt like impossible walls to overcome, such as detaching our codebase from Expo. As we enter the workforce we have built confidence in our ability to surpass barriers by continuing to actively learn and familiarize ourselves with new paradigms and technologies.

11.11  Compassion

While our project will directly help the environment and conservation of the Galápagos, it was primarily built with compassion for the guides and administrators who will be using it. The prior solution was incomplete and unreliable to the point that the guides actively hated using it. Our application is usable and intuitive and should dramatically improve the day to the lives of those who work to maintain the Galápagos.
Chapter 12

User Manual

12.1 Web

12.1.1 Login

1. Navigate the webpage.

2. Enter your login credentials.

3. Press the login button.

12.1.2 Password Reset

1. Log out of active accounts.

2. Click the 'forgot your password' button.

3. Supply the email associated with your account.

4. Wait for an email to arrive.

5. Click the link in the email.

6. Type in a new password and submit the form.

12.1.3 Basic Use

1. Use the pane on the left side to view all reports.

2. Alternatively click on a marker on the map.

3. View information about a specific report and post comments.
12.1.4 Search Reports

1. Choose what field you would like to sort by (Text, User, Tag, or Date).
2. Input the restriction you would like to apply.
3. Press enter or accept.

12.1.5 Analytics

1. Ensure that you are logged in as a super user.
2. Click the analytics button.
3. View information about the website such as number of users, etc.

12.1.6 Import Data

1. Ensure that you are logged in as a super user.
2. Click the import button.
3. Choose the CSV that you would like to import. Ensure that it contains all fields with the exception of show.
4. If there are errors with fields, ensure they are spelled correctly and that the file is not saved as a UTF-8 csv.
5. Note how many reports were added and how many were duplicates.

12.1.7 Export Data

1. Ensure that you are logged in as a super user.
2. Click the export button.
3. Choose to open or save the report.

12.1.8 Tag Management

1. Ensure that you are logged in as a super user.
2. Click the tag button.
3. Use the buttons to the right of each tag to edit or delete them.
4. Alternatively, add new tags with the box at the bottom of the list.
12.1.9 User Management

1. Ensure that you are logged in as a super user

2. Choose between adding a single new user or a group of users.

3. If adding a single user:
   
   (a) Input the required information.

   (b) The user will be emailed with password reset information.

4. If adding a group of users

   (a) Prepare a CSV file of the users you would like added.

   (b) Choose the CSV and press submit.

   (c) If there are errors with fields, ensure they are spelled correctly and that the file is not saved as a UTF-8 csv.

   (d) The users will be emailed with password reset instructions.

   (e) Note which users were not inputted into the database.

12.2 Mobile

12.2.1 Installation

iOS


Android


12.2.2 Use

1. Login

   (a) Open the application

   (b) Enter your login credentials (emailed to you by the guide administrators).

   (c) Tap Log in.

   (d) If login failed

      i. If you forgot your credentials, tap Forgot password?

      ii. If your internet connection failed, reconnect to wifi and try again
2. Create report

(a) Open the application.

(b) Log in

(c) Long press on the map to open the Report screen.

(d) Select the report type corresponding to your trip.

(e) Fill out the fields in each section of the report.

(f) Under Privacy, toggle each item corresponding to which fields you want to be public or hidden in your report.

(g) Add attachments
   i. Click "Add attachments"
   ii. Select "Camera Roll" or "Camera"
      A. Camera Roll
         • Pick a photo OR video from your camera roll to attach
      B. Camera
         • Take a photo using your phone camera
         • Tap "Submit" in the top right corner to finish

(h) Upload Report
   i. Open the application.
   ii. Log in
   iii. Create a report.
   iv. Navigate to Outbox in the tab bar.
   v. If you no longer wish to submit a report listed, drag left on it and click "Delete."
   vi. If you want to upload a report, tap the upload icon in the top right of the screen.
   vii. If you would rather wait until you have a better connection, do so, and upload when you're ready. They will be saved until you're ready to submit them.

(i) View Reports
   i. Open the application
   ii. Log in
   iii. From the map view, click on any pin to see a snippet of the reports description.
iv. Alternatively, if you would prefer to view reports in a list view, you can press the icon in the top left of the map view to view all reports in a table.

v. From the table view, simply tap on a report.

(j) Comment on reports

i. Open the application

ii. Log in

iii. View a report

iv. Press the comment field at the bottom of the screen.

v. Type a comment.

vi. Tap the send button.

(k) View your submitted reports

i. Open the application

ii. Log in

iii. Navigate to the "Settings" tab.

iv. Press "Submitted Reports."

(l) Change the duration of reports you want to see

i. Open the application

ii. Log in

iii. Drag the slider under "Report Timeframe" to the number of days in the past you want to see reports from.
Chapter 13

Development Manual

The entire code base for our project is hosted on GitHub and was passed off to the Frugal Innovation Hub at Santa Clara University for future development in coordination with the Ministry of Tourism and the National Park of the Galápagos. The project is documented online in a ”wiki” page, but some key portions are highlighted in the following section.

13.1 Web

13.1.1 Libraries and Dependencies

Libraries and dependencies are located in a requirements.txt file. Once you are in a virtual environment for the project, you can run pip install -r requirements.txt to install them.

13.1.2 Code Structure

- API
  - permissions
    * This file details special permissions that are used to restrict access to the API.
  - serializers
    * This file holds the different serializer classes. These convert data from JSON form to internal database form and vice versa.
  - tests
    * This file holds tests that can be run to ensure that the API is working properly.
  - urls
    * This file defines all of the URLs which users can go to in order to view a webpage in the API app.
  - views
* This file defines the templates rendered for each URL and handles some of the configuration for the pages.

**Portal**

- Templates
  * These are the HTML templates shown to the user when they connect to the Django project via a web browser.

- models
  * This file holds all of the Django models used to generate our database.

- urls
  * This file defines all of the URLs which users can go to in order to view a webpage in the portal app.

- views
  * This file defines the templates rendered for each URL and handles some of the data processing for the pages.

**Static**

- Cache
  * This folder holds the compressed CSS and JavaScript files.

- Portal
  * map
    - This file holds the initialization function for rendering Google Maps on the main page.

  * scripts
    - This file holds all of the JavaScript functions and jQuery selectors used to provide functionality.

  * styles
    - This file contains the CSS rules which dictate how certain elements should look on the webpages.

**Web**

- settings
  * This file defines the settings for the Django project. It is crucial that this is reviewed before deploying.
13.1.3 Installation and Configuration

This installation assumes you have already done the following:

- Cloned the project to your development machine
- Installed python3, pip, and virtualenv

Please note that instructions for setting up the tools and project may be different if you are running on Windows.

```
# Navigate into the web folder
cd GalapaGuide/web/

# Pull the latest code from GitHub
git pull

# Create a Python 3 virtual environment for the project
virtualenv -p python3 env

# Activate the virtual environment
source env/bin/activate

# Download the libraries and dependencies
pip install -r requirements.txt

# Make a local directory
mkdir local

# In the local directory, create a file called config.ini
touch local/config.ini

# Determine and insert the necessary configuration settings into config.ini.
# A sample of what should be included is below:

[database]
name = XXXXX
user = XXXXX
password = XXXXX
host = XXXXX
port = XXXXX

[settings]
debug = XXXXX
```
key = XXXXX

[ email ]
address = XXXXX
password = XXXXX

Note that database section helps with connecting the Django project to the database, the settings section sets values of certain variables in settings.py, and the email section enables the Django project to send emails for changing passwords. Django will give warnings or errors if these are not properly configured when you try to run the project.

13.1.4 Run

JavaScript and CSS files need to be compressed before running the project. To compress those files, run

```python
python manage.py compress
```

You are now ready to run the Django project. To do this, run

```python
python manage.py runserver
```
13.2 API

This section will offer a brief overview of all of the calls that are available through the website API.

13.2.1 Users and Authentication

POST api/token/

Returns a token that allows a user to secure a token for accessing other API calls.

GET api/users/

Returns the information on the current logged in user based on sent token.

GET api/users/[user_primary_key]/

Returns the information on the user specified by the given primary key.

13.2.2 Reports

GET api/report/

Returns information on all reports currently in the system.

POST api/report/

Allows the user to add a new report to the database.

GET api/report/[report_primary_key]/

Returns the information of the report with the given primary key.

DELETE api/report/[report_primary_key]/

Allows a user to delete a report with the given primary key.

GET api/report/detail/[report_primary_key]/

Returns all information associated with a report including comments and usernames.

13.2.3 Search

GET api/search/date/[date]/

Returns all reports that have been made between now and that date, inclusive. Date should have the form YYYY-MM-DD.
GET api/search/tag/[tag_primary_key]/

Returns all reports that have been made that are attached to that tag.

GET api/search/text/[search]/

Returns all reports that contain the submitted text in the description field.

GET api/search/user/[username]/

Returns all reports that contain the submitted text in the description field.

13.2.4 Comments
GET api/comments/[report_primary_key]/

Returns all comments associated with a report.

POST api/comments/[report_primary_key]/

Allows a user to save a comment to the report with given primary key.

13.2.5 Files
GET api/files/[report_primary_key]/

Allows a user get all files attached to the a report with the given primary key.

POST api/files/[report_primary_key]/

Allows a user to attach a file to a report with the given primary key.

13.2.6 Tags
GET api/tags/

Returns all the tags in the system.

POST api/tags/

Allows the user to add a new tag to the database

GET api/tags/[report_primary_key]/

Returns all the tags that are associated with a certain report.

DELETE api/tags/delete/[tag_name]/

Deletes the tag with the given name
PUT api/tags/modify/[tag_name]/

Allows the user to modify the tag with the given name to have a new name.
13.3 Mobile

13.3.1 Libraries & Dependencies

A list of all libraries in use can be found in “package.json,” but three of the most common are outlined here.

- **React Native**
  - React Native allows for a single cross-platform solution for native iOS and Android development using JavaScript. The React Native library provides access to several native UI elements such as “Alert,” “Text,” or “Button.”

- **Native Base**
  - Native Base provides several more UI elements than the React Native library, in this application it is used mainly for “Input” fields.

- **Expo/ExpoKit**
  - Expo is a library that simplifies development by hosting react native builds online. It allows you to rebuild your project during runtime, greatly speeding up development, and also provides several useful APIs to access things like Camera permissions and image pickers.
  - To run a pure Expo project, you must download the expo app and scan the QR code generated by "npm start" in an un-detached expo project. This project, however, is detached using ExpoKit.
  - ExpoKit is a native library (on iOS, a CocoaPod) that bundles all of the Expo APIs natively with your offline application. When the application is run in Debug mode, it runs similarly to a pure expo project, but can also be bundled independently.

13.3.2 Code Structure

- **Interface Files**
  - All files within screens/ correspond to a screen on the application.
  - Most of these screens contain a "DataManager" instance which they use to get and send data.

- **Components**
  - ‘DataManager’
    - Abstracts all data management for the application to a single place.
    - Handles all the local storage, API calls, and Cloud Storage calls.
– ‘CloudStorage’
  * Interfaces with ‘Firebase’ for attachment storage.

– ‘StringManager’
  * Handles localization for user-facing strings within the app.
  * Exports ‘LocalizedString()’ which takes the path within the localized strings file and any parameters to pass in to that string.

– ‘IconWithBadge’
  * Defines a custom icon that has a badge icon on it. The component takes a ‘badgeCount’ property which defines the number displayed.

– ‘TabBarIcon’
  * Defines an icon specifically for a tab bar which contains an ‘IconWithBadge’ in order to display badge icons on the tab bar. This also takes a ‘badgeCount’ property.

• Navigation

  – ‘AppNavigator’
    * This defines the layout of the application according to ‘react-navigation’ conventions.

• App

  – This is the entry point of the application, it simply renders ‘AppNavigator’.

13.3.3 Localization

GalápaGuide is Localized in Spanish (es) and English (en) using the react-native-i18n library. Under the locales folder each language has its own strings file in json format. Each class in GalápaGuide that displays strings to users does so through StringManager, which exports LocalizedString which handles localization and looks up a string from the correct json file based on the system preferences of the user.

13.3.4 Installation

# Pull the latest from GitHub
git pull
# Download all the node modules from the project directory
npm install
# Link the necessary libraries with the native code
npm link
# Download the native libraries for iOS

cd ios
pod install

**13.3.5 Build & Run**

**iOS**

# Download all the node modules from the project directory

npm install

# Link the necessary libraries with the native code

npm link

# Build the latest jsbundle

expo publish

# Download the native libraries for iOS

cd ios
pod install

- Open "ios/galapaguide.xcworkspace"
- Debug mode
  - Debug mode allows for refreshing the app live without having to rebuild the entire project. This app will not work offline.
  - From Xcode, edit the current build scheme. Select "Debug" under "Build Configuration".
  - From your project directory start the expo server by running:
    
    npm start

  - Build and run from Xcode as normal (click "Run")
- Release mode
  - Release mode allows for a fully independent, offline application. This will store your javascript bundle on-device, but will not allow for the same kind of rapid development in that you must rebuild the code-base every time you want to build your application.
  - From Xcode, edit the current build scheme. Select "Release" under "Build Configuration".
  - Build and run from Xcode as normal (click "Run")
Chapter 14

Conclusion

14.1 Obstacles Encountered

Throughout the development of the system, we encountered many challenges that impacted our work on the system. Infrequent communication with client was one major obstacle we faced. This made it difficult to ensure that all features were being implemented in the way which the client expected them to be. We also faced a rapid development timeline, which meant we were not able to integrate some of the recommended requirements into our system. Lastly, when we visited the Galápagos Islands, there was extremely limited internet on the islands. As a result, we were unable to fix issues with the system while we were on the island.

14.2 Future Work

As we finished our development work on this application, it is anticipated that work on this project will continue through the Frugal Innovation Hub. Below are some of the improvements that could be made in future iterations of the system:

Mobile:

- Taking videos within the app
- Filtering reports by tags
- Editing reports
- Support for Android
Web:

- Localizing the website
- More analytics and graphs to go with them
- Sanitizing inputs to avoid malicious comments or tags
- Ensuring duplicates of tags are not allowed via the tag page

14.3 Lessons Learned

As we developed this system, we learned many important lessons that will stay with us for future projects. One of the key lessons we learned is that task administration is important. Initially it was not clear what needed to be done and by whom. With the introduction of Trello as our task management solution, we became more organized. When we visited the Galápagos Islands, we also learned that in-person meetings are invaluable. We gained more insight in a two hour meeting over coffee compared to most of our previous remote meetings. These insights helped guide the development of the application and ensured that we were meeting the expectations of our client. Lastly, we learned that it is sometimes necessary to stay out of the policy decisions. Given the number of organizations who are involved in this endeavor, we realized that many of them have different priorities. Because of this, we let them debate on these choices and had them notify us of how they would like to see these decisions integrated into the application.