6-11-2020

Mobile Application to Distribute Water Quality in Rural Nicaragua

Rachael Freitag

Sarah Ortiz-Jones

SANTA CLARA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: June 11, 2020

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Rachael Freitag
Sarah Ortiz-Jones

ENTITLED

Mobile Application to Distribute Water Quality in Rural Nicaragua

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

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor

N. Ling

Department Chair

Mobile Application to Distribute Water Quality in Rural Nicaragua

by

Rachael Freitag
Sarah Ortiz-Jones

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 11, 2020

Mobile Application to Distribute Water Quality in Rural Nicaragua

Rachael Freitag
Sarah Ortiz-Jones

Department of Computer Science and Engineering
Santa Clara University
June 11, 2020

ABSTRACT

The organization ASDENIC provides the citizens of northern Nicaragua with many resources to help them become more informed farmers and improve the quality of their lives. ASDENIC currently holds weekly meetings to distribute information such as water quality, weather conditions, and other important agricultural updates. However, technological limitations still prevent Nicaraguan farmers from getting updates from ASDENIC in real-time. We have partnered with Santa Clara University's Frugal Innovation Hub and ASDENIC to create a mobile application to distribute necessary information to communities in rural Nicaragua in real-time.

Our mobile application is a platform for community leaders to share relevant information about water quality and for community members to stay updated and share any water incidents with the leaders. This application works to reduce the need for weekly town meetings to distribute this information.

ACKNOWLEDGEMENTS

We want to thank Allan Baez Morales and the Frugal Innovation Hub, Raúl Díaz and ASDENIC, and our advisor Professor Angela Musurlian for all they have done to help us through this process.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

ASDENIC is an organization based in northern Nicaragua that aids in the social development of the area. As just under half of Nicaraguan citizens live in rural areas and of these people four-fifths depend on agriculture for their livelihood, the services that ASDENIC provides are important. They provide programs to teach and encourage farmers to better utilize their land and resources. Through the Frugal Innovation Hub, we have partnered with them to fulfill their need for a mobile application to distribute information, particularly water quality updates, weather forecasts, and farming advice.

1.2 Motivation

The ability to access accurate information about water quality is something that many people take for granted. This information is not easily accessible to everyone, including the people of small, rural communities in northern Nicaragua. Currently, ASDENIC holds weekly town meetings to verbally deliver important updates about the water quality in their region to representatives of each community. Committees have been formed to regularly visit the local water sources and test for quality. These committees then share this information with the representatives at the weekly meetings. The people living in these communities greatly benefit from this information about water quality because knowing what is in your water can prevent waterborne illnesses as well as keep crops healthy. Although the weekly town meetings currently work as a means of communication, this current solution is not the most efficient or the most timely. Water quality testers are not able to immediately share urgent water quality findings because they have to wait for the next in-person meeting, possibly causing harm to many people if they are not informed about the quality of their water in time.

1.3 Solution

At the beginning of this school year, the Frugal Innovation Hub asked us to create a multi-platform application on behalf of ASDENIC, so that town officials could distribute this valuable information, rather than hosting weekly meetings. Not only will this application cut out the need for meetings, it will also provide information to the people a lot faster than waiting for the next meeting. This mobile application has the potential to improve the lives of many local farmers in northern Nicaragua because they will have access to real-time water quality information.

When first analyzing the problem and trying to identify the best solution, we came up with the following needs that were most important to ASDENIC. It was important that the application be intuitive and simple to use so that anyone can use it with little instruction. As it is now, the town officials and townspeople have the ability of two-way communication during their in-person meetings. Instead of simply providing an information delivery platform, we planned to incorporate two-way communication within the app. Multiple interfaces are also necessary since some individuals need to input data and some simply need to read the information shared. Different administration levels are also necessary because access levels vary. Some people need to grant others administrative access, while others just need access to input data.

We also recognized that consistent communication with ASDENIC throughout the development process would be vital to keeping our client's needs and wants in mind. We decided that a customized mobile application would be better than other solutions available because we would be able to personalize it exactly to their needs–using their language, local communities, and adhering to the resources they have available. In the end, we knew it was most important to empathize with our potential users, which can be challenging at times, but necessary to make an application that successfully solves their problem and makes their lives easier.

Chapter 2

Project Analysis

2.1 Requirements

These are the requirements that we have elicited from our client for this application. We've broken these requirements down into functional, non-functional, and design constraints.

Functional

- The system should display water quality news reports posted by administrators. This is necessary to eliminate weekly in-person meetings that are currently held just to distribute this information.
- The system should support signing up for and logging into accounts using a phone number and password as login credentials. This is necessary because not all users will have emails.
- The system should give administrative rights to certain users.
- Users should be able to report water quality incidents.
- Administrators should be able to read and review water quality incidents, and be able to either call the reporter, respond to the reporter via an in-app message, or post a formal water quality report about the incident for all users to see.

Non-Functional

- The system displays information reliably and accurately.
- Administrators can easily post reports.
- The user interface is easy to use and intuitive for Nicaraguan community members.

Design Constraints

- The user interface must be in Spanish that is specific to Northern Nicaragua.
- The application must be accessed via an Android application that can be downloaded from the Google Play Store.

2.2 Use Cases

This is our use case diagram (Figure 2.1) that demonstrates all of the actions users can take when they use our application. There are three different types of users that can use our application: general users, administrators, and the main administrator. The main administrator can do everything a general administrator can do, but can also grant other users administrative access.

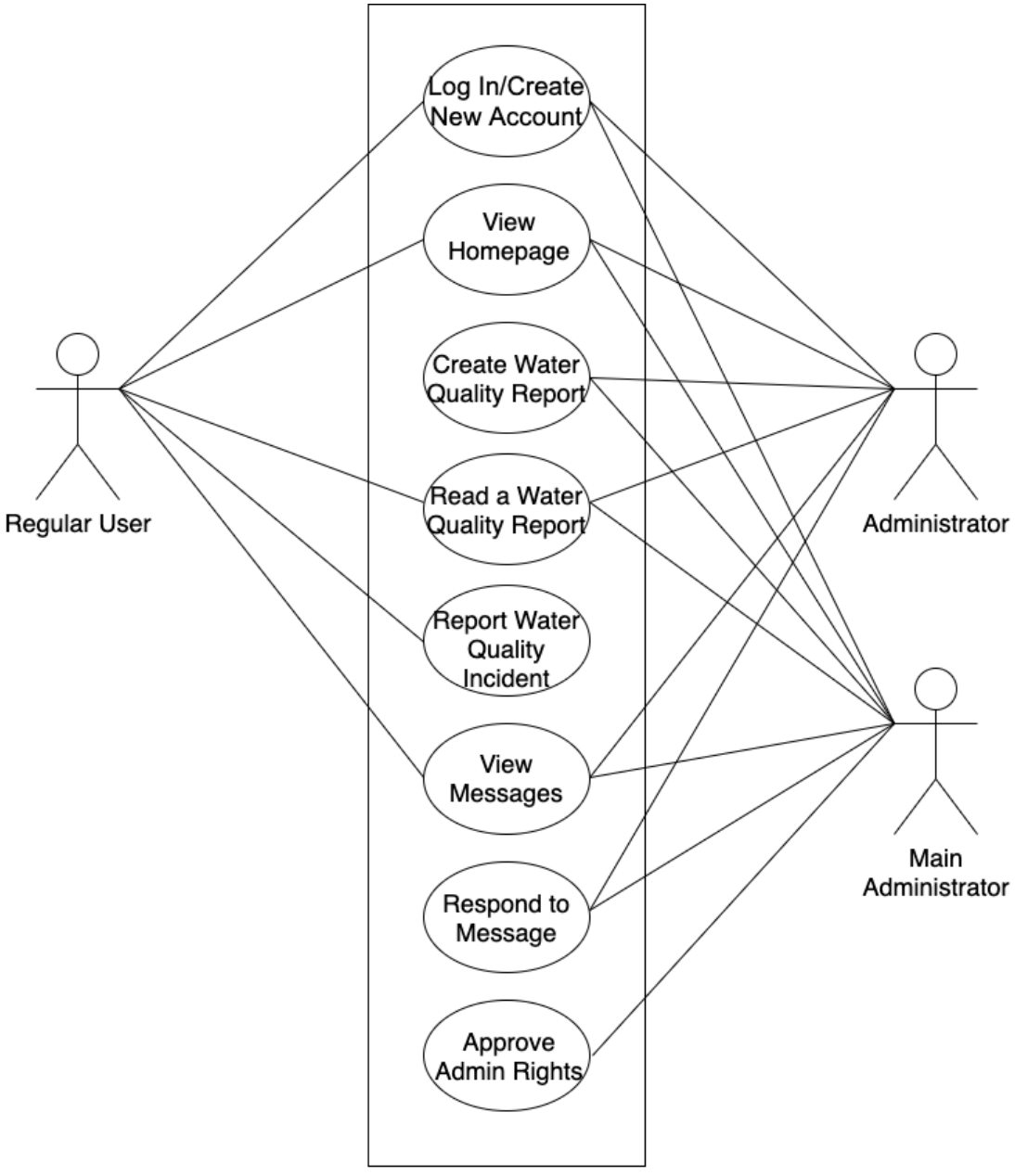

Figure 2.1: Use Case Diagram

Log in and/or Create an Account

- **Goal:** Anybody should be able to download the app and make an account. Certain authorized people will be given administrative privileges.
- **Actor:** General User or Administrator or Main Administrator
- **Pre-condition:** An individual has downloaded the app from the Google Play store and opened the app.
- **Post-condition:** That individual is taken to the homepage after creating an account and logging in.

View Homepage

- **Goal:** An individual can view their homepage where they can see water quality reports. They have the option to click on those reports to read further or depending on admin rights, complete other action available.
- **Actor:** General User or Administrator or Main Administrator
- **Pre-condition:** The user is logged in.
- **Post-condition:** The user is directed to a water quality news report, report a water quality incident, giving admin rights, creating a new water quality report or their inbox.

Create a Water Quality Report

- **Goal:** An administrator can post water quality news reports.
- **Actor:** Administrator or Main Administrator
- **Pre-condition:** The user has logged in and has administrative rights. The user has then clicked on ”add report.”
- **Post-condition:** The administrator has posted a water quality news report and it can be viewed on the homepage.

Read a Water Quality Report

- **Goal:** The user can read a report about water quality.
- **Actor:** General User or Administrator or Main Administrator
- **Pre-condition:** The user has logged in and clicked on a water quality report.
- **Post-condition:** The user has read a report about water quality and returns to the homepage.

Report Water Quality Incident

- **Goal:** The user can report a water quality incident they encounter.
- **Actor:** General User
- **Pre-condition:** The user has logged in and clicked on ”report a water quality incident.”
- **Post-condition:** The user has described the water quality incident and pressed submit. The submission will be read and reviewed by an administrator who will decide if it should be looked into and/or posted as a news report.

View Messages

- **Goal:** The user can read messages that have been sent to them.
- **Actor:** General User or Administrator or Main Administrator
- **Pre-condition:** The user has logged in and clicked on the envelope symbol in the upper right hand corner of the homepage.
- **Post-condition:** The user has read their messages and returned to the homepage.

Respond to Messages

- **Goal:** The user can respond to messages that have been sent to them.
- **Actor:** Administrator or Main Administrator
- **Pre-condition:** The user has opened a message and clicked on ”respond.”
- **Post-condition:** The user has written their response and clicked ”send.”

Approve Administrative Rights

- **Goal:** The user can grant other users administrative rights.
- **Actor:** Main Administrator
- **Pre-condition:** The user has logged into their account and clicked on ”give others admin rights.”
- **Post-condition:** The user has selected the user(s) they would like to give administrative rights to and clicked ”save.”

2.3 Activity Diagrams

General User Activity Diagram

Figure 2.2 displays how a general user maneuvers through our application. If the user already has an account, the user can log in with their phone number and password. If not, they must sign up for an account and then log in. Once logged in, they are taken to the general user homepage where they can read water quality reports, report a water quality incident, and view any responses to their incident reports from administrators. They can log out on most pages.

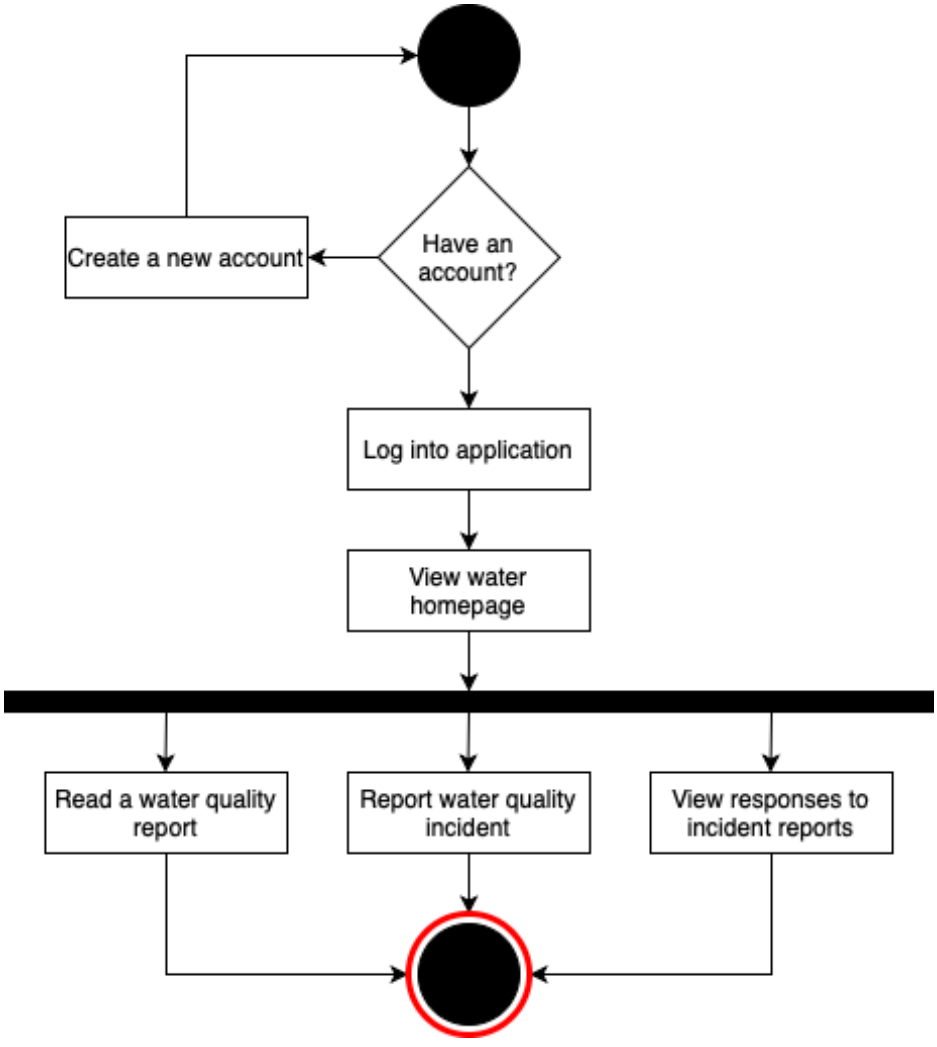

Figure 2.2: General User Activity Diagram

Administrator Activity Diagram

Figure 2.3 shows how an administrator will maneuver through our app. If the administrator already has an account, the user will log in with their phone number and password. If not, they must sign up for an account and then log in. Once logged in, they are directed to the administrator homepage. Here they can read water quality reports, create new water quality reports, and/or view and respond to incident reports. They are able to call the user who sent the incident report in, respond with a message, or use the message to pre-populate a new water quality report for editing and posting.

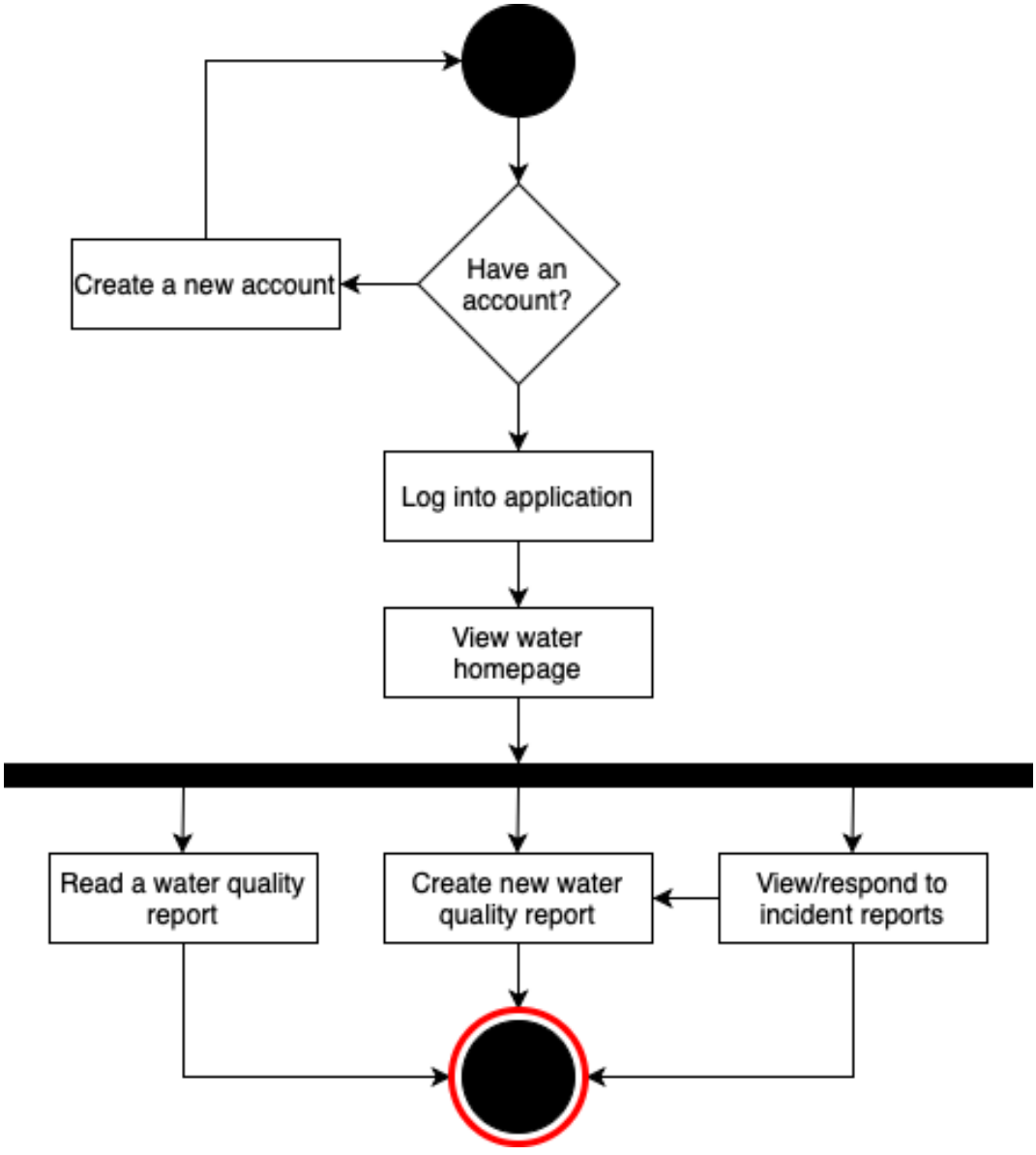

Figure 2.3: Administrator Activity Diagram

Main Administrator Activity Diagram

Figure 2.4 displays how the main administrator will maneuver through our application. It is identical to an administrator, yet they have the added capability to give or remove administrative rights from users.

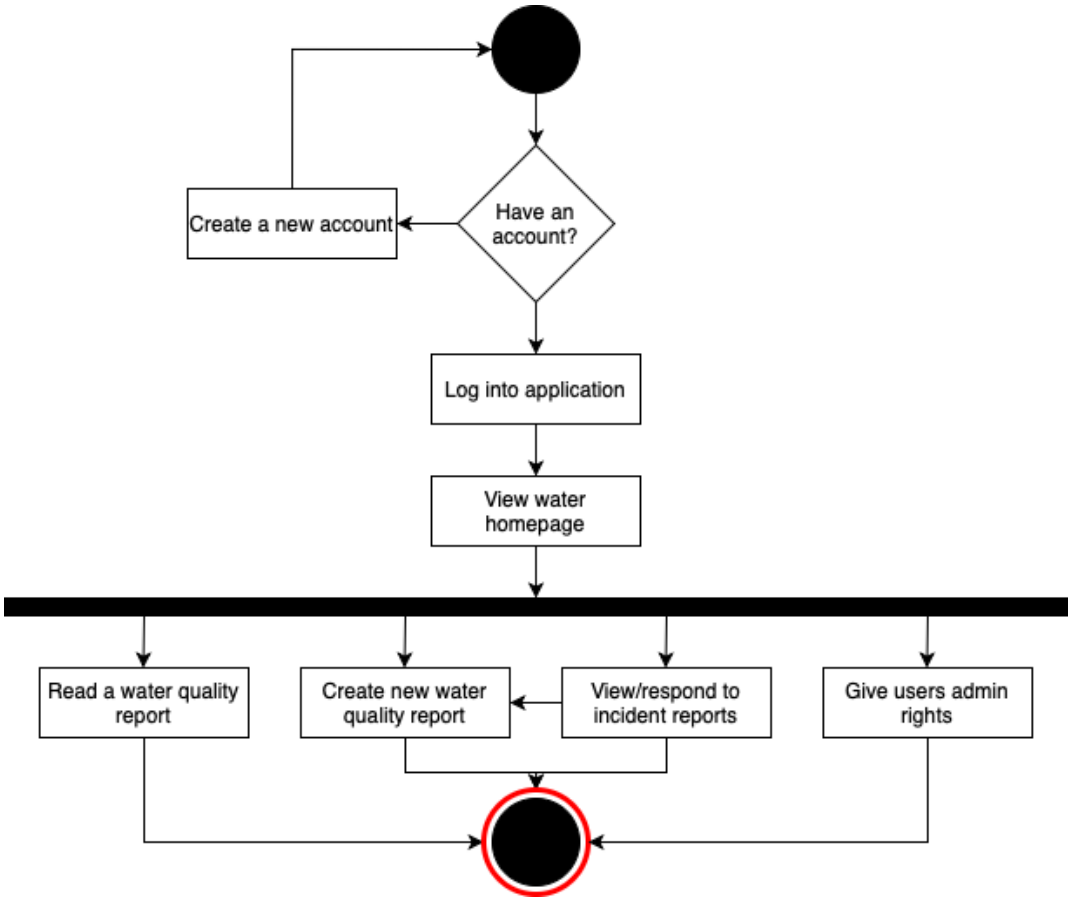

Figure 2.4: Main Administrator Activity Diagram

Chapter 3

Design Decisions

3.1 Architectural Diagram

The Architectural Diagram (Figure 3.1) is a visual abstraction of the way in which our application will function.

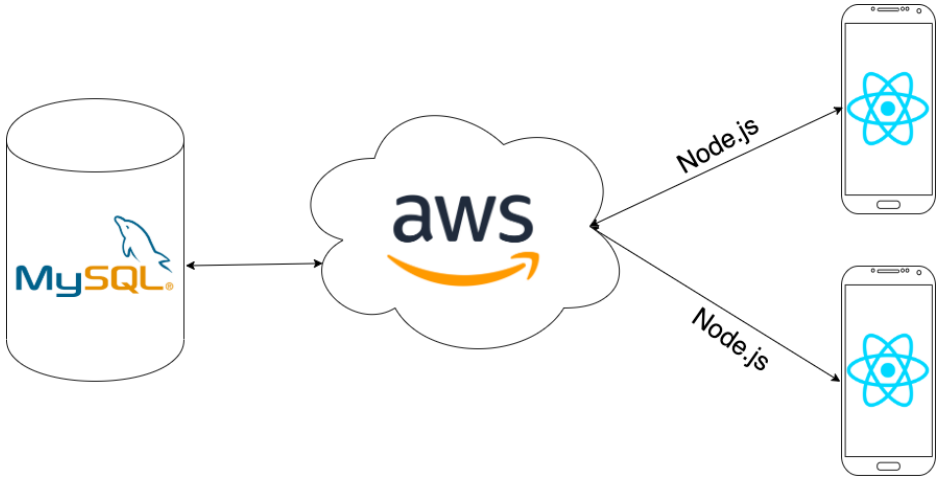

Figure 3.1: Architectural Diagram

We chose to use a data-centric architecture for our system, which can be seen in Figure 3.1. We used React Native as our front-end framework, Node.js as our server side scripting language, and MySQL hosted on AWS as our database management system. Data will need to be accessed by all users to read water quality reports and their messages, but any personal information, such as phone numbers, stored on our database will only be used for user authentication. MySQL hosted on AWS serves as our database and server which will store user information and act as the gatekeeper between the database and clients.

3.2 Technologies Used

To build our application, we used React-Native as our front-end framework, Node.js as our server side scripting language, and MySQL hosted on AWS as our database management system.

React Native

React Native is an open source mobile application framework that allows for cross-platform application development. As the community members have both Apple and Android cellphones, this enables our team to build both a native iOS and Android app.

MySQL Hosted on AWS

MySQL is an open-source relational database management system. AWS, or Amazon Web Sevices, provides on-demand cloud computing platforms and APIs to individuals and companies on a metered pay-as-you-go basis. Using MySQL hosted on AWS allows us to store information reliably and inexpensively.
Originally we had planned to use Firebase as our database, but we found it difficult to use for anything other than user authentication because we had never used it before and we lacked the resources to be able to teach ourselves in such a short period of time. Firebase is a mobile and web application development platform owned by Google.

Git

Git is a version control system where we keep our code properly updated between teammates. It keeps track of copies of our code throughout the design process.

3.3 Design Rationale

React Native

We chose to use React Native because we knew that our client needed a cross-platform application, with most users having Androids. React Native allowed us to build both native iOS and Android applications without having to start from scratch on each. We also chose React Native because we have limited implementation time and it is an easy-to-learn framework.

MySQL Hosted on AWS

We needed a reliable, easy-to-use database management system that we could use for both user authentication and storing water quality reports and messages between users. We originally chose to use Firebase because we thought it was a simple platform to understand and use. Additionally, it offered many functionalities and worked seamlessly with React Native.

However, we found it difficult to use for anything other than user authentication because we had never used it before and we lacked the resources to be able to teach ourselves in such a short period of time. After many unsuccessful attempts at using Firebase, we switched our database management system to MySQL hosted on AWS. We decided to switch to MySQL in particular because we have both used it before and there are ample resources available online. We decided to host our database on AWS because it allowed us to reliably and inexpensively store data.

Git

We had both used Git before and it reliably keeps track of our code history.

Chapter 4

Planning Phase

4.1 Test Plan

Testing

We began with unit testing the different components of our application separately before adding it completely. When adding a new feature, we ensured that it worked smoothly with the application as a whole before finalizing through integration testing. This sometimes meant adjusting dependencies or troubleshooting database conflicts. As our application is being finished, we have been doing end-to-end testing to verify that our app behaves exactly as we expect during a full usage. We will also have our client test it to complete user acceptance testing. This will help uncover any bugs found in practical usage.

Validation

It will be important to have every functional requirement checked in this testing. This means validating the information our application is grabbing from the database is correct and that it works in many different environments.

Verification

- Have fluent Spanish-speaker review all user interfaces to ensure proper verbiage is being used.
- Meet with client periodically to ensure our system is meeting their needs.

4.2 Risk Analysis

Shown below, Table 4.1 displays possible events that may have occurred throughout the development of our system. We have included the impact value associated with each risk, which we based off of the probability that the event will occur and the severity of it occurring. Mitigation techniques are also included, which in theory reduced the probability of each risk occurring.

Risk Analysis Table

RISK	CONSEQUENCES	PROBABILITY	SEVERITY	IMPACT	MITIGATION
Unfamiliar Technology	Team members must learn technologies needed before actually implementing them	0.8	7	5.6	Learn about enviornments before development. Research aspects that we are unsure about.
Team Member Illness	Team will have less communication between partners - one partner is unable to work on project	0.6	9	5.4	Try to stay healthy Keep communication open if this does happen, so both team members are aware of each other's tasks
Missing Requirements	The final system does not satisfy all of client's requirements - unhappy client	0.5	7	3.5	Constant communication with client from the beginning. Implement critical requirements first, then focus on others.
Platform Compatibility Issues	Developed application does not function as intended on both Apple and Android phones	0.3	7	2.1	Test earlier and continuously on both phone types to ensure proper function. If it is not, make changes right away.
Bugs in Code	System does not work as intended	1.0	2	2.0	Write code well to begin with. Write code that is easy to debug.
Insufficient Time	System is not completed on time	0.2	9	1.8	Create project schedule and stick to it. Keep other teammates accountable.

Table 4.1: Risk Analysis Table

4.3 Development Timeline

This is our original development timeline for this project including the deadlines for our documentation requirements, initial system, and final system. The deadlines we set here were meant to help us meet our deadlines and complete all requirements on time.

Although we tried to stick to our timeline as best we could, we definitely strayed from the path laid out. We now realize how difficult it is to stick to a timeline made for a project we have never completed before. Had we more experience in the development process of an application beforehand, we may have been able to create a better timeline from the start.

Development Timeline

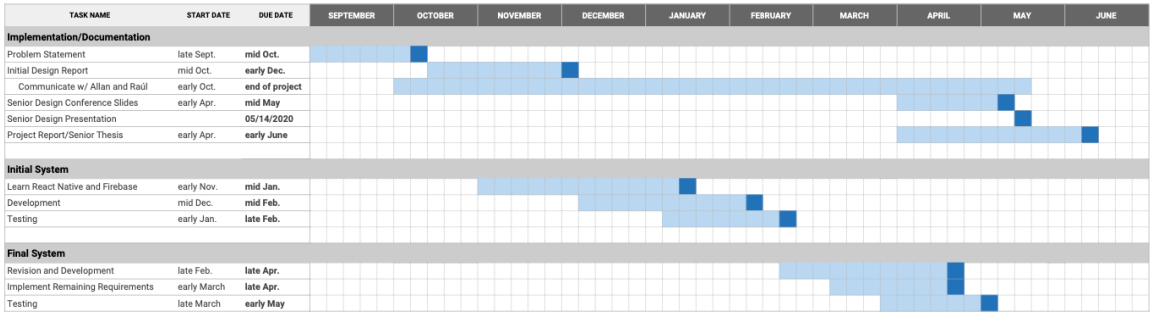

Figure 4.1: Development Timeline

Chapter 5

Other Considerations

5.1 Ethical Considerations

When creating an application for a real client, there are many ethical issues that need consideration. One major concern is the privacy of each user. As users will be sharing personal information, such as their phone numbers and a password, anonymity is necessary. Only administrators have access to a general user's phone number if they need to contact them to clarify or further explain an incident. General users do not have access to any phone numbers as there is no reason for them to have this. Passwords are encrypted as they are added to the database.

5.2 Social Considerations

Because we were creating an application for a group of users that currently interact mostly in-person, there were many social factors we had to consider. Current in-person meetings are not only a means to distribute water quality information, but also used as learning opportunities. With our mobile application, we not only want to get important information into the hands of Nicaraguan farmers faster and more efficiently, but also allow for constant communication between community members and ASDENIC representatives. Taking this into account, we decided to include two-way communication within our app through both incident reports and messaging. Our application is meant to make the people of rural Nicaragua better farmers by making them more informed farmers, and not to eliminate any form of social interaction.

5.3 Geographical Considerations

As our application is for people of rural communities in Nicaragua, there were a couple aspects of our application that we needed to customize due to this. One being the language our applications interface is in. This required us to get help in translating any words or alerts of the interface from English to Spanish. While we could do basic translations, it became difficult because we are not aware of Spanish jargon. We

also decided to use phone numbers for the user login instead of an email. This was chosen due to the fact that not all of the people using the application will have an email.

5.4 Usability

Usability is always a factor that needs to be considered when creating a mobile application. In our initial discussions with our client, we were told that the average person living in rural Nicaragua does not own a computer and are not used to using complicated mobile applications. They use their smart phones mostly for communication amongst one another and access to the internet. We, along with our client, believed that an easy-to-use mobile application would be the best way to distribute information for usability reasons. For this reason, when designing our application we aimed for a user interface that was simple, consistent, and easy to use. We did this by using very few colors, clear wording, and uncluttered screens. In order to promote a good user experience, we set up steps in a straight-forward manner so that it is obvious what step you are meant to do next.

Chapter 6

Implementation

6.1 General Conceptual Models

Before developing our app, we created high-fidelity mock-ups to show our client and make sure we were building exactly what they wanted before actually building it. These mock-ups gave us and our client a glimpse into what our app would later look like. Originally, we planned to make an app that distributed both water quality information and information about local weather conditions, but, unfortunately, due to unforeseen circumstances, we had to cut down on the scope of our application and remove the weather distribution portion of our app. However, we did work with ASDENIC and the Frugal Innovation Hub to design a weather distribution section that would meet their needs. These conceptual models represent our original design with the weather portion of the app still included.

Figure 6.1: Welcome Page

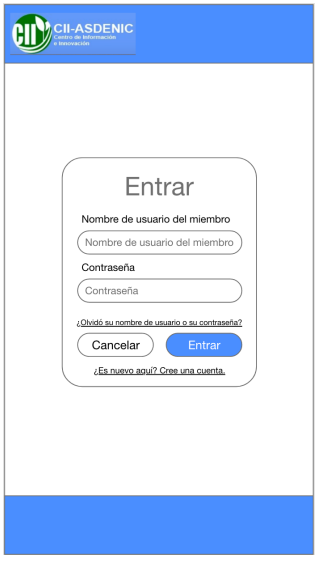

Figure 6.2: Login Page

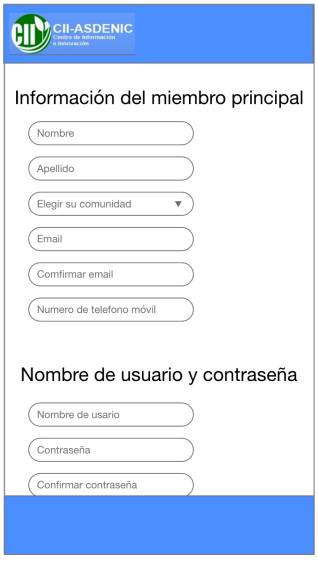

Figure 6.3: Create New Account Page

6.2 Weather Distribution Conceptual Models

In our original conceptual model, once a general user had logged into their account, they would be taken to the **Homepage**, shown in Figure 6.4, where they would be given the option to select ”Agua” for water quality news, or ”Clima” for weather condition updates. In our final implementation of the app, once a general user logs in they will be brought directly to the **Water Page** shown later in the **Water Quality Section** instead of the **Homepage** shown in Figure 6.4.

Clicking on ”Clima” from the homepage would take the user to the **Weather Page**, shown in Figure 6.5, where the user has the option to read weather reports or ask weather-related questions. Administrators would have the option to post a weather report by clicking on the button on the upper right-hand corner. This button would only be displayed for administrators. Users have the option to ask questions about the weather by clicking on the link at the bottom of the page. They would be taken to the form shown in Figure 6.6 where they can submit their question.

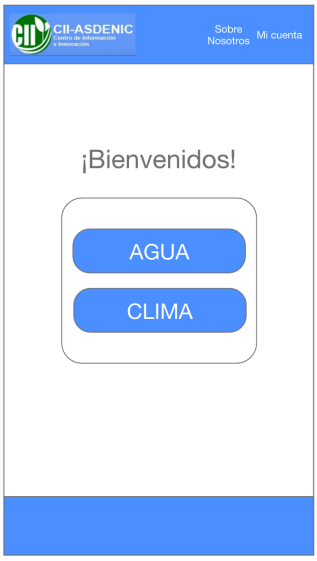

Figure 6.4: Homepage

Figure 6.5: Weather Page

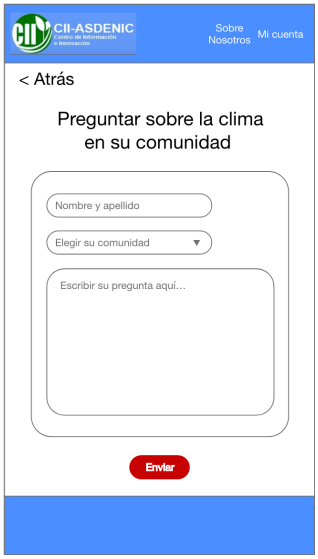

Figure 6.6: Ask Weather Question Page

6.3 Logging into Your Account

In our conceptual models and the final implementation of the app, we aimed for a user interface that is simple, consistent, and easy to use. The following screens were taken from the finished application. As you can see, we followed our conceptual models closely, but had to make minor adjustments.

The **Login/Create New Account Page**, Figure 6.7, is what every user, including administrators, are taken to when they open the app. If they already have an account, they will click on ”Entrar” and be taken to the **Login Page** in Figure 6.9. If they do not have an account, they will have to create a new account by clicking on ”Crear una cuenta”, filling out the information in Figure 6.8, and clicking ”Enviar.” If the new user would like to request administrative access, they would simply click the box that says ”¿Necesitas derechos de administrador?”

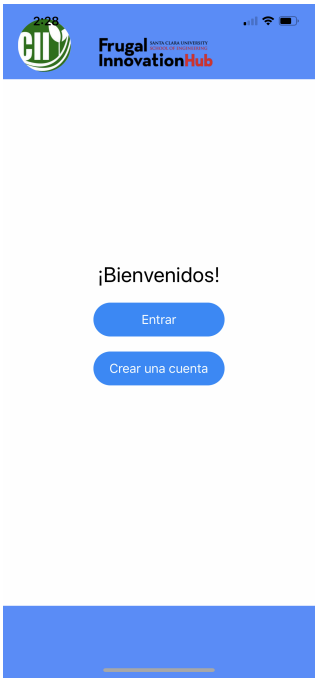

Figure 6.7: Login/Create New Account Page

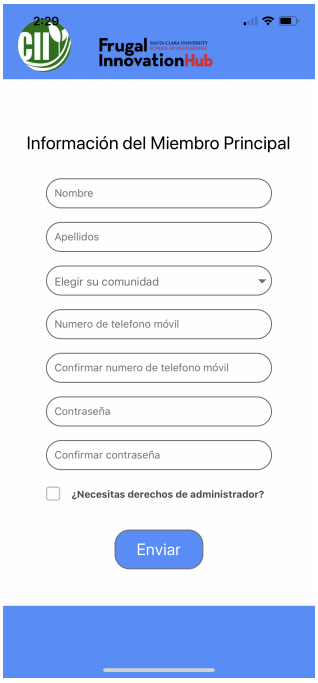

Figure 6.8: Create New Account Page

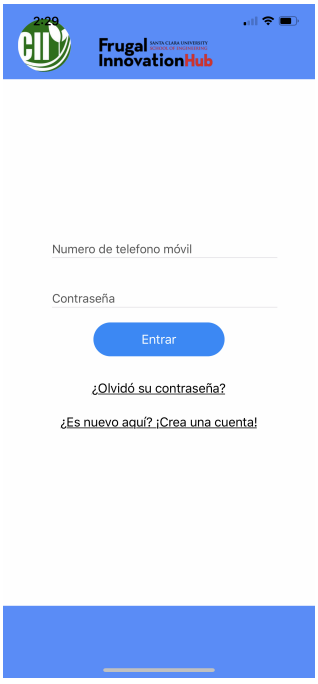

Figure 6.9: Login Page

6.4 Water Quality Section

As mentioned earlier, in our original design users would be taken to a homepage similar to the one in Figure 6.4. However, in the final implementation of the app, once a user logs in they will be brought directly to the **Water Page**. The homepage for administrators is shown in Figure 6.10. From this page, administrators will have the option to post a water quality report by clicking on the plus-sign in the upper right-hand corner, where they will be taken to a form similar to the one shown in Figure 6.11. This button will only be displayed for administrators.

The homepage for main administrators is shown in Figure 6.12. From this page, main administrators can also create new water quality reports, as well as grant other users administrative access by clicking on the link at the bottom of the screen. After clicking on that link, they will be taken to the screen shown in Figure 6.13, where they will be able to see which users have requested administrative access and choose which users to grant administrative access to. This feature will not be available for general users or general administrators. The homepage for general users is shown in Figure 6.14. From their homepage, general users have the option to submit water quality incidents by clicking on the link at the bottom of the page. They will be taken to the form shown in Figure 6.15 where they can submit the report. This button will only display for general users.

Figure 6.10: Administrator Homepage

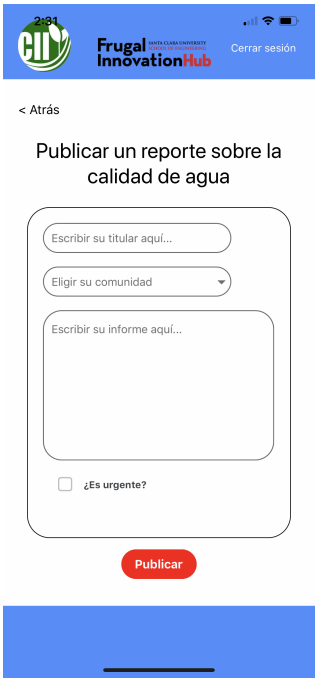

Figure 6.11: Create New Water Quality Report

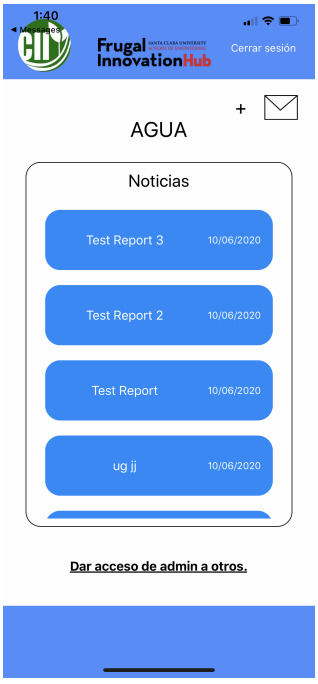

Figure 6.12: Main Administrator Homepage

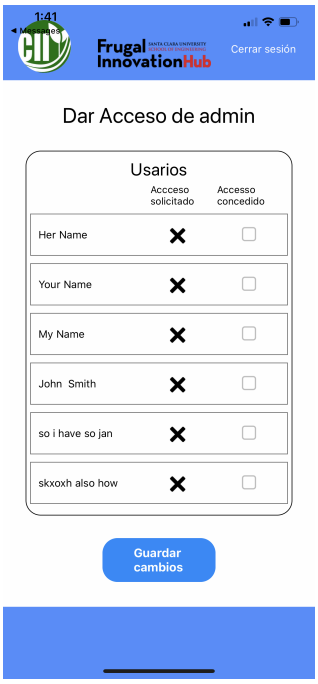

Figure 6.13: Grant Administrative Access

Figure 6.14: General User Homepage

Figure 6.15: Create New Water Quality Report

6.5 Messaging

Administrators and general users can both access their inbox by clicking on the envelope symbol in the upper right hand corner of their homepage. Every type of user will be taken to the **Inbox** shown in Figure 6.16. If a general user clicks on one of their messages, they will be taken to the screen shown in Figure 6.17. General users will not be able to respond to messages sent to them by admins.

An administrators inbox will hold all water quality incidents that have been reported by general users. When an administrator views one of their messages, they will be taken to the screen in Figure 6.18. From here they can either call the user who reported the incident, respond to the user who reported the incident via the app, or publish the incident as a formal water quality report.

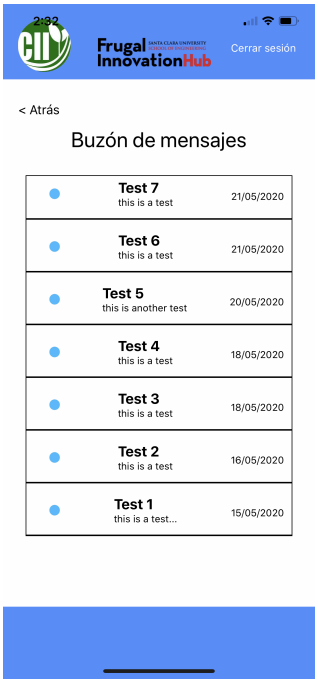

Figure 6.16: Inbox

Figure 6.17: View an Individual Message as a General User

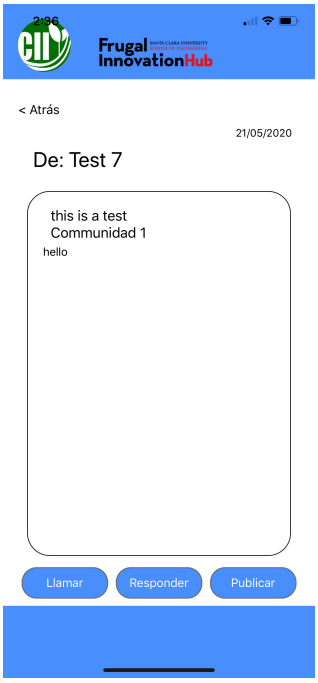

Figure 6.18: View an Individual Message as an Admin

Chapter 7

Conclusion

7.1 Risk Resolution

In reviewing our risk analysis table created at the beginning our project, we realize we were affected by every single one of them and more, regardless of our planned mitigation techniques. To be expected, using unfamiliar technology was a risk that we were very wary of. In order to combat this, we tried to learn about each technology we planned to use before any implementation even began. We predicted that one of our team members may get sick as that is just life, so we tried to stay as healthy as possible. We definitely did not predict the national pandemic that began just before the major implementation of our project started. Seeing that there was ample possibility a key component might be missing from our final application, we attempted to be in constant communication with our client so we could gain an understanding of exactly what they expected. We understood that the most important part of this project was ending with a functional application. This meant prioritizing critical components and understanding which features would have to be put to the back burner. We also tried our best to stick to our development timeline which we had built some buffer time into. While these mitigation techniques to resolve any risks were helpful, there is no way to predict exactly what would happen over the course of development.

7.2 Obstacles Encountered

As first time app developers, we did not realize the understanding needed of certain technologies to create a full-stack application. Beginning with Firebase as our database, we struggled to connect it to the front end of our application. Although we had researched it before implementation, it is hard to predict how exactly a technology will work unless one has prior hands-on experience with it. This led us to switch databases to MySQL, one that both team members had previous experience with.

There were viruses and bugs throughout the development process as well, both in our code, but also in our personal lives. We found bugs in our code which sometimes kept us focused on one small aspect of the application for too long. The unprecedented COVID-19 virus and other personal illness also halted development at times. Unfortunately, these had great affects on the finalized product. It impacted the efficiency of our team as we were not used to working separately. We noticed the biggest effect in not being able to pair program as were used to before being socially distanced. We got stuck on issues often when we were not together, which led to major time loss as we were not able to collaborate as easily.

In experiencing these obstacles, we were forced to decide which features were most important - usually deciding to focus on the critical components first.

7.3 Lessons Learned

We learned quite a few lessons throughout the app development process. The first lesson we learned was when working with our clients, it was important for us to completely understand what features they wanted in the application and for them to understand what we could do with the technology and resources available to us, in order to design the most illustrative conceptual models. Often times when discussing the design of the application, features were understood differently by different parties, leading to the need for clarification before committing to a design.

Another problem we faced was focusing on the details before designing and implementing at a high-level. Although it's intuitive to want to focus on the details, such as simple user-interface nuances, it's more important to first create a backbone for the application. Looking back now, the first thing we should've done was get every component of our stack fully working before starting with the user interface and other small details. That way we would have run into the problems we had with Firebase earlier, and we could've changed course earlier in the development of our app. We also should've started development earlier, even if we didn't have all of the design details worked out with our client. That way, we would've had more time for implementation.

Finally, we learned that an easy way to get over technological hurdles was to switch tasks when we got stuck. When you work on something for too long you often get tunnel vision and can't identify the problem. Often times when the the other person took over the task we were stuck on, they were able to fix the problem very quickly. We definitely work better when we are in the same room and are able to bounce ideas off of each other and quickly ask for help when we need it.

7.4 Next Steps

As mentioned before, unfortunately, we had to cut down on the scope of our application and remove the weather distribution portion of our app in order to give our client, ASDENIC, a fully-functional app. The next step for the application is for ASDENIC and the people of Nicaragua to use it to distribute accurate, real-time information about water quality in their area. Hopefully, another group of students will continue building the weather distribution side of the app for their senior design project next year.

Appendix A

Installation Guide

This installation guide will show you how to download the repository for this app from GitHub and run it locally on your computer.

A.1 To Download and Use the Beta Test of the App on Your Phone

1. Download 'Testflight' from the iOS App Store or the Google Play Store.

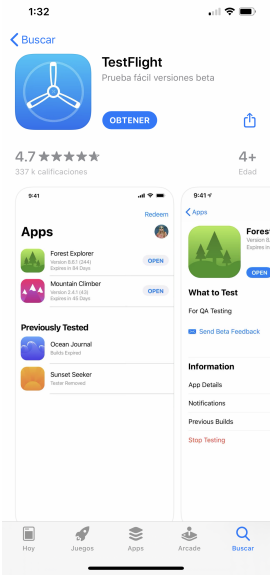

Figure A.1: Download Testflight from App Store

2. Open Testflight and click ”Redeem” in the upper right hand corner.

Figure A.2: Open Testflight Application

3. Enter the invitation code sent to you. If you do not have an invitation code contact Allan Baez Morales at the Frugal Innovation Hub.

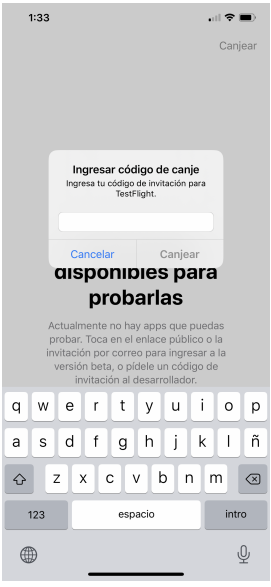

Figure A.3: Insert Unique Code to Testflight

4. The app will download onto your phone and you can use it!

A.2 To Download and Edit the App

Clone the Git Repository to Your Local

1. Go to https://github.com/srortiz/Senior-Design and select the button that says ”Clone or Download” on the right.
2. Copy the link to your clipboard.
3. In terminal, go to the directory that you would like your local repository to be in.
4. In terminal, type ’git clone’ and then paste the link and press enter.
5. Your code will now be on your local in a folder called Senior-Design.

Download Necessary Dependencies

1. Download Node.js and node package manager here: https://nodejs.org/en/download/
2. Now that you have npm downloaded, you can use it to install other dependencies.
3. In your command line, go the your Senior-Design directory and type in the following commands:
   - npm install -g expo-cli
   - npm install -g mysql
4. You may need to download additional dependencies later.

Install MySQL Workbench

1. Go to https://dev.mysql.com/downloads/workbench/ and download the correct version for your OS.
2. Once downloaded, open MySQL Workbench and click on ”Database” >”Connect to Database.”
3. Email professor Angela Musurlian for the correct credentials.
4. You can now access and edit the database!

Run the Application

1. Download Expo onto your smart phone from the Apple App Store or Google Play Store.
2. In terminal, go to the Senior-Design directory.
3. Type the command, ’node server.js’
4. Open another terminal and go to the same directory.
5. Type the command ’expo start’ and you should be brought to a site on your web browser. Use your phone camera to open up the QR code.
6. You should be brought to the app on your Expo app on your phone.

Appendix B

Works Cited

Asociación Para El Desarrollo Social De Nicaragua. Asociación Para El Desarrollo Social De Nicaragua: ASDENIC. https://www.asdenic.org/

Larsen, B. (2013). *Economics of Sanitation Initiative - Nicaragua: The Cost of Inadequate Sanitation*. Water and Sanitation Program, World Bank.

Walker-Leigh, V. (2011, August). *Country profile - Nicaragua*. New Agriculturist. http://www.new-ag.info/en/country/profile.php?a=2143