

VIRGINIA TECH



Blacksburg, VA 24061  
CS 4624 – Multimedia, Hypertext, and Information Access  
Fall 2021

**Final Report**

**WildLife Tracker App**

**Instructor:** Dr. Edward A. Fox

**Client:** Luis E. Escobar

Authors:

Ziwei (James) Kou

Muyang Wang

Boshen Zhang

Kevin Zhang

Dec. 8, 2021

# Table of Contents

<b>Table of Figures</b>	<b>3</b>
<b>Table of Tables</b>	<b>4</b>
<b>1 Abstract / Executive Summary</b>	<b>5</b>
<b>2 Introduction</b>	<b>7</b>
<b>3 Requirements</b>	<b>8</b>
3.1 Website	8
3.2 App	8
<b>4 Design</b>	<b>9</b>
4.1 Design of the App	9
4.2 Design of the Website	9
<b>5 Implementation</b>	<b>10</b>
5.1 IOS Application	10
5.2 Website	11
<b>6 Testing</b>	<b>13</b>
6.2 Common Users	13
6.2.1 Website	13
6.2.2 App	14
6.3 Advanced Users	14
6.3.1 Website	14
6.3.2 App	14
6.3.3 Assessment	14
6.4 Target Table	16
<b>7 Users' Manual:</b>	<b>17</b>
7.1 IOS Application	17
7.1.1 About Us	17
7.1.2 Upload Tasks	17
7.1.3 History	18
7.1.4 Map	18
7.1.5 Settings	20
7.1.6 Login View	24
7.2 Website	28
7.2.1 Home Page	28
7.2.2 Map Page	29

7.2.3 Spreadsheet Page	31
7.2.4 App Page	32
<b>8 Developer's Manual</b>	<b>33</b>
8.1 IOS App	33
8.1.1 Environment	33
8.1.2 Clone the project to the local environment	33
8.1.3 CocoaPods and Firebase	33
8.2 Website	34
8.2.1 Firebase install	34
8.2.2 Firebase Storage and Firestore	34
8.2.3 Mapbox	35
8.2.4 Implementation Detail	35
<b>9 Lessons Learned</b>	<b>39</b>
9.1 Timeline	39
9.2 Problem / Solution	40
9.3 Future Work	40
<b>10 Acknowledgements</b>	<b>41</b>
<b>11 References</b>	<b>42</b>

## Table of Figures

Figure 1: Survey	13
Figure 2: Zoom out map view	18
Figure 3: Zoom in view with one pin clicked	19
Figure 4 Settings View	20
Figure 5 list of security questions	21
Figure 6 Entering security answer	22
Figure 7: Entering password	23
Figure 8: Success Prompt	24
Figure 9: Login View	25
Figure 10: Invalid Password	26
Figure 11: Password Reset	27
Figure 12: Home Page	28
Figure 13: Map page	29
Figure 14: Pin of map	29
Figure 15: map page when logged in	30
Figure 16: Spreadsheet page	31
Figure 17: Spreadsheet page when login	31
Figure 18: App Page	32
Figure 19: Git repository	33
Figure 20: Firebase Storage	34
Figure 21: Structure of Firestore	35
Figure 22: code used to import mapbox	35
Figure 23: code for logo, title, and navigation bar	36

## **Table of Tables**

Table 1: UX Assessment Table	16
Table 2: Problems and Solutions	40

# 1 Abstract / Executive Summary

In the previous semester, Dr. Luis E. Escobar from the Department of Fish and Wildlife Conservation initiated a project called Wildlife Diseases Tracker App to help with his research studies. The original project aimed to provide a free-of-cost mobile application with a website to accomplish early detection of wildlife diseases. During this current semester, our goal was to update and finalize a new version based on the previous project and work on the continuation of the smartphone application and the website to track wildlife diseases in real-time.

The project has two divisions: an easy-to-use mobile application and a website. The goal for the mobile application is to update and expand it with the capacity to take photos and get the geolocation and date of the photo. It is able to record metadata and upload data to the server. The website is updated with the functionality to summarize the data and images collected for analysis, mapping, and creating a more aesthetically pleasing user interface. We wished for the implementation of a cross-platform mobile application; however, due to technical difficulties and time constraints, we could only update and finalize the existing IOS version of the smartphone application.

For our IOS application, we have completed an improved interface while conducting tests and fixing existing bugs. We also improved the existing Upload functionalities by adding more input fields and adding new functionality including map and login credentials. A Face ID/Touch ID implementation for users was also created for easy login access. As for the website, an updated version of the home page was constructed with basic information about our project, and instructions for downloading and using the application. We also updated the map and the spreadsheet, so it displays downloadable data to users. To test, we released a beta version and sent out questionnaires and surveys for students and collaborators to examine functionalities, aesthetics, and accessibility.

The final deliverables for the project include a final report that explains details of our project and a final presentation slide deck which we used to display what we have accomplished throughout the semester to the class, our client, Dr. Escobar, and Professor Fox. In this final report, we discussed everything we have achieved throughout different phases of the project, from the beginning to the end, with specific details. The order is as follows: We first introduce our project and talk about the project overview. Next, we analyze the requirements and expectations from

our client, Dr. Escobar, and discuss how they lead to our designs and actual implementations. Following is the plan for testing our updated versions of the website and the IOS application. For ease of use and future possible pick-up of the project, we also introduce the Users' Manuals and Developers' Manuals sections for users and future developers, respectively. We finish this final report by talking about some lessons we have learned through the entire project, future implementation directions, acknowledgments, and references. Our presentation to the class and the slide deck worked as an alternative and rather a quick way to show what we have accomplished during this semester in a limited period of time. We have covered most of the aspects in our presentation; however, due to the time constraints, what we discussed in the presentation was not as specific and detailed as what we have written in this final report.

As for the expectation towards our project, we hope that the mobile application and the website can be useful for early detection of emerging diseases of wildlife origin, such as COVID-19 and Ebola. The app can help detect and contain diseases that can affect humans, wildlife, pets, and livestock globally and thus should be treated seriously. We would also like to see our effort made in this project helpful to both students participating in related studies and to our client, Dr. Escobar.

## 2 Introduction

Many emerging diseases, such as COVID-19, Ebola, and the avian flu, have origins in wildlife. We, as educated scholars, need to dive into the reasons behind and come up with effective strategies to handle situations, especially when the emerging disease is zoonotic, like COVID-19.

There are many causes why we are facing the challenges of zoonotic diseases. According to Faunalytics, “One of the biggest reasons that we face these threats is because of our encroachment into wild spaces. Deforestation to make way for grazing land has displaced many wild species and brought unknown numbers of viruses in closer contact with people like never before.” [1]

As human beings invade nature, we also capture and kill wildlife animals and then bring them back into human areas for different purposes such as consumption, companions, trophies [1, p. 4]. Such activities are extremely dangerous, as the risk of disease uncertainty remains unknown. COVID-19 is a good example.

Prevention, detection, and control of these potential emerging diseases become paramount. According to Robert G. McLean from the University of Nebraska, “Theoretically, the most effective prevention and control methods are obviously to prevent the introduction or emergence of wildlife diseases of public health or animal health importance rather than attempting to control them after they have become a problem. Prevention of introduction is a daunting task because of the many sources and routes of introduction into the United States, and early detection is one of the keys to prevention and rapid control or containment.” [2]

Even though the monitoring process can be expensive and time-consuming, it is still crucial to monitor and track wildlife diseases. As McLean suggested, early detection is one of the keys to prevention; citizen science brings an opportunity for this. With the use of knowledge in the Computer Science field, the previous group that started this project from scratch focused on the development of a mobile application freely available to the public to track infectious diseases in wildlife [4]. Our task is to develop a better version by updating and finalizing, based on previous work.

## **3 Requirements**

When we talked to Dr. Escobar, he gave us specific requirements for both website and app: updating the previous group's work [4] to make it run better and make it look better too.

### **3.1 Website**

For the website, Dr. Escobar asked us to make the page look better and more comprehensible to new users coming into the website. One of the things that he asked us to do is to make an instruction page on how to download the app as it is not on the app store, and downloading the app is a confusing process. Additionally, he wants the CSV file to be more noticeable, and be downloaded easily. We then discussed how it could have multiple web pages to alleviate that problem, and Dr. Escobar said he liked the idea. We then focused on making multiple pages that solve his requests.

### **3.2 App**

Dr. Escobar stated he wants a better way to collect the data the user wants, which is up to us to implement. Another thing he wants us to implement is by using the photo the user shot, we would be able to extract information from the photo that could be helpful to use, ex., the location of the user shooting the photo. Lastly, he wants a better way to improve storing information from the user to be stored into the website.

## **4 Design**

### **4.1 Design of the App**

For the most part the app remained similar to the previous group's app [4] with some more tabs added and more design overhaul. We added two more tabs that the previous app didn't have, one of them being the map, and a setting functionality. The setting was used in conjunction with the new login tab, where the user is able to have a security question and be able to change their password on that tab. Another thing we added was a map that worked similarly to the map on the website, where the user can scroll around and check the pinpoints of the data they've uploaded as well as from other users. Aside from that, most of the tabs and functionality remain the same. The upload tab can take a picture that users want to upload as well as information they want to give be stored in the database. The about us tab is used to provide information on the project as well as provide information about Dr. Escobar so that people can contact him or know more information about him. Lastly, the history tab shows a list of the users' uploads with metadata showing on each item.

### **4.2 Design of the Website**

There was only one page for the user to view the website from the previous group work, and it looked cramped. Thus, we decided to make multiple pages with multiple separate functions so that the user is able to discern what they are able to do on each page. On the home page, we want to have a description of the project overview and have contact information for Dr. Escobar, so that people can contact him for questions or ask him anything about the project. We have a download instruction page to provide explicit instructions and pictures for the users so they are able to understand how to download the app. Next, we planned the map page that displays the last group map on the entire page, making the map more readable and having the user be able to scroll and zoom the map to find points where other users upload their data. Lastly, we planned a page where the user can download a CSV file and browse data that has been sent to the website in a table form.

## 5 Implementation

### 5.1 IOS Application

Since the goal of this project is to help with Dr. Escobar's research, our design of the application was mainly focused on how to optimize the experience of tracking wildlife diseases. Based on the IOS application made by the former group, we have updated and finalized the app according to the client. Several main features were added including 1) map, 2) login credentials, and 3) detail view. Because this is a continuing project based on an existing application, we have also fixed several UI bugs in order to provide users with a better experience when accessing the app.

Two-tab bars were added to the main screen of the application, one of them being the map functionality. As a result of group discussion, we wanted to grant users more access to the information about diseases and therefore reduce the time when collecting data from the app. We decided to make a map that includes different markers representing each entry uploaded by users. This way, users can view the information in a way different from a list. Each marker is also accompanied by relevant information including location, experience level, and upload time. Whenever a user uploads an entry about the disease, it is automatically updated on the map.

Another functionality we decided to implement is the login credentials. We added another Settings tab bar for this functionality. Users can set a password and security questions on the settings page. In order to successfully unlock the app, users also need to verify the password correctly. We implemented this function to avoid the situation when users mistype their passwords and therefore are not able to open the app. Another handy feature we implemented is an option to show passwords entered. If the toggle for Show Entered Values is on, the user input shows as consecutive dots instead of the actual input. We did this in order to protect users' privacy. Users do not need to fill in login credentials the first time they log in to the app. After they set the password, users have the option to reset the password as long as they have the security questions entered correctly.

Thirdly, we implemented the detail view for the records list. When using this application, users were not able to access much information, like contact information, specific location, and additional notes. After the implementation, when the user clicks on the item in the history list, the user can get a more detailed understanding of what each record looks like. We also decided to

convert longitude and latitude to a specific location that is human readable. By doing this, the user can have a clearer idea of where the image was uploaded.

We also decided to fix several UI bugs including some mistyping on the screen. We noted that this app is not working properly when the phone is in dark mode. After several instances of testing and some fixes, the app worked both in bright and dark mode.

## 5.2 Website

The previous group that worked on this project created a functioning map giving information about photos stored in the database with the help of Firestore. Therefore, our main focus was to provide more information about our project[10] with the website. You can visit our website at <https://recorder233.github.io/4624website/home.html>.

Compared to the previous website [9] which only contains a map, a short description, and a download link for data, all in a single webpage, we created a web page with the following:

- Home page
- Map page
- Spreadsheet page
- App page
- Admin login page

The home page is an introduction to our project. It provides some necessary information which includes the purpose of our project, basic information about the disease, and how users can help us.

The map page is dedicated to the map. It fetches pictures stored in our cloud storage and displays them in the corresponding location on the map. To achieve this, the **Mapbox SDK** and **Firestore SDK** were used. Mapbox is a map provider that allows its user to create custom maps. The previous group used it to make a map with pins indicating the location of a picture that was taken. The previous group created a cloud storage representation using Firestore, although we can not log into the account that owns the cloud storage. Firestore provides a specific API and we have the JSON token necessary to access and change the data, so we decided to keep using it.

The spreadsheet page supports downloading the data in the form of a CSV file. It also displays a table of pictures in our database with the time the picture was taken and the contact information of the uploader.

The app page provides instructions about how to download our app. Similar to the home page, it tells the user how to download XCode which is used to install the app. It includes pictures, guiding the users to make the download process easier.

# 6 Testing

The main focus of the testing is to make sure that the app is easily downloadable for all users when following the directions, and that the app itself and the website can be navigated with ease. Additionally, we needed to make sure that the users are able to discern where to find specific information or functions for the app and the website. However, we also needed to make sure that the app has met all the minimum requirements for the client so that they are able to use the app in the future for its main purpose. As such, we conducted tests on two separate groups. One group includes our peers and friends in order to get insight on the usability of the app and website. The second group provides deeper insight on the functions of each tab, seeing if they meet the requirements. To evaluate the data, we provided questionnaires to both groups. Figure 1 shows the survey employed.

The image shows a survey form with the following sections:

- Title:** App and Website
- Introduction:** This survey will be conducted to evaluate how well the website and app function and how friendly it is to new users.
- Question 1:** How would you rate navigating though the app? \*  
A 10-point Likert scale with radio buttons. The scale is labeled "1" to "10". The left end is labeled "I don't know where I am" and the right end is labeled "I know where everything is".
- Question 2:** Was there any issues when navigating the app?  
A long-answer text field with the placeholder "Long answer text".
- Question 3:** How difficult is it to upload your picture? \*  
A 10-point Likert scale with radio buttons, labeled "1" to "10".

Figure 1: Survey

## 6.2 Common Users

### 6.2.1 Website

For testing, we decided to organize supervised testing and unsupervised testing. We would be observing some groups and asking them directions and feedback while they figure out how to

navigate through the website, while the others are given instructions and the website to see if they can solve it by themselves. The main purpose of the supervised testing is to observe some actions that users might not have caught that could be made better, such as trying to click a link that hasn't been linked, or thinking the buttons that navigate to another page were smaller than they expected. Most manage to do well in navigating pages and downloading the CSV file, but some did struggle to make sure the app can be downloaded. After they finish downloading the app and navigating through the pages of the website, we move on to testing the app.

## **6.2.2 App**

After successfully downloading the app, the users do a similar test as they did on the website. They navigate through the pages and try to upload the photos with information to be sent to the website. They won't be checked if the function works, but if they can, do the task given to them. After completing the task, they are given a survey. They are to determine, on a scale from 1-10, for each task, if they are satisfied with the website and app.

## **6.3 Advanced Users**

### **6.3.1 Website**

The more advanced users, those who are in the Wildlife Disease Team, check to see if the information provided meets their requirements and needs and if the implemented functionalities work as intended. They do tasks the common user would do, just to make sure everything functions well. For the website, it is the same with the common group as they need to make sure the instructions are clear and the CSV file is downloadable. They also check the CSV table to make sure that the data provided shows up and that the data shows on the downloadable page. Our testing showed that all of the deliverables look good and the information the CSV provided looks good. We just needed to improve some grammar and develop a more aesthetically pleasing interface.

### **6.3.2 App**

The users we examine the mobile application with also test the functionality of the app, seeing if it contains the information they wanted to include with the photo, and sending it to the website. Additionally, they check the types of information that can be sent through and how the login feature works.

### **6.3.3 Assessment**

We evaluated the test by use of a UX table, where we were obtaining the averages of each test based on the scores given in the questionnaire. We had a target level to reach, and we compared what we obtained from the questionnaires to our expectations. This provided a good sense of the

usability of each function and regarding the design of the website and app. It helped with ideas to improve each part of the app and website.

UX: Goal	UX: Measurement	UX: Metric	Baseline Target.	Target Level	Observed Result
Download the app by following the website instructions	Questionnaire	Using a questionnaire with a scale of 1-10 focusing on questions 1, 3	7/10 While the instruction looks clear there might be some hiccups.	8/10 Max Score: 9 Min Score:6	Some struggle with downloading X-code; may need to provide more clarity.
Upload the photo with information	Questionnaire	Using a questionnaire focusing on question 2	9.5/10 The images we choose are quite clear and the instruction is simple.	10/10 Max Score: 10 Min Score: 10	Seems to be relatively clear. Needed some time to figure it out.
Lookup Other people	Questionnaire	Using the questionnaire question 4	9/10 A little clunky moving the map but should make sense.	9/10 Mac Score:10 Min Score: 8	The only issue seems to be when the pins are too close to one another.
Download the data needed	Questionnaire	Using a questionnaire focusing on questions 5	10/10 Navigating and downloading should be	10/10 Max Score: 10	Could make the blocks smaller on the webpage but overall very solid.

			relatively clear.	Min Score: 10	
The photos uploaded have the information needed	Questionnaire	Questionnaire focusing on 6 and 7	8/10 All the data is there, it could be formatted a little more differently.	8.5/10 Max score:10 Min score: 8	The data looks fine, but we might want to change some of the ways we display the information.

Table 1: UX Assessment

## 6.4 Target Table

The target table is used to report our findings after testing, figure out the user's average scores, and note down any additional observations that we noticed during testing. We had 4 people testing for the common user perspective and 1 person for the advanced user perspective. We had two people in common do the test without our supervision while the one person who was in our advanced group did their testing with supervision. We did have everyone test the same functionalities. The common users checked to see if the function worked and the advanced users provided any insight into if it was working the way they wanted. The table has been split up to note the following information:

**UX Goal:** This section notes what task we wished to accomplish while the users test the app and the website.

**UX measurement:** This section notes how we would evaluate our tests. In all cases, we are using the questionnaire to users for evaluation.

**UX metric:** This part specifically aims at how the scores are to be divided from the measurements. In this case, we have specific questions pertaining to specific actions on the questionnaire, and we would use those scores from those specific questions to evaluate the average for their corresponding test.

**Baseline Target:** This is what we think the predicted average score would be. This score is what we think the usability and clarity of each test are at right now. If the target score is higher, then we exceeded our expectations. If lower, then we did not implement the functionality well.

Target Level: This is the average score that we calculated for each person. For the sections that have two questions that pertain to the test, we would get the average scores of both questions then get the average score of the results.

Observed Results: Any additional information that we noticed during the supervised testing is noted there.

## **7 Users' Manual:**

### **7.1 IOS Application**

The goal of this application is to provide a platform for users in the field of wildlife conservation to use this application flawlessly. This is an application that enables users to upload their observations to the database and therefore to the website. Moreover, while the application is providing the basic functionalities, we wanted to make the application more accessible to more general users.

After downloading and installing the application, there is a small application icon named “TrackDiseases” on your screen. Clicking onto the application takes you to the login page of the application where it asks for your touch ID to unlock the application. You are able to login into the application if you give the correct touch ID. If you did not set up a touch ID on your iPhone, you are taken to the login page of the application; click on login if it is your first time using the app since there are no login credentials.

#### **7.1.1 About Us**

This page serves as a Home page when users first launch the app. There are three sections on this page including Project description, Profile, and Mange Description. By clicking on the blue button on the screen, the user is directed to another website. For a more detailed description, please refer to the report from the previous group [4].

#### **7.1.2 Upload Tasks**

The Upload Tasks bar provides users a way to upload pictures to the database, which are then displayed on the website. It asks users to upload specific information including the level of expertise, location, and additional comments. There are detailed instructions at the bottom of the screen. After the upload process, users are prompted as a sign of success. For a more detailed description, please refer to the report from the previous group [4].

### 7.1.3 History

When the user first logs in and no records are uploaded, this page does not display any entry. After the user uploads records using the Upload Tasks page, records then appear on the screen as a list. These are results fetched from the database and the user is able to navigate to obtain more detail by clicking on each item. The user is able to see the date, location, and a picture snippet for each item from the list. For a more detailed description, please refer to the report from the previous group [4].

### 7.1.4 Map

The Map function works similarly to the history function. Instead of presenting all of the records in a list view, it presents data in a map view. If the user has not uploaded any record in the Upload Task function, the map is empty for the first time but it is then updated once the user uploads any record.

As long as the history tab is not empty, there is an annotation pin appearing on the map. As shown in **Figure 2**, there are two pins on the east coast of America.

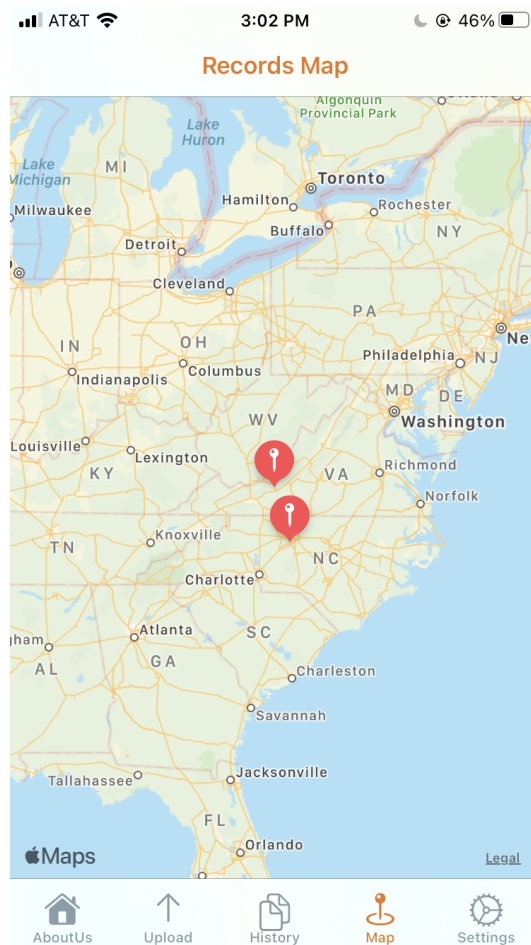


Figure 2: Zoom out map view

The map enables the user to zoom in and zoom out to navigate through the map. Moreover, when users are interested in a particular position, they can click on the annotation pin on the map and it shows the relevant metadata, as in **Figure 3**.

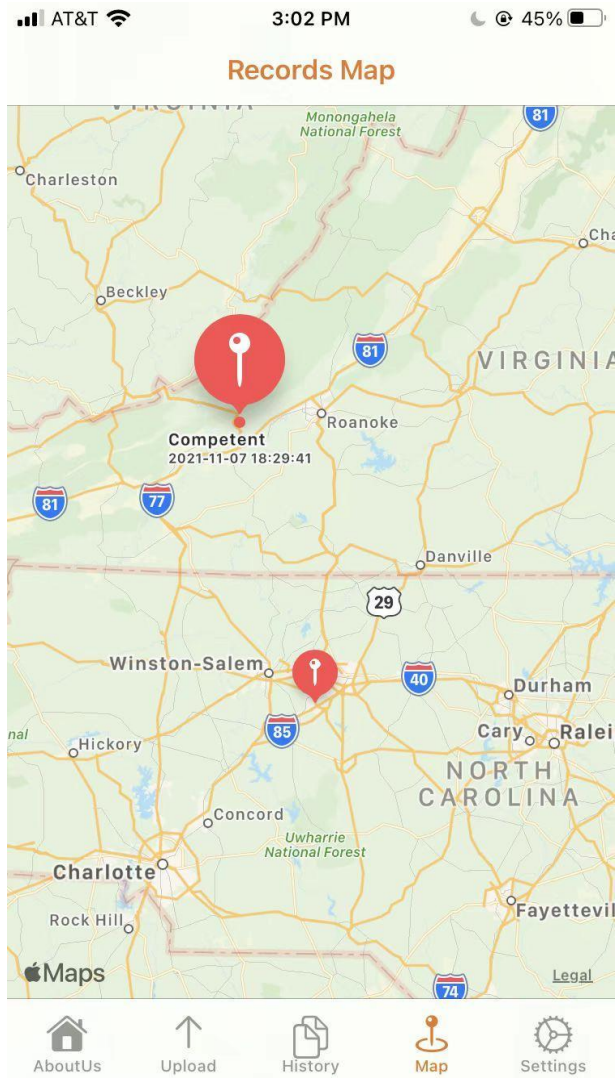


Figure 3: Zoom in view with one pin clicked

## 7.1.5 Settings

The Settings page is aimed to give the user options to create their own password unique to the TrackDisease app. Supposing the user wants to set a new password, the user needs to navigate to the Settings page, as shown in **Figure 4**.

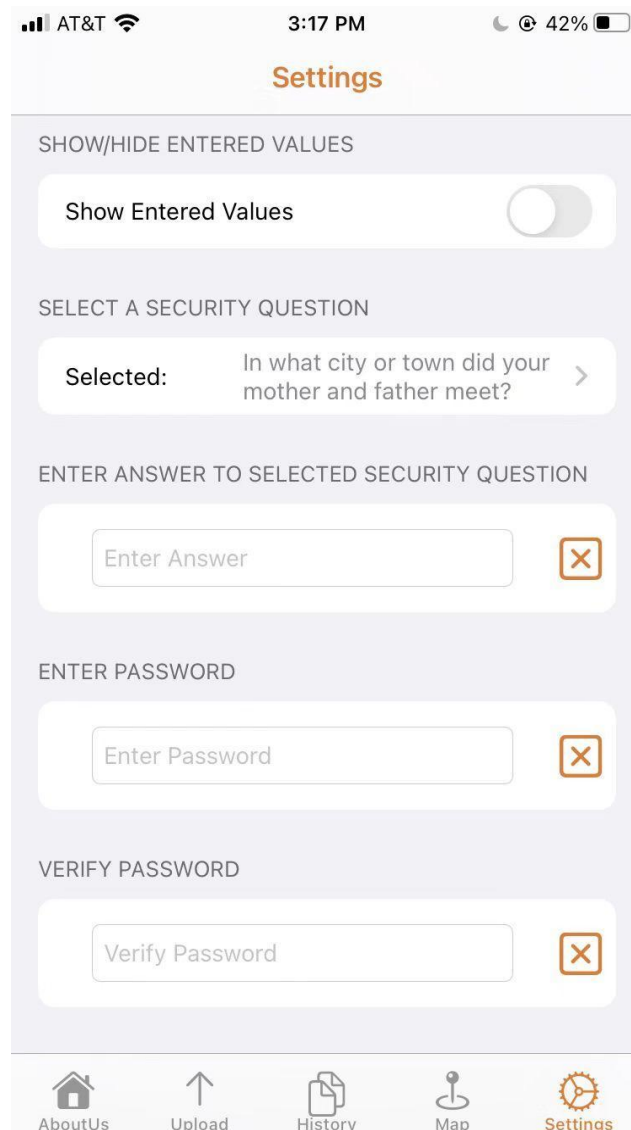


Figure 4: Settings View

They first need to select a security question by clicking on SELECT A SECURITY QUESTION. By doing this, it takes the user to a list of security questions to select from. As shown in Figure 4, we selected the first question.

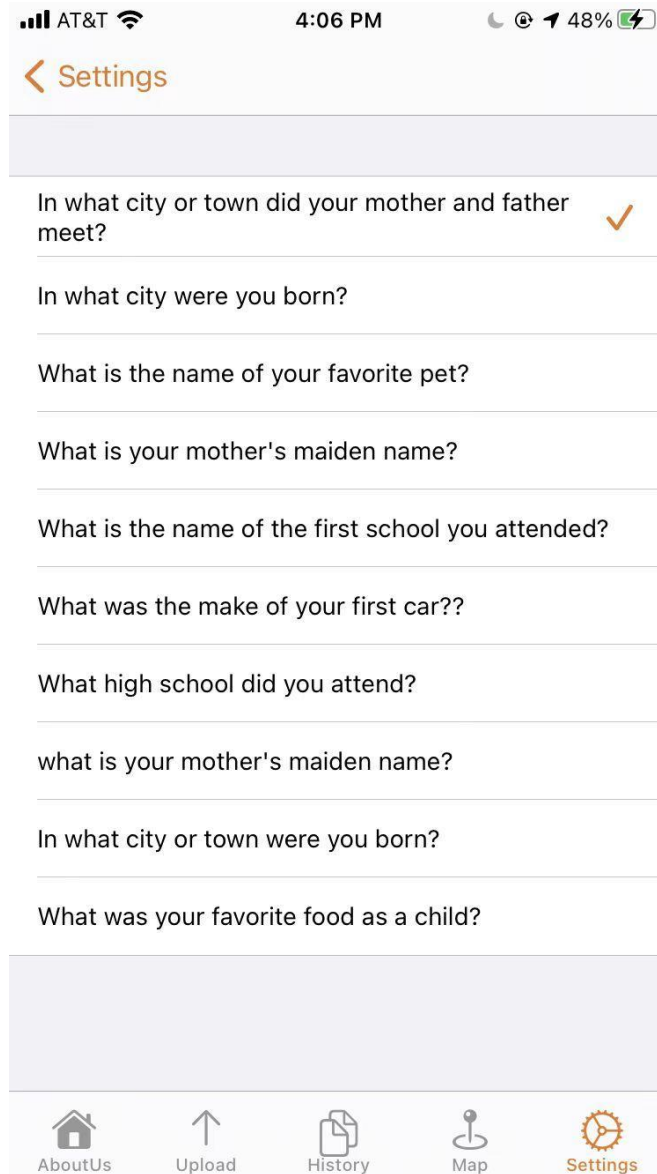


Figure 5: List of security questions

After selecting the security question, enter an answer in the following input field for recovery purposes. See **Figure 6**.

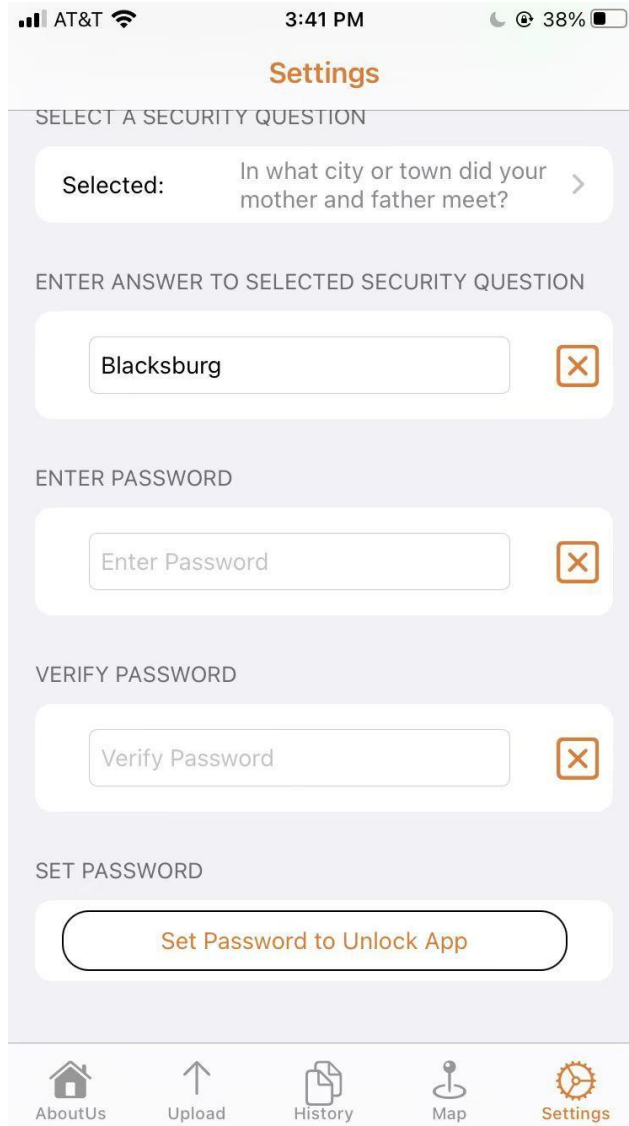


Figure 6: Entering security answer

The user then should enter the password two times in the input field. As shown in **Figure 7**, the user enters “Password” as their password.

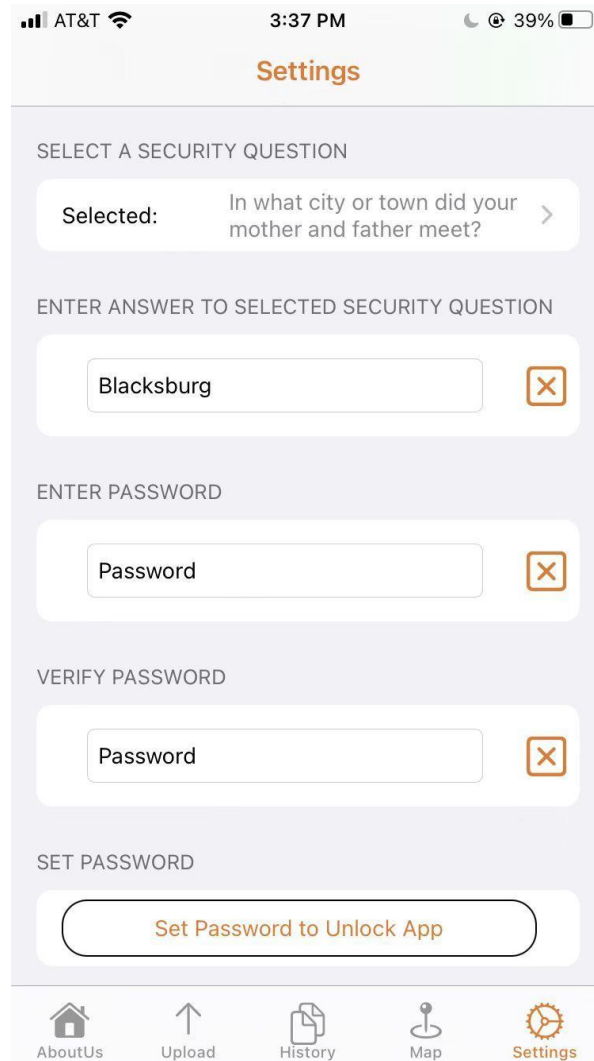


Figure 7: Entering password

The final step is to click on the button Set Password. If the previous steps were conducted correctly, a prompt then shows up to indicate success. See **Figure 8**. Two passwords have to match in order to set the password, otherwise, the prompt does not come up.

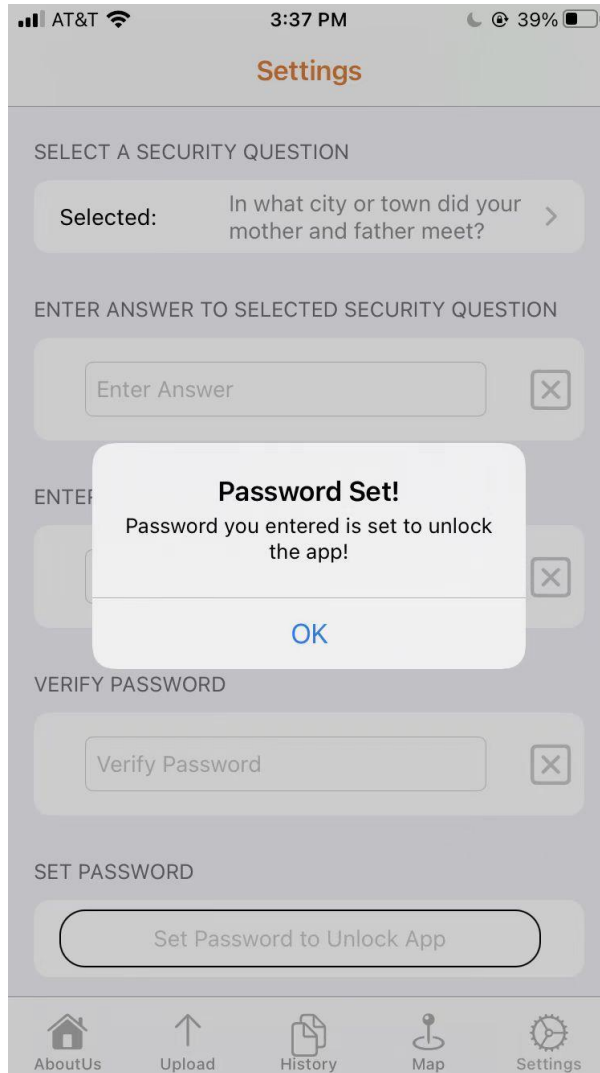


Figure 8: Success Prompt

### 7.1.6 Login View

The Login view is an extension of the settings page which gives users the ability to login and reset the password; see Figure 8.

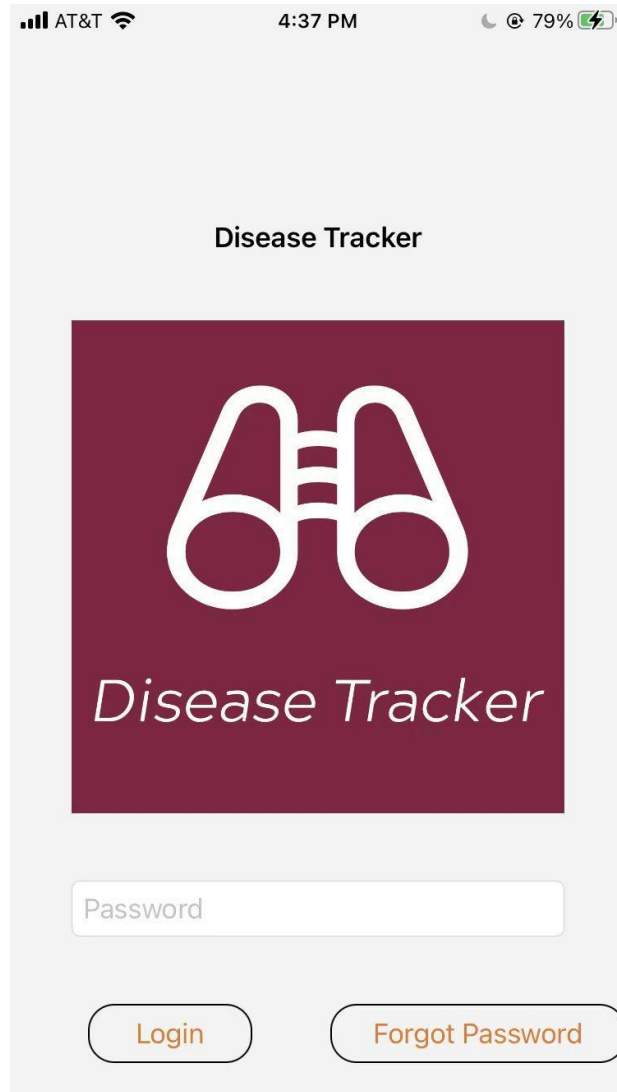


Figure 9: Login View

As shown in Figure 8, the user needs to enter the correct password in order to log in. If the user enters the wrong password, the system then prompts with an error message (see Figure 9). When the user enters the correct password, the user is directed to the home page.

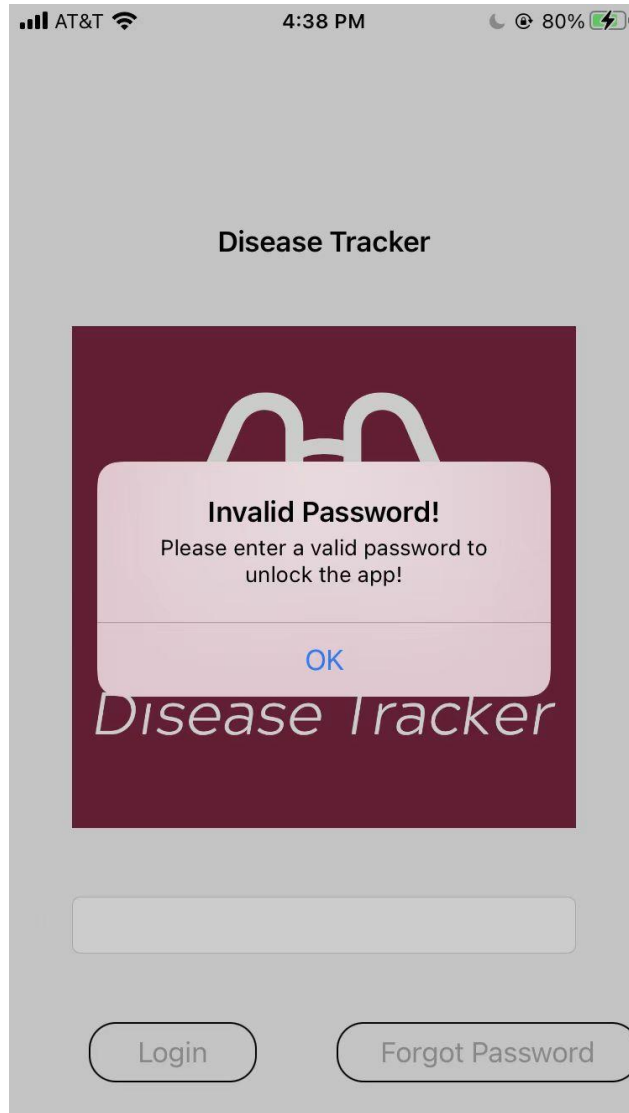


Figure 10: Invalid Password

If the user is trying to recover the password, they need to click on the Forgot Password button which directs them to the recovery page (see Figure 11). The user has to enter the correct answer to the security question to navigate to the settings page.

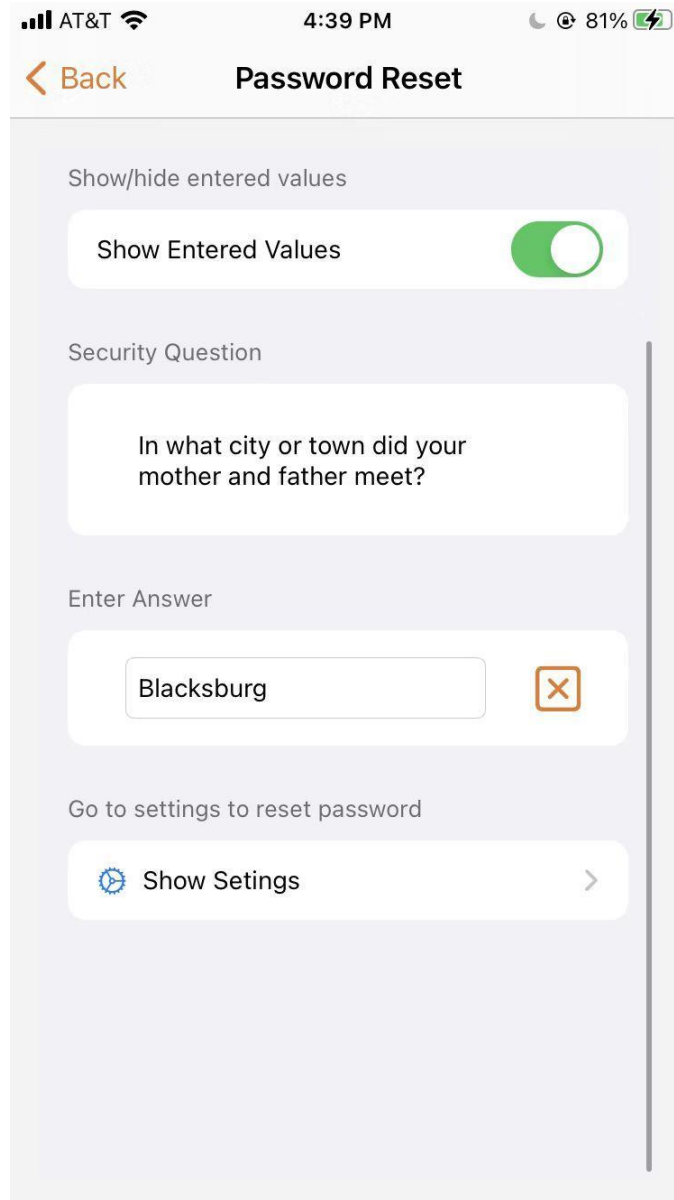


Figure 11: Password Reset

After the above steps, the user should be able to log in with a new password. Users could reset the password in the future anytime they want to.

## 7.2 Website

The website allows the user to view basic information about our project, a map with all the data we have in the database, a spreadsheet of the data, and instructions to download the IOS app. If the user is an administrator, they can remove data from the the database in the spreadsheet and the map. The website can be accessed at: <https://recorder233.github.io/4624website/home.html>

### 7.2.1 Home Page

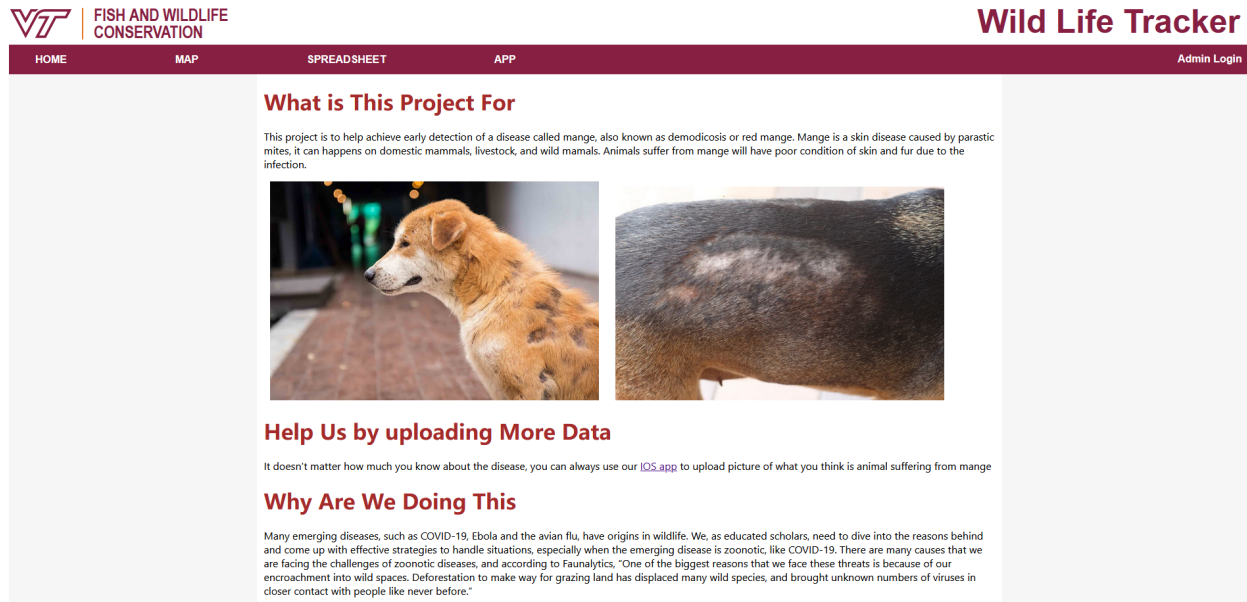


Figure 12: Home Page

The home page is a brief introduction of our project; it has 4 parts:

- The purpose of the project
- How the user can take part in our project
- Why we are doing this project
- Contact information

## 7.2.2 Map Page

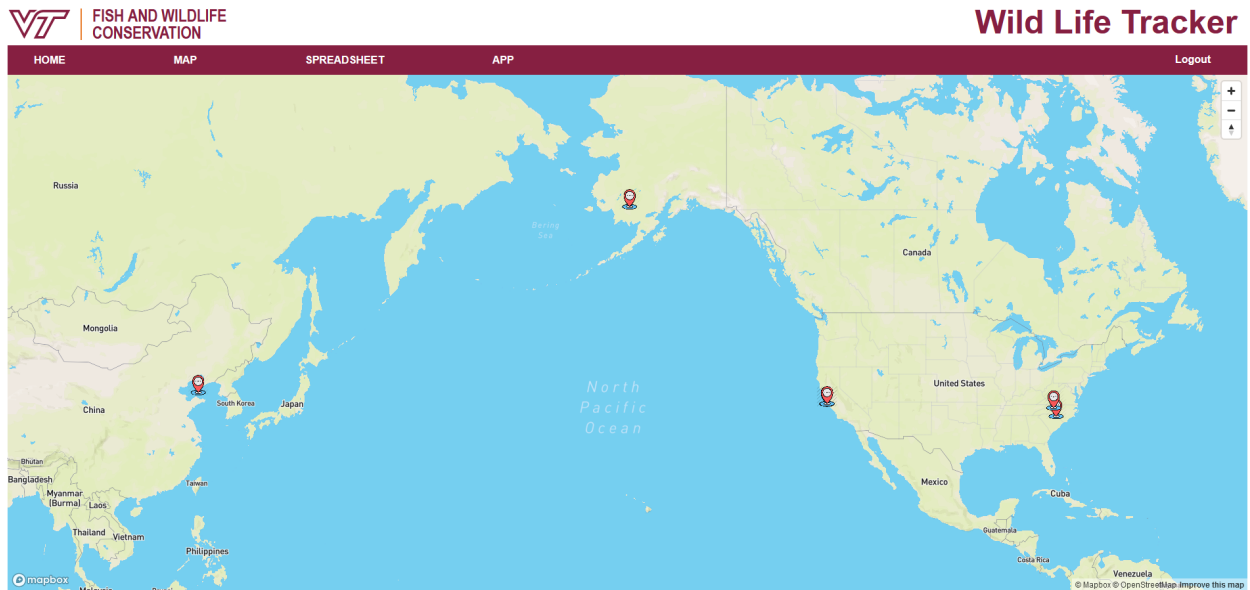


Figure 13: Map page

The map page shows a map with pins. Each pin represents a photograph taken at that location, as recorded in our database.

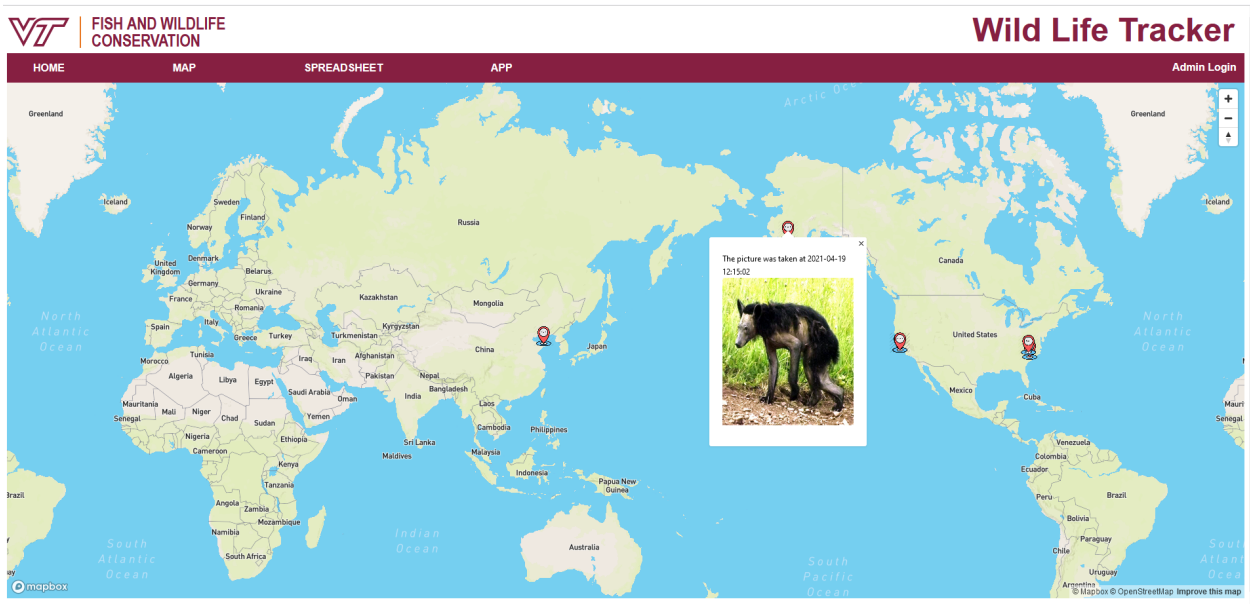


Figure 14: Pin of map

When a user clicks on a pin, the website shows a pop-up that can display the picture and the time the picture was taken.

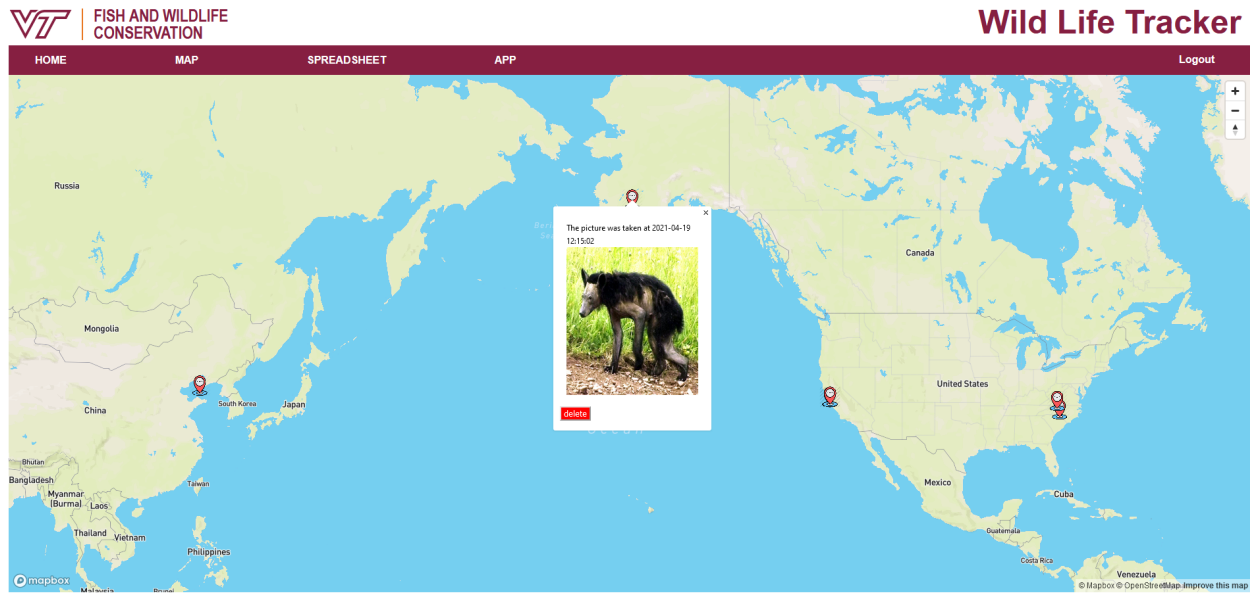


Figure 15: Map page when logged in

If the user has logged in as administrator, at the bottom left corner of the pop-up, a red delete button appears. When it is clicked a confirmation alert appears. If confirmed, the picture is removed from the database.

## 7.2.3 Spreadsheet Page

Time	Uploader	Picture
2021-11-07 22:16:07	Test2	
2021-11-28 23:37:57	Boshenz@vt.edu Boshen Zhang	
2021-11-07 18:30:54		

Figure 16: Spreadsheet page

The spreadsheet page has a link to download all the data in the database as a CSV file and a table of all pictures with time and uploader's contact information.

Time	Uploader	Picture	
2021-11-07 22:16:07	Test2		<a href="#">delete</a>
2021-11-28 23:37:57	Boshenz@vt.edu Boshen Zhang		<a href="#">delete</a>
2021-11-07 18:30:54			<a href="#">delete</a>

Figure 17: Spreadsheet page when login

When a person logs in as administrator, a delete button appears, allowing the administrator to delete the post from the database.

## 7.2.4 App Page

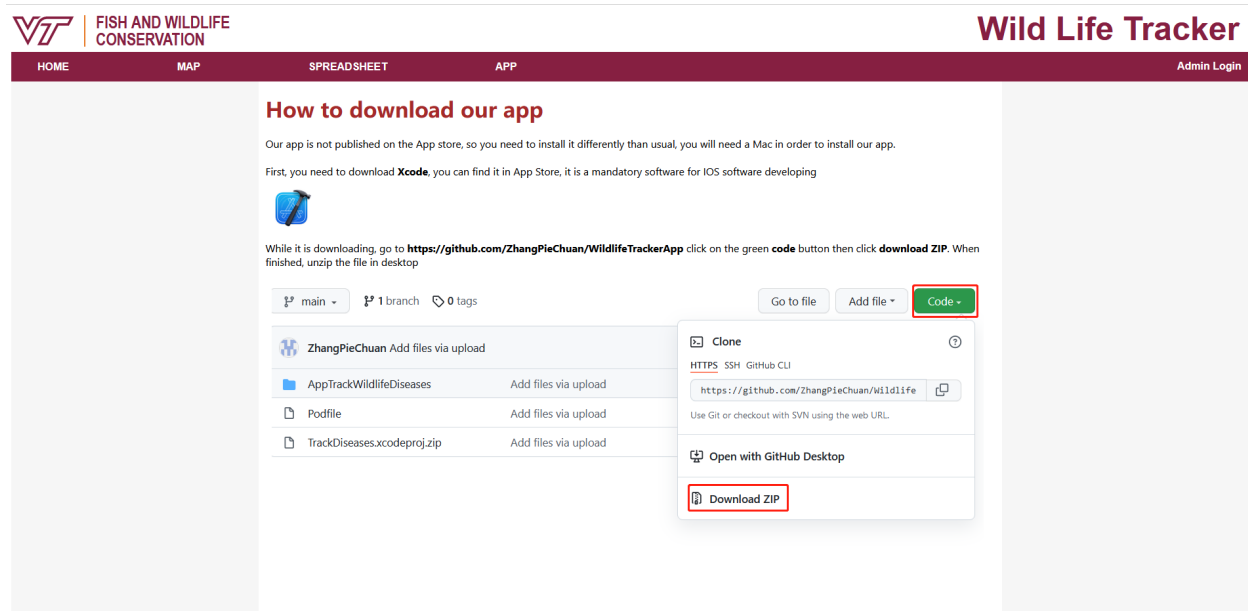


Figure 18: App Page

The app page includes instructions to download and use the app. Our app is not published on the app store, so currently, the user needs to install the app as a developer, which is rather complicated. This page contains a detailed guide about how to install Xcode and other dependencies of our app, and then install the app on the phone and configure it correctly.

## 8 Developer's Manual

### 8.1 IOS App

The updated app utilized Mapkit in the Apple preinstalled environment. The **Mapview** is a customized Swift file called **MapViewAll.swift**. This is a non-draggable, customizable **Mapview** that can display all of the annotations from Firebase. We also made some changes to the content view to cope with two views, **LoginView** and **MainView**. The latter is the place to display all of the tabs after the authentication, while the **LoginView** is displayed before the authentication.

#### 8.1.1 Environment

We are using the same environment as the last group. Please refer to the report from the previous group to find the specific environment for the installation [4].

#### 8.1.2 Clone the project to the local environment

Xcode is only compatible with macOS. You need to download the Xcode from the app store to edit the code. After the installation is complete, clone the project to your local computer by accessing the git repository [6]. You then find the following files in the git repository, as shown in Figure 19.

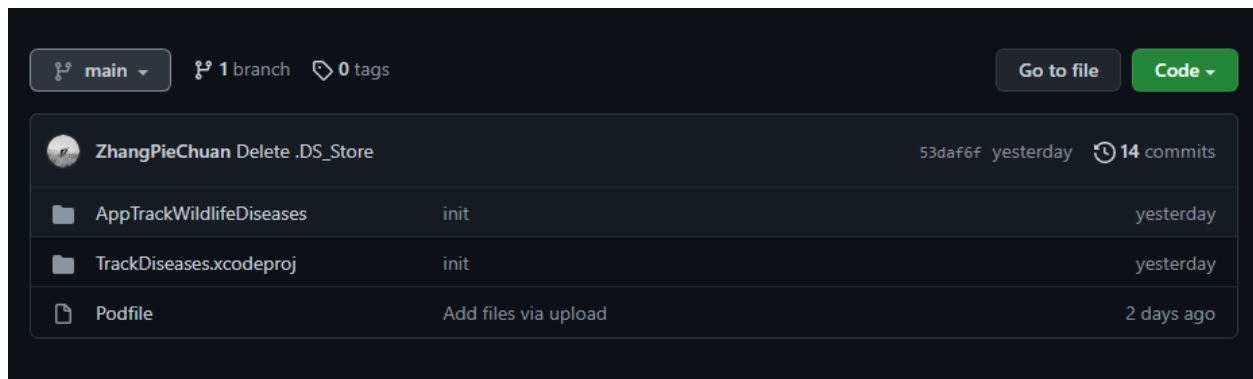


Figure 19: git repository

Click on the green Code button and select download Zip to clone the project to the local machine. Or we can directly clone the project from the Xcode. Refer to the report from the previous group for detailed instructions [4].

#### 8.1.3 CocoaPods and Firebase

We are using Cocoapods for the dependency configuration for Firebase. Refer to the report from the last group for detailed instructions on how to install and initialize Cocoapods [4]. After the

installation of the dependencies, there are three new files created, including **Pods**, **Podfile.lock**, and **TrackDiseases.xcworkspace**. Whenever trying to launch the project, please ignore the **TrackDiseases.xcodeproj** and use the newly generated **TrackDiseases.xcworkspace** instead. You can access and edit the Swift file using the workspace file. At this point, you have finished the installation of all the dependencies and can test the app with the preinstalled simulators.

## 8.2 Website

### 8.2.1 Firebase install

Our website is currently using online resources of Firebase code. If this online resource does not exist anymore in the future, please install Node.js and Firebase 8.3.0 on your server. For Node.js, please follow the download instructions on their official website. After installing Node.js, you can use “`npm i firebase@8.3.0`” and “`npm install -g firebase-tools`” to install Firebase. As for instructions regarding Firebase[7].

### 8.2.2 Firebase Storage and Firestore

Our project depends on the server built by the last group using Firebase which we do not have direct access to. It might be a good idea to set up a new server in the future.

There are two services of Firebase we are using. One is Firebase Storage, as it is used to store the picture the user uploaded.

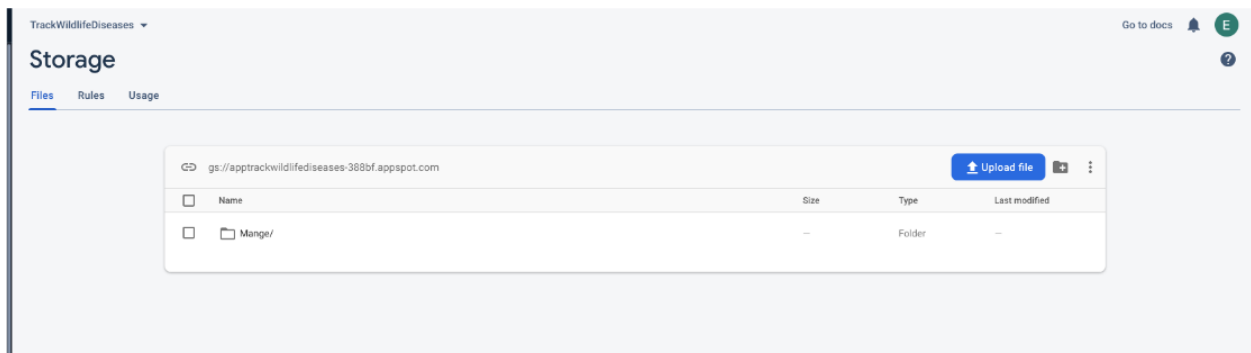


Figure 20: Firebase Storage [4]

The other one is Firestore. It is a different service from Firebase Storage, though the two have similar names. Firestore is a No-SQL cloud database. It stores the information the user uploaded along with the picture.

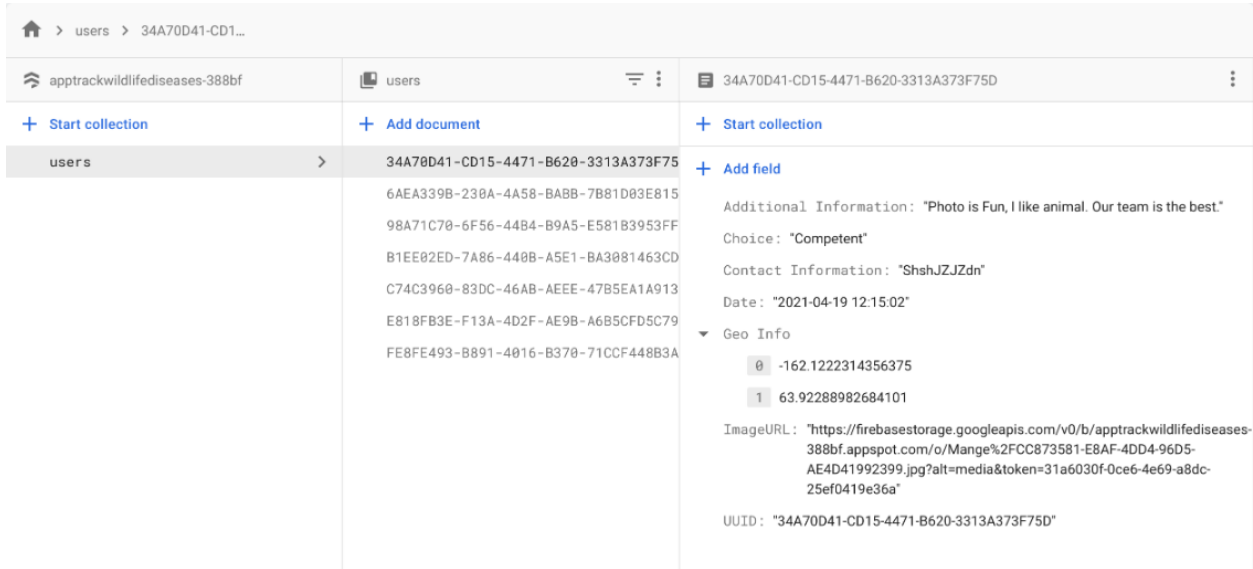


Figure 21: Structure of Firestore [4]

### 8.2.3 Mapbox

Mapbox is a powerful tool to create custom maps. It is used on our map page. To use it, we need to include the following code shown in Figure 22. For instructions about Mapbox, please visit their manual [11].

```
<meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no">
<link href="https://api.mapbox.com/mapbox-gl-js/v2.1.1/mapbox-gl.css" rel="stylesheet">
<script src="https://api.mapbox.com/mapbox-gl-js/v2.1.1/mapbox-gl.js"></script>
```

Figure 22: Code used to import Mapbox

### 8.2.4 Implementation Detail

The code consists of:

- HTML files
  - home.html
  - map.html
  - spreadsheet.html
  - app.html
  - login.html
  - logout.html
- Javascript file

- CSS file
- Multiple pictures

Each HTML file is responsible for a webpage. Every HTML file except `logout.html` always contains a logo of the Fish and Wildlife Conservation Department of Virginia Tech which also acts as a hyperlink that can jump to their website. In addition, there is a title and a navigation bar to navigate between the web pages.

```

<!-- start of vt logo -->
<div>
  <div id="vt_logo" class="vt-logo">
    <a href="https://fishwild.vt.edu" class="vt-logo-link">
      
      <span class="vt-logo-divider" focusable="false" aria-hidden="true"></span>
      <span class="vt-logo-text">Fish and Wildlife Conservation</span>
    </a>
  </div>
  <a class="title">Wild Life Tracker</a>
</div>
<!-- end of vt logo -->

<!-- start of navigation block -->
<div class="navigation">
  <ul class="navlist">
    <a class="navtab" href="/home.html">HOME</a>
    <a class="navtab" href="/map.html">MAP</a>
    <a class="navtab" href="/spreadsheet.html">SPREADSHEET</a>
    <a class="navtab" href="/app.html">APP</a>
    <script>
      login_check()
      document.write(login_tab(loginCheck))
    </script>
  </ul>
</div>
<!-- end of navigation block -->

```

Figure 23: Code for logo, title, and navigation bar

The login/logout tab in the navigation bar depends on two functions in our JavaScript code: **login\_check** and **login\_tab(bool)**. Here **login\_check** checks the user's cookie to see if the user has login information in it; if so it sets the global variable **loginCheck** to true, otherwise set to false. The **login\_tab** returns a string containing HTML code to display login or logout in the navigation tab depending on its input.

For the home page and the app page, there is no other JavaScript used.

For the map page, the functions **showMap**, **getData**, and **deleteFromMap** are used.

- **getData** connects to the Firestore and gets all data stored in it, and then stores the result in a global set **listSet**.
- **showMap** calls **getData** first, then use **Mapbox** to load the map. Inside the on(‘load’) function every dataset in **listSet** is converted to a “**feature**” object; the “**feature**” is then added to the map as “**Layer**”, which can be shown as a pin and popups to indicate the location of every picture in our database. If the user is logged in, the feature also contains a delete button. When clicked, it calls the **deleteFromMap** function.
- **deleteFromMap**: The delete first shows an alert to ask if the user really wants to remove that data. If the user confirms, it removes the **Layer**, then deletes the data from Firestore. Note that this function only removes picture references from Firestore; the picture itself is still available in Firebase Storage.

For the Spreadsheet page, **DownloadCSV**, **convertArrayOfObjectsToCSV**, **buildTable**, and **getData** are used.

- **convertArrayOfObjectsToCSV** is the helper function of **DownloadCSV**; it converts the data in **listSet** into a CSV file
- **DownloadCSV** starts the download of the file
- **buildTable** first calls **getData**, then generates a table in HTML, and then adds it to the webpage. Note that buildTable does not put every information item in the table; it only shows time, uploader, and the picture. If the user logged in as admin, an extra column with the delete button also appears, similar to the delete button in the map. It allows the removal of the row from the table and Firestore.

For the login page, the functions **Login**, **Encrypt**, and **addCookie** are used.

- The **Login** function checks if the input of the user matches the username and password stored in the code. For better security, the stored username and password are encrypted with SHA-256. The correct username for login is **admin**. You can change it by changing the global variables **stored\_username**, **stored\_password**. If the user entered the login information correctly, it adds two cookies that store the encrypted username and password, and then redirect the user to the home page. Otherwise, it informs the user that the user name or password is invalid.
- The **Encrypt** function takes a string as input and returns a string that is encrypted using SHA-256.
- The **addCookie** function takes two parameters. It adds a cookie that expires after a year that has the name of the first parameter and the value of the second parameter.

The Logout page is very simple. It contains nothing but a **logout** function. When the function is called, it sets the username and user password cookie to an empty string then redirects to the home page.

The CSS file contains every style used in our website. To have a clear view of what each line of style is for, we recommend using the inspect tool of your browser rather than directly reading the file.

## 9 Lessons Learned

Throughout the project, we have encountered a lot of issues, new coding techniques, and even overthinking way too hard on some aspects of the project. We lost a team member during the middle of the semester causing our team to re-evaluate our timeline while having to drop a requirement given to us by Dr. Escobar. Many obligations caused us to readjust our schedule and re-evaluate the feasibility of getting this project done in time and caused us to have deadlines that were creeping upon. Lastly, we were overthinking a lot of instructions while doing things that might not be necessary for the project. As a team, we encountered a lot of hardship, but we persevered and were able to give a decent website and app for Dr. Escobar to use.

We've learned many valuable experiences that could help us in the future. One major thing that we feel has been the most important lesson is learning how powerful keeping communications with one another can truly be. When we have time issues or have assignments for other classes, we contact each other to make sure that we know what we should expect ahead of time. We were able to make up for some time by making sure we communicated what needs to be done, what should have been started, and overall making sure everyone is responsible for their tasks. Another perspective is that we have strengthened our connections and perseverance during the semester. We learned that perseverance can support us to overcome obstacles, especially when we lost a team member and the tasks were becoming overwhelming. It was the spirit and our connections that motivated us to successfully accomplish our semester project.

### 9.1 Timeline

Problems/Solutions: The previous group used two tables to show individual problems and solutions. We might use that style as it shows how we tackled each problem that arose along the way.

September

**9.20:** Reviewing current app/website models

**9.30:** Have the first 20% of idea designs and new features complete

October

**10.3 - 10.9:** 40% of designs complete

**10.10 - 10.16:** 60% of designs complete

**10.17 - 10.23:** 80% of designs complete

**10.30:** App & Website complete

November

**11.11:** Begin Testing / Interim Report Complete

**11.20:** Final Project complete & creation of final presentation

**11.30:** Finish presentation & Report

## 9.2 Problem / Solution

Problem	Solution
Losing a member who would help implement the application from iOS to be used in multiple different types of devices	We discussed the situation with the client and figured that we should only focus on what we are told to implement. We can leave more advanced implementations for possible future works.
Conflicts with other classes and time overlapping	Strong communication within the group to make sure to help out with members of the group who are struggling with time. Additionally, we adjusted the schedule to accommodate those changes.
Trying to host the server and get the app onto the app store so people can download it	We were overthinking things and should focus on the task and requirements and shouldn't think too big.

Table 2: Problems and Solutions.

## 9.3 Future Work

In the future when the next team works on the project, we would like to see some changes made to the app and the website. For the website we would like to connect the corresponding data on the map to its corresponding data on the spreadsheet, so that by clicking on the data point, we can bring up the data on the spreadsheet for the user. Additionally, we would like the spreadsheet to have a variety of string options the user can use, such as sort by oldest, newest, by country, etc. For the app, we would like to see the app be published so that users can access it more conveniently. One way of publishing the app would be using the Testflight provided by Apple which is an online service for over-the-air installation and testing for IOS, but the issue is that it requires an Apple developer account which could cost a good amount of money. Aside from that, we would like to implement a helper function that detects and reports incorrectly added data. Lastly, we would like to see this app be available on all OS platforms like Android.

## 10 Acknowledgements

We would like to thank Professor Luis E. Escobar for allowing us to collaborate on the project, as well as providing us with help and ideas to make the project better. We would also like to thank Mariana Castaneda-Guzman for helping on the project as well. Lastly, we would like to thank Dr. Fox for allowing us to work together, helping along the way, and providing us with feedback for our project.

- Dr. Luis Escobar

Department of Fish and Wildlife Conservation

[escobar1@vt.edu](mailto:escobar1@vt.edu)

<https://fishwild.vt.edu/faculty/escobar.html>

- Mariana Castaneda-Guzman

Department of Computational Biology and Disease Modeling

<https://www.linkedin.com/in/mariana-castaneda-guzman>

- Dr. Edward A. Fox

Office: 2160G Torgersen Hall

Phone: (540) 231-5113

Email: [fox@vt.edu](mailto:fox@vt.edu)

Address: Dept. of CS, 1160 Torgersen Hall, Mail Code 0106, Virginia Tech,

Blacksburg, VA 24061

Laboratory: Digital Library Research Laboratory - Torg 2030, x3615.

## 11 References

- [1] Faunalytics, *Fundamentals: Zoonoses*. Faunalytics. (2021, February 10). Retrieved November 11, 2021, from [https://faunalytics.org/fundamentals-zoonoses/?gclid=CjwKCAiAm7OMBhAQEiwArvGi3BPh\\_KFWe\\_b1xft6l6Or6lET\\_b4e2Ajkj8pq7SPfbC7RyqPgBguFTBoCSV0QAvD\\_BwE](https://faunalytics.org/fundamentals-zoonoses/?gclid=CjwKCAiAm7OMBhAQEiwArvGi3BPh_KFWe_b1xft6l6Or6lET_b4e2Ajkj8pq7SPfbC7RyqPgBguFTBoCSV0QAvD_BwE)
- [2] McLean, R. G. (2007, June). *The introduction and emergence of wildlife diseases in North America*. DigitalCommons@University of Nebraska - Lincoln. Retrieved December 15, 2021, from [https://digitalcommons.unl.edu/icwdm\\_usdanwrc/710/](https://digitalcommons.unl.edu/icwdm_usdanwrc/710/)
- [3] Morey J. Haber. (2021, October 5). *Security questions pose a high risk: Learn Tips & Tricks to mitigate the threat*. BeyondTrust. Retrieved December 15, 2021, from <https://www.beyondtrust.com/blog/entry/reused-security-questions-can-pose-a-high-risk-learn-tips-tricks-to-mitigate-the-threat>
- [4] Ji, S., Lyu, J., Vu, J., & Zhang, T. (2021, May 7). *Apptackwildlifediseases*. Handle Proxy. Retrieved December 15, 2021, from <http://hdl.handle.net/10919/103257>
- [5] Apple. (2021) *Documentation*. Apple Developer Documentation. (n.d.). Retrieved November 21, 2021, from <https://developer.apple.com/documentation>
- [6] Zhang, B. (2021). *IOS GitHub page*. GitHub. Retrieved December 15, 2021, from <https://github.com/ZhangPieChuan/WildlifeTrackerApp.git>
- [7] Firebase. (2021). *Firestore documentation*. Google. Retrieved December 15, 2021, from <https://firebase.google.com/docs/>
- [8] CocoaPods. (2021). *CocoaPods.org*. CocoaPods.org. Retrieved November 21, 2021, from <https://cocoapods.org/>
- [9] *Track wild life diseases website*. Track Wild life Diseases Website. (2021). Retrieved December 15, 2021, from <https://trackwildlifeanimaladmin-58tz.web.app/>
- [10] Wang, M. (2021). *Website GitHub Page*. GitHub. Retrieved December 15, 2021, from <https://github.com/recorder233/4624website>
- [11] *Map: Mapbox GL JS*. Mapbox. (2021). Retrieved December 15, 2021, from <https://docs.mapbox.com/mapbox-gl-js/api/map/>