Requirements Specifications

All Ears

Date: 11/20/2020

Sponsors: Dr. Chris Doughty and Ms. Jenna Keany

Faculty Mentor: Mr. Tomos Prys-Jones

Team Members: Bailey Erickson, Savannah Fischer, Zhijun Hu, Elijah Macaranas, and Jared Weinberger

Version 2.2

Accepted as baseline requirements for the project:

Client: _____
Client: _____
Team Lead: _____

Table of Contents

1. Introduction

African forest elephants are a threatened subspecies of elephant that inhabit the densely wooded rainforests in west and central Africa. These elephants remove some of the younger trees freeing up more space that allows the larger trees to grow which results in the net draw-down of carbon dioxide from the atmosphere. The elephants' choice of habitat, coupled with poaching, makes counting the elephants in a traditional manner extremely difficult.

Our sponsors, Ms. Jenna Keany and Dr. Christopher Doughty, are both studying how African forest elephants affect forest structure, climate, and ecosystem function. They both track the poaching of elephants using tools like the database project MIKE[1]. They are involved in research that helps find new and better ways to estimate the populations of forest elephants. One of the biggest issues that they are facing is the lack of public awareness of African forest elephants, poaching, and the effects of each on the climate and the elephants' habitats. Poachers are getting better as technology advances, making them harder to find and counteract. Ms. Keany and Dr. Doughty work to support large conservation groups that include a focus on African forest elephants.

In an effort to assist with our sponsors' endeavors, we will create a web-based and mobile application called All Ears to educate the public. Our applications will include information about the current statistics and dangers of poaching. It will also show African forest elephants' current populations and the benefits of their continued survival. and provide donation opportunities for the public to assist anti-poaching organizations. A carbon-calculation tool will be included in order to estimate how many elephants would be required to be saved in order to offset these emissions. Which will also include the donation recommendation that will be calculated through this travel information. The desired outcome is the survival of forest elephant populations, which will lead to forest maturation and carbon sequestration. Graphical representations of data collected from MIKEs will be utilized by our application to present a more easily understandable picture of the threat posed by poaching. A stretch goal of our application will also be to integrate a previous application called Animaps that will be used for displaying fauna data based on geographical regions of a map when selected. The application will contain links to pre-existing conservation organizations, where the donations can be made.

In this document we will explore the problem statement in more detail. As well as explore how our application is addressing these issues. Then we will look at the functional, performance, and environmental requirements in a logical and detailed fashion, followed by the potential risks we face. Finally we will walk through the expected timeline of our project. By signing this document we agree to the requirements that will be included in the finished product by April 2021.

2. Problem Statement

Dr. Chris Doughty and Ms. Jenna Keany are both researchers working at the Megabiota lab. Dr. Chris Doughty is an assistant research professor in ecoinformatics at Northern Arizona University and Jenna Keany is a PhD student working in the lab as well. Their work involves using LiDAR[2] data to evaluate how forest elephants impact forest structure and how ECOSTRESS[3] data can predict temperature stressors in the tropics. The goal of their research is to determine trends in elephant populations, poaching, and carbon emissions. Their intention with this project is to support elephant conservation groups. They have collected all of this important information and they wish to share it but do not have the platform to do so effectively. Ultimately what their business lacks is a customized way of displaying their data and information to help educate the public about their work in conservation and the role elephants play in drawing down carbon emissions.

Our sponsors recognize an unmet need for an educational tool aide in teaching and informing the public on their area of studies of African forest elephants. There is currently nothing on the market that is meeting their requirements. One of the most prevalent issues they have encountered is awareness; awareness of the specific species of elephant, the impact that they have on the environment, and the impact that poaching has on the elephant populations and the atmosphere.

3. Solution Vision

In this section, we will look more closely at the capabilities that our application will need to have as well as the solution that we have created. We have compiled a list of all of the basic functions that our applications will include. Along with the detail of the architecture system that we will implement.

Dr. Doughty and Ms. Keany are looking for a publicly available and uncomplicated application for both the web and a cellular device.

- Educating people about African forest elephants
- Displaying accurate elephant population estimations based off map data
- Showing the number of poaching incidents and the approximate locations of the incidents
- Teaching people about how forest elephants allow forests to mature and thus decrease the amount of carbon in the atmosphere
- Donation calculator based on an equation provided to us by the clients that considers the distance traveled and the mode of transport to estimate a donation amount.
- Page of links to pre-existing elephant conservation charities for people who would like to make a donation (This feature will be accessible without the carbon calculator)
- Administrative login for updating the MIKEs database information on poaching incidents

Our main priority for our application will be to educate the public on forest elephants and their effect on the atmospheric carbon. However, encouraging our users to donate to endorsed anti-poaching organizations is our secondary priority as it helps in the effort to halt the illegal poaching of elephants in the African rainforests.

We have also analyzed the projected cost of the project and we believe the application's financial cost will be relatively low. For our application, data storage will be the largest cost and it should not be much at all. The data storage is currently estimated to be approximately five dollars a month with Digital Ocean’s Droplet.

According to the above requirement and existing requirements of the client, we decided to use the frontend and backend separation architecture. It is a kind of structure that separates the frontend system and backend system and uses a tool to achieve the

purpose of frontend, backend interaction. In case to avoid excessive consumption of resources and improve developer efficiency.

3.1 Frontend System

The frontend is the part of the application that the user directly interacts with, which takes responsibility for the view layer and controller layer. Using Flutter is an optimal choice in order to meet the clients requirements of cross platform capabilities. Our frontend system will contain functions such as; the informational poaching incident map, a carbon calculator, and the endorsed donation site links. The communications from the backend to the frontend will provide the applications with the following functions:

- The educational page allows user to interact and view
- Interactive map for the distance calculator
- Carbon Calculator is based on the formula that the client provides.
- Donation Links for anyone
- Admin Login and change password

3.2 Backend System

The backend is the part that users of the application cannot see, which takes responsibility for the management system layer and database layer. The backend should include a server, models, management system, and database. The server will be constructed on the Flask framework that relies on the Python language. The model contains the functions that would be supported for the frontend system and will hold on the backend, such as editing operation. A management system is a system that allows content creators, editors, and publishers to submit, modify, approve, and publish content. The database is the place that holds all the data that collects from external resources. backend developers build and maintain these components to provide multiple frontend supports. The backend server will receive requests from the frontend and find the program responsible for processing the request or get the data from the database to the backend. Finally, it returns the appropriate result to the frontend application. From the backend, it can provide the specific functions below:

- Frontend and backend interaction.
- The database saved all the map tool required data and then displayed it to the frontend.
- Admin editing data.
- Manipulate data.

3.3 Frontend and Backend Data Conversions

As mentioned above, a frontend system sends a request to the backend system, and the backend system will grab data from the database then send it back to the frontend system as displayed in Figure 1.1. However, the problem is the backend system needs to transmit a variety of data formats to adapt to the frontend system, so it needs to use the JSON tool to convert and use it to represent various complex data, such as objects, arrays, collections.

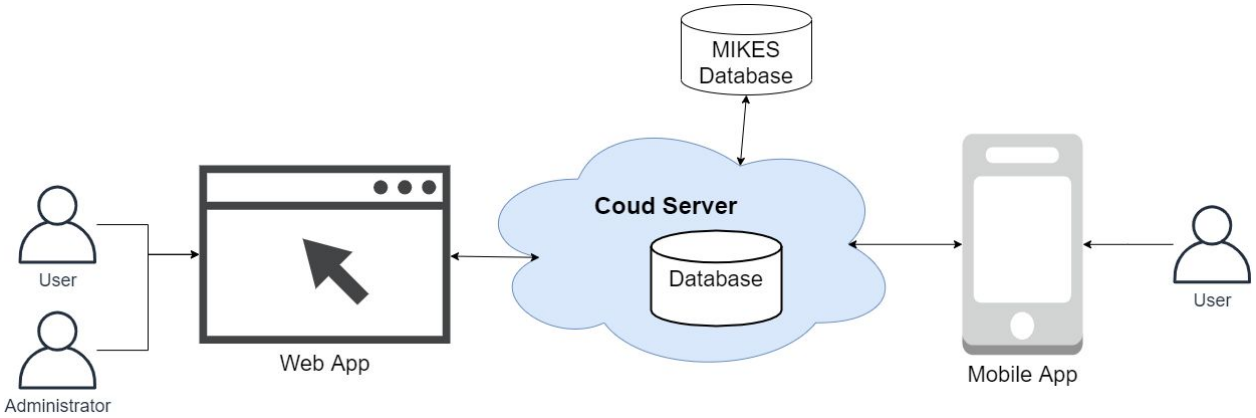

Figure 1.1 System Diagram displaying how the system will communicate using the backend-frontend interaction

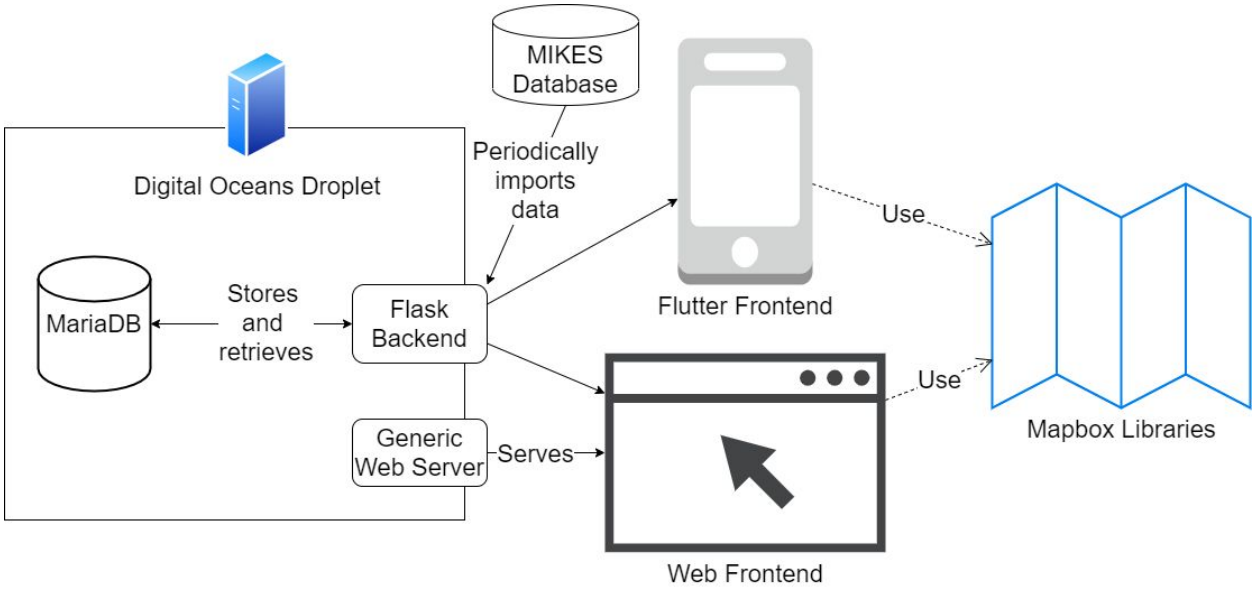

Figure 1.2 System Architecture Diagram showing how the entire system will be structured using all of the components of our system.

4. Project Requirements

In order to bring our product to fruition we must first detail the functional, performance, and environmental requirements. Our functional requirements, which will be for both our mobile and our web applications unless stated otherwise, will include:

1. **Integrated and Interactive Mapping System**: The user will need to be able to view a map with estimated elephant populations as well as map a trip for the carbon calculator.
2. **Built-in Carbon Calculator Tool**: This will be integrated into its own section of the mapping tool which will help calculate donations.
3. **Suggested Donation Links**: We will provide links to pre-existing elephant conservation organizations, allowing the user to easily make a donation.
4. **Administrative Login from the Web**: Administrative login to change and update the data from the MIKEs database.
5. **Backend Database Management**: Will provide up-to-date (within a year) information for our systems to run on.

In order for our system to run on devices and have passable performance and response time, our performance requirements for our application will include:

1. **Data Accuracy**: Ensure the application is supplying the user with the most accurate information.
2. **Timing Characteristics**: Ensure the user gets a reasonable response time.

The environmental requirements, constraints that All Ears must work around concerning client requested and technological necessities are as follows:

1. **Cross Platform Compatibility**: Working on Android and iOS platforms, including iOS iPad as a stretch goal.
2. **Data from MIKEs Database**: Must use data from MIKEs database by downloading the CSV datafile and uploading through the administrative backend.
3. **Easy Access to Donation Links**: Must have easily accessible donation links for the user to donate to a list of pre-chosen conservation groups.
4. **MIKES Data Collection and Information Display**: Must be able to download MIKES from an Administrator while developing the program for testing purposes

The next section will provide more in depth details on and explore the functional requirements of our application.

4.1 Functional Requirements

The functional requirements of our application are the requirements that our clients have expressed the most interest in making sure that we include. We start with the more broad topics and then narrow them down to specifics within each of the sections. First, we will look at how we will display the poaching data.

4.1.1 Display Elephant Poaching Data -- Interactive Map

Our application must allow users to view the general regions where African forest elephant herds exist - exact locations will not be provided, as this could prove counterproductive to our goal of working against poaching. We will be using Mapbox to create an interactive map that will allow users to select the various regions to display the locality elephant information. The map also will display a graph that represents the poaching incidents according to a timeline of approximately twenty years for the specific region that the user is currently viewing, so long as the data is available.

4.1.1.1 Selectable Countries in Map

One of the main features that our Interactive map must include is selectable countries. By default, the map will display all countries located in Central Africa, the primary location where African forest elephants can be found. Users will be allowed to tap each country. After a user selects a country, the map will zoom in on that region and display each region's elephant population information and poaching statistics.

4.1.1.2 Poaching Incidents Graph

Once the user selects a country, the map will zoom in to the specific country and will bring up a small window that contains a bar graph of poaching incidents based on the year. The graph will show the number of poaching incidents per ‘bar’ when the user presses on a selected ‘bar’ in the graph, and the trend of poaching during recent years. When the user taps another place without closing the popup window, the popup window will auto shrink until the user re-clicks it.

Use Case: User clicks on the popup window to view the poaching incidents graph

- **Actor:** user and administrators.
- **Description:** create a graph of poaching incidents that display it on the window.
- **Precondition:** the map will zoom to the country that the user selected. The elephant's population data will be displayed on the same layer of the map.
- **Postcondition:** A popup window appears with the content of a graph of poaching incidents during recent years.

Main flow:

1. The user selects one country on the map
2. The system receives data from the back-end and displays general information of locality Elephant
3. The user clicks on one region of the country
4. The system brings up a popup window that displays the graph of poaching incidences.

4.1.2 Display Environmental Impact -- Carbon Calculator Tool

In order to aid in the decision making of donation amounts our application will have an integrated carbon calculating that can suggest a donation amount.

4.1.2.1 Calculate Flight Distance

The user will select two or more cities with airports, from starting to finish from a drop-down list that lists them alphabetically, which will result in an estimated displacement distance using an airport distance API. The user will also be able to select countries with airport(s) on the map. To add more airports for intermediate flights, the user can add up to 5 flights together. Next, using MapBox, the system will display the route of the flights to the map, then the API will use the chosen airports to estimate distance traveled by plane. Subsequently, we implement the formula provided by our clients to calculate carbon (CO) footprint by miles traveled by plane.

4.1.2.2 Offset CO footprint from flight information

The application will implement a formula from the clients to calculate elephant carbon sequestration so that it can determine how many elephants will counteract the carbon footprint that results from the user's inputted travel information. And from that, the application will estimate how much money should be donated to an anti-poaching organization so as to fund the efforts necessary to save the approximate amount of elephants calculated.

Use Case: User inputs flight information to use carbon calculator tool

- **Actor:** user
- **Description:** A tool that prompts the user to input flight information from two or more cities with airport(s) which calculates the offset CO footprint which results in the estimated money needed to help the elephants that are counteracting the carbon footprint.

- **Precondition:** The user goes to the CO calculator page
- **Postcondition:** The system successfully displays the amount of money needed to help the elephants

Main Flow:

1. The user selects a country or a city through the map or otherwise, through the dropdown menu
2. The user selects a "**+**" icon to add the city to the calculation
3. The user selects a calculate button
4. The system displays flight route on the map
5. The system calculates how much carbon is produced by the flight and how much elephants are needed to counteract the carbon produced and the money needed to help said elephants
6. The system displays all information on number 5 as a popup with a donation prompt

4.1.3 Suggested Donation Links

Our application will also provide the links for the user to go and donate to either from the home screen or from the carbon calculating tool. Since our application will not be handling the user's personal banking information we will link them directly to the donation pages.

4.1.3.1 Donation Prompt

For users to donate before or after calculating their CO footprint offset, they can go on the page listing the anti-poaching organizations, then users will be presented with a donation form with or without their calculation from the CO calculator for a voluntary opportunity to visit the individual donation portals of the organizations listed below.

4.1.3.2 Organizations

- WildAid (Scott G's pick)
- Wildlife Conservation Society
- Tusk - Advancing Wildlife Conservation Across Africa
- ESPA - https://www.speciesprotection.com

Use case: User donates to certain organizations using donation links

- **Actor:** user

- **Description:** Display a donation prompt to the user which will redirect them to the donation page of organization they select
- **Precondition:** Users must be willing to donate to an organization and/or have the carbon offset calculation from the Carbon Calculator tool
- **Postcondition:** Users are redirected to the organization's donation page of their choice

Main Flow:

1. The user selects a donation button or icon from the CO calculator tool or any page with the donation button
2. The application displays a donation prompt with range of prices to select from or with the calculation from the CO calculator, with the list of organizations
3. The user selects an organization from the list
4. The application redirects the user to the donation page of the organization showing the amount selected or the amount calculated from the CO calculator tool

4.1.4 Provide Current and Accurate Data -- Administrator Data Curation

In order to keep the data for this application up-to-date and accurate, we will need a trusted administrator to curate our datasets. This includes adding new data from the MIKES database, which has no public API, and removing incorrect data. This will only be implemented in the web application.

4.1.4.1 Access Restriction

In order to access the part of the website where one can change the stored data in any way, one must first enter a master password. After one has entered the password, one can access the administrative functions mentioned in the following sections.

4.1.4.2 Data Upload

An authenticated administrator can upload a CSV file with the same format as one from the MIKE database available. This file will then be parsed and added to the application database. Duplicate records will be automatically removed, so one can either upload all of the data from the MIKE database with the new entries included or one can upload only the new entries, which one must filter for oneself. If the file is not in the format mentioned above or is not a CSV, it will be rejected.

4.1.4.2.1 CSV Upload

One can upload a CSV file from the MIKE database and have its records entered into the application database. Any uploaded records which have the same unique ID (the MIKE site ID and the year) as a record in the database will replace those records in the database.

4.1.4.2.2 File Rejection

Any file that is either not a CSV file does not have the same structure as the CSV available from the MIKE database available on the 5 November 2020 from the URL: 190307_PikeStatsUpTo2018FusionTableFormat.csv will be rejected.

4.1.4.3 Data Revision

An authenticated administrator can view the records in the application database and make changes as needed.

4.1.4.3.1 Manual Record Entry

The administrator can add single records to the application database.

4.1.4.3.2 Manual Record Correction

The administrator can correct fields in single records in the application database/

4.1.4.3.3 Manual Record Deletion

The administrator can remove single records from the application database.

4.1.5 Provide Current and Accurate Data -- Application Database

The database must be altered from its default state in order to read and handle CSV files, as well as to function most effectively in the use scenarios implicit with creating a web and mobile application - ideally expecting traffic and application component usage from various devices simultaneously throughout the world.

- Use CONNECT[4] or a similarly functioning database engine[5] to assist with the reading of CSV files and the writing of their information into the proper database table(s)
- Organize information from CSV file uploaded from associated web page into organized database table specific to MIKES database information
- Database properties:
    - Utilization of a database firewall, as well as a master password or SSH key to secure from unauthorized access and alteration.
    - Minimization of unnecessary redundancy
    - Configure database for optimal performance

- Modify to allow for greater number of connections over greater number of machines
- Facilitate better handling of fast connections/disconnects by increasing related configuration information.
- Ensure simultaneously running fast queries are better managed by altering the associated settings.

4.2 Performance Requirements

Accuracy and speed will be the two qualities by which our application will be measured apart from basic functionality. The quality of performance will also determine the usefulness of the app. A calculator program is useless if it takes five minutes to compute 1 + 1 on a modern computer. One could say that it has that functionality, but because of the performance costs that functionality is negligible.

4.2.1 Data Accuracy

As stated above, the data accuracy of our project will be determined by those for whom we are creating this application who are experts in the field. The accuracy of the data presented will depend on their curation of the application's database and the MIKES database primarily from which our application will get its data. We, as the developers of this application can only ensure that the data served from this application will be the same data that the administrator manually enters into the database or that an administrator uploads in the form of a CSV file. There will be no discrepancies between an uploaded CSV file and the data in the application's database. Our team will test this by selecting a MIKE site at random and making sure that the numbers displayed in the application are the same numbers that are in the uploaded CSV.

4.2.2 Timing Characteristics

The response time of our application will depend largely on the quality of the internet connection between the backend mobile whichever frontend, web or mobile, is being used. Therefore for all performance stipulations, we will require that the internet connection throughput for all parties be at least 100 Mb/s for download and 10 Mb/s for upload.

Additionally, the database will not be directly considered in this section, as there will be no direct user interaction with it. Its performance is integral to the rest of the application because the performance of any part of the application involving poaching data will almost directly depend on it. Therefore, any requirements placed on the parts of the application that depend on the database will effectively constrain the database as well.

4.2.2.1 Display Elephant Poaching Data

It will take new users 10 seconds or less to access the fully-loaded interactive poaching map and 3 seconds or less to access the poaching graph of a country or MIKES site after that. Experienced users should reduce those times to 3 seconds or less and 1 second or less, respectively.

4.2.2.2 Carbon Calculator Tool

The carbon calculator tool has two components: calculating flight distance and the carbon footprint of that flight. A new user will take 40 seconds or less to calculate their flight distance and calculate the carbon footprint of that flight. An experienced user will reduce that time to 15 seconds or less.

4.2.2.3 Suggested Donation Links

A new user will be able to find the links suggested for donation within 10 seconds after opening the application. An experienced user will be able to find the links within 5 seconds or less after opening the application.

4.2.2.4 Administrative Features -- Login

A new administrator will be able to login and gain access to the restricted administrative section of the application within 20 seconds. An experienced administrator will be able to do the same within 10 seconds.

4.2.2.5 Administrative Features -- Data Upload

A new authenticated administrator will be able to upload a CSV from MIKES database within 15 seconds or less. An experienced administrator will reduce that time to 10 seconds or less.

4.2.2.6 Administrative Features -- Data Revision

A new authenticated administrator will be able to add a new record within 40 seconds or less. An experienced administrator will reduce that time to 20 seconds or less. Deletion for both administrators will only take 10 seconds or less. Changing a record will take 40 seconds or less for a new administrator and 20 seconds or less for an experienced one.

4.3 Environmental Requirements

In this section, the constraints of the project are discussed, required either directly by the clients or by the necessary technology for our application’s development.

4.3.1 Mobile Platform Compatibility

Flutter is a great frontend mobile application framework to use in our application with numerous capabilities that benefits our application. However, it still comes with a few setbacks that needs to be looked at.Our clients imposed us that if possible, we should build a mobile application for every iOS and Android devices but Flutter can only go so far back in the iOS and Android versions.Having discussed it with our clients, we came to the conclusion that we build the applications for both operating systems for devices that are at least 5 years old or newer. For iOS, we have iOS 9.0 which was released September 16, 2015 and for Android, we have SDK 23 which is Android 6.0; also released in 2015 a month after iOS 9.0's release.

4.3.2 Data from MIKEs Database

Our team discussed with our clients when we showed them a basic prototype and see what they specifically wanted to be included in our application and one such requirement is that based on the MIKES database, we look into each region in the dataset and specify which conservation group is in each region. The problem is the dataset lacks that information and at the moment, we might have to hard code the locations of each organization to match the dataset the Administrator will provide.

4.3.3 Easy, Consistent Access to Donation Links

The purpose of All Ears' application is to educate the general public and allow them the opportunity to get involved with anti-poaching efforts through donation recommendations and access to donation links to prominent anti-poaching organizations. If the donation links cannot be easily found or are not obvious in their inclusion throughout the application, the application will not meet the whole of its goal. Donation links will be available on every page of the application, and will also have a dedicated page for links only, as well as general descriptions of the anti-poaching organization that they are for.

4.3.4 MIKES Data Collection and Information Display

All Ears will be utilizing data collected from the MIKES database in order to educate the public about regional elephant poaching statistics. To do this, the developing application must be capable of downloading the CSV file containing poaching information MIKES provides and integrating it into its personal database. The data should then be used to display the information in an easily navigable, easily readable format to the application's users.

5. Potential Risks

There are usually several potential risks involved in software development that are unavoidable and need to be addressed whenever one occurs. Here, we identify some of those. First we will consider the likelihood of each of them actually occurring, ranging from low to high, their severity, also from low to high, its consequences, and our team's solution in a case where one of these risks happen to occur after the application deployment.

Risk	Likelihood	Severity	Solution
An administrator forgets their user credentials	**Low**: People often forget their credentials, and the master credentials are shared among different researchers	**Medium**: When an administrator cannot access the data upload section of our app, it can mean that the application is providing outdated information to users	The administrator emails an All Ears team-member in order to have their password reset
A non-administrator user has access to the site/application's administrative privileges	**Low**: Only our team members and our clients know the master credentials	**High**: The non-administrator can upload an incorrect file type or provide a wrong dataset for the applications, spreading misinformation	The system will send an alert to the actual Admin notifying them of any non-admin access to the website

A user is redirected to a wrong website	**Medium**: The organization updates their website or we get the URL wrong	**High**: May lead the user's donation to be redirected to a wrong website	Contact our clients for updated URLs for the organization(s) and double check that we input the URL right in the source code. The user will be able to flag or report this to the admin
Incorrect Implementation of the MIKE's Database	**Low**: The organization may change the file type or provide a wrong dataset	**Medium**: This can lead to incorrect calculations of the recommended donation amount	We will restrict the upload of a non-CSV file by maintaining the current dataset and we can only trust the admin will upload an accurate dataset as it is from a 3rd-party organization
Incorrect Calculation of a Donation Recommendation	**Low**: Our team/clients get the calculation or formula wrong	**Low**: The wrong recommended donation amount will be shown, but users have the choice to donate a separate amount to the recommendation	Make sure that the team understands how the formula works and double check with the client if our calculation works. The user can flag the calculation as off through a button as to notify the admin of a potential incorrect calculation

As seen from the table above, every potential risk we have can come from one security risk, when an Administrator forgets their credentials to access the website. One can also say our application relies heavily on the dataset the administrator provides. Without it, it will render our application's purpose as useless because of our main purpose —to help the elephants.

Another potential risk that might affect our system indefinitely is having an incorrect calculation of the donation from the CO offset calculator tool. This could stem from

several factors such as getting the flight distance data wrong to plainly just getting the formula wrong.

Finally, the risk of not being able to meet our clients' requirements because they are essentially the core of our entire project, providing us with the vision and mission to educate the commonplace about the carbon elephants in the rainforest can offset. These risks can be avoided by regularly checking with our clients regularly and seeing if we're on the right track with their requirements.

Now that we have listed the potential risks we might bear in the future, we now outline our timeline over the coming months.

6. Project Plan

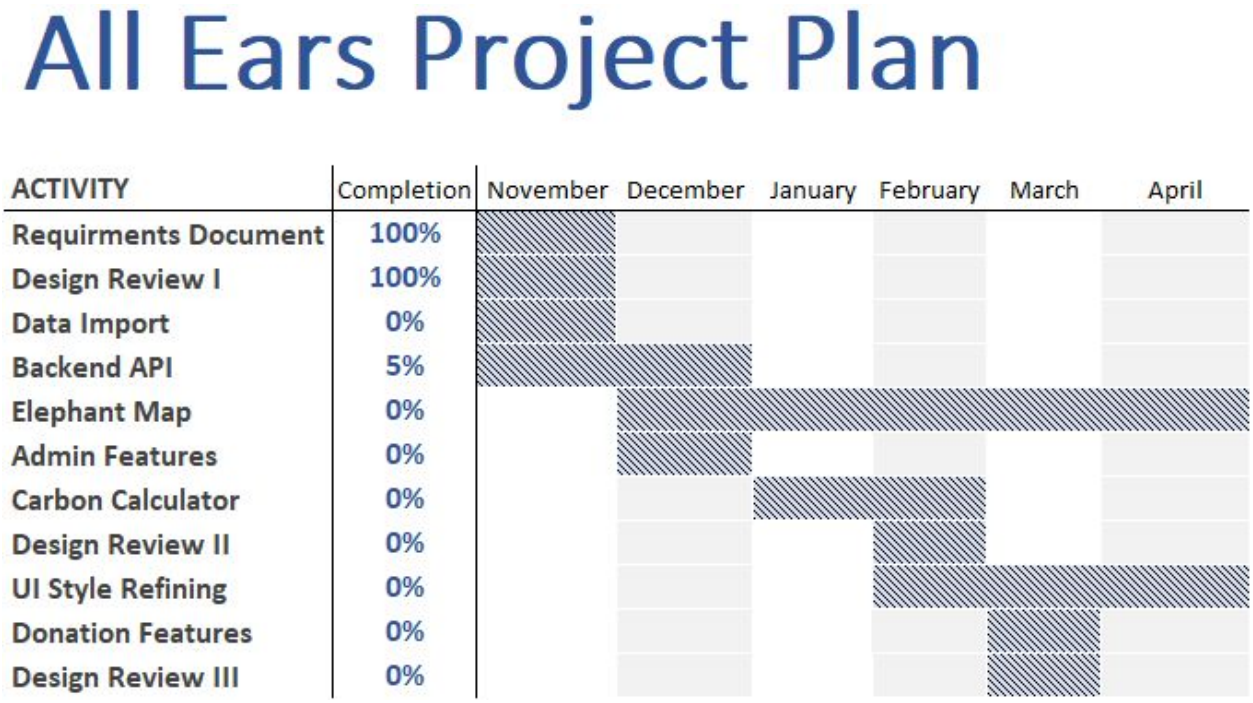

Figure 6.1: Gannt Chart showing All Ears' project plan for the following 6 months.

We have ten major tasks in the development of our application listed in the Gannt chart in Figure 6.1. We have incorporated enough time for bugs, errors, and revision in our planning time to ensure that we will be outputting a high quality product. Here are the tasks:

1. Requirements Document - This task represents this document.
2. Design Review I - We have already completed Design Review I.
3. Data Import - We will begin working on the data import functionality during the week of November 9 and will finish it the next week.
4. Backend API - We have already started developing our backend API and will have it finished by the end of December.
5. Elephant Map - We will implement all of the features of the elephant map over the next 19 weeks starting the week of November 9 on mobile devices.
6. Admin Features - We will add in the rest of the admin features and restrict general user access on the week of November 23. The task will only take two weeks.

7. Carbon Calculator - Development will start on the carbon calculator for mobile platforms will start in the first week of February and will last four weeks.
8. Design Review II - We have allotted ourselves three weeks for the second design review.
9. UI Style Refining - Starting in the middle of March and lasting for the next 7 weeks will be a team effort on creating and refining the user interface for our application after the application is mostly functional.
10. Donation Features - We will implement donation features on all devices starting the second week of March and lasting for four weeks.
11. Design Review III - We have allotted ourselves three weeks for the third design review.

7. Conclusion

The continued survival of African Forest Elephants is imperative for African rainforests to thrive, and thus for their continued contribution to the steady removal of carbon emissions from the air. The true scale of African Forest Elephant poaching is often unknown by members of local and world-wide populations. Educating the public about the positive environmental impact of African Forest Elephants, the threats they continue to face, and ways of contributing to their protection through anti-poaching organizations is important to protecting the species.

In order to create a functional website capable of educating the general public about the benefits of and threats against African Forest Elephants, our application is required to

- Educate application visitors about general African Forest Elephant information and about how forest elephants help decrease the amount of carbon emissions in the atmosphere
- Display accurate elephant population estimations based off of map data
- Show the number of poaching incidents and their locations
- Host a donation amount calculator based on an equation, provided to us by the clients, that takes into account distance traveled by plane to come up with an estimated donation amount.
- Institute a donation page for people who want to donate to pre-existing conservation groups in Africa.
- Utilize an administration login to be used for updating the application database using MIKEs database information on poaching

To implement these necessities, All Ears will build upon frontend and backend separation architecture. The front end will display the general education information to the public, host the Carbon Calculator tool to help users understand the benefit of increased forest elephant populations, provide the ability for data alteration and file uploads to be performed by those with the proper credentials, and direct users to anti-poaching donation opportunities and organizations. The back end will store data relevant to mapping API, such as elephant populations and the number of poaching incidents and elephant deaths recorded from the MIKES database, both by region.

By taking these anticipated components, frontend and backend, and reducing them into smaller essential tasks and expectations, All Ears has created the foundational action-plan for designing, constructing, and realizing the African Forest Elephants' educational, multi-platform application.

Apart from frontend and backend expectations, our final application will be made to work with Android and iOS devices with software from the last five years at the latest. A database engine will be used in conjunction with MariaDB in order to expedite the reading and writing process between a relational database and CSV files. The interactive map used to focus on elephant populations by region must be capable of not only zooming in on the user-desired location(s) when directed, but must also contain region-specific information about elephant populations and poaching, accessible when selecting those regions. These are key topics of our document, illustrating the more pointed nature of the performed evaluations. Concerning current progress, the backend API is currently under construction and should be finished by the end of December, which will facilitate connections from backend to frontend and data exchange both ways as necessary.

All Ears is confident that the foundational design for their mobile and web application is complete and the team is ready to begin development of an African Forest Elephant and carbon-emission educational application supporting anti-poaching organizations by encouraging the public to donate.

8. Glossary

---

[1]MIKE: Monitoring the Illegal Killing of Elephants is a database containing the regions involved in specific poaching incidents, as well as those incidents that involve the death of elephants

[2]LiDAR: Light Detection and Ranging is a method of remote sensing that uses a laser that pulses to measure ranges to the Earth's surface.

[3]ECOSTRESS: ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station is a way of accurately measuring plant temperature and creates amazing heat map images of the earth's surface based on plant temperature.

[4]CONNECT: a Database engine that focuses on reading from CSV files

[5] Database engine: also known as a Storage Engine, a Database Engine is an underlying software component utilized by database management systems to read, create, update, and delete data from the database