# Software Design Document

## All Ears

**Date:** 2/5/2021

**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

# 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 abundant younger trees that clutter their forest habitats, freeing up more space in the process and allowing larger trees to continue to grow and the surviving younger trees to mature. This results in a greater net reduction of carbon dioxide from the atmosphere than would occur without their influence. African forest elephants are important to the health of the forests they live in, but the elephants' choice of habitat and the prominence of poaching makes keeping track of the elephants’ population sizes in a traditional manner extremely difficult.

Our sponsors, Dr. Christopher Doughty and Ms. Jenna Keany, 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 Monitoring the Illegal Killing of Elephants database or MIKE and 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 the importance of African forest elephants, African forest elephant poaching, and the effects of the species and threats to the species on the climate and the elephants’ habitats. Dr. Doughty and Ms. Keany work to support large conservation groups that focus on African forest elephants.

In an effort to assist with our sponsors’ endeavors, our team is in the process of creating a web-based and mobile application called All Ears with the primary goal of educating the public. Our applications will include information about the current statistics and dangers of poaching, 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 so members of the public can calculate their air travel emissions of $CO_2$ over user-defined distances. The calculator will then suggest an amount to donate to conservation charities in order to save enough elephant years to offset these emissions. The user will be directed to a donation page that will contain links to pre-existing conservation organizations, where the donations can be made.The desired outcome of this application is that it function as a tool to assist in the the continued survival of forest elephant populations, which will lead to forest maturation and carbon sequestration. Graphical representations of data collected from MIKE will be utilized by our application to present a more easily understandable picture of the threat posed by poaching. A stretch goal of our product will be to integrate an application called Animaps to display fauna data based on geographical regions of a map when selected.

There are specific requirements that our applications must meet to be considered successful. For key user requirements, we must have an integrated and interactive mapping system, a built-in carbon calculator tool, suggested donation links, an administrative login, and a backend database management system. Our primary performance requirements are ensuring data that the user sees is accurately imported from the MIKE database and that the applications respond to the user within a 7 second timeframe. Finally, our environmental constraints include a cross platform compatible mobile application, easy to access donation links, and the ability to collect and display information from the MIKE database accurately.

Further on in this document, we will detail the overview for implementation and examine software decisions, specifically how they will be implemented in each component of the applications, and the planned architecture of our final product. We will then discuss the modules and the interfaces that will be used in the development in both the applications, as well as the plan for implementing all of the modules and the timeline we have developed for completing them.

# 2 Implementation Overview

In this section, we will go over the necessary components of our application that we must pursue in order to get our clients their goal product. We will look at our overall goals of the project and discuss the overview of how we will implement each component. Our final product will be an application accessible both on the web and with mobile devices with the goal of:

- Educating people about African forest elephants through interactive tools
- Displaying accurate elephant population estimations based off of map data
- Showing the number of poaching incidents and general locations of the incidents, so that they are not traceable to specific locations
- Teaching people about how forest elephants allow forests to mature, thus decreasing the amount of carbon in the atmosphere
- Supplying a carbon calculator to estimate $CO_2$ emissions based on air travel between two specific locations
- Utilizing the same carbon calculator to recommend a donation amount to an anti-poaching organization that will ideally aid in the preservation of enough elephant years to offset the emitted carbon from the distance travelled
- Providing a 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, and the amount will be flexible and not pre-specified by the donation calculator)
- Allowing administrative login for updating the MIKE database information on poaching incidents

Our team will be building upon the frontend and backend separation architecture. The front end will display the general education information to the public, host the carbon calculator tool, provide options for administrators to update MIKE data and file uploads, and direct users to anti-poaching donation links.

For the mobile version specifically, we will be using Flutter. Flutter is a UI software development framework that our team is utilizing in order to build the front-end UI components of our application. Flutter_maps is a package for Flutter that will also allow us to configure a map on a specified section or page of our application. For the web-based portion of our application, Vue.js, a JavaScript framework for UI development, will be used to construct the necessary UI components.

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 MIKE database, both by region.

Flask is a micro web framework that provides structure for our application. This is important specifically for the development and utilization of our application’s backend, such as database management and connectivity.

# 3 Architectural Overview

Together with Dr. Doughty and Ms. Keany, we developed an overall design for our final product, which will include a mobile and web application for displaying the negative impacts of poaching and give users an opportunity to offset their carbon emissions by donating to elephant conservation organizations.

## 3.1 System Components

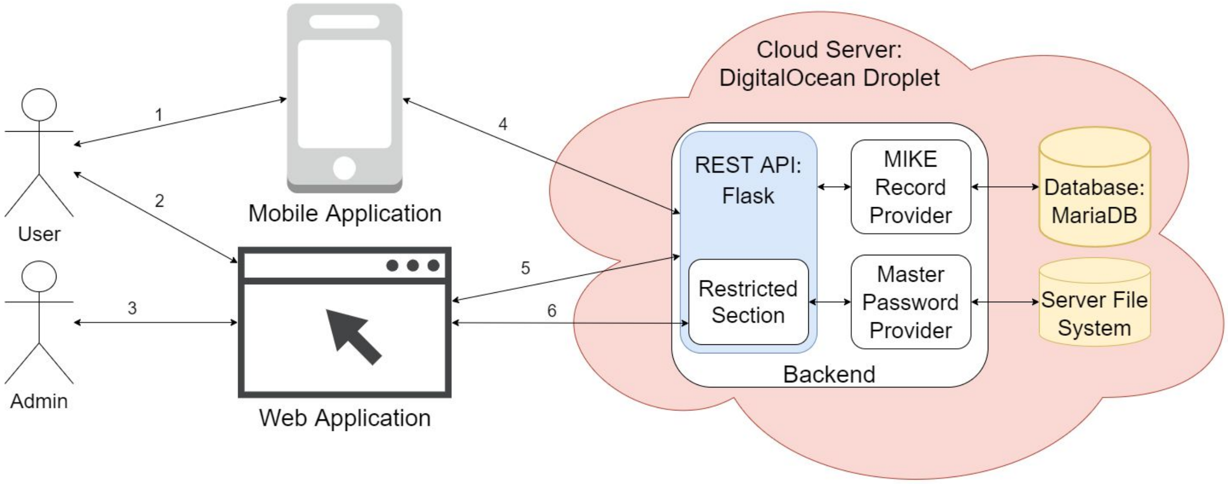


**Figure 3.1**
A high-level architectural diagram of our application (described further below)

As displayed in Figure 3.1, our final product will be made up of three components: a web application, a mobile application, and a backend to support both. The web and mobile applications will interact directly with the users as shown in interactions 1 and 2 (the double-sided arrows) and compellingly display the information that is critical to our product. The backend will provide the necessary data for the mobile and web applications to display as seen in interactions 4 and 5.

The mobile application will provide its users with information about African Forest Elephants and how they are integral to the ecosystem of the region, an interactive map showing the countries where they are being poached, and graphs for each country showing elephant poaching numbers over time. It will also provide suggestions for users to donate to various charities that combat forest elephant poaching based on the users' carbon emissions from their air travel.

The web application will have all of the functionality of the mobile application with an additional function: it will contain all of the administrative features of our application. Administrators will use the password-protected admin portion of the web application to curate the elephant poaching data and to make sure the data stays current. They can login, manually correct errors in the dataset, and add incidents that may not be currently included in the data provided by the MIKE database, an elephant poaching database which will serve as our application's primary data source. An administrator can also use the admin portal in the web application to trigger an automated update of the application database, where the dataset provided by the MIKE database is uploaded into the local database utilized by our application.

The backend, as mentioned above, will support the mobile and web applications by providing the necessary data for the two to display poaching statistics and other relevant information. It will also provide a major part of the security for the admin section of the web application, validating or rejecting login attempts in order to restrict who is allowed to add and modify database entries. In order to change anything in our product's database, the web application will need to request that the backend make changes and update the database.

## 3.2 Component Communication

The mobile and web applications communicate with the backend via a REST API as seen in interactions 4, 5, and 6 in Figure 3.1. A REST API is a special way that applications with a similar structure to ours communicate using the Web. Clients request the server to send out information or perform an action. The greatest characteristic of this strategy is that each request from a client (our web and mobile apps) contains everything necessary for the server (our backend) to interpret it, meaning that the server does not need to keep track of its clients.

When either the web or mobile apps display the poaching map and subsequent graphs, they need to send requests to the backend to acquire the relevant data (interactions 4 and 5). When a user tries to gain access to administrative features (interaction 6) by logging in, the web application will need to send the password to the backend to have it checked. Every time the web app tries to do any administrative action, it will need to send a token which will need to be validated before the backend will fulfill its request. When poaching data is to be updated, added, or deleted, the web application will need to send a request to the backend to update the application database.

# 4 Module and Interface Descriptions

In our web application and mobile application combination, we have three major components, each of which will be discussed further on in the document, that all have their own specific functionalities. Our first major component is our user interfaces, which will be very similar to each other and display all of the same information to users on both the web and users of the mobile application. We also have a mapping section which is what will be the most interactive part of our product. Finally, we have the back-end database system, in which only users with administrative privileges will be able to log on using a web portal with a corresponding password to edit and add data to the database that feeds necessary information to both the web application and mobile application.

## 4.1 User Interface

Both the mobile and web applications will be for the target groups of elephant conservation organizations, people that care about African Forest Elephants, and researchers. In order to make an educational application to cater to these groups we need to keep a simple and user-friendly design. For simplicity of design, we opted for a simple home screen that contains a navigational menu for the rest of the application. For our mobile application, We will have four main pages (Figure 4.1.1): an about page, a carbon emissions calculator page, a donations link page, and an educational interactive elephant map page. For the web application however, we will have an extra page exclusively for administrative use (Figure 4.1.2). All of the links shown in the homescreen of the application will also be accessible through a drawer menu at the top left corner of the page which can be clicked and expanded for easy navigation of the application. The web application will share a similar feature.

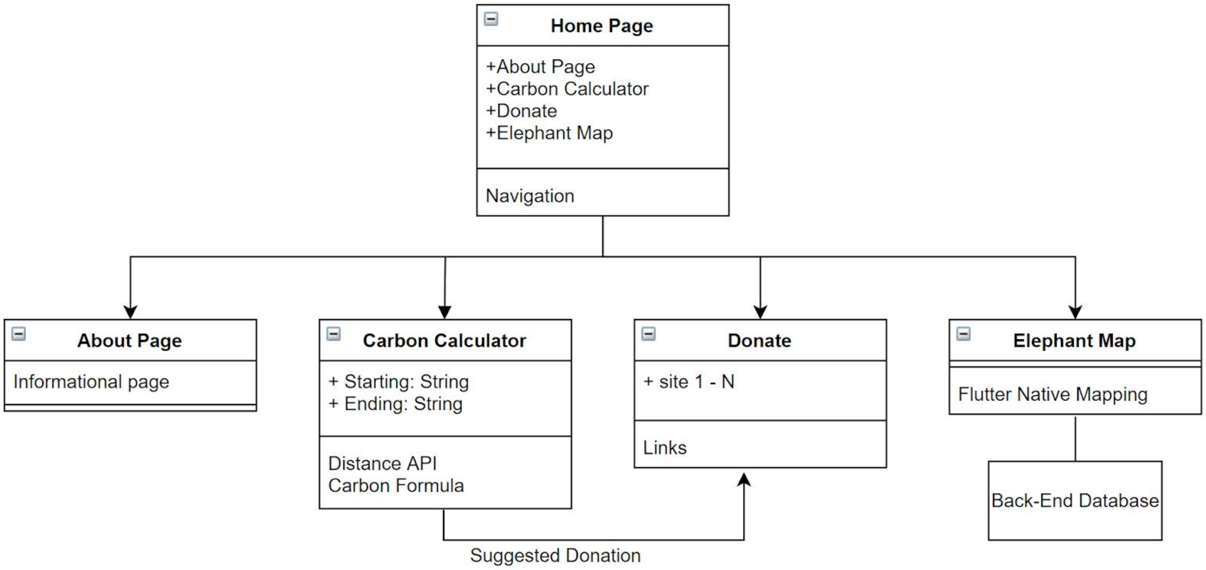


**Figure 4.1.1**
UML diagram showing the mobile application's general layout and its nodes

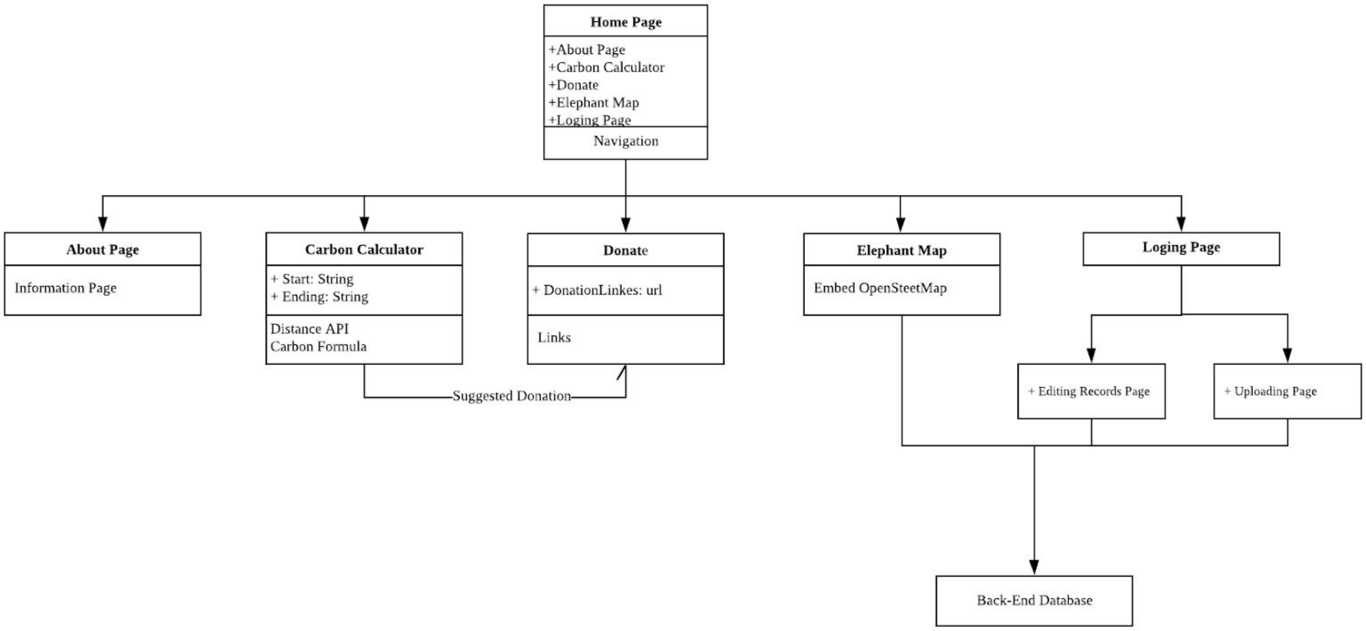


**Figure 4.1.2**
UML diagram showing the web application's general layout and its nodes

### 4.1.1 About Us Page

The about us page will consist of all of the following comprehensive descriptions:

- An easy-to-follow description of the African forest elephants, their populations, their contributions to the sequestration of atmospheric carbon, and their dropping population numbers due to poaching events.
- Our sponsors work as researchers at the MegaBiota labs and their vision for the applications(mobile and web).
- It will highlight the need for conservational efforts to help populations of forest elephants.
- It will include our application's necessary information that provides a general instruction to use our application efficiently.

None of these explained features will require any active input or output from either the user or the application. It is, essentially, a display page.

On the web version, this page will use the standard HTML code to write, which contains basic tags such as <head>,<body>, and <picture> to display the content. While on the mobile version, this page will use the standard Flutter code to display the information using Text() and TextStyle(). The content of this page for both web and mobile should be very straightforward, which should allow users to access it easily.

### 4.1.2 Donations

The donations page will contain links to the donation pages of several pre-selected Forest Elephant conservation groups, as well as a description of each of the group's ideals and missions to help the user make a more informed decision about the conservation groups that they plan on donating to. This page will not take the users' payment information at any point. Upon clicking each link, the app will redirect the user to the organization's donation website using an external browser. All Ears chose to do this in this way so as not to have any issues with privacy concerning the need to create accounts or having to take, keep, and maintain private user information and plan for the security risks associated with it.

### 4.1.3 Admin Login Page (web-only)

The Login page component is created for an administrator to easily manipulate vast amounts of elephant data through the web front-end and back-end database. This component will contain three web pages that allow users to login, modify elephant data to MIKE database, and import CSV files to the database to use for the elephant map.

## 4.2 Mapping

Mapping is crucial for both our mobile and web applications. As this provides the most interactive parts of our applications, we decided to have two maps: the elephant map, where the educational data about elephants is displayed to the user, and the carbon calculator map, where users can select countries as points for the calculator. Both will be discussed thoroughly in this section.

### 4.2.1 Elephant Map

The most interactive part of the application will be the elephant map page. Users can select certain regions in Africa on the screen and look at estimated elephant populations and poaching events in the area. This only takes a click of the user on the highlighted countries of Africa and the application will then proceed and search the database and bring back information on populations, poaching events in the area and be able to bring back all that information and potentially be able to create some graphs of the trends from the information. All of the data that the elephant map will be displayed is from our backend database that is updated annually through an upload of the MIKE database on the web front-end.

This module will contain multiple map container layers. The main one will import from Leaflet tools to display the central Africa zone. And a second map layer would display each region's forest elephant information when the user interacts with it. On the other hand, the Elephant map module will query different types of data from the backend database to allow the display of these data into the minor map layer.

Example code of import map from Leaflet:

```
<template>
  <div>
    <l-map>
    </l-map>
  </div>
</template>
```

### 4.2.2 Carbon Calculator Map

The carbon calculator page will have a map option that will use a distance API to track the distance between plane flights taken by users and calculate a suggested donation amount. This portion of the application will take in the input of the starting point and the ending point of their flight path. To select a starting and an ending point, the user will input key words or letters in a search bar for a suggested list of airports. The user can also utilize the map by selecting a country and choose an appropriate airport to select starting and ending points. Using those inputs along with a distance API and an emissions equation provided by our sponsors, the application will then bring back a pop-up of the trip summary, their carbon emissions, and a suggested donation amount estimated to be necessary in order to counter their carbon footprint.

The suggested donation pop-up will have the option to ‘donate’ which will then send the user to the donation links page in order for them to choose an organization and make their donation using their suggested donation amount. As an interactive part of the application, depending on the number of elephants that needs rescuing, the pop-up or the map will display some graphics of elephants on the screen.

This map will contain a simple map of the entire world where users can choose any country as a starting point and an ending point. Upon selecting a country, it will zoom in in that specific country, and will show a list of accessible airports for a starting point for the calculator. Selecting an ending point will work the same way. This module mainly utilizes the distance API to locate airports in the world. Unlike the elephant map, this is independent and does not rely on the database for inputs; it relies on user input only.

## 4.3 Backend

The backend component of our system is responsible for making the elephant poaching data available to the other web and mobile components of our system. It will therefore be divided into two parts: the publicly accessible REST API, which serves out the data, and the MariaDB database, which stores the data safely.

### 4.3.1 REST API

The REST API, implemented in Python’s Flask server framework, is responsible for sending the data out to various clients (our web and mobile applications) on request. It will serve the records in JSON form, a text standard for sending objects, to all that make a request to a certain URL. Additionally, it will add, delete, and make updates to the database if an incoming request is made to certain URLs with the correct credentials, i.e. the master password.

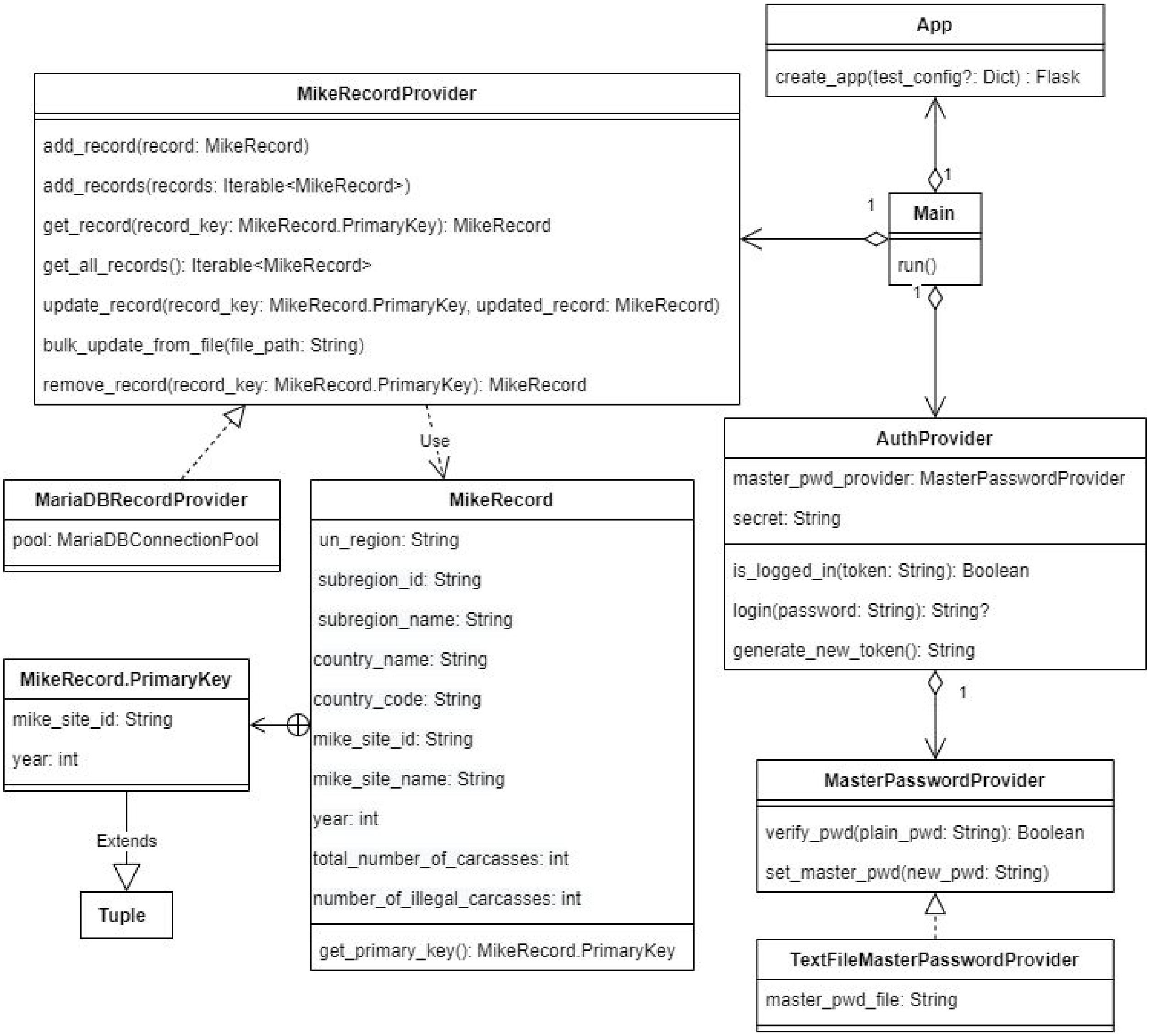

App
create_app(test_config?: Dict) : Flask
MikeRecordProvider
add_record(record: MikeRecord)
add_records(records: Iterable<MikeRecord>)
get_record(record_key: MikeRecord.PrimaryKey): MikeRecord
get_all_records(): Iterable<MikeRecord>
update_record(record_key: MikeRecord.PrimaryKey, updated_record: MikeRecord)
bulk_update_from_file(file_path: String)
remove_record(record_key: MikeRecord.PrimaryKey): MikeRecord
1
Main
run()
1
1
Use
MariaDBRecordProvider
pool: MariaDBConnectionPool
MikeRecord
un_region: String
subregion_id: String
subregion_name: String
country_name: String
country_code: String
mike_site_id: String
mike_site_name: String
year: int
total_number_of_carcasses: int
number_of_illegal_carcasses: int
get_primary_key(): MikeRecord.PrimaryKey
AuthProvider
master_pwd_provider: MasterPasswordProvider
secret: String
is_logged_in(token: String): Boolean
login(password: String): String?
generate_new_token(): String
1
MikeRecord.PrimaryKey
mike_site_id: String
year: int
Extends
Tuple
MasterPasswordProvider
verify_pwd(plain_pwd: String): Boolean
set_master_pwd(new_pwd: String)
TextFileMasterPasswordProvider
master_pwd_file: String

**Figure 4.3.1**
The class diagram for the REST API

Our application is run in the Main class with its pseudo-method run(). The term pseudo-method is used because the Main class is technically nothing but a set of commands in a file, rather than an actual class due to the quirks of both Python and the framework we are using, Flask, which is not an object-oriented framework. Main will use the create_app() method in App to load the configuration for the server and define the URL routes and the application's responses.

### Public Actions

In order to serve MIKE records, Main uses the MariaDBRecordProvider, which interfaces with the database, to serve out records.

### Administration Restriction

To do any administrative action, i.e. any action that will change the database in any way, the app will use the AuthProvider. The first request to gain administrative access will contain a password to be verified by AuthProvider. It will check with the MasterPasswordProvider to see if its hash (the result of a password going through a one-way mathematical function) matches the stored hash of the master password on file. If the two hashes match, the AuthProvider will generate a token, which is a string that the client must send with every request. AuthProvider will then verify this to see if the client is an authenticated administrator. Contained within the token is the amount of time that it is valid, which is currently set to 30 minutes. The client will be responsible for asking for a new token if it wishes to remain logged in, in which case the AuthProvider will, after checking the validity of the current token, generate a new one for the client and send it back.

### Administrative Actions

When a client is logged in it can change the database in a few predefined ways. It can upload a CSV with the same format as the MIKE database, request that the application to directly update from the MIKE database, or it can send a request containing a list of MIKE records to be added, updated, or deleted. In the first case the app will put it in a temporary file location and the MikeRecordProvider will read the file and upload it into the database. The second case will be similar to the first; instead of the client sending a file to be scanned and uploaded into the database, the server itself will download the CSV of the current MIKE database and use MikeRecordProvider to scan that file and load it into the database. In the last case, where the user sends the three lists, the application will use MikeRecordProvider to add, modify, and delete records contained in the three lists.

### 4.3.2 Database

Our product's database component, which will be implemented using MariaDB, is responsible, most prominently, for storing African forest elephant poaching information, such as country locations and the amount of elephant carcasses found in there, and allowing the values to be accessed by both the web- and mobile-applications intrinsic to the proper functionality of our project as a whole. This information will be utilized by the Mapping components in our web and mobile applications in order to provide an interactive data display showing highlighted countries that lie within the habitat-range of African forest elephants and the number of reported incidents in those countries.

The main MIKE database, available online, will be available for comparison to the All Ears database through administrative action, though only when the administrator is active on a web-application component of our team's product while active on the machine where the All Ears database is hosted. From there, the administrator may enter or upload data manually, so long as it matches the format of the table it is being entered into. They may also set a process in motion where all data on the MIKE database is saved to the local machine and then the relevant information is compared to what already exists in the local database. 'Relevant information' is determined by the set of specific countries our sponsors have indicated should be recorded; the information from those countries and those countries only are recorded into the local database from the MIKE datasheet and utilized in the All Ears web and mobile applications.

If information is present in the MIKE datasheet that matches the unique identifier of a local database entry, which is the MIKE site ID and the year the data was recorded (as no pair is or will be repeated in the whole of the datasheet), the MIKE record will replace the local database entry. Datasheet records that do not appear in the local database will be added, and entries in the database that do not appear in the datasheet will be ignored and left in the database, as they are likely manual entries inserted by the administrator. This will ensure that any errors in the data, altered by MIKE officials in later uploads of their datasheets, will be reflected in our database.

The web- and mobile-applications will then, when necessary (such as when prompted by user-interaction with the Mapping API), pull data from the local database to display poaching statistics.

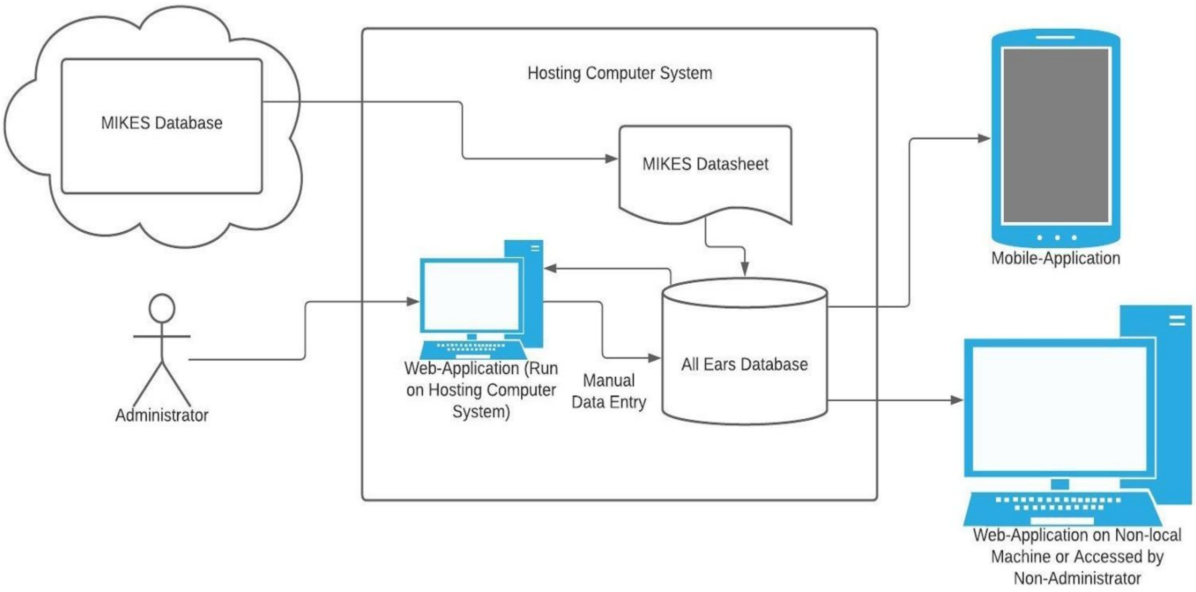

Hosting Computer System
MIKES Database
MIKES Datasheet
Administrator
Web-Application (Run on Hosting Computer System)
Manual Data Entry
All Ears Database
Mobile-Application
Web-Application on Non-local Machine or Accessed by Non-Administrator

**Figure 4.3.2**
Diagram showing the interactions between various product components and the product database

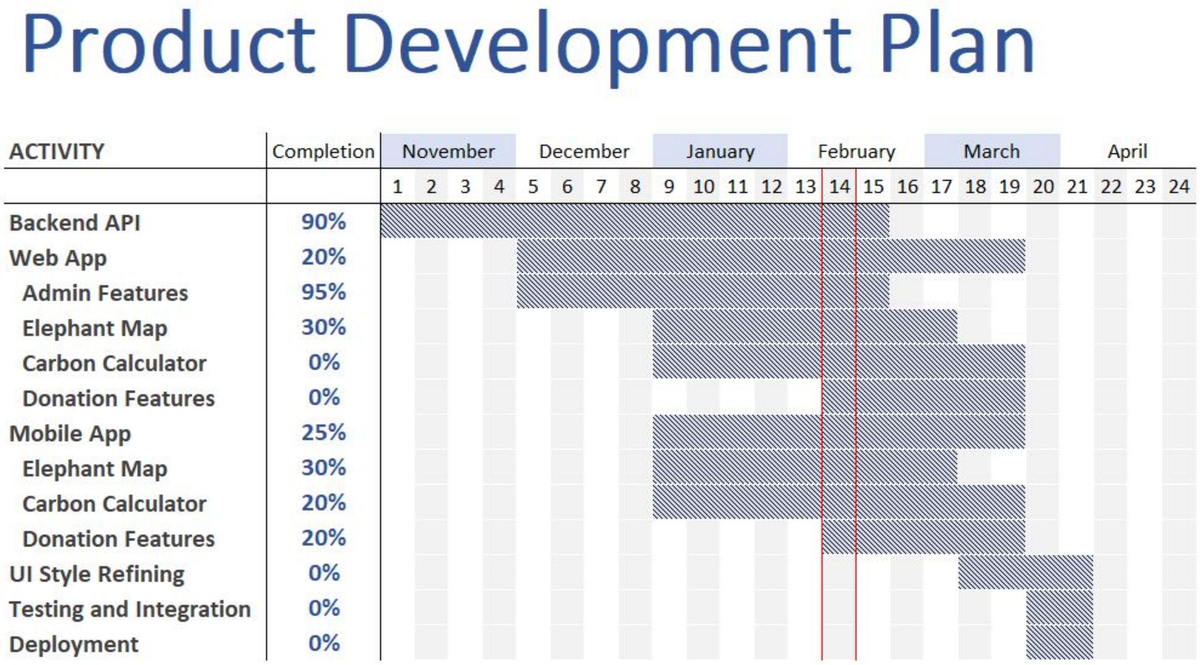
# 5 Implementation Plan

In order to develop each piece explained in the above section, we split each module into its own section: the mobile frontend, the web frontend, and the backend database. Each member of our team has found their place in production between one or more of the modules. Looking at the Gantt chart, figure 4.1 will show the implementation plan. Each modules' progress percentage is based on the timeline, and each module has a specific start and end time. The left bar of this chart shows the completion of the module work in percentage. Consider from a general view; our implementation could be split into two phases: the development phase and the testing phase.

The development phase will contain Vue, Flutter map, Flutter, Flask, MariaDB, and DigitalOcean Droplet components. We are currently working on all these modules at the same time. This is because while each module intersects with each other, they are independent enough to be developed and tested relatively independently and only depend on each other in certain circumstances. According to the Gantt chart, we will finish implementing our back-end API and admin feature in the mid of February. Then we decided to focus on elephant maps for both web and mobile development during March. With the All Ears elephant maps module complete, the rest of our modules should be done before the end of March. In conclusion, the ultimate goal of the development phase is a high-quality demo. Which will be fully developed and tested before mid of April, and it will include the below characteristics:

- A new front-end GUI for mobile and web application (End of February)
- An All Ears Elephant map to demonstrate the forest elephant's detailed information when a user interacts with the map (mid of March)
- A carbon calculator to display the straight-line distance between two points and carbon emissions. (end of March)
- The thoroughly tested and debugged demo (end of April)

After each module is fully done by timeline, we will test each individually. Since we are creating the product that researchers will use, we can split it into host testing and user testing. Host testing can check if the product fits requirements when interacting with users (usability). And user testing will test if our product meets our client’s expectations. We have set aside the last week of March and the first week of April to test our application to make sure that each of the components work well together and in the environment we plan to deploy them.



# Product Development Plan

| ACTIVITY | Completion |
|---|---|
| Backend API | 90% |
| Web App | 20% |
| Admin Features | 95% |
| Elephant Map | 30% |
| Carbon Calculator | 0% |
| Donation Features | 0% |
| Mobile App | 25% |
| Elephant Map | 30% |
| Carbon Calculator | 20% |
| Donation Features | 20% |
| UI Style Refining | 0% |
| Testing and Integration | 0% |
| Deployment | 0% |

**Figure 5.1**
Gantt Chart displaying the progress of our project

# 6 Conclusion

An article by Jeffrey Gettleman published in 2012 suggests that the population of African forest elephants dropped over from 700,000 to 100,000 in just a few decades, with about half of the remaining elephants residing in Gabon. These African forest elephants help the African forests thrive by making more space for trees to grow and draw down carbon dioxide levels in the atmosphere.To help prevent these environmental engineers from being completely eliminated, All Ears is designing an application to educate the general public about how important African forest elephants are to the stability of the global climate, to monitor the population density per sponsor-specified region of African forest elephants, and to mark the number of poaching events each year.

With our application, we aim to assist in the prevention of the indiscriminate poaching of African forest elephants by drawing attention to several pre-selected organizations selected by our sponsor that are most concerned with the safety of this subspecies of elephant. Overall, our applications will have these main components and features:

- **An About page**, which contains general information about our cause, our sponsors' research work, and the elephants. We would like for our users to read this first if they are truly interested, and possibly wish to donate to our cause
- **An Elephant Map**, an interactive map of the countries that are spanned by the habitat of African forest elephants and displays relevant information and poaching statistics about the various countries
- **A Carbon Calculator**, a calculator that utilizes a mathematical function designed by our sponsors that will determine a general estimate of the carbon emissions created by travelling a user given distance with a particular mode of transport, usually by flight
- **Donations Links**, hyperlinks to anti-poaching organizations that allow user the opportunity to donate to their cause
- **An Admin Page**, only for our web application, that will allow an administrator to update the database annually
- **Database**, where information from the MIKE database will be stored independently, along with any manual alterations made by those with access to our product's admin page, and can be accessed by our web and mobile applications to display poaching statistics and relevant information

In this Software Design document, we hope to have presented a detailed representation of what our web and mobile applications will look like along with the backend API. The implementation overview relays what technologies we will use in our development process, such as Flutter and Flask. Our architectural overview demonstrates how our users and administrators will interact with our application’s three main modules: the back-end, the web app, and the mobile app. The Module Interface Descriptions section demonstrates how each module or component is structured by showing concrete examples, such as the UML diagrams and snippets of code. Finally, the implementation plan lets us understand how we are going to be able to implement said components in the Module Interface Descriptions. We hope that with the effort in which we put forth in this document, that we are closer to providing knowledge to the people how tremendously important African forest elephants are to the world and its ecology.

## Works Cited

Gettleman, Jeffrey. “*In Gabon, Lure of Ivory Is Hard for Many to Resist*,” December 27, 2012.https://www.nytimes.com/2012/12/27/world/africa/in-gabon-lure-of-ivory-proves-hard-to-resist.html?pagewanted=all.