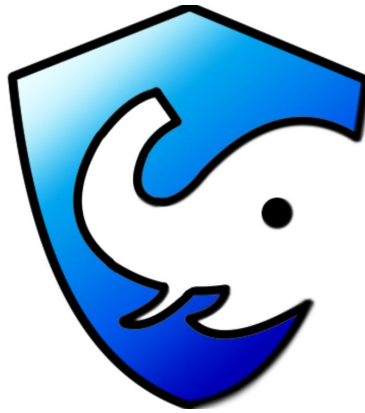


# Technological Feasibility Report



**All Ears**

**Date:** 10/9/2020

**Sponsors:** Ms. Jenna Keany and Dr. Chris Doughty

**Faculty Mentor:** Mr. Tomos Prys-Jones

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

# Table of Contents

---

1. <a href="#">Introduction</a> .....	3
2. <a href="#">Technological Challenges</a> .....	5
3. <a href="#">Technology</a> <a href="#">Analysis</a> .....	7
3.1 <a href="#">Mobile Application Frameworks</a> .....	7
3.2 <a href="#">User Interface</a> .....	12
3.3 <a href="#">Multiple</a> <a href="#">Layer</a> <a href="#">Map</a> .....	16
<a href="#">Tools</a> .....	16
3.4 <a href="#">Backend Interface</a> .....	17
3.5 <a href="#">Database</a> <a href="#">Management</a> <a href="#">System</a> .....	22
<a href="#">Technology</a> .....	22
3.6 <a href="#">Hosting Services</a> .....	26
4. <a href="#">Technology Integration</a> .....	28
5. <a href="#">Conclusion</a> .....	30

# 1. Introduction

---

African forest elephants are a threatened subspecies of elephant that inhabit the densely wooded rainforests in west and central Africa. These elephants, by removing some of the younger trees that reduce the amount of room all of the other trees have to grow, help allow their forests to mature, which in turn decreases the carbon in the atmosphere. The elephants' choice of habitat, coupled with the number of poaching incidents, makes counting the elephants in a traditional manner extremely difficult.

Our sponsors, Jenna Keany and Dr. Christopher Doughty, are both studying how African forest elephants affect forest structure, climate, and ecosystem function. They both also work to help track the poaching of elephants using tools like the database project MIKES, or Monitoring the Illegal Killing of Elephants. They are also 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 knowledge regarding African forest elephants, poaching, and the effects of each on the climate and the elephants' habitats. Similarly, poachers are getting better and more technologically advanced, 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, as well.

All Ears, in an effort to assist with our sponsors' endeavors, will create a web-based and mobile application to educate the public about the current statistics and dangers of poaching; African forest elephants' current populations and the benefits of their continued survival; and provide donation opportunities for the public to assist anti-poaching organizations, with recommendations about the amount to donate provided based on user-defined information regarding modes and distances of travel and the carbon emissions that would be released into the atmosphere. 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, and the donation recommendation will be calculated through this information. Ideally, these functions will assist in ensuring that forest elephant populations thrive, improving global environment conditions in the process. 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, displaying fauna data based on geographical regions of a map when selected using a previous application called Animaps. We will also link to several sites that the user can then make their calculated donation to.

In the next section, we will discuss the more apparent technological challenges that we will need to overcome in order to deliver our full project to our sponsors. We will first look at the most viable mobile application frameworks for the application and rate them on a scale for best useability. Important aspects of the user interface and what technologies will be available for that will follow, along with map design and the options for displaying maps in app and online. We will also analyze backend interface technologies, database management systems, and hosting services and determine which of each best suits our needs based on functionality.

## 2. Technological Challenges

---

Listed below are the more major technological challenges that will need to be resolved during the development phase of our project. In order to create a complete and well-rounded application, our group will need to implement a mobile application framework, an elegant and simple user interface, a backend interface to facilitate communication with the mobile and web applications and the database, a database to store pertinent data for use by the application and the users, and a hosting service to store the backend components of our final product.

1. Mobile Application Frameworks:
  - a. Choose a mobile application framework to develop in for both Android and iOS systems
  - b. Display data compellingly and aesthetically for our users
2. User Interface (UI):
  - a. Institute a live and interactive data mapping display with multiple layers
  - b. Utilize visual specifications to design the user interface for our application
3. Multiple Layer Map Tools:
  - a. Support map display and accompanying animations
4. Backend Interface:
  - a. Implement standardized communication methods between our web and mobile applications and our backend data source, such as an application programming interface (API)
  - b. Retrieve data from the seldomly updated databases MIKEs, and make them accessible to our application and users
5. Database
  - a. Keep data readily accessible
  - b. Allow data to be updated
  - c. Team familiarity with backend components is preferable

## 6. Hosting Service

- a. Ensure components can be maintained at a low cost
- b. Employ a hosting service to encapsulate the backend
- c. Improve the security of held data through the use of a firewall
- d. Team familiarity with the backend hosting service is preferable

In the following technology analysis, we will review each of the potential technological solutions to the corresponding challenges in order to fulfill the requirements of our project.

## 3. Technology Analysis

---

### 3.1 Mobile Application Frameworks

A mobile application framework is a collection of code that provides structure and built-in functions which are meant to ease the process of mobile application development. Facebook, Instagram, Google Ads, Trello, and various other well-known mobile applications all use mobile application frameworks. Each framework is convenient and efficient in its own way, dependent on how and for what purpose each is used for. Deciding on one will be a key part of our project because it is essentially the face of our application. By evaluating the needs of our desired final product and the capabilities of the available frameworks, our team should be able to determine a framework that will work best for our developing project. We are judging the below options with these certain criterias:

- Cross-Platform, assessed from a scale of 0 to 2. Zero being the lowest means that the Mobile Application Framework at hand absolutely does not have any cross platform capabilities whatsoever. 1 implies that it has some cross platform capabilities but comes with certain limitations if applied. Can it develop applications for both iOS and Android devices without having to recode everything? 2 being the highest meaning that the framework can be worked with across Android and iOS devices
- Ease of Development, assessed from a scale of 1 to 3. One being the lowest specifies that the framework is not optimized for efficiency during development. Does it have features to speed up the development process? 2 probably means that it meets the project's requirements we need but having the score of 3 denotes that the framework's certain features not only meets our requirements but also goes far and beyond with its capabilities
- Member Experience, assessed from a scale of 0 to 2. Zero signifies that members have no experience using the certain framework. 1 meaning that only a few or some of our members have prior experience utilizing the framework. 2 being the highest points that all of us have used the framework before, are familiar with its structure, and know our ways around it

- Community Support, assessed from a scale of 0 to 2. Zero as an assessment means that little to no users use this platform, only providing mediocre or little online documentation, and scarce resources can only be found online. 1 having some of said traits but still lacks efficiency in finding resources. 2 connotes to the framework having a huge user base, has an easily accessible online documentation, and tons of resources, videos, tutorials, etc. are readily available online

### 3.1.1 Mobile Application Frameworks In-Depth Analysis

For our in-depth analysis, we handpicked these 4 mobile application frameworks to work with and analyze. Each of them has distinct qualities that may provide useful to our project. Using the criteria above, our team will line it with our project's requirements and needs.

#### Android Native

A mobile application framework that is exclusive to the development of Android applications. It can be written using Java, Kotlin, or C++ using the Software Development Kit(SDK).

- This platform is only exclusive to Android devices and is not compatible with the development of iOS applications. If we want to spread awareness to all potential users as much as possible, it's better to avoid this framework Cross-Platform Score: 0
- Using a plugin, Android Native can be programmed in multiple languages ranging from Kotlin, Java, to dynamically typed languages like Python. So there is no need to learn a new language if we ever decide to program in with this framework. If we were only gonna develop for android devices, we would have given it a higher criterion score but as such, there's no use in doing so Ease of Development Score: 1
- As all team members have taken a class for developing mobile apps or had to develop in it for a project, we could say that the team is somewhat familiar with the framework, although only using it with the Kotlin programming language Member Experience Score: 1



- As it is used for multiple languages, there are tons of materials online providing video lectures and tutorials for each language to design and develop an Android application in. The online documentation is pretty versatile and provides documentation across all Android Devices and core topics regarding development such as Cameras, User location, etc. Community Support Score: 2

## Flutter

A mobile application framework is an open-source UI SDK developed and designed by Google in May 2017. Examples of applications that use this framework include: ebay, Baidu, Alibaba, etc.

- Flutter excels in its ability to enable program portability across numerous platforms such as Android and iOS devices without having to do much work to convert an application to one platform to another, a simple icon change and you're done. It is designed to provide portability by maintaining User Interfaces across devices Cross-Platform Score: 2
- Flutter is based on Dart, a much more efficient and readable programming language than Java, its predecessor. It also allows flexibility with Widgets for easy UI design and customization that translates to every device it was created in. Additionally, it comes with some widgets that already have an API attached to it, allowing us to use them for our project. Another key feature Flutter has is the Hot Reload feature which allows users to receive instant feedback in the development procedure Ease of Development Score: 3
- Our members are also somewhat familiar with this framework as this was required for us to learn in the Mobile Development class and we all agree that Flutter should be on the list of frameworks we should work on Member Experience Score: 1
- Flutter in the last couple of years is still gaining traction, developers building more apps with it, and enlarging the Flutter community as it is. There are tons of resources and tutorials videos for it because of its popularity lately. Its online documentation is cataloged by sections, clearly specifying widgets and the syntaxes it is to be written on, and videos for each and one of them for people to use as a guide Community Support Score: 2

## React Native

An open-source mobile application framework created by Facebook in March of 2015 to develop applications across multiple platforms. It is widely used with popular applications such as Facebook, Instagram, Skype, etc.

- React Native uses React along with the Native language to create applications for both iOS and Android devices by using JavaScript bridges but that doesn't necessarily mean code portability. Even if we can create an Android application using this, we still need to create a separate application for iOS devices as the application is not a standalone one which is terribly inefficient with the group of our size Cross-Platform Score: 0
- Same with Flutter, React Native comes with its own Hot Reload Feature, allowing for instant feedback. It also comes in with ready-made foundational applications components to speed up the development process. The components, however, cannot be used to replicate complex UI designs, as they are displayed differently in different screens Ease of Development Score: 2
- React Native is based on the language, React which is written on JavaScript. Our team only has two members who have some experience programming in JavaScript and they are already in charge of the web development side of the project, and the rest has little to no experience regarding this framework. Member Experience Score: 0
- React Native's online documentation emphasizes more on basic core components than UI components as it relies on Native's UI components instead. It has a huge user base on stack overflow with over 86,000 topics being discussed concurrently. Several videos and tutorials are also readily available online Community Support Score: 1

## Ionic

An application development platform and an open-source SDK for hybrid mobile applications.

- Using a one shared codebase, our team can build cross platform applications for iOS and Android devices using web technologies such as HTML5, CSS and JavaScript. There is a slight take though. It is proven to have performance issues in older devices due to the lack of certain technologies like Overhead Cordova and Cross Walk which is designed to cooperate with older generation devices Cross-Platform Score: 1.5

- Although there is no Hot Reload feature for instant feedback during development, Ionic offers a diverse library of mobile-optimized UI components and tools for building fast and interactive applications Ease of Development Score: 2
- Same with React Native, we might have people already familiar with web development but they are already in charge with the web application, and we cannot afford to lose time learning new technologies and would rather stick with the experience of members with certain technologies Member Experience Score: 1
- As Ionic is a true cross platform mobile application framework much like Flutter, it has a huge user base and is also getting traction from the mobile development community. Its online documentation is user and beginner friendly and also exhibits each component's usage and syntax, providing accurate information for each. There are tons of topics available online, including topics in stack overflow, tutorials in YouTube, and free/paid courses Community Support Score: 2

### **3.1.2 Mobile Application Framework Verdict**

Our team rated above mobile application frameworks based on each of its cross platform capabilities, its ease of development, the member's familiarity with the framework and its community support.

Android Native is insufficient in its development capabilities for mobile applications, specifically concerning its inability to construct an iOS application. React Native is lacking in UI customization flexibility and would require independent creation of two applications instead of one in order to ensure the Android and iOS distribution of our project.

Flutter and Ionic have the ability to fulfil critical needs of our project, such as code portability across devices and efficiency and versatility towards user interface construction - accomplished through unique built-in libraries and front-end capabilities that will help our final product appear aesthetically pleasing to future users and attract more user traffic. Ultimately, Flutter was chosen as the framework with which to develop and design our mobile application because of the performance issues Ionic has in older devices. Without knowing the demographics of our users and the devices they will be using, Flutter is the safer option over Ionic for a more comprehensive global reach for our final product.

## 3.2 User Interface

The user interface is a medium for communication between humans and machines. The user issues an introduction to the machine. The user issues an instruction to the machine, and the machine immediately starts a process, responds to information, and gives feedback to the user. The user can make the next operation decision based on the feedback received. User interface design for digital products is mainly concerned with multiple prototyping design tools, layout, information structure, and the display of interface elements on display screens and various terminal platforms. In addition, user interface developers also need to decide using which of map tools for the user interface if the application has the GPS map functionality.

For the purpose of our application, we distribute the user interface into two subsections to ensure each process is met with the final goals for our application. Visual specification and multiple layers map tool section points to different concepts of user interface. Visual specification is focused on the default settings about the interactive design part. In detail, visual specifications containing the options of prototyping design tools, navigation types, color, text, view, etc. are unified and standardized. On the other hand, the multiple layers map tool section will determine which map interactive tool can be used in our application more properly. Based on the features that we desired, the list of characteristics represents which one is best to choose for our application, some of these characteristics may not support all of the options and each characteristic has scored between 0-5.

- Legibility, Readability, and Comprehension: determined the pay-reward analysis test for our application. Pay means how much entanglement do users need to experiment on this application. Reward means is there anything worth it?
- Identifiability: determined the identification ability of our application for users. In User interface, it often refers to the user's identifiability of the interactive design
- Compatibility: determined how smoothly the chosen tool can combine with our application, a tool having high compatibility often means the learning curve of this tool will be decreased
- Stability: determined the stability of a tool, a tool with high stability means an enhancement of update capability
- Interactivity: determined the interactivity between tools and plugs. High interactivity means that plugs can be used anywhere

### 3.2.1 Interface Design Tool

#### Sketch

- Sketch is a lightweight and efficient vector design tool. On the basis of vector editing, it provides basic bitmap style support such as blur and color adjustment, and supports functions such as rectangle tools Interactivity Score: 3
- The interface is simple and beautiful, with clear functions. No floating panel, select an object/layer and the corresponding inspector will be displayed Stability Score: 4
- However, the disadvantage of Sketch only supports application on Mac Compatibility Score: 1

#### PhotoShop

- PhotoShop is a bitmap software developed by ADOBE company. The main functions of the software include adjusting the color of the image and synthesizing the image. Mainly used for graphic design, web design art and photo studio postprocessing, and also the powerful layer functionality is provided Interactivity Score: 5
- It can support both the windows and Mac platform. Besides, PhotoShop is too complex to let the first time user have the hardest learning curve Compatibility Score: 4

#### MockPlus

- Simple to operate, quick to get started, easy to interact(just drag and drop), diverse functions, rich component resources, diverse preview methods and export types Interactivity Score: 4
- MockPlus is a design collaboration platform that focuses on one-stop product design processing. It provides product development teams with full-process collaboration, high-fidelity design and design specification management support Stability Score: 5
- On the other hand, MockPlus provides a basic version for customers that is free to use Compatibility Score: 4

## 3.2.2 Navigation

### Label Menu

- The label menu is simple and efficient and is easy to operate Legibility Score: 4
- This menu remembers the current entrance location, it jumps easily at each entrance without getting lost Identifiability Score: 4
- The label menu displays the content information of the most important entry directly Stability Score: 5
- However, the label menu also has drawbacks because if there are too many function entries, this option can be cumbersome and impractical Readability Score: 1

### Drawer Menu

- The drawer menu separates content to a “drawer”, that means the navigation modules are divided into primary and secondary. Compared with the commonly used tabbed navigation, the primary and secondary distinctions between the modules of the navigation drawer are more obvious Identifiability Score: 5
- When the application is launched, each module entry of the tabbed navigation is visible. The module switching of tabbed navigation is easy, usually by clicking or sliding operation Legibility Score: 4
- When entering a module, the entrances of other modules are still clearly visible Readability Score: 3

### Springboards

- The springboards navigation is to aggregate all the main entrances on the page, allowing users to make choices. Each entrance is shown clearly Legibility Score: 3
- It is easy to find and remember the location of each entrance conveniently Identifiability Score: 3
- Such an organization methods cannot allow users to see the content for the first time, and the pressure of selection is high Readability Score: 2
- The applications that use this kind of navigation have become fewer, and are often used on secondary pages as a graphical form of content list presentation, or it is an aggregation as a series of tool entrances Compatibility Score: 1

## List Menu

- List is a mode that can carry the most information in all navigations. Like the label menu, it is not only clear in structure, but also easier for users to understand. Sometimes it is better than a label menu in terms of hierarchical attributes Legibility Score: 5
- The other good news is that the list menu would be able to hold the longer titles content since each row just represents one topic Identifiability Score: 4
- However, there are few drawbacks of List Menu that include: when there is too much content at the same level, users are prone to fatigue when browsing; Low typesetting flexibility is an issue; Entrances can only be distinguished by arrangement order and color Readability Score: 1

## Conclusion

Based on the characteristics mentioned above. We decided to use the MockPlus as our user interface design tool since it has the features of general higher scored of each characteristic: the highest scored of stability(support for whole development team), good scored of compatibility(since the basic version of MockPlus is free), and also good interactivity(multiple plugs and icons).

Considering the number of functions and content display size that our application needs to implement. The choice for the navigation menu is the label menu. A small number of functions can be quickly switched Users will not lose their way in the process of using It has the maximum clear level. In addition, label type also provides a better user experience. For example, when the one-handed mobile phone user encounters an application that is tabbed navigation, it can be operated more conveniently without having to use the other hand.

### 3.3 Multiple Layer Map Tools

According to the application requirements, a multiple layer map tool that is based on WebGIS technology needs to be used in our application. Network Geographic information systems that combine Web technologies and GIS technology, the most exciting figure of people is WebGIS(Earth Geographic Information System). WebGis, in short, is a new technology that uses Web technology to expand and improve geographic information systems. Using the request/response mechanism through the HTTP protocol. It has the ability to convert user interaction, can transmit, and display multimedia data on the browser or application. And the information in GIS is mainly spatial data that needs to be represented in graphics and images. Nowadays, there are many types of map tools that apply the WebGIS technology, and we need a general view to ensure which tool is best for our application.

#### Mapbox GL

- On the other hand, the disadvantage for Mapbox GL is that it cannot support a floating true 3D model graph Legibility Score: 2
- Mapbox GL is a JavaScript SDK provided by MapBox, which can be used for the development of various frontend geographic information. It runs on a mobile browser that supports WebGIS Compatibility Score: 4
- Smooth map browsing, rich visualization effects; easy to use, good scalability, and numerous plug-ins to meet the various needs of developers Interactivity Score: 5

#### Openlayers

- OpenLayers makes it easy to put a dynamic map in any web page. It can display map tiles, vector data and markers loaded from any source. OpenLayers has been developed to further the use of geographic information of all kinds. It is completely free, Open Source JavaScript, released under the 2-clause BSD License (also known as the FreeBSD). The pros include a mainstream projection coordinate system is supported Interactivity Score: 3
- Compatibility with various other tools exists Compatibility Score: 5
- The cons also consider as poor visual expression is available with Openlayer Stability Score: 2
- This Map design tool offers poor memory release and optimization Identifiability Score: 2



## Leaflet

- Leaflet design adheres to the idea of simplicity, high performance and good usability. It can operate efficiently on all desktop and mobile platforms  
Compatibility Score: 5
- It will take advantage of HTML5 and CSS3 on modern browsers, and it also supports access by old browsers too. Support plug-in extension, there is a friendly easy-to-use API document and a simple, readable source code  
Interactivity Score: 4
- The Cons include Leaflet does not use WebGL for rendering(), which results in a bad visual expression and no hardware acceleration is used, which results in slow response times  
Stability Score: 2

## Conclusion

Based on the advantages and disadvantages of the previous list, the best tool option for multiple layers mapping is the MapBox GL. The GL refers to WebGL, a 3D drawing protocol that allows such as JavaScript and OpenGL ES to be combined to provide hardware 3D accelerated rendering for HTML5 Canvas elements. Most of the PC and mobile browsers can support WebGL, which means that, when applied to map layering, hardware-level rendering graphics capabilities are beyond the limitations of previous browsers. The large-volume Geographic Information System data rendering and interaction issues have been solved by MapBox GL.

## 3.4 Backend Interface

Our application depends on serving and visualizing data that will be updated periodically, which makes a backend API very important. In order for our API to work, we need both a database and a backend application to interface with it and serve its data. We are considering using various backend technologies such as NodeJS Express, Go, Flask, ASP.NET Core. The first crux that we come to is deciding what kind of API we want to have.

### **3.4.1 Backend Communication API**

There are two major communication standards that we must consider. The question of whether to choose SOAP (Simple Object Access Protocol) or REST (Representational State Transfer) has been plaguing API designers ever since REST's debut in 2004. We will analyze each API type and answer that question for this project. We will use these criteria in choosing between them: which protocol introduces the least amount of complexity in solving our current issue and how easily does it integrate with most backend technologies?

SOAP is a heavyweight interface, which ironically has "simple" in its name, that exchanges its messages in XML, a standard for encoding data as text. It is often used for distributed applications and works well for ordering efforts between various computing nodes. Our application simply requires that our backend get data objects from our server to our frontend applications. For this purpose SOAP is overkill. It can however be implemented with about equal effort on most backend technologies that are out there.

REST is a simple interface choice that typically sends its objects as either XML or JSON, another standard for encoding data in textual form. One requests the data over HTTP GET or POST requests with parameters and it responds with the requested data or stores data that is being given to it. It is incredibly lightweight and simple and is the industry standard for applications like ours. It can be implemented on any framework out there due to its simplicity.

Given the above, we believe that a REST API is best for our application due to its simplicity and ease of implementation.

### **3.4.2 Server Technology**

Now that we have chosen an API, we now move to the question of which backend server technology we will use. While a REST API is incredibly easy to implement, some frameworks implement it more easily than others. The purpose of our backend is to grab data from the database and send it in a useful form to our frontend mobile and web applications. The criteria for our backend technology is as follows:

- Ease of Development: Ease of development will be scored from 0 to 3 (higher is better). A score of 3 represents a framework that our team can learn and easily use. They don't need to know it inside and out to use parts of it. The framework does not require lots of quirky, hard-to-understand, and/or repetitive code to function. Its code is eminently readable and in a programming language already known by our backend/database developers. A score of 2 represents a framework that is mostly understandable and readable (this includes the readability of the programming language). A score of 1 represents a framework that has some idiosyncrasies and requires our backend programmers to learn a significant amount of the framework to utilize its parts. A score of 0 represents a framework that is written in a programming language unknown to our backend developers and requires that the programmers know most of the framework inside and out to use it effectively
- Community Support: Community support will be scored from 0 to 2. 0 indicates no or very few guides, examples, and answers on StackOverflow, a large Q&A forum for programmers. A score of 1 indicates some guides, examples, and StackOverflow answers. A score of 2 indicates many understandable guides, examples, and StackOverflow answers
- Cross-Platform Capability and Ease of Deployment: Cross-platform capability and ease of deployment will be scored from 0 to 2. 0 indicates that the framework cannot be deployed on one of the major server-side operating systems (Linux and Windows Server) or cannot be deployed on a serverless system (a service that runs one's app in a virtual environment whenever it is requested from the internet). 1 indicates that the framework can be deployed on all of these platforms, but requires considerable configuration both in the framework's code and the host. 2 indicates that the framework that can be deployed on the platforms listed and requires some configuration on the host
- Performance: All of the frameworks being considered are all performant enough for our project and are used in industry, so this criterion will only be used as a tie breaker if frameworks are equal in all other criteria. They will not be benchmarked unless the comparison becomes necessary. When they are benchmarked, they will be compared based on the average number of requests satisfied per second and the average time it takes to satisfy a request

All of the frameworks will be compared against each other by each criterion based on how well they answer the questions posed in each category. With this in mind, we now consider our options.

## ASP.NET Core

ASP.NET Core is an open source cross platform web server supported by Microsoft. It can be used by any language that runs on the .NET Core platform, such as C#, Visual Basic, and F#. It is highly structured and is always a major contender among larger backend frameworks. Here is how it measures up to our criteria:

- Most of our team members would need to learn C# and probably become familiar with Visual Studio. While C#, the most used language for this framework, would be very familiar, one still must learn a new language along with the framework. Also, because the framework is highly structured, developers must become familiar with a lot of middleware and framework tooling in order to use it effectively Ease of Development Score: 0
- ASP.NET Core is a cross-platform, though it is harder to deploy on Linux than on Windows Server. It is easiest to deploy as a reverse-proxy server behind a general purpose web server like NGINX or IIS. It can be also deployed to serverless services such AWS Lambda or Google Cloud Run Cross-Platform and Ease of Deployment Score: 1
- This framework has an extremely wide community with a lot of resources, including an understandable, well-written user guide from Microsoft Community Support Score: 2

## Flask

Flask is a relatively lightweight Python web framework that is often used for creating APIs. It is also a very structured framework but is not as structured as frameworks like ASP.NET Core or Spring.

- Flask is written in Python and has clear and understandable syntax. Everyone on our team has had experience in Python. One can learn the framework incrementally, so one does not need to know the entire framework to utilize a part of it Ease of Development Score: 3
- A Flask application can be deployed on hosting services like the serverless AWS Lambda, Linux, or Windows servers. It is easy to deploy with extensive guides for each deployment option Cross-platform and Ease of Deployment Score: 2
- Flask has extensive documentation and a very understandable guide on its website. It has many questions and answers on StackOverflow, so one can often find that one's questions have been asked and answered by others Community Support Score: 2

## Express

Express is a lightweight framework built on top of NodeJS. NodeJS is a platform to run Javascript outside of the browser. It runs on a single thread and handles multitasking by using an asynchronous event loop. Express is high performance and widely used in modern web backends, especially REST APIs like the one we will be building.

Measured against our criteria, Express is a strong contender.

- Express is written in Javascript, a simple language usually used in browsers. Javascript can be prone to errors brought on by its weak type system. One can use other languages that compile to Javascript like Typescript, which requires almost no new syntax knowledge. The framework is simple and extensible, meaning it is easy to learn. One can develop it easily in any text editor Ease of Development Score: 2
- Express applications are easy to deploy and can be run on serverless services, Linux, and Windows servers Cross-Platform and Ease of Deployment Score: 2
- Express documentation is very extensive and one can get examples and guides from multiple sources. There is massive community support that one can draw on for help during development Community Support Score: 2

## Go/Chi

Go is a language built by Google which has syntax like Python and performance like C++. It has a built in and high level HTTP library, meaning that it can technically be considered a web framework in and of itself. That framework is extended by Chi, a package that includes the ability to structure ones applications similarly to Express.

- Go is an entirely new language that the team will need to learn. It has some concepts that are not common to any other mainstream programming language which can make using Go very difficult at first. Its general syntax is simple and understandable, but still has a learning curve. Because of its simplicity, the actual programming framework will be easy enough to learn, but only after one understands the language itself does this play a factor Ease of Development Score: 0
- Go applications are easy to deploy and can be run on serverless services, Linux, and Windows servers Cross-Platform and Ease of Deployment Score: 2
- Go's community is smaller than some of the others on this list, but it is extremely supportive and Go has a few well written guides and many answered StackOverflow questions that can guide one during development Community Support Score: 2

## Conclusion

ASP.NET Core and Go, while having many valuable features, have too steep of a learning curve to be chosen as our server technology. They have the lowest scores. This brings us to Flask and Express. Flask wins over Express by one point in the ease of development section. They are equal in all other categories.

## 3.5 Database Management System (DBMS) Technology

Storing and accessing data will be an intrinsic part of our developing product; users should be able to access information about forest elephant populations in specific regions of Africa, as well as poaching statistics. The clients also want the system to perform basic mathematical calculations to determine how much of a negative impact on the environment certain modes of travel cause over user-selected distances and how many elephants would be necessary to offset this. DBMS exist to enter, remove, and otherwise manipulate databases and the information stored within. It can also be used to define various rules for manipulating specific data in a database. Considerations for this technology include:

- Ease of development, scored on a scale of 0 to 2. 0 represents a lack of compatibility with computing languages in consideration for the construction of the main project, which are Python and SQL. 1 marks the compatibility with some, if not all, of the languages in consideration, and 2 represents complete compatibility with all languages in consideration for construction of the application
- Cross-platform compatibility, scored on a scale of 0 to 1. 0 is an observed incompatibility with one or more of the platforms our team requires (Windows, Mac, and Linux) and 1 is the compatibility with all required platforms
- Community support, ranked from 1 to 2. A score of 2 is reserved for the DBMSs with easily available online documentation, free live support, and large followings on StackOverflow (over 1000 tagged questions for that database system)

### 3.5.1 Relational vs. Non-Relational Databases

There are two main types of databases: relational and non-relational. Relational databases store their data in tables and rows and links information from different databases through the use of keys, while non-relational databases can store data according to key-value pairs, entities that can be made to hold string fields and object data, relations between entities, or some columnar structure. Non-relational databases are incredibly flexible and scalable, making them preferable for large systems where database entries will regularly expand across multiple machines. Relational databases provide more structure and grant users the ability to craft complex SQL commands to query desired data and query it faster because of the limitless indexing capabilities. Because the larger project will rely on those with administrator privileges only to update information stored within the database, and because it is highly likely that most alterations to the database will not be new entries added to the data stores, but will instead be data alterations and replacements, there is little need for a highly scalable database. Users will also expect data to be delivered fairly quickly, data related to donation information, carbon footprints, and elephant and elephant poaching statistics, a database system with a quick querying capability and easily accessible (highly indexed) information is preferable. Therefore, only relational databases will be considered for All Ear's developing product.

### 3.5.2 Relational Database Options

We will be considering relational databases that are already widely used in production and are trusted for their reliability and performance throughout the industry.

#### **MariaDB/MySQL**

A popular SQL database management system utilized by Facebook, Twitter, YouTube, Spotify, GitHub and more.

- This relational database management system utilizes C, Java, SQL, C++, Python, Tcl, PHP, and Perl. Existing backend-oriented team members have a familiarity with Java, Python, and SQL at the minimum and would require minimal training to begin development with this system in regards to language. This system is also credited as being easy to learn for the complete beginner Ease of Development Score: 2
- MariaDB/MySQL is available on Linux, Windows, and Mac. It can also be connected to both Android and iOS apps Cross-platform Compatibility Score: 1

- At the completion of this document, there are currently 625 pages tagged for this DBMS on StackOverflow; its GitHub page has site links to dedicated, live QAs for beginners; and documentation is available online for free at mariadb.com  
Community Support Score: 1

## PostgreSQL

A highly-customizable relational database management system that provides full SQL functionality and support for advanced data-types like arrays and user-defined types.

- This DBMS supports Python, Java, C#, C/C++, Ruby, JavaScript and Node.js, Perl, Go, SQL, and Tcl. Existing backend-oriented team members have a familiarity with Java, Python, and Node.js, which would expedite learning to use this system Ease of Development Score: 2
- PostgreSQL is available on Mac, Windows, and Linux. It is also able to connect to both Android and iOS apps Cross-platform Compatibility Score: 1
- Live support is available for anyone willing to create an account on postgresql.com, which appears to be free to make. At the completion of this document StackOverflow has 8398 pages of questions tagged for this DBMS. Comprehensive manuals are available at postgresql.org/docs Community Support Score: 2

## Oracle

A DBMS most commonly used for online transaction processing and data warehousing.

- Development languages of Oracle DBMS include C, C++, Java, COBOL, PL/SQL, and Visual Basic. Oracle requires a plugin in order to be used with Python-based systems. The backend development team has greater familiarity with C, C++, Java, and SQL overall, which may assist with the development process. No boolean type is available, however Ease of Development Score: 1
- This DBMS runs on both Linux and Windows, but may require a virtual machine to be used on Mac. Oracle Database Mobile Servers support Android and iOS development Cross-platform Compatibility Score: 0
- Oracle Database has 8494 tagged questions on StackOverflow at the completion of this document. Free documentation is available at docs.oracle.com, but it appears that Oracle support is for paying clients only Community Support Score: 1



## SQLite

Less powerful than many other SQL DBMS, but a common usage is to provide data storage capabilities to mobile applications. It is marked as being compact, reliable, and self-contained.

- SQLite has bindings for many programming languages, including, but not limited to, C, C++, Java, JavaScript, Scheme, and Visual Basic. Use of SQLite with Python requires the use of PySQLite and/or a specific library module when working with Python, which introduces a small amount of complexity. The team members responsible for backend development have some experience with C, C++, Java, and JavaScript at the very least, which would expedite some of the time spent learning this particular DBMS Ease of Development Score: 1
- SQLite is included, by default, in many operating systems, but can usually be installed on Linux, Mac, and Windows systems that do not already have it installed. Android has an in-built SQLite Database package. SQLite can also be used to develop for iOS Cross-platform Compatibility Score: 1
- Documentation is available at [sqlite.org/docs](http://sqlite.org/docs). The official site directs users to a forum for support, as the official mailing list has been deprecated. The only other support option recommended is for paying clients. 5671 tagged pages for SQLite are available on StackOverflow Community Support Score: 1

## Conclusion

Oracle DBMS and SQLite fall short of a few necessary criteria for our project, namely documentation, ease of development, and, in the case of Oracle, cross-platform compatibility. For this reason, Oracle and SQLite shall no longer be considered for use by All Ears for the construction of our final product. The two highest-scoring DBMSs were PostgreSQL and MariaDB; each had attractive support and documentation access, as well as compatibility with important computer languages and the required multiple development platforms. In the end, MariaDB was chosen by our group for its more well-known role in many popular applications, as well as team familiarity with its parent application, MySQL.

## 3.6 Hosting Services

A backend interface and database pairing is useless if one does not have a place for it to run. This makes the choice of a hosting service very important to our application. Our app requires that we have a database and a REST API for data provision and so we must choose a hosting service or hosting services that can accommodate those things. All hosting options being considered can host our application and provide adequate security options. We have team members that have extensive hosting administration experience, so the only consideration that we must make as we choose among the following options is cost. According to our sponsors, we need to keep our budget under \$10 a month, so no service will be considered that charges \$10 a month or more. The service with the least monthly cost is the strongest competitor.

Among the following options there are two general hosting options that we could follow: serverless hosting, where the host only runs our application when it gets requests from the internet, paired with a managed database or a dedicated cloud server, where our app and database are hosted on the same machine.

### Google App Engine and Hosted DB

The Google App Engine is a serverless hosting solution, and only charges one for the time one's application is active. It costs \$0.526 per hour of single core usage plus \$0.0071 per GB of memory per hour. This cost will likely come out to less than a dollar per month for the Flask backend application. It will however require a hosted database, which would cost \$7.67 per month, which is below our budget.

### Google Cloud Compute Engine

Google Compute Engine is a small dedicated cloud server with 1GB of memory and 1 vCPU (virtual processing core). It is priced at \$5.42 per month, which is well within our project budget.

### AWS Lambda and Hosted DB

AWS Lambda is another serverless solution with a hosted database from AWS. It charges \$0.20 per one million requests and \$0.0000166667 per GB of memory per second. This would also probably come out to less than \$1 per month when hosting our Flask app. We will additionally need a database solution from AWS as well. That costs \$8.760 per month, which is within our budget.

## **AWS EC2 Instance**

An AWS EC2 instance is a small dedicated cloud server with 2GB of memory and one vCPU. It is priced at \$11.75 per month, which is too expensive for our project.

## **DigitalOcean Droplet**

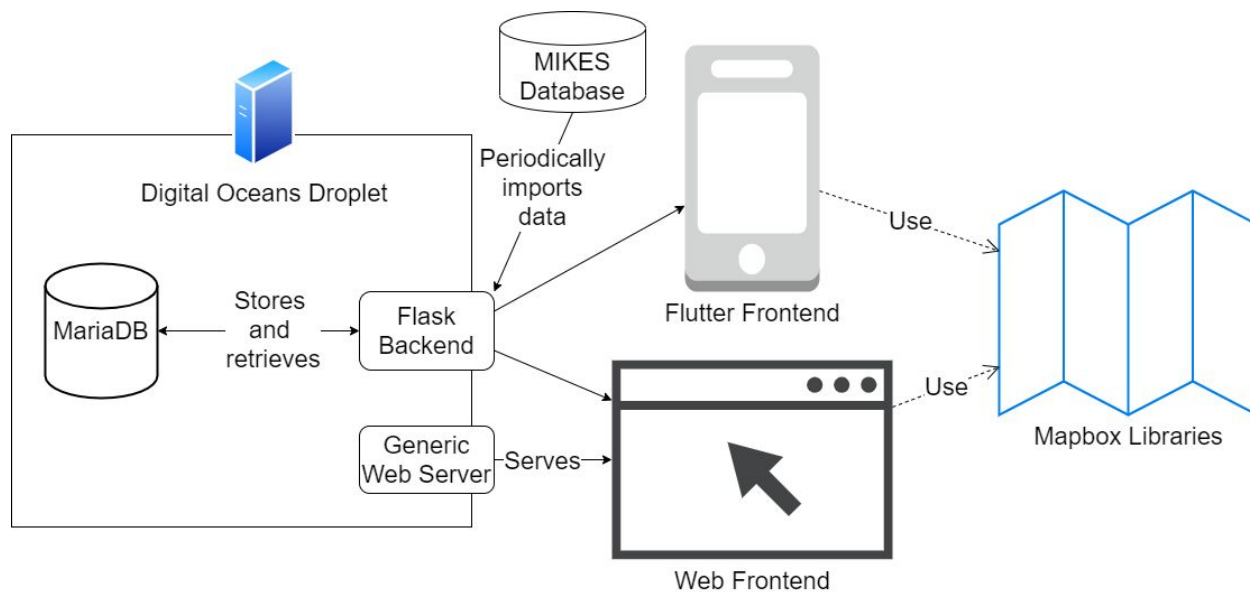
A DigitalOcean Droplet is a small dedicated cloud server with 1GB of memory, one vCPU (virtual processing core), and 25GB of storage. It is easy to administer and can handle our application. It is priced at \$5 per month, which is well within our budget.

## **Conclusion**

All options will host our application and run it effectively, so pricing becomes our primary choice motivator. DigitalOcean's Droplet is the least expensive, beating out Google Cloud Compute Engine by providing the same service for \$0.42 less. While it was not a deciding factor, the experience of one of our team members with administering DigitalOcean Droplets is certainly a bonus.

## 4. Technology Integration

After choosing out various technologies that will help us to realize our application, we now need to see how they all fit together. We intend to use MockPlus for fast prototyping and User interface design. MockPlus can implement all of the design principles we listed above. We will then implement this UI in code using Flutter and basic website technologies (HTML, Javascript, CSS). Thanks to Flutter, we will not need to implement a design in both Android Native's Kotlin and Apple's Swift. They will in turn use libraries from Mapbox, which has specialized packages for Flutter and can also easily be used in HTML.



**Diagram 1: System Diagram**

As seen in Diagram 1, MariaDB, Flask, and the basic web server will be hosted on a DigitalOcean Droplet. Flask will easily communicate with MariaDB through the Python MariaDB connector library. It will periodically receive data from MIKES and update its local database for faster data provision. It will then serve this data via a REST API to both the web application and the Flutter mobile application. Both the mobile application and web application will be using Mapbox for map displays.

Upon purchase of a DigitalOcean's Droplet environment and installation of an operating system, All Ears team developers will utilize the Secure Shell network protocol to install MariaDB onto the virtual machine. After installation, the database should be ready for general use. In order to then connect Flask to the MariaDB database on the hosting platform, we will be required to install the latest versions of Python and the MariaDB Connector onto the virtual machine, as well, to facilitate communication between the Python-based Flask framework and MariaDB.

## 5. Conclusion

---

All Ears ultimately seeks to create a cross-platform application to help spread awareness to concerned citizens and policy makers about what elephants can do for the environment and urge the public to contribute to the cause of saving elephants from the threat of poaching. The major decisions required to be made to construct our project, such as selecting a mobile framework; contemplating UI design choices; selecting a back-end application; choosing a DBMS; and selecting a hosting service for our applications, have been compiled this Technology Feasibility Report for use as an outline and a guide for the duration of the project.

To summarize our proposed solution for each technological challenge, our team proposes a confidence level chart to gauge how difficult or smooth each solution will go as we move forward. A confidence level of five being the highest, and one being the lowest.

<b>Technological Challenge</b>	<b>Proposed Solution</b>	<b>Confidence level</b>
Mobile Application Framework	<b>Flutter:</b> Robust built-in libraries, Cross-capabilities	5
User Interface	<b>MapBoxGL:</b> WebGL, 3D-accelerated rendering	5
Back-End Application	<b>Flask:</b> Simplicity, Extensive support structures, Familiarity	5
DBMS Technology	<b>MariaDB:</b> Extensive support structures, Familiarity, Security	5
Hosting Services	<b>DigitalOcean Droplet:</b> Cost-Effective, Familiarity	5

As shown, our team is quite confident that we will be able to put the puzzle pieces together and complete the project but there is still the possibility of changing our solutions or adding more challenges based on the clients' alterations with the requirements in the future. As we move forward, there will surely be more challenges we have to face and to tackle said hardships, we might want to repeat the same process we did here so we can analyze each and every detail we want to focus on to meet the requirement.

It is not only with the project's requirements that we need attention to but also the quality of our work, our team's collaboration on the project, and making major decisions in emergency situations - such as handling the loss of a team member or navigating a complete overhaul of the project. To confront these kinds of hardships, we need the help of each team member to discuss the problem at hand and ask for assistance from our mentor and clients if deemed necessary.

All Ears is positive that we have made the right decisions moving forward, and we believe that each of the solutions our team carefully settled on has what it takes to finish the project. In the event that some of the technologies chosen prove to be unsuitable for the development of our larger product, several alternatives for every chosen solution are available, and each are ultimately capable of providing the functionalities our project requires.