Ecolocation

Requirements Document

December 6, 2017

Client/Mentor:
Dr. Chris Doughty

Team Members:
Brenden Bernal
Chandler Hayes
Michael Hartzell
Anthony De La Torre

Version:
2.0

Accepted as a baseline requirements for the project:

Client Signature	Date	Team Lead Signature	Date

Table of Contents

1. Introduction

People are drawn to charismatic animals but do not understand that they have an indispensable role in contributing to their environments. For example, the elephant is an endangered species, and if it were to go extinct, it would have dramatic effects on the environment. Since it is one of the largest animals on land, it has an equally large impact on its ecosystem. It can cover larger distances than most mammals which results in it distributing more nutrients throughout its ecosystem. Due to poaching and habitat loss, the elephant population has decreased by 95% [1]. The extinction of elephants would result in many species lacking resources that elephants indirectly provide through ecosystem services.

Elephants portray how animals with greater mass are more likely to provide more ecosystem services such as **nutrient dispersal**. Nutrient dispersal is the distribution of nutrients throughout an ecosystem. This ecosystem service is vital for the growth of plants which in turn increases the productivity of the planet. The main contributors of nutrient dispersal are the bigger species in the ecosystem, but in many ecosystems, their population densities have decreased. One of the major factors that have contributed to this is due to hunting in excess. The consequences of hunting pressures results in less fertile environments for other species and plants. Biodiversity can be restored by motivating and educating the general population on the relationship between animals and their ecosystems.

Our client, Dr. Chris Doughty, has been researching the effects animals produce through ecosystem services [2]. To further analyze these effects, ecosystems are compared with how they used to be in the period before human influence. This is important because ecosystems before human existence typically had larger animals. This means these ecosystems had high nutrient dispersal and it also demonstrates how humans have impacted the environment. Accessing information on extinct and existing animals is critical in understanding the effects of changes in specific animal populations on an ecosystem. For example, our current ecosystems could still have high nutrient dispersal if hunting pressures were nonexistent.

Ultimately, understanding animals' influences on ecosystems requires a few factors: observing the effects of larger animals, focusing on nutrient distribution, and comparing ecosystems with how they were before human influence. Our goal is to increase the awareness of how important animals are for maintaining current ecosystems and to motivate people to decrease factors that limit ecosystem services. To achieve this goal we have to address current obstacles that prevent people from educating themselves on ecosystems and animals. In this document, we are defining each feature of our solution so that we can ensure our goals are met.

2. Problem Statement

In order to understand the inefficiencies of our client's workflow, it is important to understand how the work is done. Currently, our client uses Matlab programs, along with database information, to create graphical representations of animal services like nutrient dispersal. The information in the **International Union for Conservation of Nature (IUCN)** and historic databases includes **range maps**, visual representations of an animal's habitat, and masses for known mammal species. His programs use this information in mass based scaling equations, which return a numerical representation of a given species' effect. This is combined with range map information for every animal to plot each species contribution on the entire planet. To view hypothetical scenarios, he is able to alter which species are included by changing the code and rerunning the entire program. Although this is effective for our client's current work, it does have some issues.

The main problem with our client's current workflow is that it is not made for the average user. It uses databases and programming languages that most people do not have access to or experience with. Even if they are able to gain access to all needed resources, the information provided cannot be understood by the average person. A more detailed layout of the problem is:

- Average people are not able to use programs
- Currently uses all animals in the database for every calculation
- Current graphs can take hours to create
- There are multiple databases in different locations
- Historic database only available through client
- All database information must be stored locally
- Databases can only be searched using scientific names
- All animal information is very technical
- Cannot view all animals in an ecosystem

3. Solution

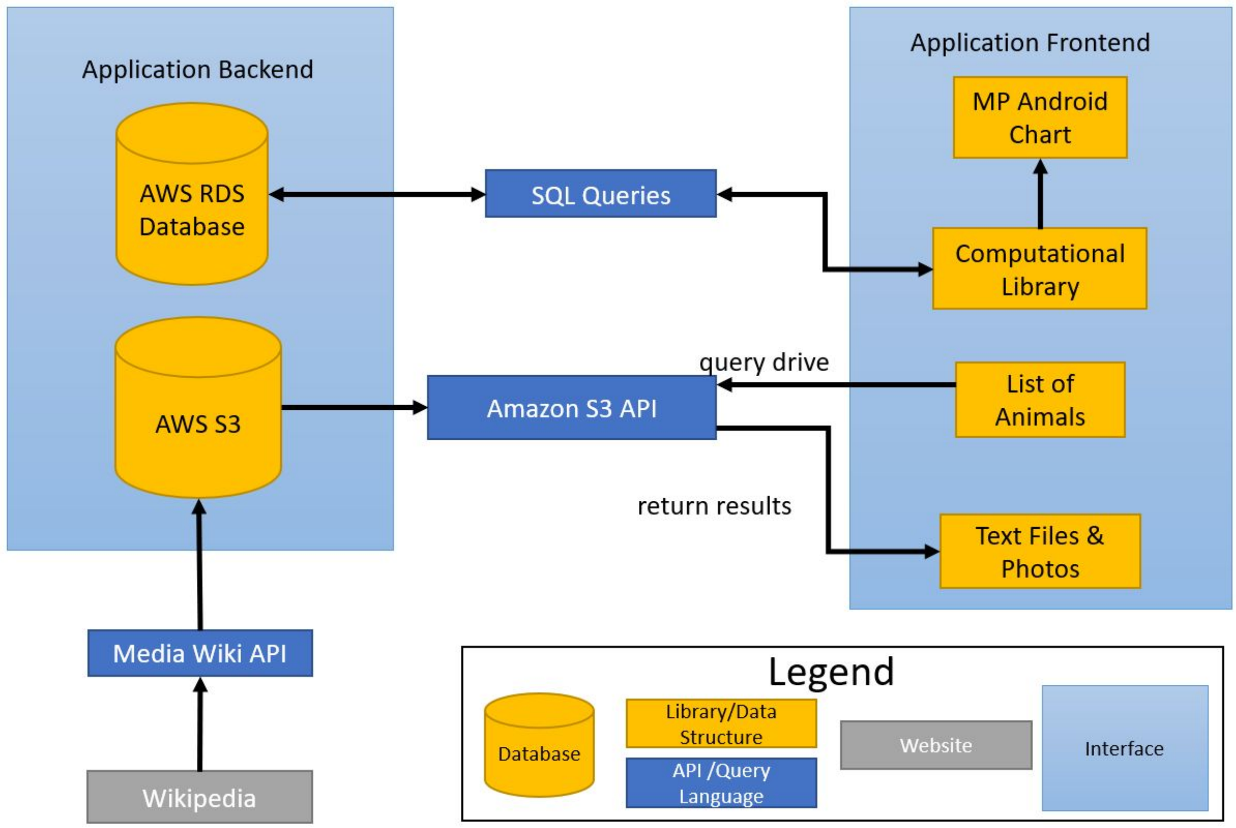
The current process of learning how animals impact their ecosystems is challenging. The two main problems in the current process are that it has **low comprehensibility** and it is **difficult to understand**. To address these problems, our solution is to create a user friendly mobile application that graphically displays how animals impact a given ecosystem. The app will focus on **nutrient distribution** (how much animals circulate nutrients throughout an ecosystem) to demonstrate the effects animals produce. There are other ecosystem services that we could have included like seed dispersal, but we decided that nutrient distribution would be the easiest one for users to understand. In a later phase in this app, additional ecosystem services may be included.

Based on a given location, the app will demonstrate these effects by showing graphs of nutrient distribution of all the animals in the ecosystem. To emphasize how changes in population numbers have a large impact, the graphs will be compared with the nutrient distribution of **historic animals** (period before humans influence) and modern animals. To enable further exploration, the user can view a list of species in an ecosystem and learn more about them by being directed to their Wikipedia page. Although, smaller animals will not be included because it would make the processing time slower and increase the amount of memory used.

Application Overview:

The app will have two main components: a backend and a frontend. The application backend will consist of the databases for hosting the species' information and the animals' pictures and summary. The first database will be hosted on **Amazon Relational Database Service** (**AWS RDS**) and contain the data from the **International Union for Conservation of Nature** (**IUCN**) and historic database. The second set of data are the pictures of the animals and their description which will be retrieved from Wikipedia and stored on **Amazon Simple Storage Service (AWS S3)**.

The second component of the app is the frontend; this is the element of the app that the user directly interacts with. The app will generate a list of animals in the user's location. The list will be used for computing nutrient dispersal of the current ecosystem and its state before human influence. The results of these calculations will be plotted onto three graphs representing the nutrient dispersal: global spatial graph, local spatial graph, and a bar graph. Furthermore, the user will be able to see the list of animals in their ecosystem. The list will contain an image and description for each species on the list. The diagram below further demonstrates how these components will be integrated.



The mobile app will have the following key features:

- Generate a list of species in the ecosystem
- Present detailed information on the species
- Produce three types of graphs on nutrient distribution
- Each graph will have a modern species graph and a historic species graph
- Enable manipulation of population numbers of specific species

Addressing Attainability

By implementing this project as an Android application, we address the attainability problem in two ways:

1. To make this information more available to the general public, we decided to create a mobile app that conveys the importance of animals in different ecosystems. This makes it more accessible because most people own a smartphone that they carry with them at all times. A smartphone is more intuitive than programming to obtain how animals impact their ecosystems. We decided to implement the app for Android phones because there

are more Android users than iPhone users. We did consider making a cross-platform application but research revealed that performance would be slower than developing for a specific platform. Since our application requires a bit of computation, we decided that native app development was the best choice.

2. Accessing these databases are crucial for the functionality of our app. It contains the data needed to compute the nutrient distribution. It also has the range maps to determine which ecosystems different species are part of and what ecosystem services they provide. Currently, we are only using two databases for our application, the mammals IUCN database and the Historic database; accessing these can be cumbersome. The IUCN databases can either be accessed through programming or directly downloaded. The first option requires programming knowledge that most people do not have while the second option requires a lot of memory. Both databases requires signing up or emailing the organization for access to along with provided reason for use. In our app, we are storing these two databases in one location on **AWS RDS**. This simplifies the process of acquiring information on both historical and modern animals as all the information will be in the same location. A later phase of the app could handle additional databases such as a database on aquatic animals.

Addressing Comprehensibility

The second issue the application addresses is making the data more understandable. With the current process there are two ways that a person could learn about animals and their relationship with ecosystems.

1. The first method is if the person has programming knowledge, then they can compute the nutrient distribution. The output of this would be a lot of data in the form of numbers. This format is difficult to understand and would require further analysis.
2. The second process would be to read paragraphs of information on different species. This method does not allow a person to clearly see the relationship between an ecosystem and its animals because it would the require the person to research each animal individually.

To simplify the comprehension of the data, we will **display the result graphically**. The app will display three graphs of nutrient dispersal. The three graphs are:

1. Selected location’s nutrient dispersal
2. Global nutrient dispersal
3. Bar graph comparing the historic and current nutrient distribution of the given location.

To summarize, we want to increase the awareness of the relationship between animals and their ecosystems. The current process is not accessible nor comprehensible. Our solution is to create an Android app that makes it easier for the average person to access and understand.

4. Project Requirements

4.1 Functional Requirements

Functional requirements are actions that the device and overall system will need to perform. The functions that our application will need to implement are the following:

- Get users location
- Retrieve data from the IUCN and historic databases
- Calculate nutrient distribution for a given location
- Provide graphical representation of data
- Retrieve images and summaries from AWS S3
- Make a list of both extinct and existing animals in an ecosystem
- Provide in-depth information on each animal

4.1.1 Location

The user must input a location to receive information on a specific ecosystem. Entering a location can be accomplished in two ways:

1. **Current Location**

   If this option is selected, then the user's location will be retrieved using the **Google Location Services API,** and those coordinates will be used to retrieve data.

2. **Manual Location**

   If the user chooses to select a specific location they will be redirected to a map where they can do so. We will be using **Google Maps API** to allow the user to choose a particular location easily. The user can select between two different input methods for entering a location: dragging a marker on the Google Map to the desired location or entering the latitude and longitude coordinates of the location.

   We decided to use Google Maps because it is commonly used, so users should be familiar with the layout and how to use it. If they are unfamiliar with Google Maps, they should be able to figure it out because it is simple to use and is intuitive. Therefore, we do not expect our users to have difficulties using this method to select a location.

4.1.2 Cloud connection

Our application will have quick and reliable access to databases that contain data on both modern and historical animals. The IUCN database will provide data for terrestrial mammals that currently exist all over the globe. The historic database contains information for species that lived during the **Pleistocene Era** which is an era without human impact.

With our app, users will not have to access these databases directly. Instead, they will have to enter an ecosystem that they want to explore by entering the location or using their current location. Then our application will query the database and provide the user with a comprehensive list of species at that location along with information on each species.

To access the information from these databases, we will retrieve the entire contents of both databases and store them on AWS RDS. This will provide reliable access to the data at an acceptable speed. For more information on specifics of AWS RDS and the speed, requirements refer to non-functional requirements.

4.1.3 Calculate Nutrient Distribution

After the app obtains a location and the data is returned from AWS RDS, we will need to calculate the nutrient distribution of the given ecosystem. The calculations will be provided by our client and mentor, Dr. Chris Doughty, in the form of Matlab code. We will need to translate the code to java and then parse the data to get the necessary input so we can get the correct nutrient distribution. With the nutrient distribution, we can create the three graphs to provide information on the effects the animals have on its ecosystem.

4.1.4 Graphs

Once the calculations are finished, the application will use graphical representations to display the results. To graph the results MPAndroidChart will be used to create and display the following graphs:

- Bar graph
    - The bar graph will display the data on the current and historic nutrient distributions.
    - If the user manipulates population numbers, an additional bar would represent the re-calculated nutrient distribution
- Spatial graphs
    - Two spatial graphs will be created to show the selected location at the center with the animal distribution in the surrounding area
        - The first will show the current animals in the ecosystem
        - The second will show the nutrient distribution from Pleistocene Era for that location
    - User will be able to zoom graph out to see the global map

4.1.5 Animal list

The application will generate a list of all existing and extinct species in a given location.
The list will be created by querying AWS RDS for all the species in the location. This list will be used to obtain all the images and summaries for each species in the list from AWS S3. The list will display some high-level information such as being color coded by threat level. The list will have the ability to be sorted several different ways including diet, endangered level, and size.

4.1.6 List Display

Each row in the list will represent an animal that belongs or used to belong to that ecosystem. The rows will be color-coded to describe the threat level of the animal. Different threat levels that we will include are the least concern, near threatened, vulnerable, endangered, critically endangered, extinct in the wild, and extinct. Each row will also display:

- An image retrieved from AWS S3
- The scientific name and the common name, if available
- A switch to make a species extinct or existent for the selected location
- Color coded to represent threat level

4.1.7 Altering Species Contribution

Allowing the users to take out selected species from the nutrient distribution calculation. This can help increase their understanding of how the ecosystem will be affected by those species disappearances. For example, the user could see how the ecosystem changes if elephants were to go extinct in an ecosystem located in Africa. This would emphasize how important elephants are to that ecosystem. The app will allow the user to:

- Select species to not include in the nutrient distribution calculations
- Select extinct species to include back in the nutrient distribution calculations
- Re-calculate the nutrient distribution after
- After re-calculations, the bar graph will include the historic ecosystem, the current ecosystem, and the altered ecosystem

4.1.8 Animal Information Page

From the screen that lists the animals in the ecosystem, our application will allow users to click on a specific animal to go to this page which will provide more information. This page will show additional information for the selected species that includes the following:

- Image of animal
- Short description
- Mass
- Population
- Diet Classification
- Link to a Wikipedia article for more information
- Endangered level

The images and short description will be taken off Wikipedia and put on AWS S3 that will then be used by the application. The IUCN and the historical database will be used to get the rest of the information which AWS RDS will host.

4.2 Non-Functional Requirements

The following are non-functional requirements. These requirements specify the execution of the functionality of the application. They include maintainability, usability, resource management, and documentation.

4.2.1 Maintainability

Both the AWS S3 and AWS RDS databases will be updated once every year to make sure that the application maintains current information. Since this information will not drastically change, we decided that once a year is an adequate period of renewal.

4.2.2 Usability

The application will be designed for use by the general public. Since the general public does not have a background in ecology or coding, we are developing the app to have simple navigation and easy to understand results. To make the critical features of the application easy to navigate, we want the user to be able to:

- Give their current location in no more than 2 clicks.
- Give a location other than their own in no more than 3 clicks.
- See graphs in no more than 4 clicks.
- View a list of local species in no more than 5 clicks.

4.2.3 Resource management

We plan on the size of the application to be no larger than 60MB. The majority of the large data is on cloud-based servers which will allow us to keep the memory size small.

4.2.4 Documentation

The application will provide detailed and thorough instructions. The documentation will explain how to use the app and how to read the graphs. The graphs themselves will be intuitive, but we will provide documentation to ensure they are understandable.

4.3 Environmental Requirements

The following are the environmental requirements for this application. These are requirements that constrain the application. It includes system deployment and development environment.

4.3.1 System deployment

We are making an Android application so we will be distributing the app on the Google Play Store because we are making an Android only application.

4.3.2 Development Environment

The Application will be developed in Android studio. We will be using MPAndroidChart library to make the graphs this library only works on Android 2.2 devices and up. We will be putting both the IUCN and historic database on AWS RDS and have both the images and summaries of all the animals put on AWS S3. The application will need an internet connection at all times due to this.

5. Potential Risks

Throughout the development of our application there are risks that we acknowledge and expect. The following provides information on these risks and how we plan on mitigating or avoiding them to ensure a well functioning and usable application.

Data Usage

Our mobile application requires an internet connection for accessing the required data. We need to make sure that we are using the absolute minimum amount of data since our app will often not be used where there is WiFi. We do not want to put an unnecessary burden on our users and use a large amount of their available mobile data since it would cause users to stop using our application. We need to keep our data usage to a reasonable limit and ensure that we are not wasting data.

Speed and Efficiency

When the user inputs their location the following occurs:

- Information is sent to AWS RDS where it will query the IUCN and historic database for information regarding animals at that location
- The query will return the desired data
- The application will parse that data and run calculations
- Results will be passed to MPAndroidChart
- MPAndroidChart will create the necessary charts
- This data will be displayed to the user

Considering the amount of steps from location input to viewing results, code efficiency is paramount to ensure a quick response time. If we have bad code and the results take upwards of 10 or 20 seconds then our users are going to get frustrated and tired of waiting. They might decide to not use the application rather than wait for results. We need to make sure we are returning results quickly so that we keep users engaged.

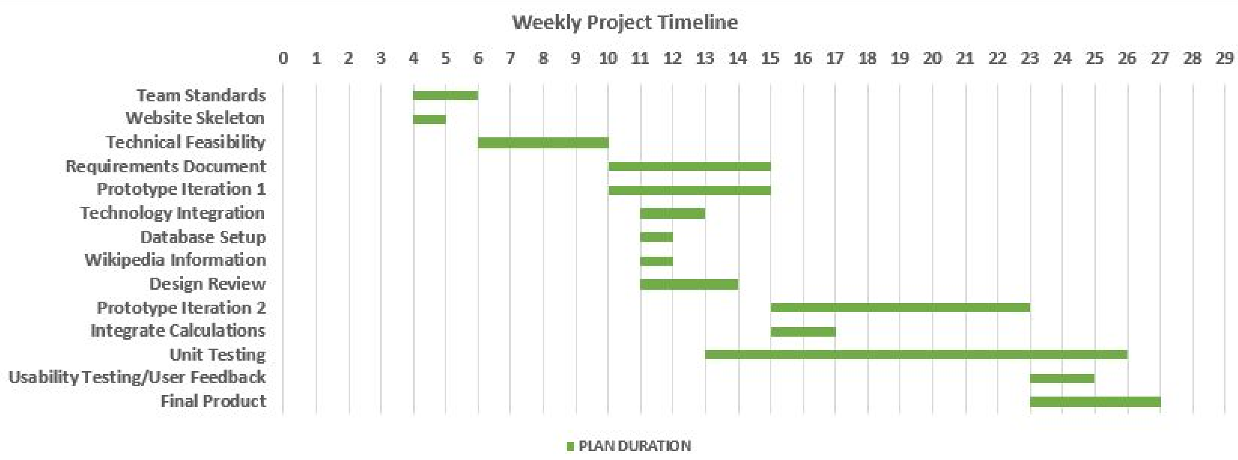
To reduce the potential for long response time we will strictly monitor the efficiency of each section of code in this section. Each section of code corresponding to the bullets above will be individually tested and re-written until they are as efficient as possible. With testing and code re-factoring this risk should not be a problem.

Stale Data and Misinformation

One of the main purposes of our project is presenting information to our users. If we do not keep our databases up to date then our information will be out of date and will present false information to our users. Also, if we have an error in our calculations then we will again be presenting false information to our users. If we are giving wrong information then our application is not serving its purpose. Another risk is that if we accidentally provide misinformation and people notice that we have wrong results we will lose credibility and lose users. We need to ensure that we are keeping our data up to date and that are calculations are returning the correct results.

6. Project Plan

Our project is being implemented over the course of eight months and is being dealt with in multiple steps. Below is an outline of out plan and information detailing when we are going to begin each step and what we plan to accomplish at each phase of development.



Currently we are working on the prototype for the application. The prototype currently consists of an outline of the screens we need as well as proof of technological integration. We are making sure all of the technologies (AWS RDS, AWS S3, MPAndroidChart) work together in Android Studio and that we will be able to seamlessly integrate into the application. Below is a list of milestones that are currently being implemented.

Database Setup

We need to setup the databases that we will be using for the application in AWS RDS so that we can access them without storing the data on each user's phone. We will be using two databases for all of our information: IUCN and historic. This milestone includes setting up accounts and permissions for access to data, as well as writing the web service to connect to the database. The actual databases will be migrated to AWS RDS by week fifteen and will be ready for the start of our second prototype iteration.

Wikipedia Information

Along with displaying a list of species at the specified location we are also showing a photo of the animal with a description. We are getting this information from Wikipedia, as it is free use, and storing it on AWS S3. This milestone includes scraping Wikipedia for all the mammals that we will be using and storing the photo and the top level description.

Technology Integration

Our application uses multiple tools, including accessing data stored on AWS RDS and AWS S3 as well as the Google Maps API and MPAndroidChart. This milestone includes incorporating all of these technologies into Android Studio and proving that they will all work together and perform as expected. This milestone includes getting the location option working for both manual and current location. It also includes pulling data from the database at chosen coordinates and returning the results. We will then use dummy data to generate graphs. The reason for using dummy data is that we are not implementing the calculations until a later step.

Integrate Calculations

This milestone will begin on week fifteen of the project which is the start of semester two. This will be the first step in the second prototype iteration. These calculations will be provided by our client and mentor Dr. Chris Doughty and they will be responsible for the data we will be graphically displaying. In this milestone we will be converting the provided calculations from MATLAB to Java so we can use them in our application. Once we have the calculations we will then parse the queried data to the proper formats so we can pass the information into these calculations and get the final results.

Unit Testing

Consistent unit testing will begin with our second prototype iteration and will continue until the final product is ready. These unit tests will be used to ensure our application is functioning as it should and to check our calculations. We need to be certain that our calculations are returning proper results for any input. Our unit tests will test all possible inputs to make sure that our application is handling the data properly.

Usability Testing / User Feedback

This milestone will begin after spring break and will go on for two weeks. This milestone includes having other people use and review our application so that we can get unbiased feedback on functionality, usability, and UI. We will be getting feedback from a number of different potential users; both technical and non-technical users. Feedback from technical users will be useful for feedback on the user interface and the user experience while feedback from non-technical users will be more useful for understanding ease of use and interpretation of results.

7. Conclusion

Animals play a crucial role in the overall health of their ecosystems. The problem is that many people are unaware of the role that mammals play and the information pertaining to nutrient distribution and animal impacts is hard to get access to and understand. Our solution to this problem is to create a mobile application for Android devices that will take data from both historic and current databases and provide all the necessary information in one location in an easy to read and understand way. The mobile application will follow this basic workflow:

- User inputs location
- Location data is sent to cloud and databases are queried
- Results are returned to application and data is parsed
- Parsed data is input into calculations
- Results of calculations are displayed graphically
- User can view graphs and list of species
- User can alter populations of species and re-run calculations
- Graphs are produced on new calculations showing effect of this change on ecosystem

Our prototype is in progress and we are on track to giving a demo in the next couple of weeks that shows the layout of the application and working integration of all the necessary technologies.

References

1. Anon. African Elephant. Retrieved November 4, 2017 from https://endangeredspeciesbiomesprojects.wikispaces.com/AfricanElephant
2. Anon. School of Informatics, Computing, and Cyber Systems. Retrieved November 3, 2017 from https://nau.edu/siccs/faculty/chris-doughty/