Software Design Doc

2/7/2025

Forest Frames & Data Integrity



Sponsored by Dr. Camille Gaillard, Dr. Chris Doughty, Dr. Duan Biggs, and Dr. Jenna Keany

Mentor: Scott Larocca

Forest Frames Members: Dalton Tippings, Daniel Austin, and Nicholas Greco

Data Integrity Members: Aidan Trujillo, Fady Zaky, and Kyle Bambling

Version 1.1

# Table of Contents

# 

[**Table of Contents 2**](#_wti665xp3fed)

[**1. Introduction 3**](#_txr0hxfzj5p2)

[**2. Implementation Overview 3**](#_gb39gqtid1nz)

[**3. Architectural Overview 6**](#_5z6djkxg0xpk)

[3.1 Architectural Diagram 7](#_etn1349ori7c)

[**4. Module and Interface Descriptions 7**](#_7vf2bzoyx2qp)

[4.1 User interface 7](#_5q6r2wm5izyw)

[4.2 Data Receiver 9](#_mi6omyr6jo4r)

[4.3 Data Verification 10](#_soydq4t2ggx4)

[4.4 Data Storage 11](#_3v2zsevd8gdy)

[4.5 Verification Status Reader 12](#_4bf6ftokqqxs)

[4.6 Interactive Map 12](#_crwqwh1q2so8)

[**5. Implementation Plan 14**](#_d3g5cv2ya3fw)

[5.1 Development Timeline Diagram 14](#_e4n9nthvyca4)

[5.2 Major Development Stages 14](#_g4gpu6ukqd6i)

[**6. Conclusion 15**](#_8ys9wykrfgbz)

# 

# 1. Introduction

In recent years, citizen science has emerged as a powerful tool for environmental conservation, leveraging the power of crowdsourced data to support wildlife monitoring, biodiversity studies, and ecosystem health assessments. This approach has been particularly impactful in countries with vast, hard-to-monitor wilderness areas. However, in many rural regions—such as those found in Malaysia or Kenya—time and financial incentives are often limited, leaving significant gaps in local ecological data collection. This capstone project addresses these gaps by developing a citizen science app that incentivizes residents in these regions to participate actively in gathering and submitting environmental data simply and efficiently.

The primary objective of this application is to make it simple and rewarding for local residents to submit verified data on flora and fauna in their surrounding wilderness areas. Through our application, residents will be able to find places that scientists have requested to be surveyed for local fauna or flora. They will then be guided to the survey area using offline mapping and guidance features to ensure that they are able to reach their destination. Ensuring residents from anywhere are able to find their way.

Using a combination of machine learning and data integration with established wildlife databases, the app verifies user submissions for accuracy and relevance. GPS locations will be attached to submissions to confirm that all users are capturing data in desired locations. Verified data not only contribute to conservation efforts but also allow for the user to receive benefits, which could range from monetary compensation to other locally relevant rewards. This approach encourages consistent engagement, offering an accessible way for locals to play a crucial role in monitoring and preserving their natural environments while aiding the wider scientific community.

By aligning with conservation goals, this app has the potential to contribute valuable insights to scientists, environmental organizations, and government agencies that rely on precise, localized data to make informed decisions and benefit communities. With this knowledge scientists will also be able to make sure that biodiversity needs and goals are being met. Through this platform, the project aims to foster an interconnected community of local citizens and scientists, creating a sustainable, data-driven approach to environmental conservation in underrepresented areas.

# 2. Implementation Overview

For our frontend, what we intend to implement is an efficient and lightweight mobile application that can run on old devices. This application will be able to connect to a remote server on our backend which will handle data processing that would otherwise be too computationally expensive on a phone. This application will seek to facilitate participation in citizen science by people who would normally be unable to participate in it. We will do this by providing an application that is able to assist regular users in getting to desired data locations, assisting in data collection, and providing verification of this data. We aim to accomplish the following goals:

**Older Android Compatibility:** To ensure the maximum number of people can participate in citizen science through our application, we are going to focus on creating an Android application compatible with Android version 12 and up. This will allow a wide range of devices to use our application without having to worry about the version. Along with this, an important aspect is to ensure this works across many devices by making it as lightweight and efficient as possible.

**Offline Data Gathering Functionality:** While some connection will be required to download and upload data for our application, it will be able to function offline for large periods of time. While a user is out gathering data, they don’t have to connect to any online services, and will only require a connection when specifically downloading data when needed and uploading gathered data to our servers. This app will allow users to pick a desired location to download offline navigation from a starting point. That way when users are traveling, they will be able to get where they need to go using pre downloaded maps minimizing confusion where a data footprint may be.

**Secure Hardware Access & Local Data Storage:** Our app will use Android system services to access the camera and microphone to gather data requested within specified coordinates and store it locally on their phone until they are ready to upload it for verification.

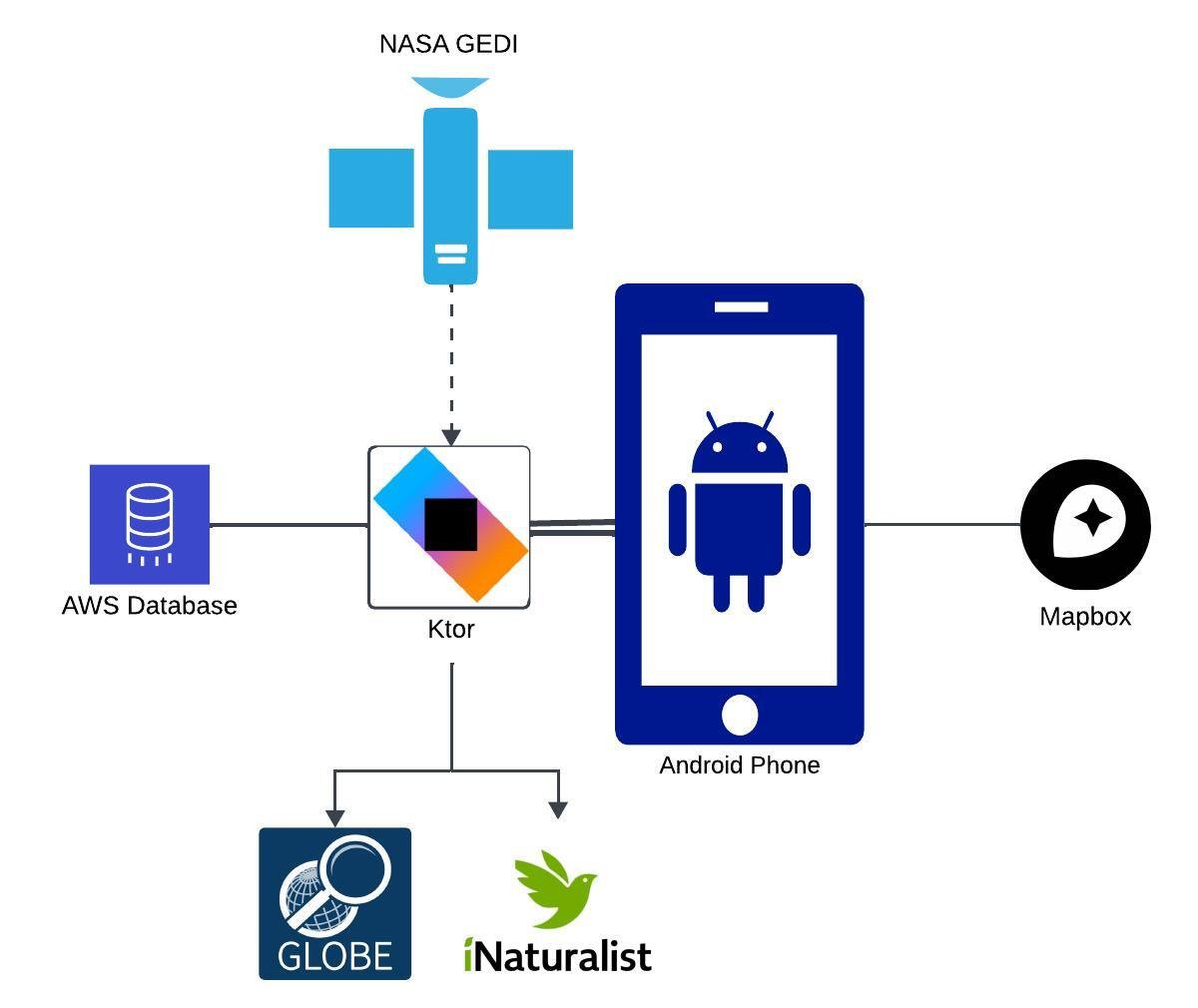
**Verification and Use of Offline Location:** Our app will use the built-in GPS system to get accurate offline location information. This will guide users to the location while also serving as a way for our app to verify the data collected by associating it with a data footprint. The GPS will serve as an important tool in getting users offline.

Our backend will consist of both a server and a database. The server will be used to verify the user-collected data using public datasets about the area that the user collected their data in. For multimedia data types, we will use multiple different open-source recognition technologies to extract data that will be comparable with those public datasets. The database will be storing all user collected data sent to the server as a backlog for future reference to data or retroactive correction. The server will also store the GEDI footprint coordinates that a user may visit so that a user's device does not need to store all of the data. This will help provide a solution to our final concern:

**Image Verification:** Verification will be conducted by using recognition/classification modules (PyTorch-Wildlife) and its datasets to cross-reference image data and get a “verification score”, which we can then use to determine if the data is trustworthy or not.

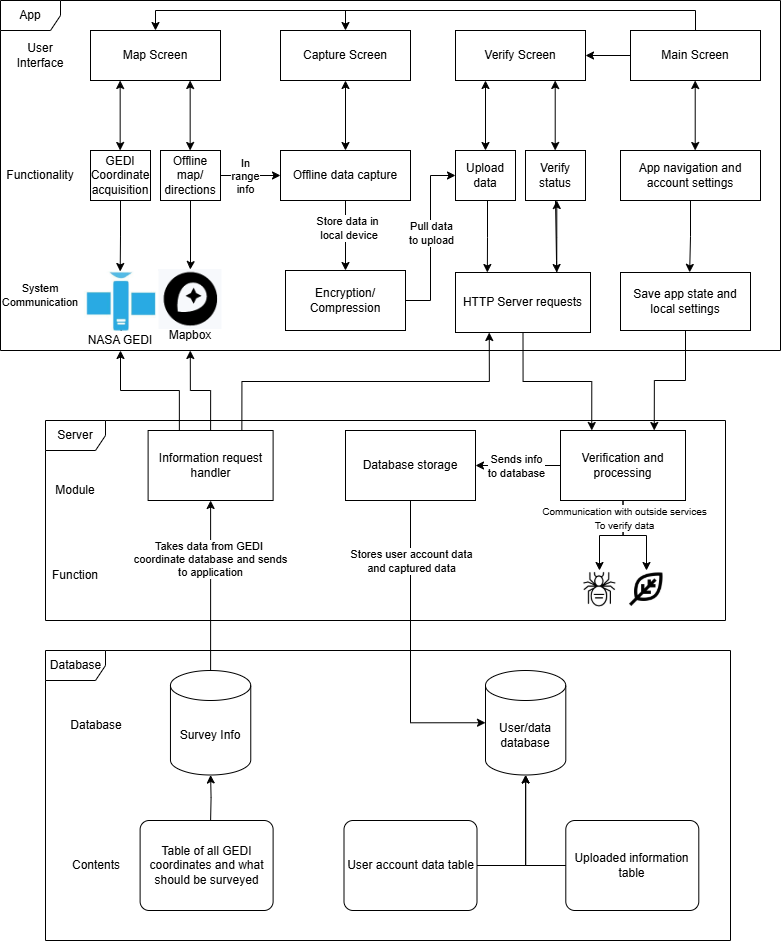
**User Data & Coordinate Storage:** GEDI coordinates will be stored in the database, but the server will do the work of processing which coordinates to send to the users phone based off of a certain radius from a starting location. This alleviates resource usage from the user's phone greatly. All other data gathered from users will be stored in the database as well, along with user information.

**Figure 1: Project’s Architecture**



# 

# 3. Architectural Overview

Below is the architectural setup of the Forest Frames system. First is a diagram showing the various components of the system and the connections between them, such as what services are provided and what information is passed between them. Following is a detailed description of the diagram and other relevant system details not clearly shown in the architecture diagram.

## 3.1 Architectural Diagram

The diagram covers the three main components of our overall system. This will keep the project modular so that user functions and interface is only within the app. Anything that requires processing and algorithms, such as verification, will be offloaded onto the server to alleviate resources on the user interface. Lastly will be the database so that we can store user data, user submissions, and desired coordinates for data collection.

On the first level of the application diagram is the **user interface** and the screens/sections of the application that the user will directly interact with. The level below describes the functionality of each screen and what it is supposed to accomplish along with any communication between the different functionalities as well. The third level describes the systems that each functionality will interact with. Such as what outside API’s/services will be used or what parts of the application will use the hardware for specific purposes.

The second major component of our system will be **the server** as shown in the middle of the diagram. This will be where all of the app’s online functions will be routed through. This component will have two primary purposes, verification of data uploaded and the passing of data to and from the database. Verification and processing of data communicates with outside services to verify data which is then stored within the database associated with a user, which includes their account settings and login information. The information request handler's primary purpose is to pull data from the database storing GEDI coordinates and the requested information of said coordinates and transmit that information to the user of the application.

The last major component of the system is **the database**, this component will only be communicated with through the server and will be the central storage for all information passing to and from the rest of the system. The first level of the database component shows the two sections that the database will be divided into. The survey info contains information on the GEDI coordinates and what was requested to be found. The other section stores the User data and associated uploaded information that is being verified/was verified until later use. The contents level describes what each table in the database should contain and how they should be organized.

# 4. Module and Interface Descriptions

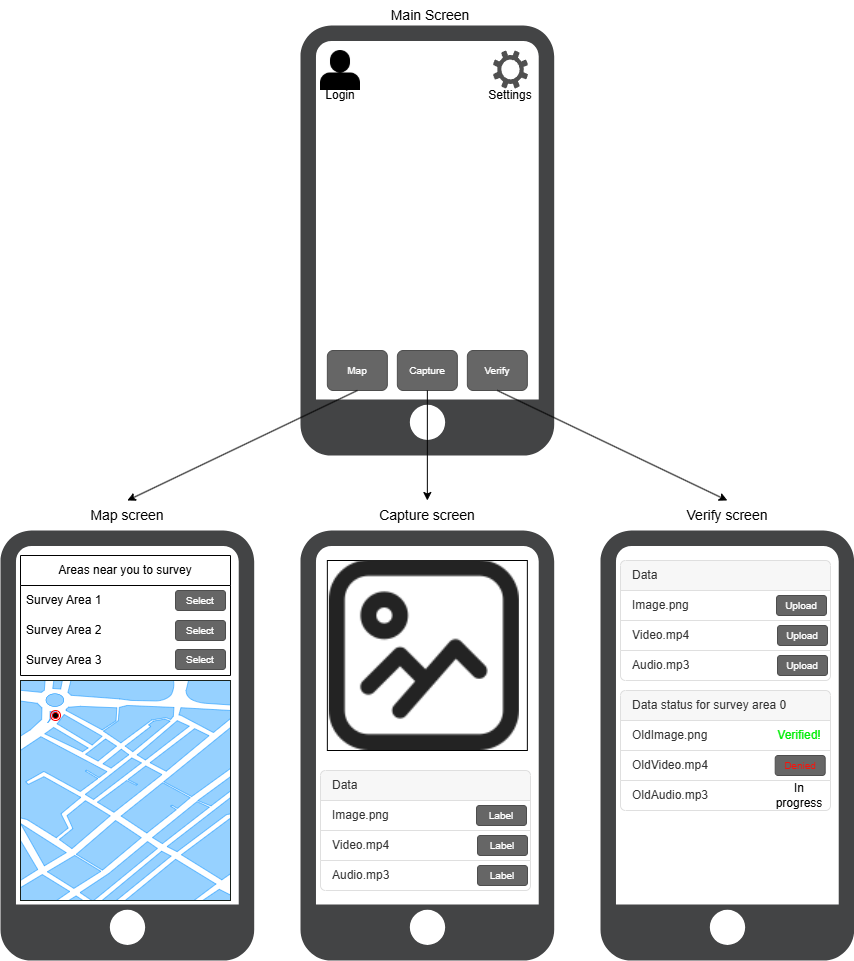
This section outlines the main aspects, or modules, of our project. These modules span from how the user interacts with the app on the frontend to how collected data is being received, modified, and stored with the server on the backend. Each module will have its general overview, a graphical model showing its layout and connectedness of its components, and detailed descriptions of those components.

## User interface

### Overview

The app interface will be made up of four primary components using XML layouts, the home screen, map screen, capture screen, and verify screen. Each screen has its own specified purpose as described below.

### Model



### Detailed Description of Components

**Home Screen** - This is the landing screen from which the rest of the app will be navigable. There will be options for app-wide settings, such as theme and other relevant options for the application. Account login/settings will also be accessible from the home screen. Navigation buttons will also be present here to travel to all other screens as needed.

**Map Screen** - This will be where the user will get a list of survey areas along with any information relevant to the user, such as what is to be surveyed and location. After a footprint is chosen, the user will then be provided with a map, which will be displayed on the map screen that the user will use to go to the survey location. Upon reaching the survey area, the user will be prompted to navigate to the capture screen to gather information.

**Capture Screen** - The capture screen will be used to capture data when inside the survey area. The user will be able to use their camera/microphone to record information, such as images of wild fauna that were requested. The collected information will be listed to the user for them to label/classify. Upon completion of the survey, the user can exit this screen with the data saved.

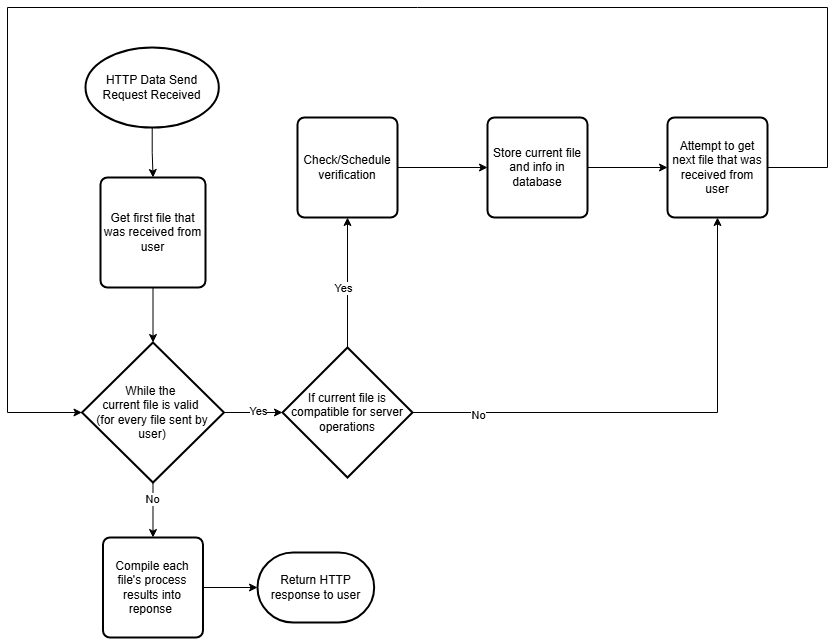
**Verify Screen** - This screen will be where the user can upload their collected data for each survey area. Here they will be able to select the areas they have surveyed and upload the relevant information. They will also be able to see the verification status of their uploaded information. If the results of verification prompt further action from the user, they will conduct said actions through this screen.

## Data Receiver

### Overview

This module, which is located on our server, is responsible for handling collected ecological data sent from the app by a user when sent for verification and/or storage. It acts as the main driver for branching off to different server operations for each received piece of ecological data.

### Model



### Detailed Description of Components

**Verify Data’s General Compatibility** - The received data’s metadata is checked to see if the server can indeed perform operations of the data type, i.e. compatibility between data and server is ensured to prevent errors. This function will run for each filename given in the HTTP request and will return True/False depending on compatibility.

**Schedule Data For Verification** - If the current file being operated on is an image and the prompt for it specifies an animal to be present, it will then “schedule” that image to be verified once a database entry has been created. Similar to prior, it will take in the filename and return True/False depending on if it will run that verification process.

**Store Data In Database** - Once prior server operations are completed on a particular file, the server will grab/update a database template for that file’s data type, also add on a verification addition if it is being verified. Once updated, it will then make a call to store this file and its information into the database.

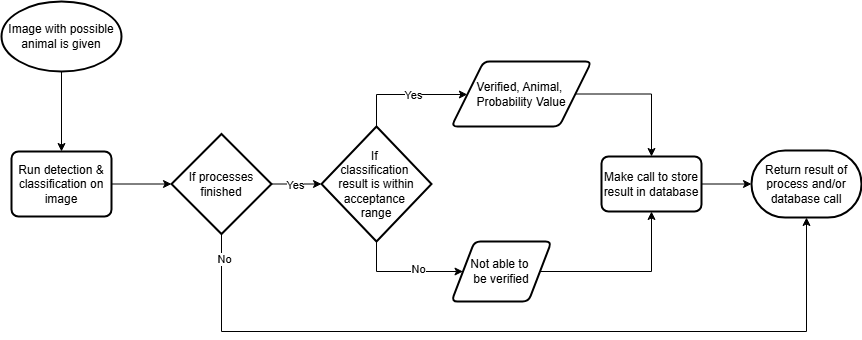
**Return HTTP Response** - Once every file has run through that process loop, a HTTP response will be returned to the user’s app indicating that their recorded files/information has been stored in our database and verifiable images are in the process of being verified. This function will take in the result from each file’s process loop (whether it was sent to be verified and/or successfully stored in the database). It will then return the formatted HTTP response, which will then be sent back by the main server driver.

## Data Verification

### Overview

This module is responsible for the verification process of images sent to the server from app users. If applicable, it will run animal detection/classification on the image, using pre-trained models from MegaDetector and the PyTorch-Wildlife modules, and will return formatted results to the database.

### Model



### Detailed Description of Components

**Detection/Classification Process** - Using the PyTorch-Wildlife module, a call for detection on the image is processed and then pipelined into the call for classification. The result of the classification is a list containing different possible animal types along with a value indicating how likely the supposed animal in the image is likely to be that type.

**Determine If Output Is Acceptable** - That prior classification result is then formatted to only contain values greater than a decided value in a config file, in order to allow for modification later preventing false positives/negatives. If an animal/value combination remains, then the highest one is chosen and determined to be the most likely to be.

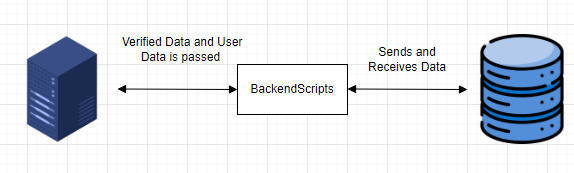
**Store Verification Result In Database** - The decided classification result is then properly formatted and a call is made to the database to store whether it was verified and its result, if applicable. It will then return a True/False depending on if the database call was successful.

## Data Storage

### Overview

The Data Storage Module is about the implementation and structure of the database and its connection to the backend server. The module takes verified data from the server backend and uses MySql commands to organize the data into tables.

### Model



### Detailed Description of Components

**Tables** - The database will be structured with 2 main data tables, User Login and Verified User Data. The User Login Table will save the username and password. This table will be completely inaccessible to the app and will only be referenced through the server for security purposes. The Verified User Data table will be significantly more complex, containing columns for actual user footage/images, user coordinates, information about the fauna being depicted, and the type of data submitted.

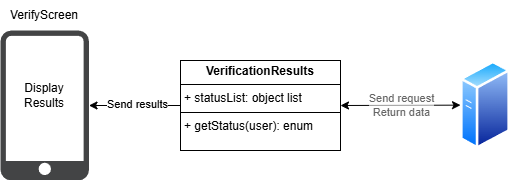
**Backend Scripts** - Multiple scripts will be written in Python and their main purpose is to communicate with the database from the server. One script will be mainly used for sending data to the database in the verified user data table, and the other will be used for retrieving data like coordinates for users.

## Verification Status Reader

### Overview

Viewable in the verification screen. This receives information from the server on the current progress of data verification for a particular piece of data.

### Model



### Detailed Description of Components

**Query Verification Status** - It will have a function getStatus that will query the server for the status of certain pieces of data the user has previously uploaded.

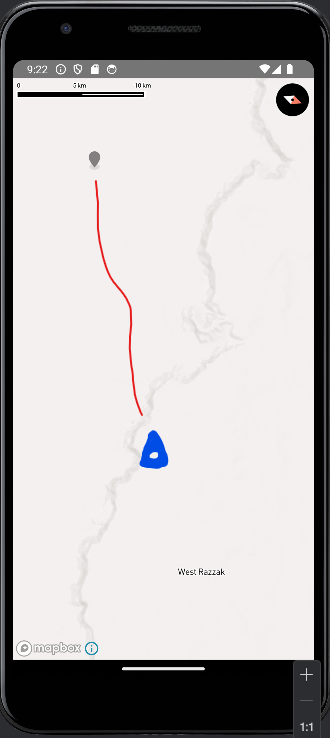
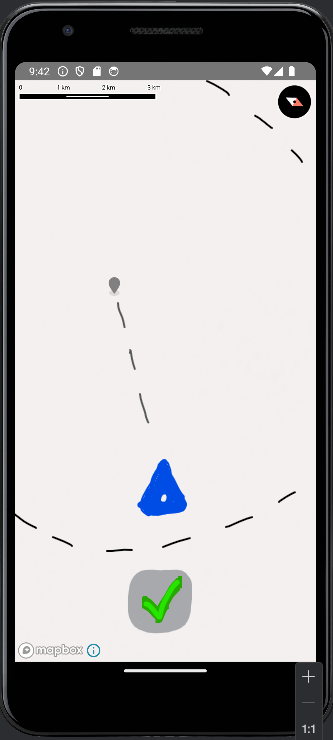
**Display Status** - These status’ will then be displayed in the verification view. Letting the user know if a piece of data was verified, still pending, or failed. If a piece of data failed, the user will be provided with options to rectify issues if any exist.

## Interactive Map

### Overview

The app will contain an interactive map which is used by the user mainly to navigate towards an acquired footprint, but also verify they are within the footprints range.

### Model

### Detailed Description of Components

**Visible User Location** - Visible on the map screen, the user's location will be displayed using Mapbox's advanced mapping interface. This interactive map provides a precise view of their surroundings, leveraging Mapbox's satellite imagery for detailed geographic context.

**Display GEDI Footprint & Route** - The map will also display the NASA GEDI carbon footprint coordinates stored in our database, highlighting key locations for users to visit. Additionally, it will provide a guided route, ensuring that even users with minimal technological experience can easily navigate to their intended destinations. Once data capture is done, the user will then be led to either the next pair of coordinates or choose to return back to their previous location.

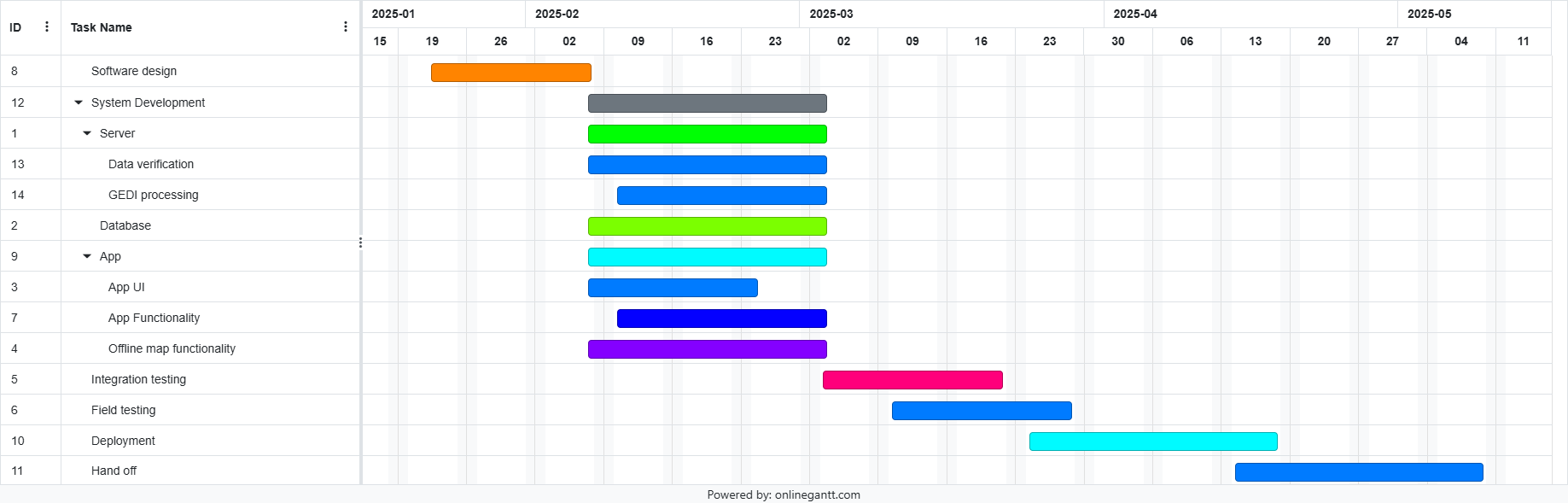
**Margin Area Creation** - Each coordinate is surrounded by a margin area, defined as a radius around the target location. Entering this area unlocks the user’s ability to begin capturing specified data, ensuring they are within the designated range for accurate data collection. Once verified to be within that area, they will then be led to the capture screen and can start recording specified data.

**Coordinate Representation** - Both the GEDI coordinates and the user’s location will be represented in latitude and longitude, ensuring seamless communication between the two and enabling precise navigation.

# 5. Implementation Plan

This next section lays out the different stages of this project. The design and requirements stages have already been completed. The main stage we will be focusing on is development. After we are confident with our app, we will send it out for testing to tweak anything important. When we have fixed anything from the testing, we will finally deploy the app to users.

## 5.1 Development Timeline Diagram



## 5.2 Major Development Stages

1. Software Design - Complete design and architecture of the system for development
2. Development (Frontend) - Develop app’s UI components
   * 1. Main - Develop main screen with settings and account management fully functional. Account functionality requires a connection to server/database.
     2. Verify - Able to receive verification status from the server, along with uploading any captured data.
     3. Capture - Able to capture data (images, videos, etc.) and store data locally.
     4. Map - Able to download GEDI coordinates and be able to guide the user to those coordinates while offline.
3. Development (Backend) - Develop components for server and database
   * 1. Server - Able to receive data, store data in the database, verify data successfully, and manage accounts. Also transferring GEDI coordinates and what is to be surveyed to the user app.
     2. Database - Able to efficiently store all necessary data and be interfaceable with the server.
4. Testing
   * 1. Integration testing - Once all components of our system complete, begin integration of all components and test their functionality in relation to each other, ensuring they work as one homogenous system.
     2. Field Testing - Have users and developers begin using the system to find potential improvements and modifications.
5. Deployment - Finalize documentation and prepare for handoff to clients.
6. Hand-off - transfer system ownership and control to the clients for future use.

# 6. Conclusion

In conclusion, our capstone project addresses a critical need in the field of environmental conservation by creating a citizen science app tailored for rural regions. The project shows the importance of local engagement in wildlife data collection, particularly in areas where ecological data is sparse, and locals often lack a reason to contribute. By creating a platform that verifies submissions against established wildlife databases and provides rewards, we are taking a big step toward making environmental monitoring both accessible and sustainable.

Our proposed solution not only fills a data gap but also creates a sense of shared responsibility among users, offering them an opportunity to contribute to conservation efforts while receiving tangible results. This document has laid out the requirements and objectives necessary to make this project, establishing a baseline for our development, and creating the tools and strategies we will use to meet our requirements.

Through this approach, we foresee creating a user-friendly, impactful app that generates valuable data for conservation. By allowing for sustained engagement and verified data contributions, our project is poised to make a long-term impact on environmental data collection in regions that need it most.