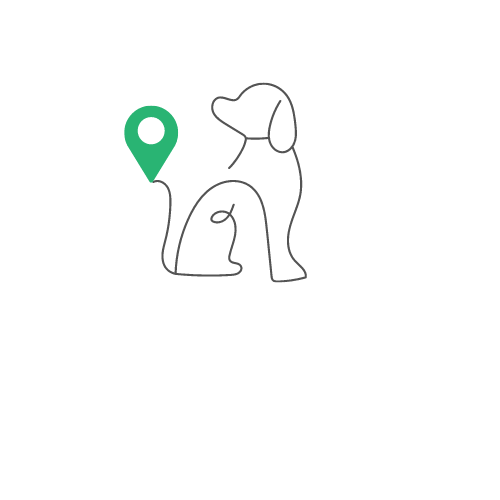
**Logo%20Main%20200**

**Boston University**

**Electrical & Computer Engineering**

**EC464 Capstone Senior Design Project**

User Manual

*Canine Tracking for SARTopo*

Submitted to:

Blair Burtan

[info@northernlightstactical.com](mailto:info@northernlightstactical.com)

by

Team #17

Canine Tracking for SARTopo

Team Members

Aanya Kutty [akutty@bu.edu](mailto:akutty@bu.edu)

Ayrton Reulet [reulayrt@bu.edu](mailto:email1@bu.edu)

Christopher Alonzo [calonzo@bu.edu](mailto:calonzo@bu.edu)

Ragib Ahsan [ragib@bu.edu](mailto:ragib@bu.edu)

#### **Table of Contents**

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

[**Executive Summary 3**](#_30j0zll)

[**Introduction**](#_1fob9te) **4**

[**2 System Overview and Installation**](#_3znysh7) **5**

[2.1 Overview block diagram](https://docs.google.com/document/d/1IsNSLwOyIQUks76q0WBB7AhzemnqjsyZ3I2hiqOlLDk/edit#heading=h.2et92p0) 5

[2.2 User interface.](#_tyjcwt) 5

[2.3 Physical description.](#_3dy6vkm) 6

[2.4 Installation, setup, and support](#_1t3h5sf) 7

[**3 Operation of the Project**](#_4d34og8) **8**

[3.1 Operating Mode 1: Normal Operation](#_2s8eyo1) 8

[3.2 Operating Mode 2: Abnormal Operations](#_17dp8vu) 9

[3.3 Safety Issues](#_3rdcrjn) 9

[**4 Technical Background**](#_26in1rg) **10**

4.1 Hardware Components 10

4.2 Software Components11

[**5 Relevant Engineering Standards**](#_lnxbz9) **13**

[**6 Cost Breakdown**](#_35nkun2) **14**

[**7 Appendices**](#_1ksv4uv) **15**

[7.1 Appendix A - Specifications](#_44sinio) 15

[7.2 Appendix B – Team Information](#_2jxsxqh) 15

# 

# **Executive Summary**

Search and rescue teams need GPS tracking collars for their dogs that work in all weather conditions, in order to track multiple dogs at once, and to tell when a dog has found something. Currently, most GPS tracking devices on the market are designed for recreational use and are limited to tracking one collar at a time. These devices are not practical for search and rescue teams that have several trained dogs working together to find evidence.

We have created a collar that is equipped with GPS and Bluetooth capabilities, as well as specific functionality that will alert the handler when a dog has found evidence. All of this information will be mapped to a mobile application accessible to everyone on the search team. The application is compatible with SARTopo, a mapping platform used by the Yavapai County Search and Rescue Team in Arizona. By connecting the collar's GPS information with SARTopo, handlers can track the location of each dog in real time and create a visual map of the search area, as well as analyze activity and location data after each search. Overall, our solution will streamline the search and rescue process, ultimately making it more efficient and accurate.

# **1 Introduction**

Search and rescue teams often have canine units trained to be a part of searches and to assist in finding evidence. There is currently no comprehensive tracking system that can effectively keep track of the activities of several canines at once. In addition to simply keeping track of multiple canines and registering their activity, we also want to connect this information directly with SARTopo, a mapping platform designed for search and rescue teams that our client is currently using.

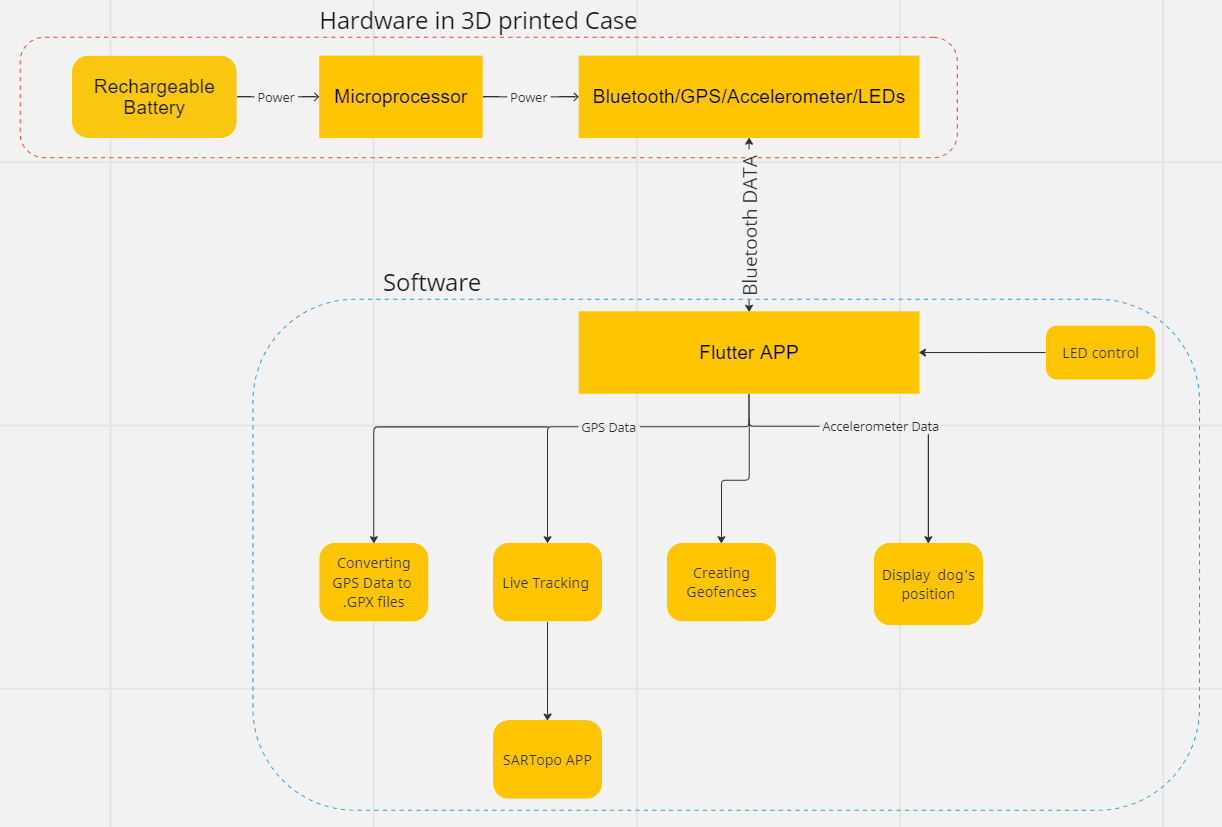
Our Client is a member of the Yavapai County Search and Rescue Team located in Prescott, Arizona. The location of our client is notable as it is a key reason for the necessity of temperature monitoring of the canines as they can often be susceptible to overheating. In addition, many of these searches are done in back country roads, leading to the necessity of the SARTopo platform in our project. By making the collars compatible with SARTopo, individuals on the search and rescue team will be able to map their searches directly to the activity of each canine. Additionally, with the use of polygonal geofences, the handler will be able to keep track of whether or not the canine is within the mapped search area. Through GPS tracking with SARTopo, as well as specific functionality such as keeping track of the canine’s physical movement, search and rescue teams will be able to execute more efficient and accurate searches.

Our solution is a two part system: a waterproof dog collar with GPS and mesh-networking radios that have a range of 1000 meters, which is then connected directly with our mobile app. The hardware is enclosed in a waterproof case in the event of a rainy search or if the dog searches in water. The GPS has a 2-meter accuracy as the location of where the dog is and where it has been is critical information for a search. The team needs to track multiple dogs at once, so the mesh-networking radio allows information to be passed from one dog collar to another. Due to the amount of time these searches can take, our device has a rechargeable battery with a twelve-hour run time. The collar uses an accelerometer to detect when the dog is sitting, a sign that evidence has been found. To ensure visibility of the collar during night activities, our collar has an RGB LED attached. To power the individual parts of our dog collar, we are using the Teensy 4.1 microcontroller. All of these hardware components are on PCBs that fit compactly in a 3D-printed waterproof case, which is then attached to the collar.

There are currently no smart collar devices on the market that are directly compatible with SARTopo, which has already been adapted by many search and rescue teams. Our smart collar and mobile application will provide a comprehensive solution and ensure safe and efficient searches.

# **2 System Overview and Installation**

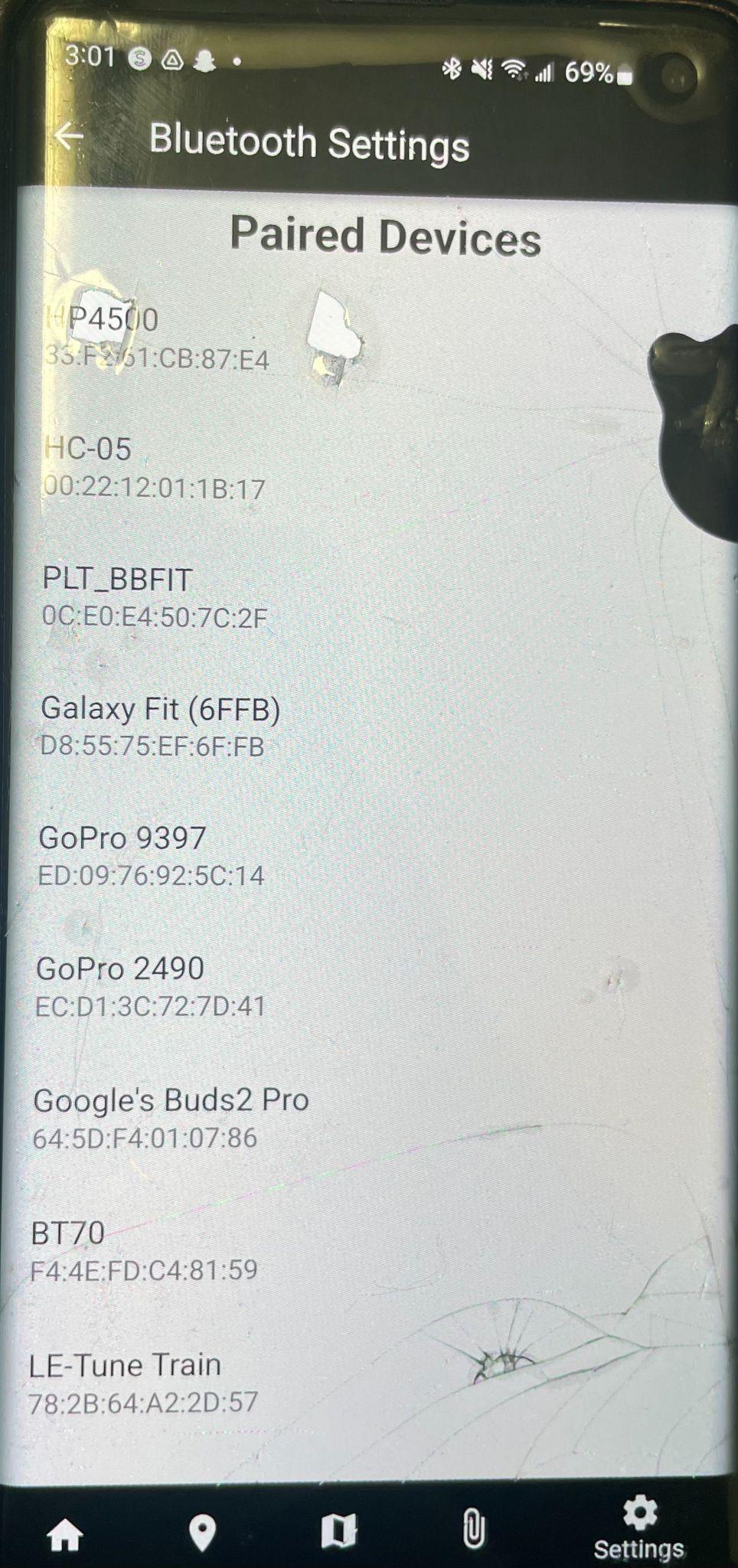
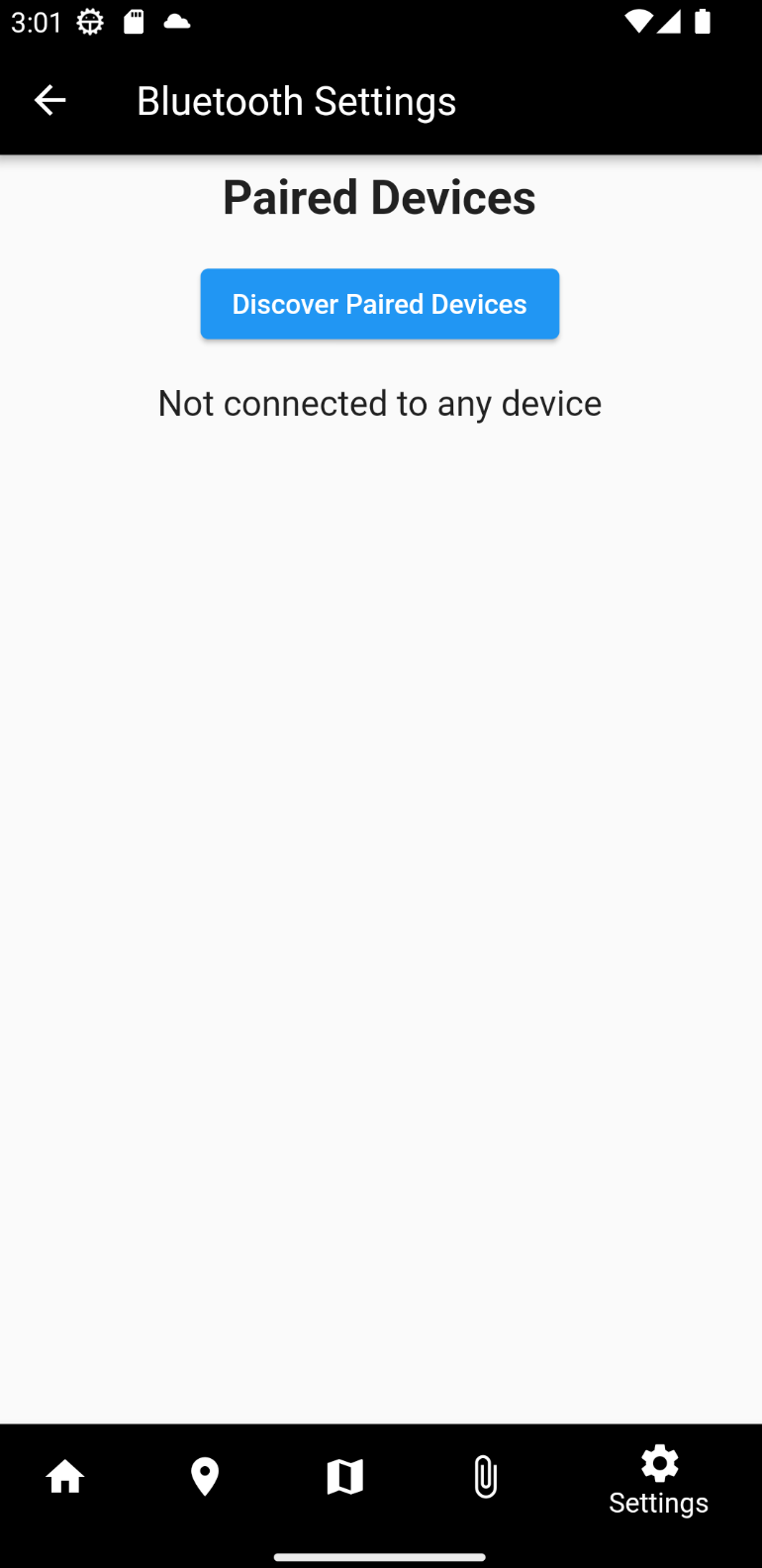
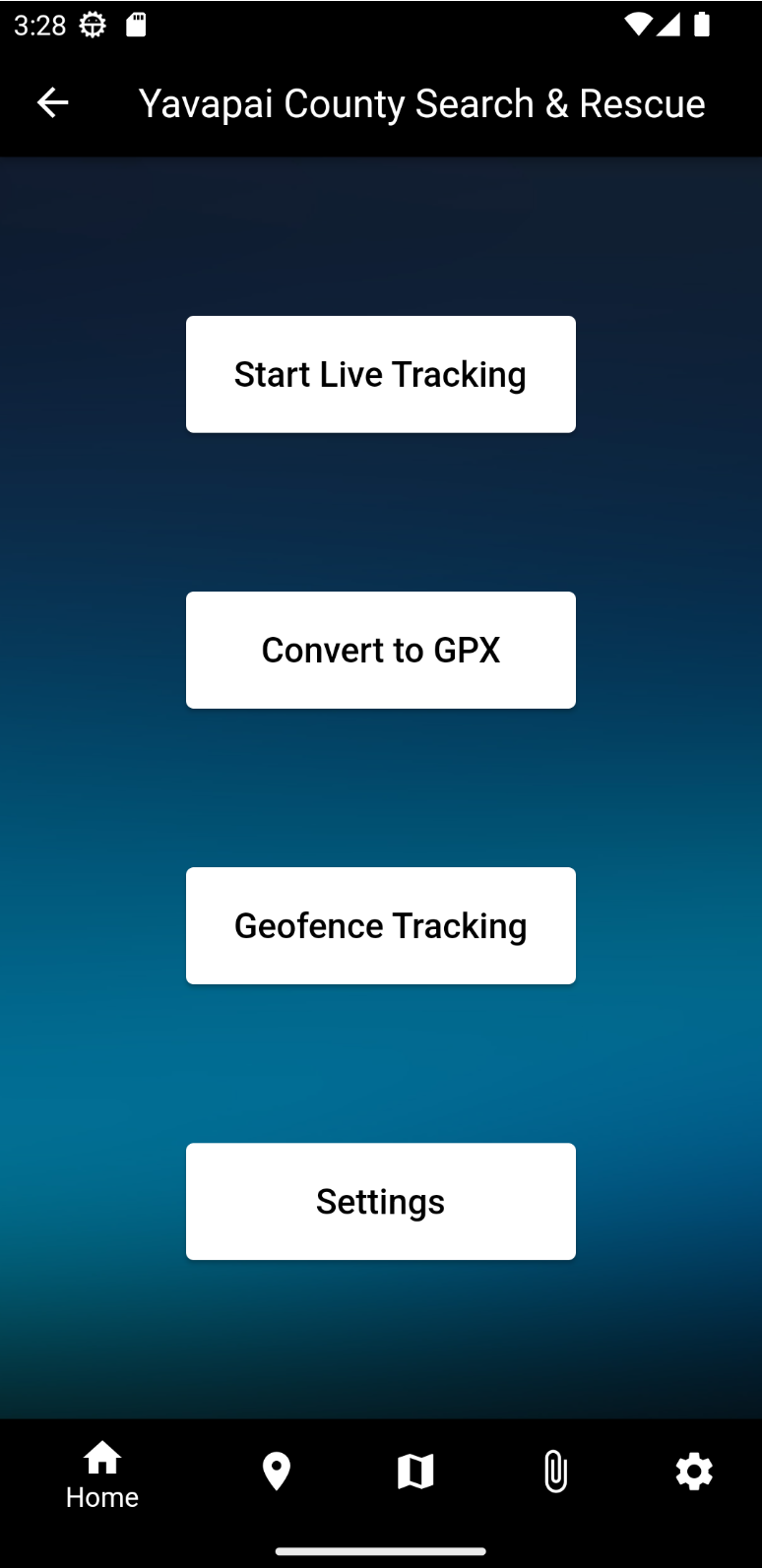
2.1. Overview block diagram



*Figure 2.1: System Overview of Hardware and Software*

Figure 2.1 is the block diagram displaying how the hardware and software function together. The rechargeable battery powers the microprocessor which powers the rest of the hardware. Data is sent between the hardware and the app via bluetooth. The GPS data is used for live tracking, the conversion to .GPX files and for the GeoFence detection. The live tracking can then be seen in the SARTopo app. The accelerometer data is used to display the dog’s position. Finally, the LEDs can be controlled from the app via Bluetooth.

## 2.2. User Interface



*Figure 2.2: Connection to device*

To connect to the device, the user must already be paired to the Bluetooth module via the settings in their phone. From there, the user can go to the settings page of our app and select ‘Discover Paired Devices.’ Select our device, which is the ‘HC-05’ and a green banner will appear at the bottom when connected as well as in a list in the settings page.

## 2.3. Physical Description

## 

*Figure 2.3: Physical device*

The physical system consists of seven major components: Teensy 4.1, MAXM10S GPS module, SMA antenna, Adafruit accelerometer, RGB LED, HC-05 Bluetooth module, and a rechargeable battery. The configuration is placed within a 3D printed case with holes for the antenna and to recharge the battery. The antenna must have visibility of the sky to connect to a satellite and send GPS data.

## 2.4. Installation, setup, and support

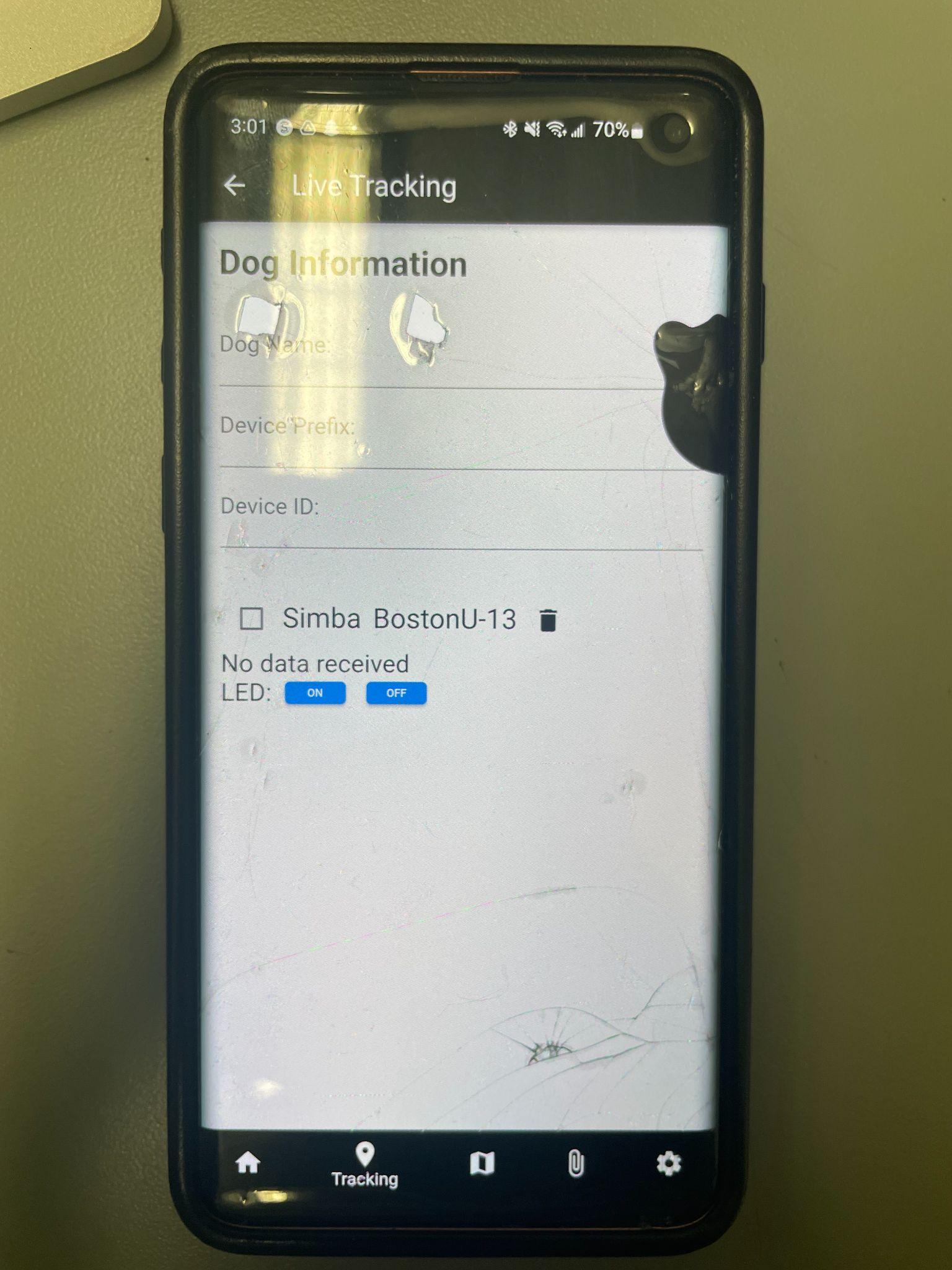
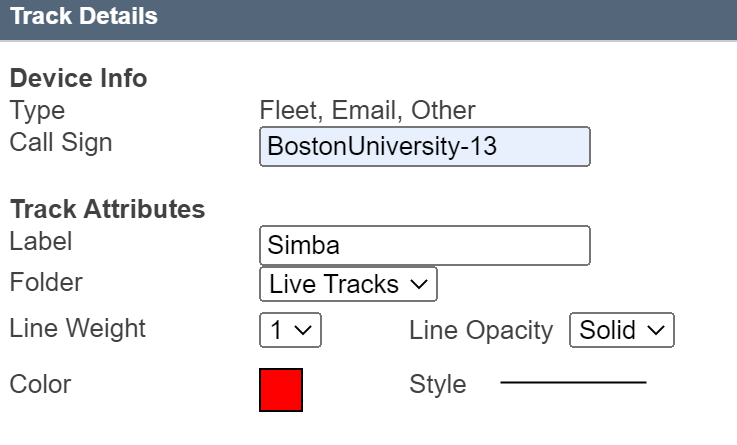
**Installation:** Each device will come prebuilt with no further assembly required, except for the need to place the device on a collar or harness.The user may need to charge the battery and plug it in to provide the system with power.

To maximize the functionality of the device, the user should consider several factors. Ideally, the device should be placed on top of the dog due to the ‘sitting’ and ‘moving’ algorithm. The algorithm may not detect if the dog is sitting if the device is placed elsewhere. Second, the device needs to be outside, so the antenna can connect to a satellite. Finally, the device needs to be within Bluetooth range of a phone with the app downloaded to connect. Additionally, the user should have a SARTopo account with live locators prepared to receive the location data.

**Setup:**

1. Download YCSR and SARTopo application.
2. Setup account and live locators in SARTopo app (Figure 2.4).
3. Pair and connect to device via settings page in YCSR app.
4. Ensure device is outside and can connect to satellite.
5. Add device information on ‘Live Tracking’ page (Figure 2.4).
   1. Ensure it is the same as the live locator on SARTopo.
6. Place on top of dog and wait for initial position results under dog information.
7. View live tracking in SARTopo app.

**Support:**For further support, users can visit our Github to determine where an error could occur.



*Figure 2.4: Live Locators in both apps*

# **3 Operation of the Project**

## 3.1 Operating Mode 1: Normal Operation

Live Tracking (Assuming device is connected)

1. When the user wants to track a dog, they should add the device prefix and device ID in both applications.
2. This information will appear in a list and be saved until the user wants to delete the device. The user can add as many devices as they want.
3. Once they click the checkbox next to the dog information, a ‘Start Tracking’ button will appear.
4. Once clicked, the user can leave the app and see the live tracking in the SARTopo app.
5. The user will then have the ability to stop tracking and delete the device once the search is over.

GPX Converter:

1. If the user wants to convert GPS data to a GPX file, then they must first select the file from their phone.
2. Then either a green banner will appear, indicating a successful conversion or a red banner, indicating a failed conversion.
3. If successful, the user can import the file in the SARTopo app and analyze the data.

GeoFence Tracking:

1. The user should create a polygon in the SARTopo app and then import it and it will save as a GeoJSON file on their phone.
2. The user can then select the GeoJSON file from their phone in our app.
3. A green banner will appear indicating that it is detecting whether the device is inside or outside the GeoFence.
4. \*Not yet implemented: In the GeoFence page on the app, the user will see all the locations where the device was outside of the created GeoFence.

Settings:

1. The user should pair the device in the settings of their phone.
2. The user should click on the ‘Discover Paired Devices’ button.
3. A list of the paired device will appear where the specific dog collar device must be selected.
4. A green banner will appear indicating that the connection was successful as well as the device name in the list of devices on the page.

## 3.2 Operating Mode 2: Abnormal Operations

Abnormal operations can occur in different parts of our system. For the hardware, the device won’t work if the battery is dead, so the user must first ensure that the battery has power. The live tracking will not work if the GPS antenna can’t connect to a satellite, in which case the user needs to ensure they are in a location accessible by a satellite or replace the antenna. The user must also ensure that they have paired the device or it will not appear in the list of paired devices within our app. As for the software, the live tracking will not appear in SARTopo if the device prefix and device ID do not match in both applications. The user can check in the list displayed on the app, and if they entered information wrong, they can delete it and re-enter the correct information. Currently, the GeoFence algorithm is not set up, so the list of locations where the device was outside of the GeoFence will not appear. The tracking of multiple dogs can lead to issues as well. If the user accidentally clicks the checkbox of an already tracked dog, then the information in the SARTopo app may be incorrect, in which case the user can delete the dog information from the app, restart the application, and enter the information again. Most of the software issues can be resolved by exiting and restarting the app. The user can then follow the steps in section 2.4 to correctly set up the system again. As dogs may move around a lot, some of the hardware may become loose, so removing the lid and checking the connections every once in a while can ensure that the device works properly.

## 3.3 Safety Issues

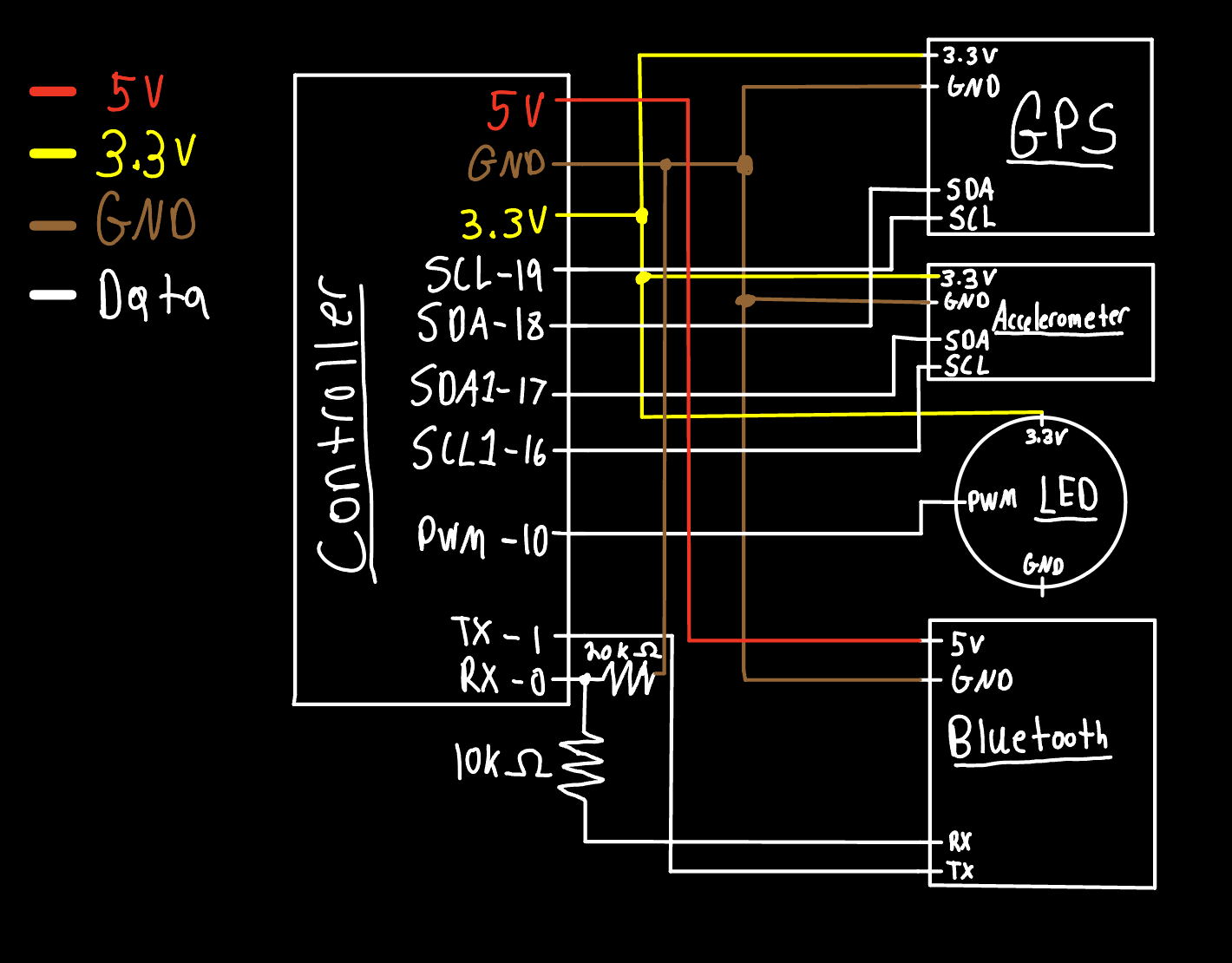
After a long period of time, it is important to check the temperature of the device. The device may get hot, especially in areas like Arizona, where this will be implemented. The battery may heat up, so users must check if the dog is uncomfortable by feeling the bottom of the 3D printed case. Charging the battery for long periods of time may also cause the device to heat up, so letting it cool down after a charge is advised. Additionally, although we attempted to make the device waterproof, it is best to prevent the device from being in water for long periods of time as water may get in and fry the system.

# **4 Technical Background**

4.1 Hardware Components

*Sensors:*

The hardware consists of several sensors including a GPS module to track live position, time, and elevation. We utilized the *SparkFun\_u-blox\_GNSS\_v3* Arduino library for its configuration. An accelerometer to track when the dog is sitting, standing, or laying down using the *Adafruit\_ADXL345\_U* Arduino library. An LED configured with the *Adafruit\_NeoPixel* Arduino library to easily find the harness or canine wearing the harness at night or in dark settings. A Bluetooth module with the *Wire*, and *Adafruit\_Sensor* Arduino library\ies for wireless connectivity with our app (*see more information on the application in 4.2*). In addition,, a lithium-ion battery is wired to both a USB-C charging port and our microcontroller to charge and power everything.



*Circuit Board:*

All of these devices will be connected to a Teensy microcontroller to read and store the sensor data, as well as wirelessly send the data to our app via Bluetooth. To store the sensor data there is a microSD card installed in our Teensy for persistent storage. Every sensor module will be connected to the Teensy on our custom-designed PCB board, ensuring a compact and efficient footprint.

*Design:*

The single PCB board containing all of the sensors and microcontroller will be placed in our custom-designed 3D printed case. This case has a cut-out for the GPS antenna, a cut-out for the battery charging port, and a hole for the LED to shine through. Each of these cut-outs will have the remaining space filled with silicone sealant to keep the hardware components inside the case safe from any dust or water. Finally, the case's lid will also be glued shut with silicone sealant to ensure the entire case is waterproof.

4.2 Software Component

*Flutter/SARTopo:*

The software for our project is consolidated into a versatile application, using Flutter and the Dart programming language, compatible with any smartphone. Upon launching the app, users are greeted with an intuitive graphical user interface (GUI), granting seamless access to all features. Once the device is paired and connected to our application, users can begin receiving and sending data between the device and the app. The Live Tracking Page allows users to enter the device information, to then be stored in a list. It’s crucial that this information matches the live locator information in the SARTopo account. Once the checkbox and Start Tracking button are selected, latitude and longitude coordinates will be sent to SARTopo via their HTTP API. The API is: '<https://caltopo.com/api/v1/position/report/$callSign?id=$userId&lat=$lat&lng=$lon>’. The call sign and user ID are the device prefix and device ID entered by the user in our app. Every time a new coordinate is received, it is sent to the live locator on their SARTopo account. The position of the dog is also displayed with a notification once the dog sits down for a couple of seconds. Users can also control the LEDs with simple on and off buttons.

Once a search is completed, users have the option to upload the GPS data to our app, which can convert it to a GPX file to be saved within the phone. The conversion happens behind the scenes using the xml builder package with specific formatting required by SARTopo.

The flutter packages which deal with GeoFences are all deprecated, so we created our own algorithm to parse through the GeoJSON file and check whether the coordinates were within the GeoFence. It’s important to note that this feature is not completed and the goal would be to notify the user when the dog leaves the GeoFence.

For the connection between the bluetooth module and our app, the *flutter\_bluetooth\_serial* package was used. This allows the user to select the module when discovering paired devices and begin receiving data seamlessly. When starting the app for the first time, all the permissions regarding location access must be accepted or the app will close. Since our app was built on Flutter, we were limited to Flutter packages and all constraints within the programming language.

# **5 Engineering Standards**

In order to ensure that our project meets the standards set by the Institute of Electrical and Electronics Engineers (IEEE), we had to follow several guidelines and procedures throughout the development process. IEEE standards provide guidelines and best practices for the design, testing, and implementation of various technologies, ensuring that they are safe, reliable and interoperable. We tried to adhere to standards throughout our process, including Consumer Electronics, Batteries, Software and Systems, Computer Technology, and more.

The following describes specific components of our project that adhere to IEEE standards:

* When planning to use Lithium ion batteries, we ensured that they weren’t volatile and could be safely incorporated into our design.
* We ensured that the power used from the Teensy to LEDs and our radio did not exceed standards set by the National Electrical Safety Code.
* All software libraries imported for our application were from verified sources.
* Designed the most feasible minimalistic and ergonomic enclosure to be secured on the harness to lessen discomfort and risk of injury
* The application is suitable to be used by people of all ages.
* We verified that any leaked voltage wouldn’t exceed the threshold set for consumer electronics.
* Made sure to use PCBs that were compatible with IEEE manufacturing standards

# **6 Cost Breakdown**

| Project Costs for Production of Beta Version | | | | |
| --- | --- | --- | --- | --- |
| **Item** | **Quantity** | **Description** | **Unit Cost** | **Extended Cost** |
| 1 | 2 | Teensy 4.1 | $35.44 | $70.88 |
| 2 | 2 | ADXL345 3-Axis Accelerometer | $6.49 | $12.98 |
| 3 | 2 | Sparkfun Max-M10S GPS Modules | $44.95 | $89.90 |
| 4 | 4 | Adafruit RGB LEDs | $1.98 | $7.95 |
| 5 | 2 | HC-05 Bluetooth Modules | $3.80 | $7.60 |
| 6 | 2 | 1000 mA Li-ion Batteries | $15 | $30 |
| 7 | 2 | SMA Antennas | $17.99 | $35.99 |
| 8 | 1 | PCB + wires + pins | $25.4 | $25.4 |
| 9 | 2 | Lithium Battery Charger Module | $0.90 | $1.80 |
| Beta Version-Total Cost | | | | $282.50 |

*Table 5.1 Cost Breakdown*

The main expenses of the project include the hardware. The breakdown above is for the two prototypes that we built, so individually it would cost $141.25 for all the parts. Most of our budget was spent trying out and finding the correct components to use. The software was free using a free SARTopo account and flutter to create our application.

# **7 Appendices**

## Appendix A - Specifications

| **Requirements** | **Value, Range, Tolerance, Units** |
| --- | --- |
| Power | 0.825 mW (all sensors + controller) |
| Bluetooth range | 200m |
| GPS accuracy | Position +2 m, -2 m |
| Battery life | 12 hours |
| File transfer rate (.gpx) | 100 MB/6.4 min |
| Waterproof case | Submerged in 3 m of water for 5 min |
| Enclosure Dimensions | 3.5in x 3in x 1in |

## Appendix B – Team Information

Ragib Ahsan

Worked on overall design of the system. Graduating senior and future Solutions Architect with Amazon Web Services.

Chris Alonzo

Worked primarily on the Flutter application and connection between software and hardware. Graduating senior and future Quality Engineer with Dassault Systemes.

Aanya Kutty

Worked on overall design of the system. Graduation senior and future Technical Sales Engineer with Texas Instruments.

Ayrton Reulet

Worked on integrating all the hardware devices and connection between software and hardware. Graduating senior and future Embedded Software Engineer with Keysigh Technologies.