



## Contents

Revision History.....	0
1 Introduction.....	3
1.1 Project statement .....	3
1.2 Purpose .....	3
1.3 Goals.....	3
2 Deliverables .....	3
3 Design .....	4
3.1 System specifications.....	4
3.1.1 Mobile/Web Application: .....	4
3.1.2 Hardware:.....	5
3.2 PROPOSED DESIGN/METHOD .....	5
3.2.1 Previously Considered Methods .....	5
3.2.2 The Current Method .....	6
3.3 DESIGN ANALYSIS.....	6
3.3.1 The Application .....	6
3.3.2 The Database .....	7
3.3.3 The Device .....	9
4 Testing/Development .....	11
4.1 INTERFACE specifications.....	11
4.2 Hardware/software .....	11
4.3 Process .....	12
4.3.1 Development.....	12
4.3.2 Testing.....	12
5 Results.....	12
6 Conclusions .....	12
7 References .....	13
8 Appendices .....	14

## Table of Figures

Figure 1 - Main Page of the Application .....	6
-----------------------------------------------	---

Figure 2 – Screen for Jobsite .....	7
Figure 3 – Database Layout .....	8
Figure 4 – Device Block Diagram.....	9
Figure 5 – Power Cutoff Circuit.....	10
Figure 6 – High Level System Diagram .....	11

## Table of Tables

Table 1 – Development Tools .....	12
-----------------------------------	----

# 1 Introduction

## 1.1 PROJECT STATEMENT

This project is about developing a tracking device that can be inconspicuously attached to a traffic sign, and can communicate information about its location with a server over a long range network. It is also the development of a mobile app/web application that can communicate with the server to pinpoint where each tracking device is located.

## 1.2 PURPOSE

The main purpose of the project is to locate lost or stolen road construction signs. This project could increase safety in construction zones because there would be appropriate signage to make drivers aware of workers. This tracker will also reduce replacement costs of road signs for our customer. Our customer is losing two to three road signs a month at about \$300 per sign. This tracker could save thousands each year by allowing them to locate these missing signs.

## 1.3 GOALS

- Customer satisfaction
- Having a small enough device that can remain hidden
- A device that can withstand harsh weather
- A device that is battery powered and can last for around 9 months
- Implement a server and a database that can store location information and be able to receive and send that information to the mobile/web application
- Implement a mobile/web application that will integrate with the Google API to show a map where the trackers are located based on the information received from the server.

# 2 Deliverables

- The Device
  - Microcontroller
  - GPS chip
  - Cellular chip
  - Battery
- The Database
- The Web/Mobile Application

## 3 Design

### 3.1 SYSTEM SPECIFICATIONS

#### 3.1.1 Mobile/Web Application:

##### 3.1.1.1 Non-functional Requirements

- The application must be mobile friendly.
- The application shall always run using a secure connection.
- The application must finish the login procedures within 20 seconds.
- The application must finish the logout procedures within 20 seconds.
- The application will use the AngularJS framework to allow the application to be contained on one page while providing smooth transitions between interfaces.
- The application must abide by the laws and regulations of applicable jurisdictions.
- The application shall not discriminate against race.

##### 3.1.1.2 Functional Requirements

- While the user is not logged in, the application shall display a login prompt.
- When the user provides valid login credentials, the application shall create a new session and enable access to the main displays.
- When the user clicks logout, the application shall clear the session and return to the login prompt.
- When the user opens the main screen the application shall display warehouses and job sites the user has permission to view.
- When the user clicks on a specific job site or warehouse, the application will display a pop up window with specific information pertaining to that job site or warehouse.
- When the user double clicks on a specific job site or warehouse the application will zoom in on the location displaying all of the tracking devices associated with that workstation.
- When the user clicks on a specific tracking device the application will display that device's ID, coordinates, battery life, workstation ID, and last time this device sent its information in a small pop-up window.
- When the user clicks on "Add Job Site" the appropriate menu for adding a JobSite will be displayed.
- When the user clicks on "Add Warehouse" the appropriate menu for adding a Warehouse will be displayed.
- When the user clicks on "Add Tracking Device" the appropriate menu for adding a device will be displayed.
- When the user clicks on "Remove Job Site" the job site will be removed and the devices that were associated with that job site will be moved to the no association category.

- When the user clicks on “Remove Warehouse” the warehouse will be removed and the devices that were associated with that warehouse will be moved to the no association category.
- When the user clicks on “Remove Tracking Device” the appropriate menu for removing a device will be displayed.
- The application must allow only one user to be logged in per application interface.
- The application must allow login with a valid encrypted username and password.

### 3.1.2 Hardware:

#### 3.1.2.1 Non-functional Requirements

- The tracking device shall use TCP to communicate reliably with the servers.
- The charge of the battery must be able to withstand usage of up to 9 months.
- The battery itself must be able to withstand harsh environments.
- The location reported shall be accurate within 5 meters.

#### 3.1.2.2 Functional Requirements:

- The tracking device shall draw less than 15 $\mu$ A when in sleep mode.
- The tracking device must report its location at least 2 times a week to database server, ideally 5 times a week.
- The tracking device shall switch to sleep mode and power down all peripherals after successfully transmitting its location.
- All components shall use 3.3V logic.
- The tracking device shall use raw GPS and voltage data to send to the database server along with the devices identifier.

## 3.2 PROPOSED DESIGN/METHOD

### 3.2.1 Previously Considered Methods

We have considered a few different methods to achieve our goals. One method we discussed was using satellite ping to directly send data about our device’s location to the database, however this had the potential of being much more expensive. Using this method, we would have to pay for each transmission of data from the satellite and it would cause more battery drain than our accepted method. We also looked at determining position using cell tower triangulation rather than GPS. However, like the previously mentioned satellite method, this would have caused poor battery performance, making it more difficult to achieve our goal of a nine month battery life. In the end, we decided to follow a method that will acquire latitude and longitude data via GPS. The device will then send this information to the database over the cell network to the application server. This utilizes less power

than the cell tower triangulation and will cost less than the satellite directly transmitting data to our database.

### 3.2.2 The Current Method

We have decided to follow a method that will acquire latitude and longitude data via a GPS chip that the tracking device will then send to the database over the cell network via the application server. This utilizes less power than using cell tower triangulation to find location and will cost less than the satellite directly transmitting data to our database, as that requires payment for every data transmission. Once the information is on the database, the web/mobile application will be able to pull the information from the database to display the information to the customer.

## 3.3 DESIGN ANALYSIS

### 3.3.1 The Application

In order to access the application, the user must first login by providing valid credentials. Once the user is logged into the system, they will be presented with the main screen of the application.

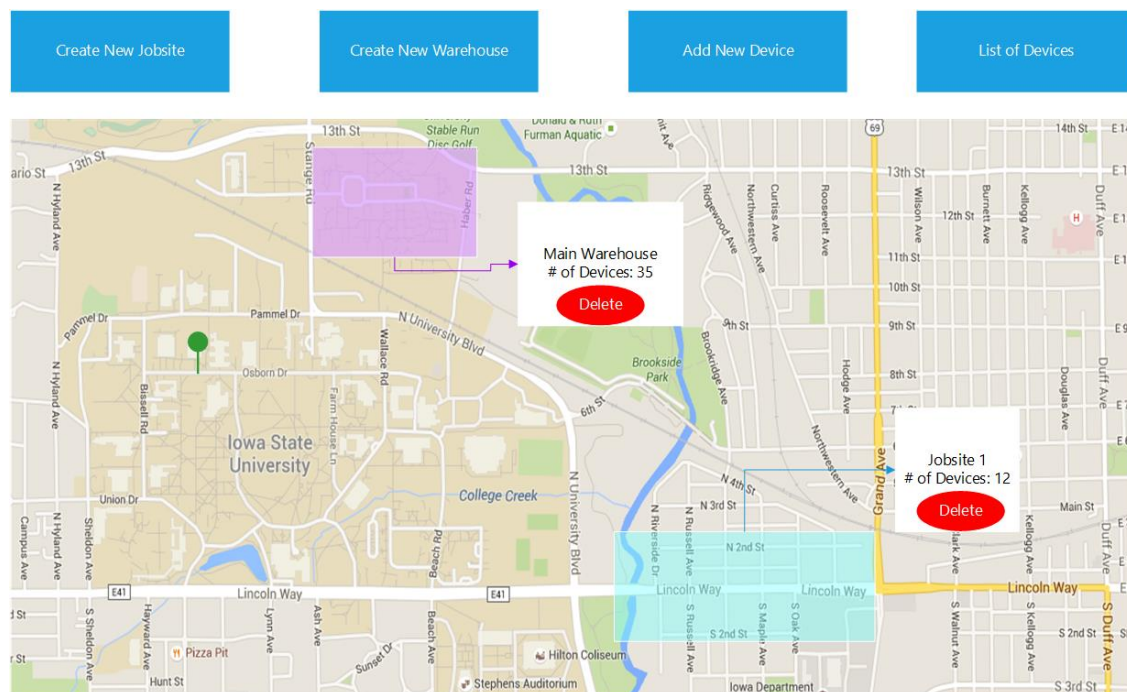


Figure 1 – Main Page of the Application

On this screen, the user will be presented with three buttons on the top of the screen. Two of these buttons are there to add a new job site and a new warehouse. Clicking on one of these buttons will bring up a popup menu that will give the

options of setting up a new job site or warehouse. The third button is the “List of Devices” button. When the user clicks on this button, a popup list of all of the devices will show. Clicking on one of the devices in the list will present the user with a new screen that is zoomed in on that particular device’s location and display that device’s information. The other information on this main screen will display the already created Job sites and warehouses as well as any devices that are not associated with either a job site or warehouse. Clicking on one of the job sites or warehouses will bring up a popup menu that will give the user information about that particular job site or warehouse, as well as give the user the ability to delete that job site or warehouse. Double clicking on one of the job sites or warehouses will bring the user to the screen for the particular job site or warehouse.

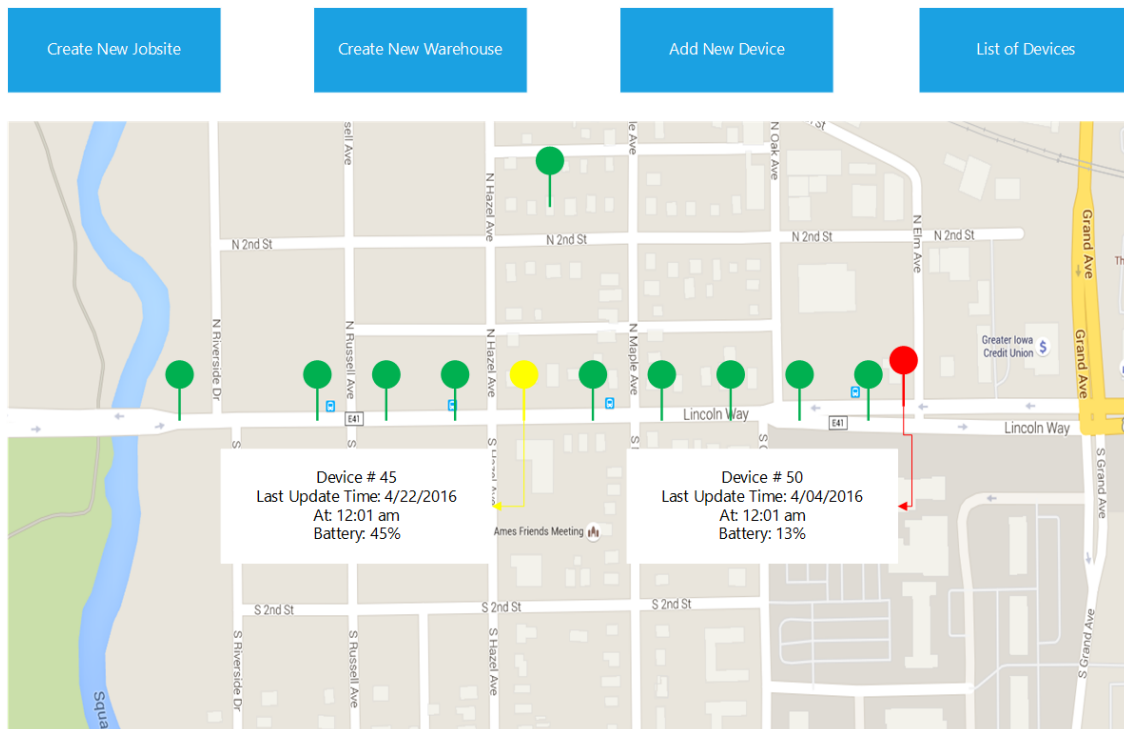


Figure 2 – Screen for Jobsite

In this screen the user will be presented with the same three buttons that were discussed above. In addition to those three buttons there be the location of devices associated with that job site. When the user clicks on a device, it will bring up a popup menu showing the user the information for the specified device.

### 3.3.2 The Database

This project will have three different main tables in our database. They are DeviceInfo, Sites, and UserInfo. There is also one more table called SiteTypes that contains the different types of sites that can be stored in the Sites table. The DeviceInfo table will hold the information for each device. Each device will be associated with a Site. There are three different types of Sites: Warehouse, JobSite,



and NoAssociation. All of the devices that are not associated with either a Warehouse or a JobSite will be automatically associated with NoAssociation. The UserInfo table will hold the information for valid users. This table will control the login functionality of the application.

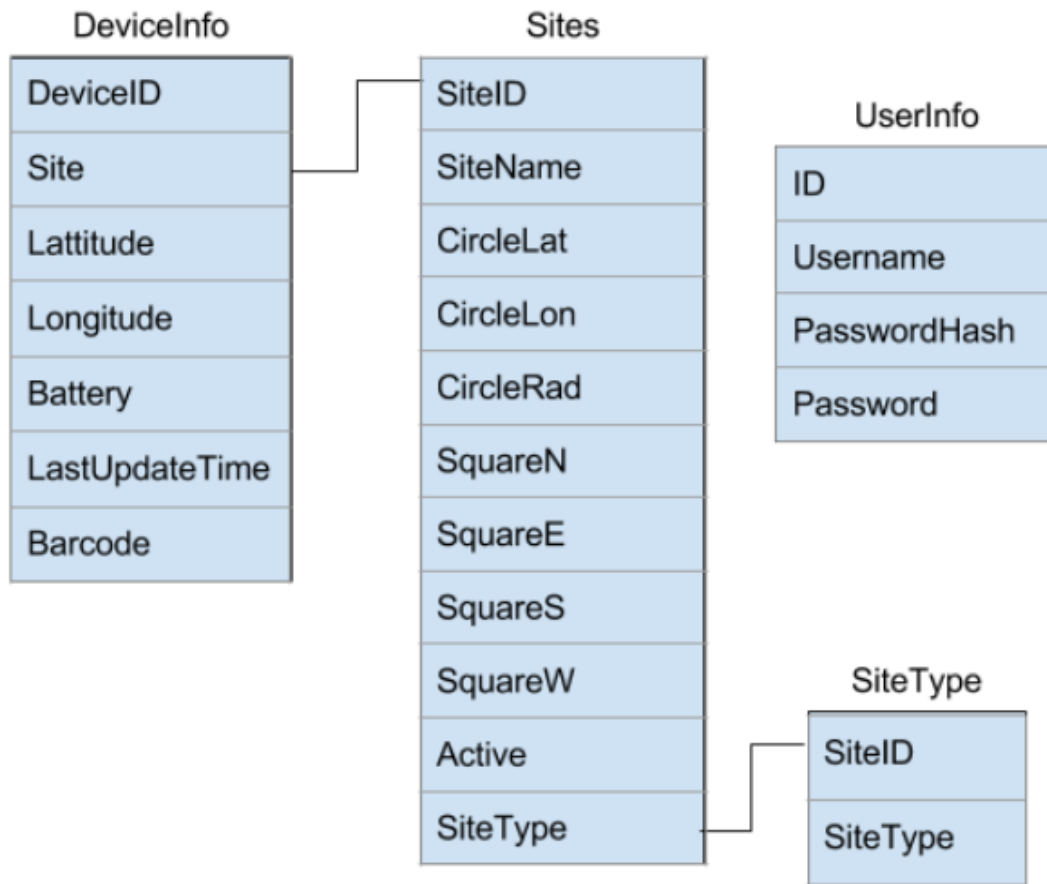


Figure 3 - Database Layout

FIGURE 3: OVERVIEW OF DATABASE

### 3.3.3 The Device

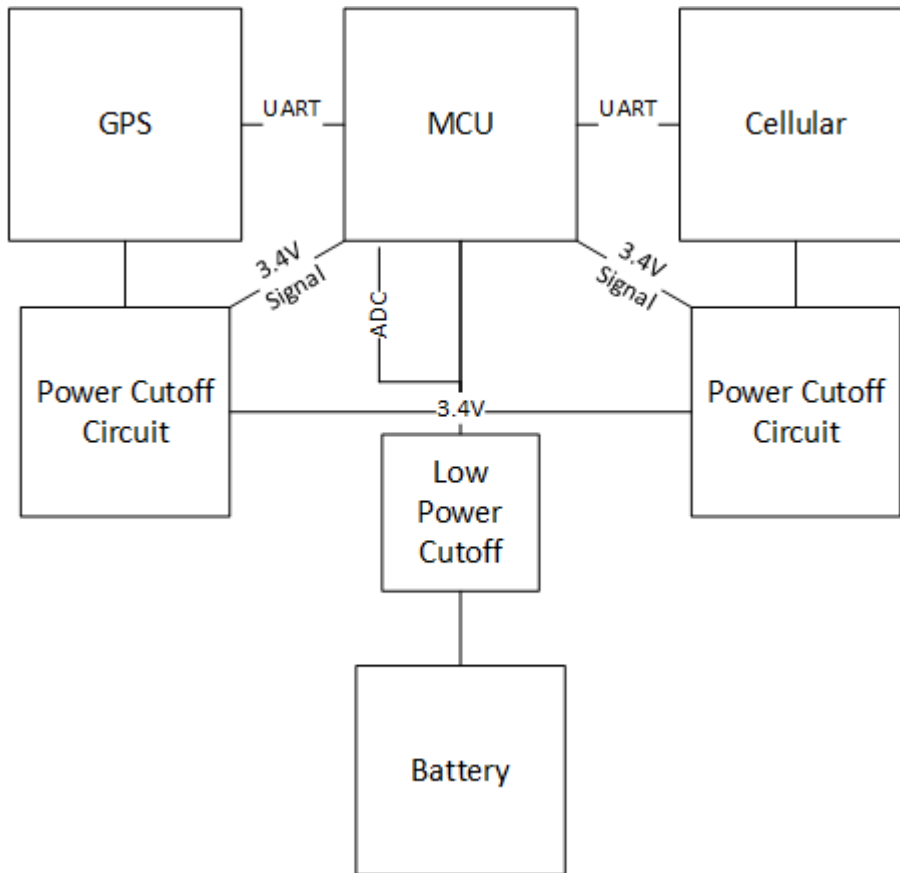
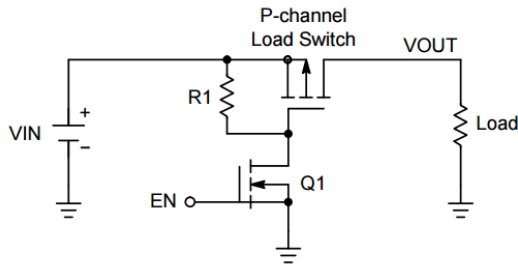


Figure 4 - Device Block Diagram

The device will consist of three primary modules: GPS, MCU, and Cellular. The device will have secondary functions controlled by the MCU to test the charge of the battery and to control power to the other two modules when they are not in use. Being able to test the charge of the battery is very important as we plan to use a lithium nickel manganese cobalt oxide cell (NMC), which can be damaged from excessive drain. The charge of the battery will also be reported with the location information so our web application can inform the user when batteries need to be changed. We also need to conserve power as much as possible, so we will cut power to the GPS and Cellular after they have shutdown to prevent them from drawing current in their off state, allowing the device to run for longer periods of time between battery changes. The circuit diagram in [figure 5](#) shows how we will

accomplish cutting power from the GPS and cellular chips.



*Figure 5 - Power Cutoff Circuit*

The device will primarily operate in a low power mode with only one timer running on the MCU and will wake up on set intervals. When the device is in its wake state the GPS and Cellular chip will be given power and will boot. The GPS takes about 30 seconds from power on before it returns valid location data. Once the MCU reads in a set of good location data, it will store the information and turn off the GPS. While the GPS is getting location information, the Cellular chip will be acquiring network connection. This has been tested and, in most cases, will take longer than acquiring location, so the MCU will have to wait for the cellular chip to indicate it is ready. Once the cellular chip is ready to transmit, the MCU will pass the location, battery voltage, and its unique identifier to the cellular chip to be transmitted to our application server where it will be converted to a format usable by our web application.

## 4 Testing/Development

### 4.1 INTERFACE SPECIFICATIONS

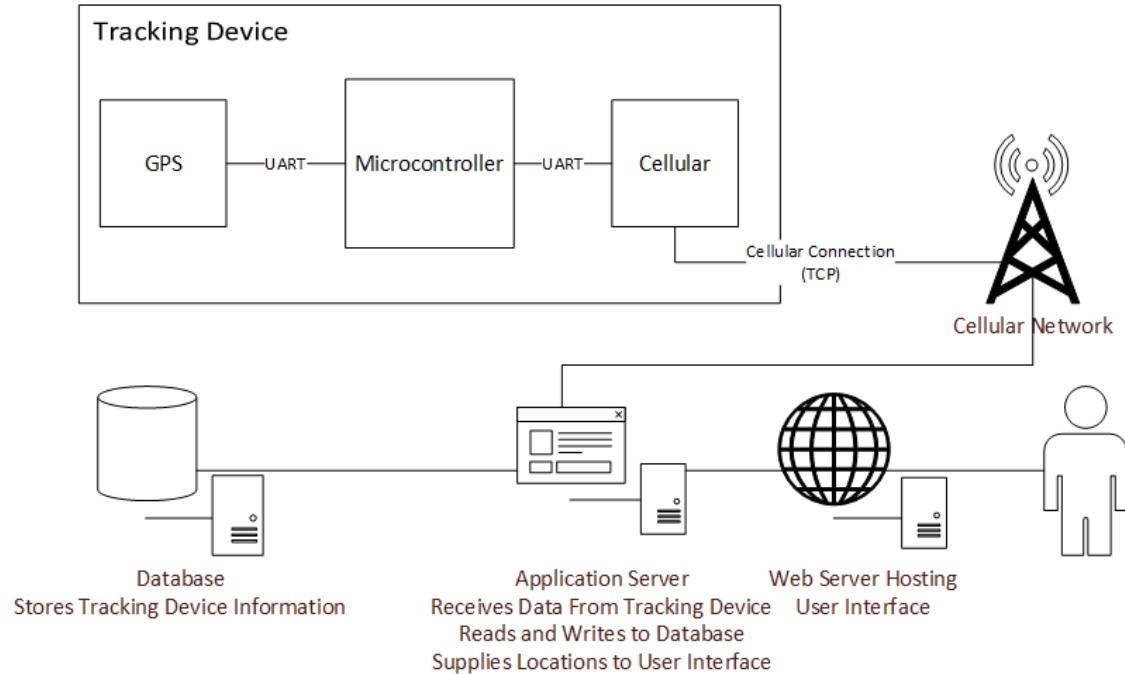


Figure 6 – High Level System Diagram

As shown in the diagram above, our tracking device will be using UART (Asynchronous Serial Communication) to interface with all of its different components. The tracking device will then interface with the application server via the cellular network. The application server will be running background code that will convert the data received from the tracking device, then store it on the server database. The application server will also be running any php code that we would use to have the web server pull data from the server database. The web server will be where the web/mobile application will be hosted.

### 4.2 HARDWARE/SOFTWARE

For development we will be using the following tools:

Tool	Reason
phpMyAdmin	Setting up and managing SQL database
Code Composer Studio 6.1.2	Coding and Debugging on MSP430
Chrome and Firefox Debugging Tools	To test the web application
Ti Launch Pad	Includes hardware Debugger for

	MSP430
Konekt Dash	Developing cellular communication

Table 1 – Development Tools

## 4.3 PROCESS

### 4.3.1 Development

The process that we will be following to create this product is a loose version of Agile. We will have weekly team meetings to report on the progress that we made over the past week as well as to come up with new goals for the next week.

### 4.3.2 Testing

Because we are using the Agile format for development, we will be doing iterative testing of both the application and hardware as we complete pieces of functionality. This will allow us to find problems early and address them while we are still in the development process. This will prevent small problems from becoming large issues near the end of the development cycle.

## 5 Results

We have been able to retrieve location information from the GPS and report the data to a console via UART. We were also able to send a message from UART to our cellular development kit that relayed the data to a provided service, proving that we could communicate information from our device.

## 6 Conclusions

We have our component parts and have tested each individually. We will continue to refine and move forward with code to interact with the hardware directly, as well as interfacing with the server, database, and web application. These will need to be ready for testing when we have the full prototype device built. Our main goals for this project are to create a tracker that is reasonably small to remain inconspicuous while attached to a road sign, make each tracker cheap enough to be worth it to use one to protect a \$300 sign, utilize the cell network to transmit location data, create a web application to mark locations of each device on a Google Map, and allow each device to operate for preferably nine months before battery must be recharged.

## 7 References

- [1] Google Developers. (2016). *Google Maps APIs* [Online]. Available:  
<https://developers.google.com/maps/>
- [2] Information Sciences Institute University of California. (1981). *Internet Protocol  
Darpa Internet Program Protocol Specification* [Online]. Available:  
<https://tools.ietf.org/html/rfc791>

## 8 Appendices