

# Pathfindr User Manual

Eric Chen 300136076

Brendan Sommers 300115531

Adam Walters 300109768

Jarrett Goodwin 300074553

Tianchen Cai 300127732

GNG 2101 Group B32

Professor: Mana Azarm

TA: Samuel Ezekiel

PM: Madeline Kyne

Submission Date: December 10, 2020

## Abstract

*This project was created to assist the librarians at the Morisset Library at the University of Ottawa to enable better navigation of the library. Therefore, it is necessary for people who may not be able to read the screen at all, to use the product and interact with the environment safely. After meeting with the client, the team designed a product that involves physical beacons which will be placed at points of interest, and a phone app that allows users to communicate with these beacons in order to reach their destination. This report outlines how to maintain and correct any issues that the user may encounter when using the product and how to use the device safely. Finally, the document assesses the final product and cites additional files that were used to develop said final product which will enable others to continue to develop this product in the future.*

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Functional Decomposition</b>	<b>1</b>
2.1	App . . . . .	1
2.2	Beacon . . . . .	1
<b>3</b>	<b>Development &amp; How it Works</b>	<b>2</b>
3.1	Target Specifications . . . . .	2
3.2	App . . . . .	3
3.3	Beacon . . . . .	3
<b>4</b>	<b>Instructions</b>	<b>4</b>
4.1	Installation . . . . .	4
4.1.1	App . . . . .	4
4.1.2	Beacon . . . . .	5
4.2	Operation . . . . .	7
4.2.1	App . . . . .	7
4.2.2	Beacon . . . . .	7
4.3	Maintenance . . . . .	8
4.3.1	App . . . . .	8
4.3.2	Beacon . . . . .	8
<b>5</b>	<b>Health and Safety</b>	<b>8</b>
<b>6</b>	<b>Conclusions and Recommendations for Future Work</b>	<b>8</b>
	<b>References</b>	<b>10</b>
<b>A</b>	<b>Appendix</b>	<b>11</b>
A.1	Bill of Materials . . . . .	11

## List of Figures

1	ESP-Wroom 32 Microcontroller Pin Mapping (1) . . . . .	4
2	Photo of the soldered protoboard placed in every beacon. . . . .	5
3	Diagram of Protoboard assembled in Figure 2 created in Circuitmaker . . . . .	6
4	Exploded view from drawing of beacon assembly made in Solidworks . . . . .	7

## List of Tables

1	Metrics and Target Specifications . . . . .	2
2	Bill of Materials . . . . .	11

## Glossary

**Arduino IDE** A compiler used to develop and upload coded instructions to a microcontroller.

**Breadboard** A base used to develop prototypes for circuits which will later be soldered permanently on a protoboard.

**Protoboard** A permanent board which can have through-hole components soldered to it and transmits electrical signals.

**Received Signal Strength Indicator** The strength received by the device from a Bluetooth transmitter represented as a number from 0 to -99.

**Universally Unique Identifier** A name for which can be used to identify information which is collected in a computer. This is similar to postal code for your house. Another name for this is a **Globally Unique Identifier**.

## List of Acronyms

**BLE** Bluetooth Low Energy. , 3

**CAD** Computer Aided Design. , 5

**LED** Light Emitting Diode. , 1, 3, 5

**PLA** Polylactic Acid. , 2

**RSSI** Received Signal Strength Indicator. , 3

**UUID** Universally Unique Identifier. , 3

# 1 Introduction

Visually impaired people struggle every day to navigate the world around them. The systems currently in use focus on going from point A to Point B, in a safe manner. These systems can be a very slow or very niche in their usage. One example is a seeing-eye dog, which can only get you safely to where you want to go but does not provide directions there. Another example is a walking cane, which has the same function as a seeing-eye dog. Pathfindr is a system that allows a user to independently navigate to the desired location within an indoor space, safely, efficiently, and with constant interactive feedback on their current location. Once arriving at the chosen location, the app will prompt you with useful information, and announcements. The basic user requirement is a device with Bluetooth capabilities, and a building with the system installed. The system can also be used by non-impaired people, to assist with navigation and accessing of announcements. Pathfindr differs from other indoor navigation systems, as it is much simpler and user-based. There are three main components to the app; navigation, announcements and settings. Each one is very user friendly and designed to be navigated visually impaired. The navigation portion allows the user to select a location to navigate to from a simple and short list.

## 2 Functional Decomposition

### 2.1 App

The Pathfindr app is an Android app with five distinct components. When opening the app, the first screen is the Home Screen, containing a short introduction to the app and buttons to go to the other parts of the app. Also on this screen is a Help button, which will redirect the user to this User Manual. Second is the Announcements screen, which displays a list of announcements sorted in reverse chronological order, otherwise known as most recent first. From this list, the user can tap on any announcement to view the details of that announcement, which includes the title, contents and a timestamp. The third screen is the Navigation screen, which is the primary function of this app. The initial Navigate screen is the Destination Select screen, which is similar to the fourth screen, Location Info, where the user is shown a list of all possible destinations and can open a page displaying basic information about each location such as the name, a description and a website. However, on the Navigate screen, there is also a "Navigate" button which will start navigation to the selected location.

The actual Navigation screen consists of a progress bar at the top of the screen, a summary of the main instructions in both text and icon form taking up the middle of the screen, the destination below that, and then three buttons at the bottom. The topmost button, 'View Instructions as Text', will display the verbally announced verbose form of the announcements as text in a dialog for users who are hard of hearing or who do not have headphones. The middle button, 'Repeat Verbal Instructions', will repeat the verbal instructions played at the beginning of each instruction in case the user did not catch it the first time. Finally, the 'Exit Navigation' button will stop the navigation process and return the user to the main Destination Selection screen.

As discussed previously, the Location Information screen allows the user to see a list of all available navigation destinations and view more information about each. The final screen is the Settings screen, where any user-changeable aspects of the app will be stored. Additionally, there is a navigation bar always present at the bottom of the app screen that allows the user to switch between all of the screens at any time.

### 2.2 Beacon

Pathfindr uses a series of Bluetooth capable beacons to map out a location and its key location points. Each beacon contains a Bluetooth transmitter, a micro-controller, four AA batteries, one **LED**, one **330Ω** resistor and the necessary wiring within a 3D-printed case. A Bluetooth signal is created and transmitted by an ESP-32 microcontroller powered by 3 AA rechargeable batteries. These batteries produce 4.5V of power. There is an **LED** which can be seen from outside the case that indicate the battery level of the beacon..

Finally, the 3D printed case is made of [PLA](#) and has four mounting screws that should be inserted into the wall as well as slits on the side of the case that enable a greater transmission distance. All electrical components should fit comfortably inside the case.

## 3 Development & How it Works

### 3.1 Target Specifications

After benchmarking versus two different systems which perform similar tasks to Pathfindr, the team generated a list of metrics and target specifications enabled the team to have something to measure the effectiveness of prototypes against. These target specifications aimed to take the benefits of both the Key2Access system and the Wayfindr system but also worked within the constraints of the budget laid out at the beginning of the project. The values tabulated are values that the team thought were feasible when considering the needs of the client and their customers.

Importance	Metric	Ideal Value	Units
5	Time from App Start to Navigation	$x \leq 15$	Time (s)
5	Customer Satisfaction	$x \geq \frac{5}{5}$	Subjective
5	Effective Range of Beacon	$x \geq 6$	Distance (m)
5	Battery Life	$x \geq 6$	Months
4	Time for Reconfiguration	$x \leq 420$	Time (s)
4	Cost per Beacon	$x \leq 30$	CAD\$
4	Weight of Beacon	$x \leq 500$	Weight (g)
4	Beacon Size	$x \leq 10$	Largest Dimension (cm)
4	Effectiveness of Notification	$x \geq \frac{5}{5}$	Subjective
3	Time to Broadcast Announcements	$x \leq 5000$	Time (ms)

Table 1: Metrics and Target Specifications

First, the team considered the speed of the application to be extremely important. When using the Key2Access system, the app was extremely slow and thus became difficult to use. The belief was: the slower the app is the more difficult and confusing it would be to use. Next, the range of the beacon was also considered to be of crucial importance, since it would be extremely expensive to place beacons every couple of metres. Additionally, a greater range would allow for more accurate triangulation to occur. Similarly, the battery life of the beacon had to be of usable length. If the batteries needed to be replaced every week then the system becomes less usable for the clients (the librarians). Other important metrics measured in time, the time to reprogram the beacon locations and broadcast announcements. For example, if the client wanted to move where the beacons were in their space then this should be relatively simple for them to do. In addition to this, the beacons should be able to send important announcement to their user's phones with a short delay. Finally, the cost of the beacons should be relatively low. This is a consequence of the budget the team had available to them, but also the fact that the team anticipated that the library would not be eager to spend a significant amount of money.



## 3.2 App

Before beginning development of the app, the team had to select the best tools available to make sure the development process would be as efficient as possible. For the development of the app itself, the team elected to use Android Studio as it is the official tool for developing Android apps as suggested by Google. To keep track of the code and its many versions, both Git and GitHub were used - the public repository containing the code for this app is available at <https://github.com/ricetech/pathfindr>.

To store the data required for the operation of the app (namely the lists of Announcements, Locations and Paths), the team opted to use Firebase Cloud Firestore - an online database which allows all installations of the app to have access to the same set of location, path and announcement data, all of which is synced live to each device. This provides the additional benefit of allowing updates to location data (for example, changing the website associated to a location) without having to send out a new version of the app since it would only require a modification to the online database.

A side benefit of using Firebase is that it allows access to a wide variety of powerful tools such as Firebase App Distribution, which allows the team to distribute app prototypes efficiently, as well as Firebase Crashlytics, which provides quick feedback from app crashes so that the team can quickly rectify bugs without waiting for reports to come in from app users.

The actual development process of the app began with searching for the libraries necessary to make the app work as intended. The team discovered the [AltBeacon Android Beacon Library](#), which was compatible with the iBeacon format that our beacons broadcasted. Additional libraries used included the Android libraries for triggering text-to-speech and the phone vibration.

## 3.3 Beacon

To develop the beacon, the team first had to determine what would be needed inside each one. Each of the components corresponds to a specific and crucial function which enables the use of the application. First, the ESP32 micro controller which can produce and transmit [BLE](#) signal to the users' phones. This allows for the app to track the users' position and give them directions to their destination. Next, there is a protoboard which has been soldered with all the necessary components to mechanically and electrically communicate with the employees at the Morisset Library. Finally, the batteries which will power the micro controller and all the additional components that are on the protoboard will be placed inside the top case.

The first component is the Bluetooth transmitter, the group chose to use a ESP32 micro controller, as it was the most economically feasible microcontroller with [BLE](#) capabilities. Additionally, the ESP WROOM-32 has dual-core processors allowing for two programs to run simultaneously. The first step in developing the beacon was to understand the layout of the pins on the micro controller, using the diagram featured below.

Using this pinout diagram, the team would be able to send signals to the [LED](#) on the protoboard. Next, the team used several different pin numbers to assess how each pin functioned, and which would be needed to make the connections. In order to ensure the ESP32 transmits a Bluetooth signal to the Pathfindr app, an iBeacon scanning app (nRF Scanner) was used to find the [RSSI](#), the [UUID](#) and the strength of the signal. This Bluetooth signal was produced using pre-written code for iBeacon signals which can be accessed by downloading the correct ESP32 libraries. These can be accessed by opening the preferences tab in the Arduino IDE software, then inserting "https://dl.espressif.com/dl/package\_esp32\_index.json" in the Additional Board Managers URL section towards the bottom of the tab. Next, the source code for this example can be found under Examples > ESP32 BLE Arduino > BLE\_iBeacon.

This code includes a globally defined variable known as the "BEACON\_UUID". The name of this variable will be different when it is retrieved by another bluetooth device. In order to connect two devices together, one will need the [UUID](#) retrieved by the phone. In order to collect this, use the nRF Connect App on an Android phone and record the [UUID](#) it produces on the screen.

Upon completion of the ESP32 set-up, the next focus was on developing the wiring to ensure we could power the ESP32, along with have functioning status [LED](#).

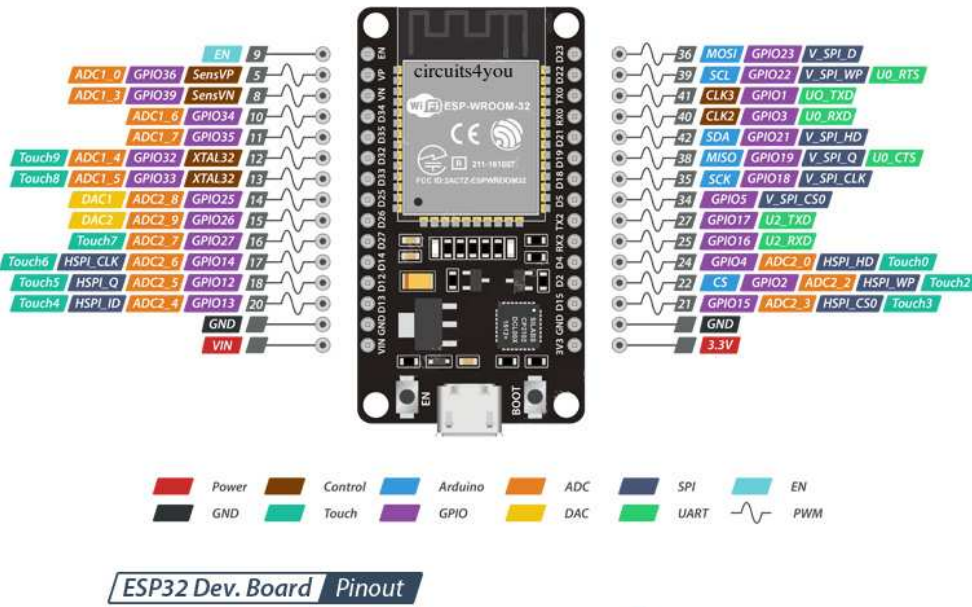


Figure 1: ESP-Wroom 32 Microcontroller Pin Mapping (1)

Finally, the team had to take the two previous developments and combine them into a single Arduino IDE code and onto a breadboard for prototyping. This can be done by assigning each of the loops to one of the two processors. This enables the loops to run in parallel and results in a far more efficient program. Following this, the microcontroller can be placed onto the protoboard without issues since the iBeacon signal does not require any additional components to function.

## 4 Instructions

### 4.1 Installation

#### 4.1.1 App

The plans for the final production version of the app include an Android app distributed via the Google Play Store, and potentially an iOS app distributed through the Apple App Store. However, for the prototype version of the app, it is only available through the Firebase App Distribution platform for Android devices. Follow these steps to install the prototype version of the app:

**IMPORTANT:** These instructions must be executed on the Android device that you wish to install the app on. Additionally, you MUST be logged into your Google Account on that device.

1. Open this link and sign up using the Google Account that you wish to use with the app: <https://appdistribution.firebase.dev/i/f9a9852aaf7a54c2>  
Note: The provided Google Account MUST be linked to the device.
2. You will receive an email from Firebase at the email which you provided in Step 1.
3. In the email, click the link. You will be taken to the registration page to test the app.
4. After completing registration, download the app from the website.
5. (Optional) Download the App Tester app so that the app can auto-update itself.

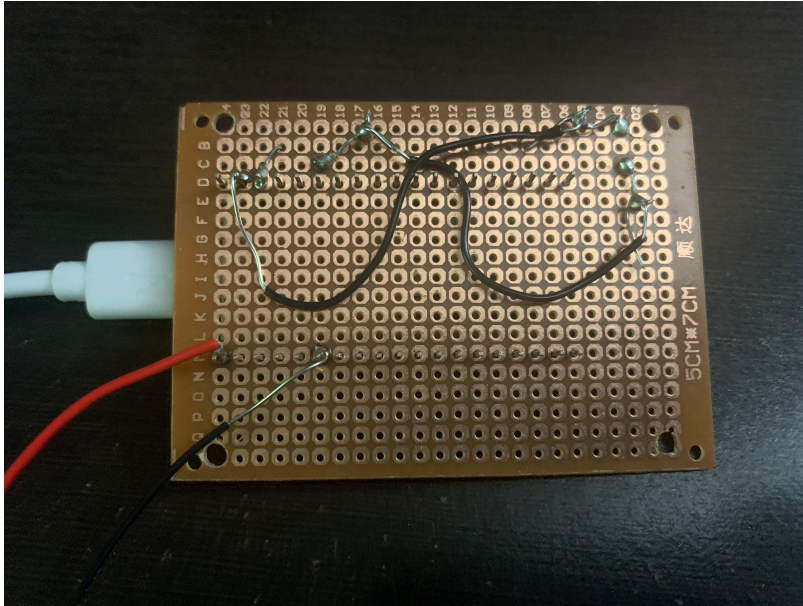


Figure 2: Photo of the soldered protoboard placed in every beacon.

6. When the download completes, open the APK file from the notification that appears in the device's notification tray.
7. You may be asked to reconfigure your device settings to allow installation of apps from unknown sources. You must do this before you will be able to install the app.  
Because each device is different, you will have to look up specific instructions for how to do this on your device. Your device may also provide a shortcut for you directly to change these settings.
8. After granting the necessary permissions, install the app using the APK file or App Tester (if downloaded in Step 5).
9. Open the 'Pathfindr' app normally like any other app installed on the device.

#### 4.1.2 Beacon

Before beginning to assemble the beacon casing, one should solder and assemble the protoboard that will be placed inside. The beacons are powered through three 1.5V AA batteries which produce a signal of 4.5V. The power from the battery pack should then be connected the 5V pin, on the left of the ESP32. Then the grounded wire should connect to the GND pin on the left. Next, in order to power the LED, connect pin 21 to the resistor which should then lead to the positive terminal of the LED. Then the negative terminal should come back to the ground pin on the right side of the ESP32. By the end the end the protoboard should look something like this:

It is possible to power this protoboard with 3V of power, or 2AA batteries. The only change needed is to change the power input from the 5V pin to the 3V3 pin, labeled 39 and 2 respectively in the figure above.

First, to assemble the beacon the components needed must be acquired from the Bill of Materials (A.1). Additional files, such as the CAD models can be acquired from the [MakeRepo repository](#). Next, the following tools will be required to assemble the beacon:

- Flathead Screwdriver
- Screwdriver (Square Interface)

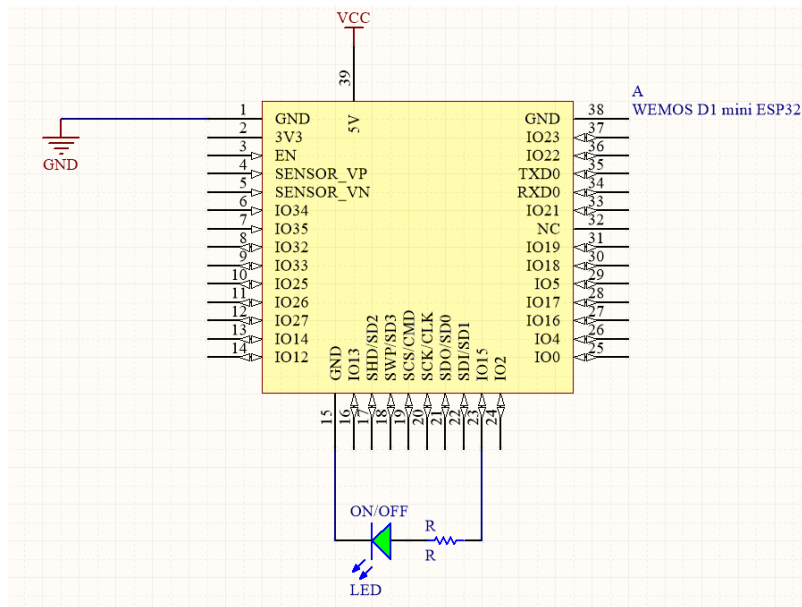


Figure 3: Diagram of Protoboard assembled in Figure 2 created in Circuitmaker

- Soldering Iron
- Small Rubber Mallet

How to assemble the beacon cases:

1. Use the soldering iron to carefully insert the heat inserts into their positions. The face of the heat inserts should be flush against the face of the base. For further instructions [This Video](#) can be used as an additional reference.
2. Use the rubber mallet to insert the hex nuts into position. These should be parallel to the face of the beacon base and do not need to be sunk into the base.
3. Insert the pan-head screws through the holes in the protoboard by screwing them into the hex nuts. There should be clearance between the surface of the base and the soldered components on the protoboard.
4. Apply the adhesive to the bottom of the battery holders making sure that the wires connecting to the protoboard do not encounter significant tension.
5. Place the cover of the beacon over the other components, and line up the screw holes. These screw holes are the same holes you inserted the heat inserts into. Insert these screws into the heat inserts until they are tight against the beacon case.

To set up the beacon portion of Pathfindr there is a multi-step process. The first step is to decide where the key locations are within the building. Once these locations are mapped out, a beacon must be placed at each point of interest, as well as placing beacons every five metres in between each location. Once the beacons are in place and turned on, go to the app and assign each beacon the the location it is associated with. To ensure safe travels between locations, make sure to minimize hazards along the main routes, and safe placement of the beacons. I.e. do not place a beacon above a stairwell, instead place it before the stairwell to give a safe warning of the upcoming location. Once completed make sure to test the system to find dead zones and other issues with the layout.

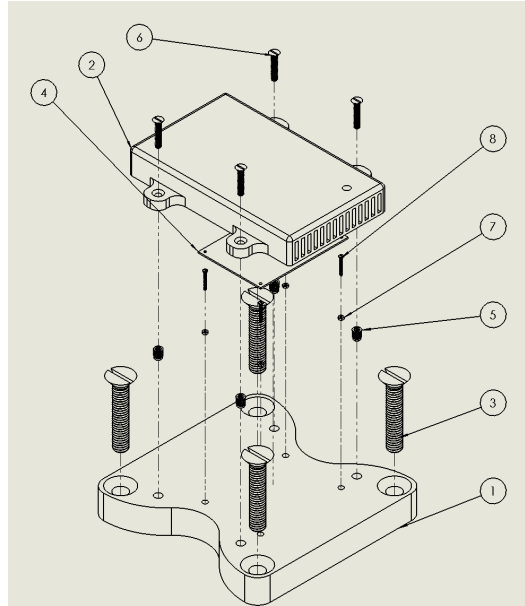


Figure 4: Exploded view from drawing of beacon assembly made in Solidworks

## 4.2 Operation

### 4.2.1 App

When the app is first opened, there are 5 main screens. To navigate between the screens there are buttons along the bottom in a navigation bar which is always visible along with large buttons on the home screen. To navigate to the announcements screen, the user will need to press the announcements button, and then they will be able to see all the relevant announcements sorted from newest to oldest. If the user wants to get more information about certain announcements they can tap on it, which will open another screen displaying the contents of the announcement. To navigate to the navigation screen, it is the same process as before. Once in the screen, the user will be prompted to choose a location to navigate to. The navigation will use audio and visual notifications to alert and direct the user to the desired location. Once the user arrives to their chosen location, the location information screen will automatically open. Similar to before to access more information about a location, the user can also open the corresponding web page. The location information screen can also be accessed using the buttons at the bottom of the screen, along with on the homepage. The last screen is the settings, where the user can adjust settings to their desired preferences. At all points throughout the app, it is always possible to press the home button and return to the very start.

### 4.2.2 Beacon

there are three steps, when install Beacon in the library. first, Determine the number of beacons according to the size of the library. and conduct on-site inspections and find places where signals are easily blocked, and add beacons here. After decide how many beacons are needed in the library; beacons need good positions to maximize its effectiveness. The main aisle should be evenly arranged according to the Bluetooth transmission range, and place more beacons in places frequently used by users. second, Install batteries for each beacon and verify that each battery is installed correctly and has power. This step can be tested by connecting with phone or use the LED on the beacon to confirm. Third, located beacons at the places where signals are easily blocked, in those shielded area make sure Signal coverage area is connected to other areas. If the goal cannot be achieved, then replace beacon until it works. After successfully dispose every beacon, the connection with the mobile terminal should be tested. when the Bluetooth network has been created successfully, the tester should choose a destination by phone randomly to test Functional integrity.

## 4.3 Maintenance

### 4.3.1 App

Once the App has been downloaded to user's mobile devices there is very little maintenance that needs to be done by user. However, the support team will regularly release updates for the product. When using the software it become stuttering or unavailable. Please restart the software, and the app will report the crash to the support team automatically. The support team will fix this bug on next update. After changing the battery or moving the position of the beacon every time, maintenance staff should open the App, and go to the setting part to relocate the beacon's location. After upload the new beacon location, the maintenance staff should return to the main page and open navigate part to make sure every path is correct and safe. It is worth noting that every change of the beacon location and the path change will be upload to the online database for later inspection.

### 4.3.2 Beacon

Once the beacons have been installed there is very little maintenance that needs to be done. The batteries last for approximately 6 days so it is advised that clients check the beacon battery led, making sure all the beacons are in proper working order to ensure maximum signal strength. When relocating the beacons it is essential that the beacon is fully powered off and the batteries have been removed. The client can then remove the beacon from its current placement by removing the 4 screws securing the beacon to the wall then simply lift the beacon and move it to the new desired location. Before securing the beacon to the new location you must update the beacons paths (see App Maintenance 4.3.1) and ensure the path will be free of all obstructions. Once the path is safe to navigate and updated properly in the apps firmware, the client can secure the beacon with the same 4 screws removed as it is explained in Beacon Installation 4.1.2.

## 5 Health and Safety

Any health and safety risk that may be associated with the Pathfindr system would stem from any user related errors. The system provides the shortest path from your position to any point of interest designated by the user, these paths will be free of any static obstructions but the user must be somewhat aware of any dynamic obstructions such as other people, trolleys or any unforeseen objects that may be left on the floor. There should not be any health and safety concerns when it comes to the beacons as they are battery powered and have enough ventilation to disperse heat in a timely manner. Additionally, risk is minimized by allowing customers to use their own mobile devices and are encouraged to maintain social distancing as with the current state of the pandemic that is essential.

## 6 Conclusions and Recommendations for Future Work

Throughout the semester our group used the engineering design process to plan and execute the development of a product. We began by empathizing with our clients and getting to know their every need. The information we gathered along with the benchmarking of similar products gave us the foundations to start coming up with design ideas. This included brainstorming sessions and lots of sketching. We then took these ideas and combined them into one idea: our first prototype. This prototype served as the starting point for all future development to grow from through client feedback and testing. Prototype two and three served the same function but were of higher fidelity and were more similar to the desired final product. Throughout this process the team learned that one must readily adapt to changing circumstances, to ensure constant forward growth. We also learned that communication is crucial to ensuring the team is able to solve all conflicts in an efficient manner. To improve on the product it is recommended to redesign the beacon to maximize the battery life. This can be achieved by changing the micro controller to one that draws less power, or to increase the size of the battery in use. Another suggestion for future work would be to re-code

the app to direct people using routes rather than to locations. This would allow more complex navigation in complex buildings.

## **References**

- [1] “Esp32 devkit esp32-wroom gpio pinout,” Dec. 2018. Last Accessed 24 November 2018.



## A Appendix

### A.1 Bill of Materials

Material	Location	Quan.	Unit Cost	Ext. Cost
3D printed case	From Makerlab	4	\$ -	\$ -
AA Battery Holder	From Makerlab	4	\$ 1.00	\$ 4.00
AA Batteries	From Makerlab	16	\$ 0.38	\$ 6.15
ESP32/ESP32S	2x Makerlab, 2x Amazon	4	\$ 9.50	\$ 37.98
Protoboards	Makerlab	4	\$ 2.00	\$ 8.00
6-32 Screws	Home Depot	16	\$ 0.12	\$ 1.92
#6 Heat Inserts	McMaster-Carr	16	\$ 0.17	\$ 2.72
#4 Screws	Home Depot	16	\$ 0.46	\$ 7.36
#4 Hex Nuts	Home Depot	16	\$ 0.27	\$ 4.32
Green LEDs	Makerlab	4	\$ 0.03	\$ 0.12
Red LEDs	Makerlab	4	\$ 0.03	\$ 0.12
Wire	Makerlab	4	\$ 0.11	\$ 0.44
Resistors	Makerlab	8	\$ 0.08	\$ 0.64
<b>Shipping</b>			<b>\$ -</b>	<b>\$ -</b>
<b>Subtotal</b>				<b>\$ 73.77</b>
<b>Taxes</b>				<b>\$ 9.59</b>
<b>Grand Total</b>				<b>\$ 83.36</b>

Table 2: Bill of Materials