Indoor Navigation for the Blind
Saleh Ahmad (stahmad2), Akhil Alapaty (alapaty2), Kushagra Tiwary (ktiwary2)

Introduction

Objective:
Blind people encounter several obstacles on a daily basis with actions we take for granted. In an interview conducted with blind people, the participants noted that the majority of the problems they faced were due to unsafe sidewalks, existence of obstacles, navigation in new spaces, people moving things around, and crossing streets [1]. To rectify some of these problems, many cities have adapted tactile paving inside public spaces and sidewalks to alert the blind of stairs, roads, and platform edges [2] [3]. However, navigating the space especially in larger indoor areas such as airports, malls, and apartment complexes still remain a challenge. Currently, this is largely done through guide dogs, assistance or the individual having to memorize the layout (usually for their homes and personal spaces) [4]. The guide dogs are extremely sensitive to the cues from the blind person and are trained to find exits, stairs and elevators, and even respond to threats [5] [6] [7]. However, wayfinding through airports and large indoor locations without assistance still remains an extremely difficult task mostly due to the complex indoor layouts, lack of directional-aid in non-visual formats and the loud constant echo-noises which can cause disorientation [1] [8]. In fact, independent grocery shopping is still one of the most functionally challenging tasks a blind and visually impaired person can do; therefore, it is essential that we design spaces and build the relevant infrastructure to be more inclusive [7]. The architecture and the lack of accessibility causes the blind person to largely avoid these spaces or only use them when absolutely necessary [1] [7] [8].

Our goal is to attempt to solve the wayfinding task for the blind and visually impaired individuals in indoor spaces such as apartment complexes, malls, etc. The wayfinding task includes locating the person and then subsequently successfully navigating them to their destination through constant vibrational feedback. To build the wayfinder application we will build “walkability” maps of the indoor spaces and use bluetooth beacons to localize the person. The blind person will be carrying a bluetooth compatible wearable-device which will connect to the bluetooth beacons and also be used to input directions and output feedback. The main idea is to be able to localize the person within the building with an accuracy of 1-3 meters using the bluetooth signals and then guide them to their destination through haptic and vibrational feedback. Our application will only handle the navigational and the associated feedback aspect of this problem.
Background:

There are a couple of companies that currently work on indoor wayfinding devices for larger indoor spaces; however, these solutions are not necessarily geared towards the visually impaired. BlindSquare is one of the world’s most popular apps the blind and visually impaired tend to use for navigation in outdoor settings; it announces points-of-interest such as street names, and saved locations and helps its users look up information regarding their surroundings with the help of GPS and third-party-apps [9]. IndoorAtlas is an application for indoor positioning system, which finds your position with the help of bluetooth beacons, WiFi signals, and barometric pressure to measure elevation for general users (is not focused on the visually challenged). This particular application can be downloaded off the Apple or Android Store; however, the application is not for the blind or visually impaired since it is simply indoor positioning technology for retail, healthcare and airports [10]. Regardless, the use of such bluetooth beacons is on the rise due to the large number of bluetooth compatible devices people use. “Indoors” is a company that uses bluetooth beacons along with its mapping systems to help blind people navigate the complicated structure of the Los Angeles Airport. Wayfindr is a similar company focusing its product on the visually impaired and operates in some stations in the London Tube. Indoors has a similar structure in place where they first create high quality maps of the system and then use iBeacons with some more input from WiFi to get the location and then subsequently provide directions [11] [12].

We plan on our application and wearable device being as compatible, cheap and user-friendly as possible to the blind person- especially from the guidance perspective. Additionally, we will also have to keep in mind that there should be no discrepancies or position inaccuracies in regards to location performance within our mapped areas, when we position our beacons.

High-level Requirements List:

- The location of the user should be calculated from at least three strongest beacon signal strengths out of all the signals received, and then must have a location precision below 5m error 85% of the time.\(^1\)
- Given a user-specified destination in an operating area map\(^2\) that is provided through the touch sensor, the system should be able to find a path to navigate the user to that destination.
- The system should be able to automatically change maps when it registers that the user has moved to a new indoor location. \(^3\)

---

\(^1\) Please refer to the appendix on how we calculated and converged to this value.

\(^2\) An operating area is defined as an indoor environment with a map that satisfy the R&V requirements prescribed below and bluetooth beacons placements appropriate for localization of the user carrying the device. In this area, the map will have specified destinations which the user can choose through the touch sensor and they will be guided to that destination. (see R&V below for more info on these sub-system requirements)

\(^3\) Given a new signal from a beacon in a different location, it should be able to detect, automatically load (given the map is already exists in physical device’s memory) and use that location’s map without any user input.
Design

Block Diagram:

Figure 1: Block Diagram

OV-1 Diagram:
Our block diagram consists of the several modular components within our system, and how they interface with each other. After formulating the walkability map on an external computer, we plan to put this information on a SD card, which will then be attached to our microprocessor. Once the user turns on the Bluetooth receiver to communicate with the 3 strongest beacons and inputs the destination point using the touch sensor, this information will be sent to the microprocessor, where all of the software processing will be carried out. This is sent back to the I/O interface, where the user will be notified of navigation errors using the vibration motor. This is conveyed in a straightforward manner by the OV-1 diagram above.
Physical Design (if applicable):

Above is the diagram of the wearable component of our product. We will construct it in this manner in order to show the proof of concept. As we can see on the top, the Input/Output interface will be outwards facing and be directly available to our users. Essentially, on the left is the Bluetooth receiver communicating with the strategically placed beacons. The vibration motors serve to alert the user of the path. The touch sensor is an input system for the user to pick the destinations- numbers of taps will correlate with different destinations. In our first prototype, we plan on uploading a walkability map onto the application beforehand. Underneath in our diagram, we have the backend setup comprising of the microprocessor, memory, and power components.

Figure 3: Physical Design
2.1 Power

This will provide power to the whole system, comprising of the microcontroller, touch sensor, vibration motor, and Bluetooth receivers communicating with the beacons.

2.1.1 Li-On Battery

The power source for the system will be a 3.7 V 40mAh Lithium-Ion battery. The Li-On battery should be able to keep the circuit charged, by powering each of the individual elements that are interfacing with the user as well as the back-end hardware components.

2.1.2 Power Switch

The power switch will allow the user to switch the device on/off. This will be a small, yet accessible mechanism for the user in order to preserve charge, as the power switch is directly connected to the Li-On battery.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
</table>
| If the system is on, pressing the power switch should turn the system off.    | 1. Probe the circuit by placing an ammeter in series with any of the other elements (vibration motor, touch sensor, etc.)  
2. Determine the current in the component of choice and ensure current is ~ 0 mA. |
| If the system is off, pressing the power switch should turn the system on.    | 1. Probe the circuit by placing an ammeter in series with the touch sensor.   
2. Ensure that touching the touch sensor should register a nonzero current value across the ammeter. |

2.2 Memory

This module will store all the information (map, beacon positions and information etc.) that will be required to run the navigation.

2.2.1 Memory Card

The SD card will store all the information (map, beacon positions and information etc.) that will be required to run the application. Since there will be multiple locations, there will be different maps stored in the SD card. When a particular beacon will connect to the system, the map belonging to that location will be used.
2.3 Processing (MicroProcessor)

The microcontroller will consist of our core software processing, which comprises of the beacon signal interpretation, determine the user’s position, and the path planning to destination module. Additionally, it also contains the I/O module interfacing with the beacons, the touch sensor while simultaneously communicating the path to the user through the vibration motor. The microprocessor will be powered by a 3.7 V battery and will draw input from our SD card. Our chosen microcontroller for this project will likely be the ATMega328P, which will serve as the processing unit between the hardware and software modules, as well as the battery and SD card. The ATMega328P microcontroller is capable of operating on voltages between 1.8 V and 5.5 V, which is why we chose a 3.7 V Li-On battery.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
</table>
| The ATMega328P should be able to communicate with the SD Card, on which we upload the walkability maps. | 1. Initialize the SPI interface and conduct a transfer of data using a script that will send/receive a byte.  
2. Test the interface with a loopback: one can wire the MOSI output pin to the MISO input pin, and check that the two bytes match each other. |
| The ATMega328P should be able to take input from the touch sensor.           | 1. Connect touch sensor to ATMega328P using serial input  
2. Write basic script that displays “Output detected” every time touch sensor is touched  
3. Leave device stationary for 10 seconds before proceeding to test |
| The ATMega328P should be able to take input from the Bluetooth receiver.     | 1. Connect Bluetooth Receiver to ATMega328 using the PB1 and PB0 pins of the controller  
2. Write a demo script that initializes the USART communication and receives data from the Bluetooth module  
3. For example, a ‘1’ or ‘2’ will be sent from a smartphone to turn an LED on (‘1’) or off (‘2’) as a means to test the connection |
| The ATMega328P should be able to provide output to the vibration motor at different patterns (used to distinguish directions). | 1. Connect the ATMega328P pins (PC3-PC0) to the inputs of the motor  
2. Write a demo script that will rotate the motors and provide a vibration that is of noticeable intensity and a different pattern. |
2.3.1 Input/Output

This block is responsible to communicate with the touch sensor and the bluetooth receivers. It will receive input signals from the bluetooth receivers and the touch sensor. The module also outputs signals to the vibration motors to give appropriate feedback to the user.

2.3.2 Beacon Signal Interpretation

This module will be responsible for choosing at least three beacons (we need at-least three beacons to get the position of the source) with the strongest signals and perform all the necessary preprocessing before sending it to the next block where they will be used to determine the position of user. The block will also communicate with the Memory module to get information about those beacons’ locations. There is no R&V since it just chooses at-least three beacons based on the strongest signal strength metric or just throws all the signals away if that condition isn’t met.

2.3.3 Position Determination

This beacon solves the trilateration problem in 2D space, with given bluetooth strength signals from at-least three beacons (we need at least three beacons to get the user location) and their respective absolute locations in the walkability map. It will also perform any coordinate transformations and corrections necessary before sending the absolute coordinates to the Path Planning Module.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>In an empty medium sized room environment with no interferences between the sender and receiver, but possible interferences with WiFi, other bluetooth devices, pillars etc. and with at least 4 Bluetooth beacons, the position determination algorithm should be able to get the location with an expected accuracy (euclidean distance) of 4.0m 85% of time. The room has to be of at least of size 6m X 5m.</td>
<td>1. Set up a unit test wrapper function around this block of code in the module and set up the appropriate verification environment.4 2. Fix the bluetooth beacons in the room to meet the R&amp;V of the operating Area Map. 3. Use the receiver and the implemented position determination algorithm to find the euclidean distance between the calculated vs. actual position. 4. Travel through the entire room at-least twice (reset the device before each run) to get at least 50 different data points. 5. Calculate the precision.</td>
</tr>
</tbody>
</table>

4 Please refer to the Appendix for more information on “test” environment and “verification” environments specifications.
2.3.4 Path Planning

This module has the pre-built environment map and the position of the user from the Position Determination block as inputs. This block will find the safest path\(^5\) from the user’s current position to their specified destination. After finding the path, the block is responsible to communicate the next series of steps to the user in an understandable way i.e. converting the path into vibrational feedback so that the user can understand and then act. After the conversion, this new information will be sent to the output block so that it can be relayed to the vibrational motor.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
</table>
| In the test environment the path planning module should be given the current user location, the operating area map, any specified destination from the list of all possible destinations and the model should be able to find the safest path every time. | 1. Set up a unit test wrapper function around this block of code in the module and set up the appropriate test environment.  
2. From a set of 5 random locations in the test environment, iterate through every possible available destination, plotting the output path into a grid map over the walkability map. The output path should be safe i.e. doesn’t navigate the user through static obstacles such as the walls, a table specified on the map, etc. |

2.4 External Computer

The maps of the environment or the operating area is generated offline. We define operating area has an enclosed area where the users can operate in since the area will have bluetooth beacons placed for localization and navigation.

The map of the environment will be created using the IEEE-RAS/MDR (Robotics & Automation Society/Map Data Representation) society’s standards in a hierarchical manner with clearly defined global maps, coordinate spaces and possible sub-maps. Please see figure X for the flowchart and design of such a map.

2.4.1 Operating Area Map

The Operating Area Map will map of the entire environment with obstacles, rooms, and pathways to walk on. The map will also include information on the position of the beacons, and will be stored in the SD card and be referred to constantly by the path planning module to localize the user. We want this map to be hierarchical in structure to include a walkability map (places the user can walk/not walk- pathways, rooms, static items), and a beacon map (locations of the beacons) so that modifications can be easily made.

---

\(^5\) We define the “safest” path to be a path that doesn’t navigate the user through static obstacles such as the walls, a table specified on the map, etc.
The test environment map should have at least one pathway, assuming a path exists, connecting two separate locations and the bluetooth beacons should be placed to have this entire area in range.

1. Set up a unit test which prints the packet that the receiver receives with the corresponding device.
2. Walk through the operating area and check the receiver output from the unit test: if we receive a signal from at-least one beacon at all possible walkable locations then the requirement is verified.

The map should have at least 2 different layers: walkability map and a beacon map.

1. Display the beacon map into a grid array and check if that layer only includes beacon positions.
2. Display the walkability map into a grid array and check if the physical space corresponds to the map.

2.5 Inputs
These serve as the physical devices that the user communicates with (touch sensor) and the bluetooth receivers and beacons that will be used to localize the user in the operating area by the microcontroller.

2.5.1 Touch Sensor
This block serves to provide the microcontroller user input. The touch sensor feedback will determine the user’s destination that the application guides to. This information is then submitted to the I/O module within the microprocessor, which is then processed in order to determine the user’s starting and end point. At this point of our project, we are using a touch sensor to relay feedback through physical taps on the sensor. Beyond the scope of this semester, we expect to convert this input feedback into a voice processing system, in order to facilitate the process for our users.

Should only be able to draw at most 5V since, we don’t want to draw too much voltage and overheat the touch sensor.

Attach a voltmeter in parallel and read the voltage reading across the touch sensor to ensure that the voltage draw is within the parameter.
2.5.2 Bluetooth Receiver

The receiver is incredibly important for our system since user location depends on the strength of the receiver. The Bluetooth receiver will be continuously detecting surrounding bluetooth beacon signals during operation.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
</table>
| The range should be at least greater than 20 meters (radius) in an indoor free, unobstructed, space environment. | 1. Place a beacon in an indoor free space environment.  
2. Walk to 20m and check the signal strength. In unobstructed space, the bluetooth receiver should detect a signal. |

Should be able to communicate with a beacon at least 5m away in the test environment.  
6

1. Place a beacon in a sample test environment.  
2. Walk to in a radius 5m from the beacon. The bluetooth receiver should detect the beacon signal throughout the walk.

2.5.3 Beacons

The beacon module will consist of at least 4 bluetooth beacons located in the operating area. The positions of these beacons will be fixed in the operating area environment, and the receiver will communicate with these devices. The locations of these devices will be fixed and registered in the operating area map. The device will use this information with the strength of the signals to determine the user’s location.

The placement and type of these beacons will be critical to the accuracy of the location of the user since there can be heavy interferences from WiFi, refraction due to pillar, walls, room size, other electronic devices, etc. The requirements below are made with these considerations in mind and we aim to have at least 4 beacons setup and simultaneously running.

6 According to [14], in a scenario similar to the test environment when the distance between the sender and the receiver exceeds 5m, the accuracy of the RSSI measurements decreases drastically, therefore, we expect the bluetooth receiver to work well within that scope.
2.6 Outputs

The path information will be relayed back to the user using vibration generated from the vibration motor. The path information that is communicated will be the next set of steps/directions that the user will need to take to get to their destination. We will encode a series of directions such as left, right, straight, backwards that the vibration motor will output to alert the user.

2.6.1 Vibration Motor

This block serves as the primary output module interfacing with the user. Once the path planning module determines the optimal path from the user’s starting point to the destination point, that information is conveyed to the microprocessor’s I/O module, which is then relayed to the vibration motor. We envision this module to communicate the path using vibrations to the user. Ideally, we want the user to be notified the next series of steps such as turn left, right, etc. and then alert them when the destination has arrived.

The vibration motor will be responsible for providing navigational feedback to the user. The vibration motor will vibrate constantly at a fixed pattern when the guidance is straight. When a turn (right/left) comes up, the vibration motor will change that pattern until the turn is detected by the system and the motor will go back to vibrating in the straight pattern. The user will be notified before about these different vibrations.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Should be sensitive enough to output vibrational frequencies between 120-240Hz.</td>
<td>1. Connect the vibration motor to a function generator and oscilloscope. 2. Sweep the generator frequency from 120-240Hz and verify the oscilloscope reading matches the input.</td>
</tr>
</tbody>
</table>
Figure 4: Software Flow Chart

This software flowchart shows how the microcontroller will process bluetooth signals, destination inputs from the user, and output appropriate route guidance after processing their location.
Circuit Schematics

Figure 5: Circuit Schematic

Tolerance Analysis

A major component of our product is the vibration motor that we intend to use as the outwards facing interface to provide route guidance to the user. The vibration motor will be used to output a distinct pattern for each of the 3 navigational directions - straight, left, and right. Clearly, this is a very important asset of our product our purpose is to provide quality feedback to the user. First, we need to determine the appropriate frequency for the vibration motor’s operation. For the vibration motor that we chose for our implementation, it is rated at 3 V at 10,000 RPM.

\[ f_{\text{vibration}}(\text{Hz}) = \frac{\text{MotorSpeed (RPM)}}{60} \]

*Equation 1: Vibrational Frequency Equation*
Using the above equation, we see that our approximate registered frequency falls at about 167 Hz. We proceeded to research optimal frequencies for similar applications in order to understand the appropriate range when dealing with frequency-related stimuli. Currently, the frequency of the vibration motor found in cell phones ranges between 130 Hz and 180 Hz [15]. The paper goes on to explain that human skin is “very sensitive to vibrating stimuli” when exposed to 230 Hz. Essentially, the optimal range, as explained by the study, for human-interaction vibration systems falls between 120Hz and 240Hz. This range is described as optimal, as the human skin will sense these signals in regards to environmental stimuli. [21] explains that the average tolerance of vibration motors falls around +/- 3200 RPM. Looking at the resulting vibration frequencies, we would see a range of 130 Hz to 220 Hz, which falls within the above listed range. Therefore, despite the tolerance that occurs from manufacturing variances, we see that the result variances still fall within the desired range. However, it is still important to include tolerances in order to keep track of the vibration motor behavior.

Cost

Our fixed development costs are estimated to be $45/hr, 10 hours/week for the three people in our group. During the 16 weeks we have available this semester, we are considering costs for approximately 75% of our project. This doesn’t include customizing our final wearable product and establishing partnerships with malls, airports, etc.

\[
3 \times \frac{45}{hr} \times \frac{10\ hr}{week} \times \frac{16\ weeks}{0.75} \times 2.5 = 72000
\]

Equation 2: Cost Analysis Equation
<table>
<thead>
<tr>
<th>Part</th>
<th>Cost (prototype)</th>
<th>Cost (bulk)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller (Digi-Key; ATMega328P)</td>
<td>$2.14</td>
<td>$1.78</td>
</tr>
<tr>
<td>iBeacon (4) (Amazon)</td>
<td>$21.99 * 4 = $87.96</td>
<td>$21.99 * 4 = $87.96</td>
</tr>
<tr>
<td>Capacitive Touch Sensor</td>
<td>$0.60</td>
<td>$0.49</td>
</tr>
<tr>
<td>Vibration Motor (Digi-Key; 1597-1245-ND)</td>
<td>$1.44</td>
<td>$1.44</td>
</tr>
<tr>
<td>3.7 Volt Li-Ion Battery 40mAh (Digi-Key; 1568-1496-ND)</td>
<td>$4.50</td>
<td>$4.50</td>
</tr>
<tr>
<td>4 GB MicroSD Card (Digi-Key; 1582-1271-ND)</td>
<td>$10.63</td>
<td>$7.66</td>
</tr>
<tr>
<td>HC05 Bluetooth Module</td>
<td>$11.11</td>
<td>$11.11</td>
</tr>
<tr>
<td>Power Switch (Digi-Key; EG4791-ND)</td>
<td>$0.53</td>
<td>$0.32</td>
</tr>
<tr>
<td>PCBs (PCBWay)</td>
<td>$3.10</td>
<td>$0.10</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$121.48</strong></td>
<td><strong>$115.04</strong></td>
</tr>
</tbody>
</table>

*Table 1: Parts and Cost*

Our clear majority of costs comes from the iBeacons that we plan on purchasing to implement our product. With a 500 meter range in free area, we determine that the cost is worth the performance. Currently, we would be purchasing from Amazon; however, as we plan on moving to bulk manufacturing, we plan to discuss with the seller (Feasycom) about cheaper prices in bulk.
# Schedule:

<table>
<thead>
<tr>
<th>Week</th>
<th>Saleh</th>
<th>Kush</th>
<th>Akhil</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/18/19</td>
<td>Buy Parts &amp; Integration of Bluetooth Receiver Module with Microprocessor</td>
<td>Buy Parts &amp; Work on Map Creation Software Module</td>
<td>Buy Parts &amp; Integration of Touch Sensor Module with Microprocessor</td>
</tr>
<tr>
<td>2/25/19</td>
<td>Integration of Bluetooth Receiver Module</td>
<td>Work on Map Creation Software Module</td>
<td>Integration of Vibration Motor Module with Microprocessor</td>
</tr>
<tr>
<td>3/4/19</td>
<td>Work on Input/Output Software Module</td>
<td>Integration of Memory Module</td>
<td>Work on Input/Output Software Module</td>
</tr>
<tr>
<td>3/11/19</td>
<td>Develop/Test Beacon Signal Interpretation Module</td>
<td>Develop Position Determination Module</td>
<td>Integration of Power Module</td>
</tr>
<tr>
<td>3/25/19</td>
<td>Complete/Integrate Beacon Signal Interpretation Module</td>
<td>Finish Position Determination Module</td>
<td>Integration of Power Module</td>
</tr>
<tr>
<td>4/1/19</td>
<td>Integrate/Test Path Planning Module</td>
<td>Integrate/Test Path Planning</td>
<td>Determine Device Case and Packaging</td>
</tr>
<tr>
<td>4/8/19</td>
<td>Work on Debugging/Testing &amp; Refining of Prototype</td>
<td>Work on Debugging/Testing &amp; Refining of Prototype</td>
<td>Work on Debugging/Testing &amp; Refining of Prototype</td>
</tr>
<tr>
<td>4/15/19</td>
<td>Demo Preparation</td>
<td>Demo Preparation</td>
<td>Demo Preparation</td>
</tr>
<tr>
<td>4/22/19</td>
<td>Present Demo/Prepare for Presentation</td>
<td>Present Demo/Prepare for Presentation</td>
<td>Present Demo/Prepare for Presentation</td>
</tr>
<tr>
<td>4/29/19</td>
<td>Deliver Presentation</td>
<td>Deliver Presentation</td>
<td>Deliver Presentation</td>
</tr>
</tbody>
</table>

Table 2: Schedule
Ethics and Safety

During the development of our system, the main safety issue we will face will arise during the testing portion of our project. In order to ensure that our project is self-sufficient for our visually impaired demographic group, it is essential that the test subjects are emulating the equivalent conditions; for example, we will utilize a blindfold so that he/she will not be able to see. Since our project, at this point, will still be in development stages, the possibility of not detecting a completely accurate path to the destination is possible. Therefore, in order to comply with what we have to ensure that our test subjects are safe and don’t come under any duress in the case that the system guides them in an inaccurate manner [13]. Taking #1 of the IEEE Code of Ethics into account, it is essential that are prioritizing the safety and welfare of users at all points of the development and utilization of our product. For example, one solution during the testing period would be to make sure of this by having someone be with the subject throughout the test, verbally/physically guiding the individual if needed.

During the use of our product, one potential concern is the fact that internal surroundings of viable locations could be slightly altered after uploading a blueprint to the system. Therefore, we must ensure that our blueprints are constantly being updated when major internal design changes are being made. Prioritizing this action will allow our users to have full confidence in our product. As seen in the IEEE Code of Ethics, the number one interest is to “…protect the safety and welfare of our users”. Additionally, we want to update our blueprints when needed as a means of transparency that we’re doing our work on the back end. This directly relates to #3 of the IEEE Code of Ethics because we must be honest and realistic with the information we are providing based off the data available to us. Moreover, it is vital that we market our product as a navigation tool which is meant to be utilized in supplement with regularly used walking aids. Industry-standard beacons tend to have a rough 1-2 meter in-accuracy, which would simply corrected by users having their canes/walking dogs/etc.

Additionally, the updated maps should follow the IEEE-RAS/MDR (Robotics & Automation Society/Map Data Representation) society whose goals are to define norms to commonly “represent and encode a map for robotics navigation”. This will make our product versatile and our maps can be used throughout the industry and the updates explained above can happen in a systematic manner. We wish to separate our maps into global and local maps and then represent them in a hierarchical manner with clearly defined coordinate spaces, grid and topological maps. Also, users might be worried about their locations and privacy since they are being tracked in public areas such as malls, airports, etc. Our duty is to inform our users that the beacons are only communicating with their wearable sensors and that the data is not being sent elsewhere.
Looking more into the elements of our design, using a Lithium-ion battery will definitely come with its own set of challenges and necessary precautions. We must account for not placing them in extreme temperatures or overheating within the circuit. Similarly, another concern is that none of the electrical components part of the wearable design should not get overheated, as that would directly impact our users. Part #6 of the IEEE Code of Ethics applies most, as we have all taken extensive electrical engineering courses and plan to conduct testing during the development in order to ensure that our designs are working as intended.

Aside from the functioning of our product, we will also encounter challenges in the form of activities that need to be completed in the senior design electronics laboratory. We will be dealing with soldering elements to our boards and using power supplies in order provide power to our product. Each of our group members has taken the Lab Safety certification and is prepared/well-equipped to deal with these challenges.
References

[10] https://www.indooratlas.com/ [Company Website]
[12] https://wayfindr.net/ [Company Website]
Appendix

Position Determination

The position determination algorithm uses the bluetooth signal strength and the beacon positions to solve the trilateration problem. There are many ways to solve this problem since it inherently comes down finding the intersection between at least 3 circles (each circle with the beacon at its center) in 2D space. However, at some configurations and strengths, the intersection of at least three circles doesn’t yield a point but an area. Moreover, there are many possible scenarios where the intersection is not accurate due to the signal strength, the number of beacons, receivers etc.

According to [14], there are a lot of methods to get the desired location and work around to these cases. The earliest one includes fingerprinting [16] where the authors reached an error of 4.15m 90% of the time, however, given the large amount of environment characterization the algorithm requires, it is not the best result. Since [16] there have been a lot of techniques developed including using WiFi signals, BLE+WiFi+fingerprinting [17] with 95% of the time below a 2.6m error in a room of size 45mx12m and using 20 beacons, BLE+Deep Learning [18] with 90% of the time below 2m error with using 10 beacons in a room of size 17.5m X 9.6m, BLE+KF [19], [20]. They have reached different precision (% of time below some distance error) values and have used a different number of beacons and computational models.

Our product may not be able to use the “state-of-the-art” technology due to many consideration including but not limited to the almost live feedback provided to the user, cost of the beacons and setup, and the computational power on the microcontroller. Due to these considerations, we set our precision goal to be within 85% of the time within a 4m error. Solving a non-linear least squares on an ATMega83 with heavy optimization is also extremely time consuming. In a raw setup (without using any optimization for the non-linear least squares) on a computer the benchmark is 90% of the time below 3.22m error in a lab environment with interference from WiFi, computers, pillars, walls, etc. In an open environment like a conference room of size 16.50m X 17.60m, the precision was 90% of the time below 7.46m. Taking into consideration the time take by the most time consuming block in software- path planning module to compute the path (even with optimization such as storing the last computer path), the time given to us to complete the product, the environments we will be operating on, and the live feedback property of our product we have roughly averaged the precision to 85% of the time below 5m. This is also considering the fact that we will be operating in large open spaces, and even implementations such as BLE+KF+Weighted Trilateration (note that we haven’t considered DL models at all because of computational constraints) give us a precision of 90% of the time below 4.6m [14].
Test and Verification Environments

We create two environments to verify the requirements for the sub systems. This is done since the user can theoretically take the module anywhere and the device should work given the beacons placement requirements and the map. Therefore, to benchmark the product we create a verification and test environment. These are the minimum properties those environments should meet:

**Verification Environment:** In an empty medium sized room environment with no interferences between the sender and receiver, but possible interferences with WiFi, other bluetooth devices, pillars etc. and with at least 4 Bluetooth beacons. The room has to be of at least of size 6m X 5m. The bluetooth beacons should be placed to have this entire area in range.

**Test Environment:** This is the environment where the user will be actually using the device e.g. Airport, malls, large indoor public spaces. For prototyping, however, we define the minimum constraints of this environment to have at least one pathway, assuming a path exists, connecting two separate locations and the bluetooth beacons should be placed to satisfy the operating map area R&V described earlier. There should be at least 4 beacons and the minimum dimensions should be 12m by 10m. We don’t have any restrictions on the possible interferences since we will not be able to regulate that in the real world, although this scenario shouldn’t be very different that the interferences faced in the verification environment.