Find All User’s ‘X’ (FAUX)



Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Samuel Richie

Senior Design I

Group #2:

|  |  |
| --- | --- |
| **Group Member** | **Major** |
| Amish Kanji | Computer Engineering |
| Joshua Radicchi | Electrical Engineering |
| Joel Gardyasz | Computer Engineering |

**Table of Contents**

1 Executive Summary …………………...…………..…………………….….….1

2 Project Description ……………………...………...........................……...…...2

2.1 Personnel ……………………………………………………...…..……….2

2.2 Project Motivation …………………………………..…….….…..………..3

2.3 Goals & Objectives ………………………………………..…...………….4

2.4 Requirements & Specifications .……………………...………………….5

3 Research ………………………………………………………………………...6

3.1 Related Projects ………………………………………………….………..6

3.1.1 TILE ………………………………………..……..……………….7

3.1.2 TrackR …………………………………….……………………....7

3.1.3 Duet …………..………………………………………….…..……8

3.1.4 Lasso Tag ……………..……………………………………….…9

3.2 Relevant Technologies …………………………………..…………......10

3.2.1 Microcontroller ……………………………….…………….….10

3.2.1.1 Relevant Microcontrollers …………...……………..10

3.2.1.1.1 TI MSP430 ………………………………….10

3.2.1.1.2 CSR1010 ………………...………………....11

3.2.1.1.3 ATMEGA328P …….…………...….……....11

3.2.2 Wireless Communication ….………………………..…….…..12

3.2.2.1 Bluetooth ……….……………………………………..12

3.2.2.1.1 Bluetooth Low Energy……………...……...13

3.2.2.1.2 iBeacon…………………….……...............15

3.2.2.2 WiFi………………………....……………….………..18

3.2.2.3 GPS/GSM……………………………….….………....19

3.2.3 LED/Speaker/Power …………………….………………..…...22

3.2.3.1 LED……………………………….…………………...22

3.2.3.2 Speaker………………….…………..………….……..23

3.2.3.3 Power…………..………….…………..……………..24

3.2.4 Protective Casing…………………………………………….25

3.3 Software Implementation…………………………………………...…..26

3.3.1 Mobile Application Design……………………….…………..27

3.3.2 Programming/SDK………………………………..…….…….28

4 Related Standards…………………………………………………….……..29

4.1 Design Impact of Relevant Standards……………...……….…...…….31

5 Constraints ………………………………………………….………………...32

5.1 Environmental Constraints ………………………..……………………….…..32

5.1.1 Distance Constraints ………………...…..…………………….32

5.1.2 Weather/Other Constraints ……….………………………………….33

5.2 Economic Constraints....…...………………………….…………………………34

5.3 Size Constraints …….……………………………………..………………...….. 35

5.4 Ethical Constraints………………………………………….……….…………....36

6 System Design Details………………………………………………………...36

6.1 Hardware Components…………………………………………………...36

6.1.1 Chosen Microcontroller………………………………………...36

6.1.1.1 Reasoning/Purpose………………………………...39

6.1.2 Chosen Bluetooth Module……………………………………………42

6.1.3 Chosen Wireless System……………………………………...43

6.1.3.1 Reasoning/Purpose………………………………….45

6.1.4 Chosen Battery…………………………….………...………….48

6.1.5 Chosen Speaker……………………………….………………..49

6.1.5.1 Reasoning/Purpose…………..…………….………...50

6.1.6 Chosen LED………………………………………….………...51

6.1.6.1 Reasoning/Purpose……………………......………...52

6.2 Software Design…………………………………………………………...53

6.2.1 Mobile Application……………………………………………...53

6.2.1.1 Chosen Operating System………………………….53

6.2.1.2 Application purpose…………………………….….....54

6.2.2 Communication Software………………………………..……..57

6.2.2.1 iBeacon Software……………………………….…….57

6.2.2.2 Microcontroller Software………………………….….58

6.3 Protective Casing……………………………………….………………..59

6.3.1 Shape and Size…………………………………………...…….60

6.3.2 Weight……………………………………………………..…….63

6.3.3 Color/Material………………………………….……...………..64

7 Design Summary……………………………………………………………..66

7.1 FAUX………………………………………....…………………...66

8 Prototype Building……………………………………………………………..66

8.1 Development……………………………………………………………….66

8.1.1 Hardware Development………………………………………..66

8.1.2 Software Development………………………………………....69

8.2 Production………………………………………………………………….72

8.2.1 Protective Case………………………………………………….72

8.3 Vendors………………………………………………….………………....73

9 Prototype Testing………………………………………………………...…....73

9.1 Testing Environment……………………………………………….....….73

9.1.1 Indoor Testing …………………………………………………..74

9.1.2 Outdoor Testing ………………………………………………...75

9.1.3 Obstacles Expected …………………………………………....77

9.2 Hardware Test Procedures……………………………………………....77

9.2.1 Microcontroller Tests…………………….…………………......78

9.2.2 LED Tests ……………………………………………………….80

9.2.3 Speaker Tests…………………………………………………...82

9.3 Software Test Procedures…………………………………….................83

9.3.1 iBeacon SDK Tests……………………………………………..83

9.4 Mobile App Test…………………………………………………………...89

9.4.1 Test Scenarios…………………………………………………..89

10 Administrative Content………………………………………………………..92

10.1 Milestone Timeline……………………………………………………….92

10.2 Budget………………………………………………………………….....93

10.3 Financing………………………………………………………………….93

11 Final Project Summary/ Conclusions……………..………………………....93

Appendix………………………………………………………………………………..95

Permissions…………………………………………………………………….95

Estimote Permission…………………………………………………..95

Punch Through Permission…………………………………………..96

Multicomp Permission………………………………………………...97

**List of Tables**

Table 1: Requirements and Specifications…………………………………………..6

Table 2: Different Bluetooth Classes…………………………………………….…..13

Table 3: Classic Bluetooth vs Bluetooth Smart…………………………………….15

Table 4: Bluetooth v4.0 vs GPS……………………………………………………...21

Table 5: LEDs and their forward voltages…………………………………………..23

Table 6: Milestone Timeline…………………………………………………………..89

Table 7: Price List……………………………………………………………………...90

**List of Figures**

Figure **1**: Legend and Block Diagram showing work division among group…...…3

Figure **2**: Top of the Light Blue Bean (with Punch Through permission…...........37

Figure **3:** Bottom of the Light Blue Bean (with Punch Through permission)..…..38

Figure **4:** System Architecture (with permission from Punch Through)………….39

Figure **5:** PCB of Light Blue Bean (with permission from PunchThrough)………40

Figure **6:** Pin layout of the ATMEGA328 chip……………………………………....41

Figure **7**: Specs of Light Blue Bean (with permission from Punch Through)…....42

Figure **8**: Estimote Sticker Beacons (with Estimote permission)..........................44

Figure **9**: Estimote Sticker Beacon PCB (with Estimote permission)...................45

Figure **10**: Estimote Beacons vs Sticker Beacons (with Estimote permission)…46

Figure **11**: Code explaining Triggers (with Estimote permission).........................47

Figure **12**: Ranging for Beacons (with Estimote permission)...............................48

Figure **13**: Picture of the CR2032, the chosen battery for the project…………...49

Figure **14**: MCKPRG1720-4011 Piezo Buzzer (permission from Multicomp)…..50

Figure **15**: Flowchart to send reminder notification to user……………………….56

Figure **16**: Flowchart to check motion of FAUX…………………………………....56

Figure **17:** 3D rendering of the case we plan on making for the Bean (Top).......61

Figure **18:** 3D rendering of the case we plan on making for the Bean (Bottom)..62

Figure **19:** The top and bottom pieces of the cover together……………………..63

Figure **20**: Simulated LED circuit………………………………………………….....67

Figure **21**: Simulated speaker circuit……………………………………………..….68

Figure **22:** Flowchart showing hardware portion of the FAUX tracking device.…69

Figure **X**: FAUX user interface

  Figure **x**: Motion mobile notification

Figure **Y**: Loss of signal mobile notification

Figure **X**: Entering FAUX range mobile notification

Figure **23**: Flowchart to change state of components.…………………………….70

Figure **X**: Flowchart for reconnect button

Figure **25**: Flowchart to accomplish tracking functionality………………………...72

Figure **X**: Near zone for application

Figure **Y**: Far zone for application

Figure **26**: Flowchart showing how we plan on testing the microcontroller……..79

Figure **27**: Flowchart showing the process of testing the LEDs……………….....81

Figure **28**: Flowchart showing the process of testing the speaker…………….....82

Figure **29**: Flowchart showcasing the use of ESTNearableZone class………….84

Figure **30**: Flowchart showcasing a stranger returning FAUX to the owner..…..85

**1 FAUX Executive Summary**

Valuables are defined as an item that is of great worth, especially, a small item of personal property. Some examples of valuables are your phone, keys and your wallet. For some individuals, keeping track of these items is a fairly simple task but for others, these items are misplaced more often than desired. The Find All User’s ‘X’ (FAUX) is a tracking device that will be attached to an item that the user considers valuable enough that if they lose it, they need to find it as soon as possible. The ‘X’ in FAUX refers to an algebraic variable that can be replaced by any item (Keys, Wallet, Laptop, etc.). For someone who misplaces their keys often, putting this tracker on your keychain will work wonders. Once the user notices that they have misplaced said item, they can open up the designed mobile app for the items and use the features presented to them in order to track their item and retrieve it.

FAUX can also be used by people who do not misplace items but need to keep track of certain valuables in case they are stolen. For example, the tracker can be placed within a laptop carrying bag or briefcase to ensure that if the item is stolen, there is a way of being able to find the location of the item and notify authorities of its whereabouts. This sense of security is a must in present day. Today, items such as a wallet and keys are required in everyday use. If either of these items are misplaced, there could be serious repercussions that follow. Strangers who find either of these could potentially steal your car, max out your credit cards and take any money you had in the wallet. Losing these items causes a state of panic in almost everyone due to the shear importance of these items in society today. Our project will be linked with a mobile application that can be accessed via any smartphone or tablet computer that will be used to locate any item that the user needs the location for.

The creation of the tracking object required hardware components like a microcontroller and LEDs and software needed for communication purposes. The only required objects for the user will be the FAUX tracking object and a smartphone and or tablet computer to utilize the application that was created to correspond with the tracker. Due to us wanting the project to be small and mobile, needing to carry around as little as possible made the project feasible and appealing for users to take advantage of FAUX.

Choosing a senior design project was not a simple task that could be accomplished within minutes. A project needed to have sufficient hardware integration as well as software implementation in order to showcase the skills acquired throughout the previous classes taken at the university. FAUX gave our group the opportunity to showboat our learned knowledge and to learn about different technologies currently available that will allow us to complete our task but also which technologies would be more suited for our projects and which technologies we should ignore. As a group, this was the first time any of us will be building a project from scratch that requires hardware and software components and ensuring both components can interact and communicate cohesively with one another.

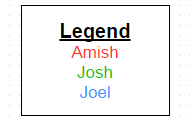
**2 Project Description**

Losing items is never an ideal situation for anybody, especially when the item holds a great value to the individual. Our project will help individuals with this problem by allowing them a quick and easy method to track down their items. We wanted the item to be small enough to be carried around on everyday items without making it a hassle for the user to have it on them the whole time. FAUX is miniature in size, allowing it to be carried around by user without it being an inconvenience. We also designed it to have a key ring hole in the casing to allow the project to be attached to keys if that’s the desired item to be tracked.

We are aware that similar projects are currently on the market but we want to have our own unique attempt at competing with similar projects. No two projects are identical in every feature so we believe that any product can be successful if correctly organized. FAUX will give users assurance that they can live worry free from misplacing items due to the fact that they have means of locating their items at any given time. A portable tracking device that could be put on a keychain, put in a bag or attached to any other item is feasible for any individual. The only other object needed to use our project would be a smartphone or even a tablet computer. In present day, it can be assumed that every individual who would use the project is an owner of a smartphone.

**2.1 Personnel**

Our group consists of three seniors, Joel Gardyasz, Josh Radicchi and Amish Kanji. We have been at UCF all 4 years of our studies. The group consists of two Computer Engineers and one Electrical Engineer. In order to be efficient with our work, we created a block diagram to divide up the work and show who is responsible for what tasks in the project. This block diagram shows that we chose Bluetooth as our form of wireless communication and Apple as our desired device with the iOS operating system. These are assumptions that we decided to illustrate for the sake of making the block diagram more detailed and easier to showcase. Wireless communication and operating system choice was decided after all the research had been conducted. Figure **1** showing the block diagram and the legend explaining who is responsible for each component is shown below.



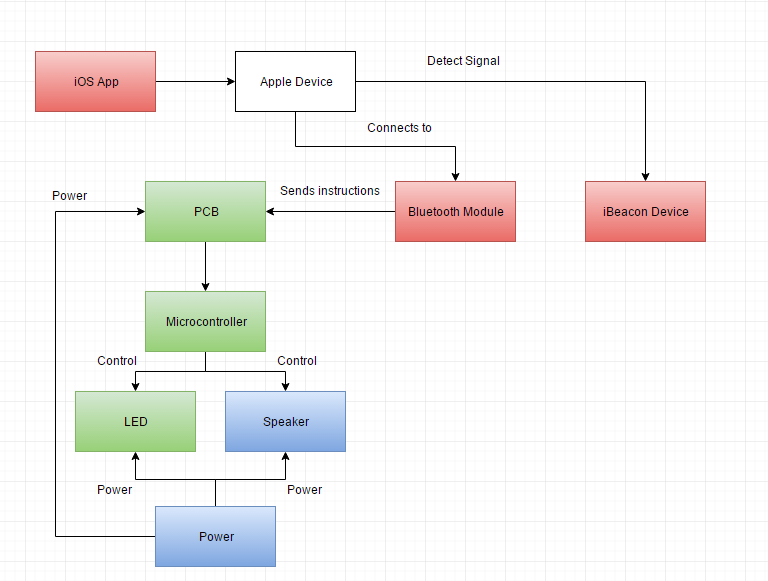


Figure **1**: Legend and Block Diagram showing work division among group

**2.2 Project Motivation**

Motivation for this project can definitely be accounted for from personal experience or similar experiences involving friends and or family. There have been countless stories between our group members where either one of us or someone we know has lost an item and bad results followed due to the item being misplaced. Most people don’t have a means of locating a lost item other than trying to remember the last memory of using the lost item or the place the item could have been misplaced at. Both of these situations are subject to human error because in some instances, the certain individual could incorrectly remember the last place they believe they were before misplacing the item which could lead in a misleading search based on the incorrect memory. We wanted to take away the chance of human error and give everyone a means of being certain of the location/proximity of the lost item.

Another reasoning for choosing this idea as our senior design project is the recent outbreak of similar projects. Tracking objects used for valuables have only recently become popular and are being seen more often. This gave us an opportunity to look at all the similar products and see what features from each of them seem beneficial and see what features we believe we can exclude from implementing in our version. Each of the already created tracking objects are unique in their own way and as a group we want an opportunity to showcase what we can make and why ours will be unique compared to other competitors on the market.

We don’t believe that any of the current tracking objects are being utilized at all. Most people encountered in a typical day do not utilize a tracking object for any of their items and some of them didn’t even realize that there was such items created and available for use. We believe that this is an upside for going into this idea because to some people, our project will be the first of any type of mobile tracking object that they have encountered before so we would get a true unbiased opinion on our project.

As a group we decided to focus on picking an object that was small in size because we wanted to choose an idea that is useful in everyday activities that people would actually benefit from having. Items today are being able to accomplish more and becoming smaller and more portable in the process. For example, tablet computers is a fairly recent technology that took a computer and the tasks it accomplishes and created a smaller and more portable option for people who prefer these aspects. People also do not like to carry unnecessary items and lug them around all day with them. Tracking objects do not have this problem because they are normally attached to the items people have to carry around with them anyways, including your keys or wallet. Also, the second component needed in order to utilize a tracker is a smartphone or tablet computer so you can access the linked application. This is also not a problem because it is safe to assume that everybody nowadays carries some sort of mobile device on them or close to them wherever they are present.

**2.3 Goals & Objectives**

The goals that we wish to accomplish with our project are for the user of FAUX to be able to lose their objects and not to freak-out over it. We believe that we are tackling this objective in the correct way. We are incorporating a mobile application into our project so that the user will be able to locate their lost items quite easily by just opening the app up and activating the speaker and LED to lead you to your lost object. We also will be incorporating a distance tracker in to the mobile application. This will be able to tell you how close or how far you are from the object that you are trying to locate. There have been too many times that people have been late to work or to school because they were unable to locate their keys. This is what we hope to accomplish by creating this tracking device.

**2.4 Requirements & Specifications**

Our project needed certain requirements and specifications in order to meet our goals and objectives. For example, due to the fact that we want the project to be portable and easy to carry for the user, we needed to have size requirements in order for it to be small enough to be portable as well as weight requirements so all users can carry it around without having to worry that FAUX will be too heavy for certain individuals. In order to make the project successful, we came up with a list of requirements and specifications that we plan to meet in our project design.

The list of requirements are shown in Table 1 below.

|  |  |
| --- | --- |
| **Required ID** | **Requirement Description** |
| FAUX-01 | The location of the object will be found by using a smartphone or tablet. |
| FAUX-02 | The tracking device required us to use either GPS, Bluetooth or Wi-Fi communication. |
| FAUX-03 | The tracking device will be of portable size and weight. The size should not exceed 4x4x4 |
| FAUX-04 | The tracker will be water resilient, and be able to withstand fall damage. |
| FAUX-05 | The tracker will be powered by a battery that will last at least six months. |
| FAUX-06 | The tracker will respond to the user’s commands sent from a smartphone. |
| FAUX-07 | The tracker will respond to the user's commands within 7 seconds. |
| FAUX-08 | Your smartphone will respond to the tracker if need be. |
| FAUX-09 | The software will be developed in Java/ JavaScript if possible but is not constrained to these two languages. |
| FAUX-10 | The tracker will produce sounds and light up when prompted. |
| FAUX-11 | The batteries in the tracker will be easy to replace. |
| FAUX-12 | The tracker will work well up to 20 feet. |
| FAUX-13 | The tracker will notify the user if the item being tracked gets too far away from the user. This will allow the user to not forget items. Distance yet to be decided but will depend on technology being used. |

Table 1: Requirements and Specifications

Table 1 shows the list of extensive requirements that we have set for the creation of our project. We believe that we should be able to satisfy all of these with no problem due to the fact that there are products with similar specifications on the market currently. If for some reason a specification could not be met, we will have to revisit the certain specification and see if our current specs will still be feasible in our opinion. That only applies if we fail to meet a specification but currently, that is the list of requirements that we will aim to follow.

**3 Research**

**3.1 Related Projects**

There are a few projects and products that closely relate to FAUX, and each of them is different in their own way. The ones that have been focused on are Duet, TrackR, Lasso Tag, and the most popular one TILE.

**3.1.1 TILE**

TILE is an application and hardware device package; it is available on both Apple and Android. This package allows the user to locate lost items that the tracker is on using Bluetooth Low Energy 4.0 radio technology. This tracker has a range of about 100 feet, and it comes with a built in speaker that produces a sound when told to. The application can also locate Tiles beyond the 100-foot Bluetooth range. If an item with a Tile device reported lost comes within range of another user's Tile application, their application will send the owner an anonymous update of that item's location.

This product has a battery life of about one year from the day that you receive the package. It uses a CR2 battery, and once this battery runs out there is no way to easily replace it. Their fix for this is that you send them your own TILE back and you can buy a new TILE at a slightly reduced cost. It weighs approximately 1.3 ounces; it has to be this lite so it does not weigh down what you are carrying it on. TILE is water resistant up to IP5 standards. This means that if you were to spill your drink on it or if it is raining out you would still be able to use it. The TILE application is very user friendly. TILE casts a signal across a 100-foot radius. To start searching for an item, select the item from the application home screen and tap the green "Find" button. You can also tap the TILE icon to show the location proximity. When you start searching for an item, your phone tracks the signal strength of your TILE. The green tracking circle fills in as you get closer and TILE's distinctive melody will play until you find the item that you were looking for. The application automatically records the last place your phone; or any other phone with the same application open, saw your TILE. Simply open up the lost item on the application and tap the map to get the approximate location and directions to the last place your TILE was known to be.

The TILE application also allows the owner of the TILE to share it with one other person to increase the range, so it is easier to find your lost item. TILE's sound is designed to be heard by a wide range of hearing levels. You can press the “Find” button on the app to have your TILE play a tune. The speaker can emit songs that range from 131 Hz to 8000 Hz. TILE’s dimensions are 37mm by 37mm by 5.3mm, that is less than one and a half inches in length and less than a quarter of an inch in thickness.

**3.1.2 TrackR**

TrackR’s are enabled by Bluetooth Low Energy, so the range of connectivity is up to 100ft. The TrackR offers multiple features that may be utilized within this 100ft, such as the 2-way ring feature and last known location feature. The two way ring feature Means that if you lost your phone you can press a button on the TrackR and it will you’re your phone ring even if it is on silent. Once a user separates from their TrackR tagged item, the user receives a notification and can see the last known location of their lost item on a map. Once it leaves the 100ft range the Crowd GPS feature kicks in which enables you to find your lost items even when you are out of the 100ft range. When an item goes missing with a TrackR device attached, all TrackR enabled phones will begin to search for that item. When another TrackR app user passes within 100ft of the lost item, that user's phone will anonymously ping our server to update the item's owner with new GPS coordinates of when & where it was last seen.

The TrackR comes in various sizes depending what you want to put it on/in such as; TrackR wallet, TrackR sticker, and TrackR bravo. Each of these is different and can be used for diverse purposes. For example their newest and best TrackR is about the size of a half dollar and the casing is made from aluminum, which makes it very durable, TrackR’s are not waterproof, but they are water resistant. And possibly the biggest advantage of the TrackR is that it has easily replicable batteries. The life of the batteries that it uses is about one year, and the replacements can be easily bought at any store. TrackR works with a paired application on your smartphone and can be downloaded on any Apple or Android device. Using the TrackR application you can pair your smartphone to up to ten TrackR devices.

This device also comes with a built in speaker that produces a noise when someone is looking for something that they lost. This speaker can produce a noise up to 85 decibels. The dimensions of their most up to date TrackR, the TrackR bravo, is 31mm in diameter and 3.5mm in thickness. This particular TrackR can be engraved with a short message on the outside of the device. This is a unique idea because if someone else finds the item you lost they can call the number that is engraved on the device.

The TrackR Company is one of the few companies that are continuing to come out with new products and new ideas so you can know where all your belongings are in your house. Their newest product that was just announced is called TrackR atlas, these are plugs that you plug-in in every room in your house, once you open the paired application it shows you the layout of your house and shows what room all of your other TrackR devices are in.

**3.1.3 Duet**

Duet is a tracking device from the company PROTAG, it uses Bluetooth Low Energy 4.0 to track your lost device. This means that the range of the device is up to 100ft. Duet comes with multiple features including and loss prevention alarm, Crowd Tracking, and the last known location. It uses a CSR1010 chip on the inside. The Duet can be paired to any Apple or Android device that is up to date. There can be up to 10 Duets paired to one device.

The Duet has batteries that can easily be replaced by just unscrewing a panel on the back. That battery that Duet uses is a CR2016 coin battery. The Duet is shaped like a square and has dimensions of 28mm by 28mm by 5.6mm. This is smaller than the most widely known tracker on the market, TILE. If someone loses their phone but they have their Duet device they would be able to press a button on the device and it will make their phone ring even if it is on silent. This way it will make one's phone easier to find. It also has a feature called “Retrace Your Steps,” this feature allows the owner to see the last place that their phone was within Bluetooth range. This way it will give the owner a place to start looking for their lost item. This device can come in various colors, this feature has no technical application, but the owner would be able to determine which device is which.

The interesting feature about this device is that it doesn’t have a speaker instead it has something that makes it vibrate. There is no explanation why they chose this over a speaker, but they seem to think that it will work better. The Duet is slightly water resistant, but not waterproof what so ever.

**3.1.4 LassoTag**

LassoTag is enabled by Bluetooth Low Energy, so the range of connectivity is up to 50ft. This device has a paired application that you can use on your smartphone, but your smartphone can only be an Apple product and it has to be an iPhone 4S or newer. At the current moment LassoTag is working on expanding to Android devices, but that have not don’t that yet. The LassoTag does come with two adhesive strips that you can place on virtually any surface you want your tag to stick to. The LassoTag gives its dimensions in quarters, so the device is three quarters thick and one and a half quarters long. This converts to about 5mm thick and about 34mm long.

The LassoTag application can track up to six different LassoTag devices. The lithium rechargeable battery in the LassoTag HD will last 10-14 days and should be charged overnight with a mini-usb cord for another 10-14 day usage. The lithium CR2032 battery in the LassoTag Beacon will need to be replaced after about 6 months under normal usage. The battery is easily replaceable; all you have to do is open up the panel on the back and put a new battery in. The LassoTag application doesn’t tell you which direction your lost device but it will tell you if you are getting warmer or colder. LassoTag HD is not waterproof, however, the new LassoTag Beacon is water resistant and in tests has been fully submerged and still works.

LassoTag is one of the only trackers that is Apple iBeacon Certified. LassoTag has a Client Center that allows you to track your stolen belongings through any of the Internet browsers. LassoTag is also very lite, it only weighs .05 ounces. One unique feature that this device offers is that it has a built in temperature sensor. When you open the application you can see the temperatures of all of your current Lasso Tags.

**3.2 Relevant Technologies**

**3.2.1 Microcontrollers**

A microcontroller is a self-contained system with peripherals, memory and a processor that can be used as an embedded system. Most programmable microcontrollers that are used today are embedded in other consumer products or machinery including phones, peripherals, automobiles and household appliances for computer systems. Due to that, another name for a microcontroller is "embedded controller." Some embedded systems are more sophisticated, while others have minimal requirements for memory and programming length and a low software complexity. Input and output devices include solenoids, LCD displays, relays, switches and sensors for data like humidity, temperature or light level, amongst others.

Programmable microcontrollers are designed to be used for embedded applications, unlike microprocessors that can be found in PCs. Microcontrollers are used in automatically controlled devices including power tools, toys, implantable medical devices, office machines, engine control systems, appliances, remote controls and other types of embedded systems. There are many different types of microcontrollers used for different tasks, we will be focusing on the type of microcontrollers that are smaller and Bluetooth enabled.

The microcontrollers that we have focused on are Texas Instruments MSP430, Atmel’s ATMEGA328P and the CSR 1010. The CSR1010 is the industry leader for Bluetooth Smart, enabling Bluetooth Smart devices to transfer simple data sets between compact devices.

**3.2.1.1 Relevant Microcontrollers**

**3.2.1.1.1 TI MSP430**

There are three very relevant microcontrollers to this project; Texas Instruments MSP430, the CSR1010 chip, and Atmel’s ATMEGA328P. All of these can be used for low powered embedded systems. The CC256x MSP430 is Bluetooth low energy enabled. The MSP430 can be used for low powered [embedded devices](https://en.wikipedia.org/wiki/Embedded_devices). The [current](https://en.wikipedia.org/wiki/Electric_current) drawn in idle mode can be less than 1 µA. The top CPU speed is 25 MHz. It can be throttled back for lower power consumption.

The MSP430 also uses six different low-power modes, which can disable unneeded clocks and CPU. Additionally, the MSP430 is capable of wake-up times below 1 microsecond, allowing the microcontroller to stay in sleep mode longer, minimizing its average current consumption. There are, however, limitations that preclude its use in more complex embedded systems.

The MSP430 does not have an external [memory bus](https://en.wikipedia.org/wiki/Memory_bus), so it is limited to on-chip memory (up to 512 KB [flash memory](https://en.wikipedia.org/wiki/Flash_memory) and 66 KB [RAM](https://en.wikipedia.org/wiki/Random-access_memory)), which may be too small for applications that require large buffers or data tables. Also, although it has a DMA controller, it is very difficult to use it to move data off the chip due to a lack of a DMA output strobe. The only problem that this board has is that it is on the larger side. For this project to work correctly the board needs to be on the smaller side, the board needs to be smaller than two inches long, or it will be too heavy to keep on your key ring.

**3.2.1.1.2 CSR101x**

Another microcontroller that is relevant to this project is the CSR101x. CSR is the industry leader for Bluetooth Smart, enabling Bluetooth Smart devices to transfer simple data sets between compact devices thereby opening up a whole new class of Bluetooth applications such as keyboards, mice, medical sensors, fitness training equipment, watches, TV remote controls, automotive keyless entry, advertising, indoor location, smart energy appliances and proximity tagging.

Some of the features that this microcontroller has are that it has BLE 4.1 radio with direct single ended 50Ω antenna connection. It also has up to 4.4V direct supply connection for Li-poly batteries, which can be easily changed. The dimensions of this microcontroller are 5mm x 5mm x .6mm. This size would be perfect for the project we have planned, but the availability of this microcontroller is very limited. You have to contact the company directly, and tell them what you plan on using their microcontroller for, then they have to review it, and then they will let you contact a distributor. Also the software development kit can only be used on a computer that is running Windows 7 or later.

**3.2.1.1.3 ATMEGA328P**

The last microcontroller that is relevant is Atmel’s ATMEGA328P. The way that we discovered this microcontroller is by examining a development board called the Light Blue Bean that is made by Punch Through. The Light Blue Bean dimensions are 1.79 x 0.80 x 0.33 inches, but if you want to make Bean even smaller, you can cut off the prototyping area that sits next to the GPIO pads. You can program this chip wirelessly by using Bluetooth, all you have to do is write your code in Arduino and send it to the Bean Loader app, which is free and easy to download. This app will then wirelessly send the code to the Bean through Bluetooth Low Energy.

This chip also has a built in RGB led, cross platform programmable pins, and an accelerometer/ thermometer. A lithium-ion coin cell battery powers this chip. This chip Compile and upload sketches with Bean Loader for Windows, OS X, iOS, and Android. Bean also has SDKs for iOS, OS X, and Android for native app development. Arduino sketches run on the ATmega328p microcontroller. This is the same part that is common to most original Arduinos and provides the most compatibility with example code.

This chip is very handy because of the programmable pins that it has, if you need anything that isn’t already on the board you can solder and wire the components directly onto the microcontroller. There are 8 I/O pins two of which are analog and 6 of which are digital. Below that there are 34 pins for the prototyping area. All of these features make this chip very close to one that we are searching for.

**3.2.2 Wireless Communication**

Nowadays, wireless communication is present everywhere and encountered/used by everyone every day. By definition, a wireless communication is the transfer of information between two or more points that are not connected by an electrical conductor. As a group, we had a lack of knowledge about how to construct a communication system and which option currently available to us will be the most ideal candidate. This technology is a necessity for this project because we need the tracker to connect to a mobile phone and be able to communicate with the phone.

We needed to conduct extensive research for this topic so we could get a better understanding on how to complete the task for this component of the project. We will begin by exploring the different options available to us for this project and look at the advantages and disadvantages to each of the different technologies and choose which option according to this. The way we narrowed down which options we believed would be the best for this project is to look at similar types of projects currently available and look at what kind of technology they used and explore those options in greater detail. Similar projects that we found seemed to focus on three different types of wireless communication: Bluetooth, Wi-Fi and GPS. These will be the three options that we decided to explore in more detail.

**3.2.2.1 Bluetooth**

Bluetooth was the first of the three communication options that we explored. Bluetooth is a wireless technology communication standard for exchanging data over short distances using short wavelength radio waves in the ISM band ranging from 2.4 to 2.485 GHz. The feature of sending data over short distances is appealing due to the fact that FAUX will have a small range to find items so a large range is not necessary for this project. Another feature that is a huge positive for this wireless standard is the low power consumption associated with it. Bluetooth will operate between frequencies of 2400 to 2480 MHz and can build secure personal area networks (PAN).

This wireless communication will allow connection between Bluetooth compatible devices. This is not a constraint because in present day, all smartphones and tablets computers are capable of communicating via Bluetooth. This is a must for our project because communication between FAUX and an application will be the only means of being able to receive the whereabouts of the misplaced item in order to recover it. Being able to be paired with a device (or multiple devices which is what Bluetooth can also offer) ensures that there will always be a communication medium open for use.

The two major features we had to take into account when choosing a wireless communication option are battery consumption and range of communication. As previously stated, Bluetooth has a low power consumption used to communicate over short distances but the exact distance is dependent on the amount of permitted power. Bluetooth created 3 separate power classes with their corresponding range. These three classes are illustrated below in Table 2.

|  |  |  |
| --- | --- | --- |
| **Class** | **Maximum Permitted Power** | **Range** |
| **Class 1** | 100mW | ~100 meters |
| **Class 2** | 2.5mW | ~10 meters |
| **Class 3** | 1mW | ~1 meter |

Table 2: Different Bluetooth Classes

Companies who decide to use Bluetooth as their means of communication typically choose between Class 1 and Class 2 because the range for Class 3 is so small that is not even worth considering using it as a means of communication. Mobile devices are installed with Class 2 so if Bluetooth was chosen, FAUX would have to utilize Class 2 Bluetooth which would give the tracker a range of approximately 10 meters or about 30 feet which we would have to discuss and decide if we believe this is a feasible range for our project.

Security must be taken into account for our project as well. Since locations are being used, we need to ensure the system is secure so only the intended users are able to access this information. The last major security breach for Bluetooth communication dated back to 2005 with no recent major concerns as of recent which is a positive. We believe this is reason enough to believe the security of this communication method has improved and we can rely on this communication method to be secure enough for our project if chosen.

**3.2.2.1.1 Bluetooth Low Energy**

As we continued researching the topic of Bluetooth we came across a variation of the wireless communication method called Bluetooth Low Energy. Bluetooth Low Energy (BLE), marketed as Bluetooth Smart, is a wireless personal area network technology that is similar to Classic Bluetooth but provides considerably reduced power consumption while maintaining a similar communication range. Bluetooth Smart’s improvement in the current technology also helped gadgets stay paired longer, which is a huge plus. For example, what if an item is misplaced but is not realized until a much later time? You need to ensure that the pairing between FAUX and the smartphone/tablet computer must continue to be linked for the longest duration possible. If the pairing is dropped, it will be impossible to use the tracker to find the item. According to studies, Bluetooth smart devices can remain paired even when they are not used for hours or days at a time. This would have two major positive impacts for our project. This first being that once the user pairs the items for the first time, we will not have to keep revisiting and refreshing the connection. This drastically reduces the chance of human error, like if they forget to refresh the connection, and makes it quite simpler for the user so they do not have to always check on the connection. The second is that the devices would not have to communicate daily in order to maintain the connection since the technology allows for the devices to maintain little to no communication but the pairing will still remain.

This new technology was first created in 2006 and was adopted into the Bluetooth standard in 2010. This concerned us because we did not know if that was sufficient enough time for items to adapt to the new technology and support this communication method. Fortunately after further research, this communication method has already been implemented in all the top marketed items including iPhones, Android phones and MacBook computers. One problem that might be encountered using this communication option is if someone has an out of date phone. Although the most popular phones support this, previous models do not. For example, Apple only started implementing this means of communication in the iPhone 4S model and every model after that. Using this technology would cut out anyone who has an old phone that cannot support this option. This will have to be taken into account when making our final decision on communication option.

Using this technology would be ideal for the size constraints of our project. The introduction of this technology makes it possible to power small devices with a tiny coin cell battery that could last for years. A coin cell battery has been a major option explored for powering our project due to the cost but more importantly the size of the power supply. As the name suggests, the supply is about the size of a coin and would be ideal for powering an item that needs to be carried around by a user. Due to the fact that the connection will stay in power-down mode (when no communication is occurring, the link will sleep until communication needs to occur and the link will wake for use), power consumption is low and happens at a slow rate so small batteries will thrive in this environment because not only is a huge power supply not needed but the battery will last for a while since the consumption happens at such a slow rate.

Now that the fundamental idea of Bluetooth Smart has been presented along with some key features that would improve our project design, we can now compare it to Classic Bluetooth in the entirety of both. Doing this will allow us to look at all of the features for both compared to each other and see which one has the advantage over the other. If Bluetooth is chosen as our means of communication, we will need to know what type of Bluetooth would be best suited for our project in order to maximize performance and efficiency. Table 3 describing the differences between the two types of Bluetooth is shown below.

|  |  |  |
| --- | --- | --- |
| **Specification** | **Classic Bluetooth** | **Bluetooth Smart** |
| **Maximum Range** | 100 meters | >100 meters |
| **Latency** | 100 milliseconds | 6 milliseconds |
| **Power Consumption** | ~1 W | ~.01 - .5 W |
| **Min. time to send data** | 100 milliseconds | 3 milliseconds |

Table 3: Classic Bluetooth vs Bluetooth Smart

Table 3 shows the specifications that we believe will be major choosing points when determining which form of communication we will choose. Bluetooth Smart shows to have advantages in all the categories we marked as important so, it is the clear winner between the two. As previously talked about in the Bluetooth section, security is an important aspect for our project as well and we must focus on keeping data between the user and FAUX private. Bluetooth Low Energy adopts the encryption, authentication and authorization security from Classic Bluetooth, which has been proven to keep data confidential.

Up to this point so far we have only discussed the advantages of using Bluetooth, weather is be the classic or smart version, but we need to look at the downsides in order to make the appropriate decision given all factors. For both types of Bluetooth, data rate is the major disadvantage of using this technology. Transmission speed for data using Bluetooth ranges from 1 to 3 Mbit/s, which is fairly slow. Even though this could be a deciding feature in some project, we decided that this disadvantage will not play a huge role in our decision. We based this on the fact that we will not have a lot of data to transmit over the link because we will only be sending a few pieces of information, like proximity, location or distance to the tracker. If the project required a mass amount of information to be sent at any given, this would be a major drawback from choosing any Bluetooth.

**3.2.2.1.2 iBeacon**

Another technology we encountered when researching about Classic Bluetooth is the beacon communication technology. iBeacon is the name given to Apple’s version of this technology, we will refer to this technology as beacon communication but name the section iBeacon to be consistent with our findings in the research. Beacon technology is a class of Bluetooth Low Energy devices that broadcast their identifier to nearby portable electronic devices. This meets the need for a user because a portable electronic device is needed for this technology and our assumption is that all users will have this. The technology behind iBeacon is a little different than Bluetooth. Beacons will transmit their universally unique identifier (UUID) to the local area. Once received, software on the receiving device (like a smartphone) will look up the beacon and can perform various functions involving said beacon.

One main feature that made beacon technology appealing for our project is the ability to implement an indoor positioning system as well. Some people rarely lose items when they are in public places because they will take extra precaution when in an unfamiliar environment but if they are at home, items could be misplaced within the vicinity and take a great amount of time to find. With the beacon technology, the broadcaster (the beacon) will broadcast its location to the receiver making finding a misplaced item indoors a simple task. Beacons can be thought as small Bluetooth Low Energy devices that broadcast a small amount of data every second. This means that all the advantages and disadvantages discussed in the previous section will be inherited by this section but new advantages and disadvantages can be considered.

As discussed above, an advantage of using beacon technology is the ability to track indoors as well as outdoors, but another advantage to using beacon technology is the security of using beacon devices. Since beacons are used as proximity detection devices that broadcast outbound signals, there is not inherent security risk in the transmission. This is because these signals are waiting to be found by the intended receiver before actions take place. The risk of this technology is within the apps that use the signals.

Every communication option talked about so far has compared advantages and disadvantages of using the option, so we will continue the trend. A major disadvantage of using beacon technology is not the battery of the broadcasting device but the receiving device battery consumption, which more than likely will be a smart phone in most situations. Reports have shown that older phones tend to draw more battery in the vicinity of beacons while newer phones can be more efficient but will still see the battery being drained slightly faster than normal. Although an improvement in technology has seem to rid this problem. There has been a drastic improvement in battery consumption for the most recent phones (iPhone 5S and equivalent phones) versus older phones. As a group, we all have phones that are considered to be in the range of when this battery consumption improvement occurred so this will not affect any of us. If other people were to use our device (if implemented with this technology), they may face the problem presented. As a group we decided to ignore this problem because as time passes, people will be upgrading to newer phones and the problem will sooner or later be nonexistent. Another disadvantage to be considered is signal interference. Beacons can be considered similar to radio transmitters that stand the chance of interference since they can be absorbed by everyday items including water and air.

This technology as stated previously can be considered as a Bluetooth Low Energy device. This implies that it is just another form of Bluetooth, so if Bluetooth is what we choose as our end product, we have three separate options to choose between. There are multiple way of implementing a beacon for this project. Beacons are the most recent technology to this date that we have explored (Apple introduced beacons in 2013 and released them to the public in 2014) so it is a fairly unexplored technology amongst the public. There are devices out on the market that are marketed as beacons which is a mini chip that will do all the process talked about in this section. Along with this, you could also build beacons using other components including a microcontroller and a USB Bluetooth Low Energy plug in.

To summarize all of the research done on Classic Bluetooth and Bluetooth Smart technology, a list of advantages and disadvantages of using this wireless communication method is illustrated below.

Advantages:

* Low Battery Consumption: Has a very low battery consumption causing the battery to not require to be a giant power supply but rather could be a coin cell battery
* Very cheap to implement
* Has low latency and time to send data causing transfers of data to not require a lot of time
* Range: Depending on the Class of Bluetooth being used, the range of communication could reach 100 meters or greater
* Integration: All smartphones nowadays are implemented with Bluetooth and can be paired with other Bluetooth compatible items

Disadvantages:

* Data Rate: The speed of transfer of data using any Bluetooth implementation is fairly slow when compared to other types of wireless communication
* Strength of Signal: Can easily be interfered with like radio signals and depending on the current power of the broadcasting the device, the strength of the signal could be altered
* Receiver Battery Life: Must take into account the amount of power being drained from the receiver
* Receiver: If Bluetooth Smart technology is chosen, people with out of date phones will not be able to take advantage of FAUX due to the fact of the lack of implementation within the hardware

Researching about Bluetooth opened the doors to all of the other expansions that we found in the process. A lot of advantages made this option appeal to our group and we believe that the disadvantages can be overlooked due to the minimal effect they will have on this specific project. We still have to consider other wireless communication options that are available before we make a final decision on which option will be best suited for our project.

**3.2.2.2 Wi-Fi**

Wi-Fi was another protocol taken into consideration as a wireless communication option for our project. Wi-Fi is a very popular technology that allows for data to be transferred wirelessly using radio waves over a computer network, like a high-speed Internet connections. Wi-Fi is also known as any wireless local area network, or WLAN, and is based on the Institute of Electrical and Electronics Engineers’ (IEEE) 802.11 standard. Wi-Fi can operate on either 2.4GHz or 5GHz super high frequency ISM radio bands. Wi-Fi provides a secure and established connection that can manage congestion on a network and reduce the errors on the connection. This form of wireless communication has become increasingly popular in smartphones or tablet computers, which are major components needed for our project.

Wireless communication was a necessity for our project. If an item is misplaced, it is safe to assume that any wired connection that could have been connected before has been disconnected. The present popularity of Wi-Fi is a major advantage for this source of wireless communication. Most places you go nowadays will have their own WLAN that anyone within the area can access. Wi-Fi also has one of the highest data transfer rates for wireless communication. Although Wi-Fi offers convenience and productivity, there are some severe disadvantages of using this as our choice of wireless communication. In the past, some of the most commonly utilized encryption methods, like the WEP, had known weaknesses and any dedicated hacker could easily compromise the network. Since then higher quality protocols have been added including the WPA (2003) and WPA2 (2004). There has been serious security breaches as recent as 2007 and these need to be taken in account. For our specific project, our application needed to send the proximity/location of the lost object to the user. If the network is easily breached and someone else was able to retrieve the location of the object, it will increase the likelihood that the object will be stolen by an anonymous person and unable to be retrieved by the proper owner.

Another major disadvantage to using Wi-Fi is battery consumption. Due to the constraints of our project needing to be small enough to put on valuables including keys and a wallet, the battery being used will be a small lithium coin battery. Studies have shown that Wi-Fi consumes more power than other wireless communication sources. The only way that our project will succeed is if they tracking object will stay alive, meaning it still has battery power. If Wi-Fi will cause a significant decrease in the battery life, this option may not be suited for the needs of this project. Below are bullet point lists of the advantages and disadvantages of choosing Wi-Fi as the wireless communication source for our project

Advantages:

* Range: Wi-Fi has a range of about 20 meters inside and a greater range outside
* Integration: All smartphones nowadays are capable of connecting to wireless networks if they sense they are within the proximity of an access point.
* Fairly low cost
* Had high data range and the signal is not easily blocked by obstacles

Disadvantages:

* Security: Known breaches in the past, we cannot let the information sent between the user and the tracking device be compromised and available to non-authorized recipients.
* Battery Consumption: Has high battery consumption causing the small battery on the object to drain faster than desired, could result in tracker not being able to fulfill its duty
* Strength of Signal: Depending how far away you are from the device, the strength of the signal could be too weak to receive data from the object which defeats the purpose of the tracker using this form of communication

After conducting our research for this wireless communication option, we believe that although there are some key features that would make this viable, there are some setbacks that we consider to be major enough to make us reconsider using this option. We had to remember that a connection would have to be established between the tracker and the smartphone in order to share locations and if the object has already been misplaced, it may be almost impossible to establish this connection. Other disadvantages that we considered have already been listed in the bullet points above. We decided to still consider this option when making our final decision but the research conducted allowed us to see the positives and negatives of using this wireless communication option.

**3.2.2.3 GPS/GSM**

Global Positioning System, or GPS for short, is one of the most useful technologies out there for location based needs. Being able to use GPS to find the exact location of something is becoming increasingly popular in technology today and it is a feature that may be a good implementation in our project. We put this under the wireless communication section although we did not believe this would be able to stand alone as the only means of communication so it would have to be paired with either Wi-Fi or Bluetooth to communicate between the transmitter and the receiver. GPS is a familiar technology to almost everyone because of all the advantages it presents. Being able to simply type in a destination location and having the technology guide you there step by step makes for any adventure to be stress free and simple. If this technology is incorporated into our project, the user would be able to pinpoint the exact location of the lost item using satellites set up for the GPS.

As all of the other communication methods talked about, the technology is already implemented in all current smartphones so we would not have to worry about the receiver experiencing issues. As a group, we had very little knowledge of how GPS worked other than it involving satellites to transmit location to a certain receiver but after some research, we did learn that the information received from the satellite is used to solve a navigation equation. The equation is showed below.



In the equation, x, y and z are the location components of the satellite and the variable I represents the satellite being used and the time sent. The number of satellites has to be at least 4 in order to get a sufficient location for the module. Although the technology is simple and most everybody knows how to use it, there are some disadvantages that we must take into account. Battery consumption has been a main focus for our project and GPS is a very power hungry technology. Most GPS systems will only last between 8-24 hours before the battery dies. This is a huge flaw for our project and if we incorporate this, we would have to explore the idea of a rechargeable battery because replacing the battery daily would be incredibly inefficient. GPS would also not be able to work indoors, so implementing an indoor positioning system would be impossible for this technology. As well as not working indoors, GPS can be affected by large buildings.

The accuracy is related to the quality of signal reception, in other words, the larger the antenna the better the signal. Again, since we have size constraints this has the potential to be a problem for us if implementing this technology. We would only be able to implement a miniature antenna to go along with the handheld project so accuracy may be an issue. These faults in accuracy are reported to cause anywhere from 5 to 10 meters off the actual location. For a location to be off by such a mass amount is major fault of the project and could lead people to believe that FAUX does not work accordingly and could take further action.

As of this point in our research, we are leaning towards Bluetooth over Wi-Fi so we decided to compare GPS to Bluetooth. Table 4 (below) showcases different specifications and will compare GPS and Bluetooth for each of them.

|  |  |  |
| --- | --- | --- |
| **Specification** | **Bluetooth (iBeacon and BLE)** | **GPS** |
| **Range** | 100 meters | Unlimited |
| **Accessibility** | Smartphones have this technology built in their hardware | Smartphones have this technology built in their hardware |
| **Battery** | Low Battery Consumption | Very high battery consumption |
| **Accuracy** | Fairly accurate, can be affected by certain factors | Accurate, can be affected by certain factors |
| **Security** | Authentication, Authorization and Encryption | Authentication and storage techniques |

Table 4: Bluetooth v4.0 vs GPS

If we decided to implement GPS, it may have been a good idea to add a GSM module as well to the project. Global System for Mobile Communications (GSM) will allow access to mobile communications network. GSM can also determine an object’s position by using signal strength and triangulation from base stations, but that would not be the purpose of implementing it with GPS. Incorporating GSM into this project would allow it to send text messages. Basically we can tell the project to send a text message to the owner if the item gets a certain distance away from the owner. This distance would be some number that we decide to be a sufficient threshold to determine if an item can be considered lost. The GSM module would need an SMS (Short Message System) module integrated or be connected to one in order to send text messages.

SMS would allow messages of 160 to 224 characters to be sent. This is completely enough characters for any message we would send to the user. An example of a message that could be send could be “You have misplaced your tracked valuable” or “The location of your valuable is” and we would include the location of the item along with the message.

To conclude this section, we will similarly do what we did in the previous sections and have two lists showing the advantages and disadvantages of using this technology.

Advantages:

* Integration: All smartphones nowadays are capable of using GPS due to the fact that they are built into the hardware
* Fairly low cost
* Range: Has unlimited range due to using satellite technology

Disadvantages:

* Battery Consumption: GPS drains the battery of the device at an extraordinary rate which could mean that we would have to deal with often battery replacement or rechargeable batteries
* Indoor Capability: Would not be able to access items indoors
* Inaccuracy: some locations may exist where GPS does not work well

After researching about GPS, we were very happy with the features and advantages that this technology would allow us to incorporate but there were still major things that we would have to take into account. The battery consumption issue was too much of an issue to ignore so we would have to consider this when deciding power supply if implemented. Indoor capability is also something we wanted to explore since it’s a fairly rare feature for trackers and using GPS would not be able to create this indoor tracking system. GPS does present us with a mass amount of advantages that would be ideal but when making our final decision on implementation, we need to look at the drawbacks as well, especially the battery consumption issue.

Now that all the wireless communication sections had been researched and all of the advantages and disadvantages had been discussed, our group decided to choose Bluetooth as our means of communication. For the purpose of this project, Bluetooth has all the features that we require since we don’t need a huge range, we need low battery consumption and the ability to pair with a phone will help our project tremendously. The option of adding a GPS and GSM module could still be added as a later feature of the project but that will only occur if we feel it is necessary to implement it. As explained, there are three different types of Bluetooth that we presented, the type of Bluetooth we chose and the product we will use will be explained later on in the document.

**3.2.3 LED/Speaker/Power**

**3.2.3.1 LED**

When brainstorming ideas about how the users of our device would locate their lost items, the idea of implementing an LED into the system came up. The goal was that if the user was searching in a dark area, that an LED lighting up would help them locate the device more easily. Another benefit of using an LED in the design would be if the user knew the color light they were looking for then they could tell the device to flash the LED, which would make searching for the device more efficient and easier for the user. Knowing the color of the light in the device helps the user look in their environment for that specific color which aids in their search. The LED would also help if the user was searching in an open field for example. The light coming from the LED would make searching for your keys in a wide area of land that much easier.

Recent advancements in LED technology have made them more efficient and cheaper to manufacture than ever before. This makes them an excellent choice for use in our system because of how little energy they use and because of how cheaply they can be purchased for our project. Most typical LEDs today operate using 30-60 milliwatts (mW). A red LED would be a good choice to use for our design due to red LEDs having the best efficiency coefficient compared to others and also requiring a small amount of voltage (approximately 2V). A 20 milliamp (mA) standard LED operating a 2V would only use about 40 mW, and the LED would only be used when the user is looking for the device. Typical lifetimes of modern LEDs are quoted between 25,000 to 100,000 hours depending on the wear and strain put on the device. The LED in our system would not be used relatively extensively, therefore, there should never be a need to replace any LEDs used in the device. A white LED would be another possible choice for our project due to the intensity of light that a white LED can produce compared to others. This would be beneficial to the user, however, the white LED does consume more power.

Table 5 shows different colors of LEDs and their minimum and typical forward voltages. The forward voltage is the voltage drop through the LED, when the nominal current is flowing. This chart will help to decide which LED is best for our design and power constraints.

|  |  |  |
| --- | --- | --- |
| LED Color | Forward Voltage (V)  @ 20 mA (Min) | Forward Voltage (V)  @ 20 mA (Typical) |
| Red | 2.1 | 2.4 |
| Yellow | 2.1 | 2.4 |
| Green | 3.4 | 3.8 |
| Blue | 3.4 | 3.8 |
| White | 3.4 | 3.8 |

Table 5: LEDs and their forward voltages

**3.2.3.2 Speaker**

An implementation of a speaker was another possible feature that would act in a similar fashion to the LED feature. It would allow the user to more easily track down lost or stolen belongings by adding a noise to the device. Allowing the device to utilize the many different senses of the human body has many advantages when it comes to looking for things. For instance, if the computer that is being searched for is completely covered by clothes, for example, then the user would not be able to locate it with the use of only the Bluetooth or the LED. The Bluetooth locator would get the user to the general location of the item but it could still be lost in a sea of other things in the room. The LED would not help in this situation either due to the device being covered. A speaker, however, that could be activated to beep, would lead the user directly to exactly where the device is located using auditory clues.

Having a speaker on the device may or may not be viable depending on power and size constraints. However, there are many small speakers available online for purchase in the range of three dollars or less, with many as low as a dollar. Most of these speakers have a power rating ranging from 200 mW to 500 mW. Acquiring one of these speakers for use in the design would not be difficult due to their low cost, but implementing one into use might prove a little more challenging. There would also need to be a way for sound to leave the housing of the device; a hole or some other form of outlet, for example, would suffice.

When researching speakers used in electronic devices, we came across what is known as the Piezo speaker. These speakers differ from the standard magnetic speaker in that they do not require a magnet to operate. Piezoelectric speakers are very easy to drive compared to other speaker designs; they use materials that demonstrate the piezoelectric effect to generate sound. When an electric charge is applied to the material, the material will produce an internal mechanical strain, of which can be used to output a sound. These speakers usually cannot produce a wide range of sounds from the audile spectrum, but they can produce a certain sound very loudly and with little effort to do so. These types of speakers are generally used as beepers in electronic devices such as watches or pagers. They are also resistant to overloads that could destroy other high frequency drivers.

**3.2.3.3 Power**

When coming up with how this device was to be powered, the first thought went to what it commonly known as the coin cell battery or the button cell. These batteries are non-rechargeable, but can last a long time in small devices that do not consume a considerable amount of power all the time. Like many of the other components for this system, these coin batteries can be obtained relatively cheaply. Another benefit of using this type of power source is the variability in the amount of voltage that can be supplied. There is a 1.5V or a 3V battery that can be chosen based on the needs of the system. Furthermore, these batteries are very small with the most common one being only 20 mm in diameter and weighing less than 2.5 grams.

Another option is to use a polymer lithium ion battery. These batteries, depending on the one used, have the potential to last much longer than the coin battery. However, there are drawbacks to this increase in battery life. For the same amount of amp-hours these batteries are generally bigger than the coin batteries. Being bigger means that they, on average, weigh more per amp-hour than the coin batteries. The weight and bulkiness of the device is a major factor that needs to be closely monitored. Every extra gram can add up and contribute to the overall weight of the system. The device also needs to remain small enough so that it can be used on small personal belongings such as keys and wallets. The lithium ion batteries are also more expensive than their coin counterparts, which must be taken into consideration when choosing parts. Also, lithium ion batteries are at a much greater risk of failures that result in a mess. Any problems that arise with the battery, such as over-discharge, short circuit, being crushed, or nail penetration can result in the battery rupturing or even starting a fire. These risks must be considered when choosing what power source should be implemented.

**3.2.4 Protective Casing**

The protecting casing is an important piece of this project. The case is so important because it will have to house all of the delicate electrical equipment. All of the competitors of FAUX have their own specific protective casing that makes them look different from the others. For example The Tile tracker has a case that makes it look like an actual tile. One of the things that set TrackR apart from the rest of the competition is that it offers their trackers in a variety of colors, and you could also get the protective casing engraved. When we build FAUX we would like to be able to something along those lines. We are going to try and make the case in the shape of an oval, which will set us apart from the competition.

There are some limitations that we put on the protective casing though; it must be water resistant, it must be relatively easy to open to change the battery, it must be clear enough to see the LED through it, and it must be thin enough so that we can hear the speaker clearly through it. The reason that we wanted the protective casing to be water resistant is so that water can’t easily get inside of the case and damage the delicate hardware. Because the part of Florida that we live in is considered one of the rainiest of the entire state, we don’t want the entire device to be ruined by a little bit of water splashing up and hitting it. Water resistant does not mean that you will be able to completely submerge the entire device in water and expect it to not be damaged at all. If that were the case then our device would be considered waterproof, but out other limitations would prevent this from happening.

The protective casing must be relatively easy to open so that it will be easy to change that battery. Some of the competitors make you send the device back to them because their batter is not accessible without special tools. This causes you to have time without your tracker, and they would have you purchase a completely new tracker to replace the one that you sent back to them. There are two methods of doing this that we are considering doing; the first one is using a couple screws to hold the two sides together so all you would have to do is remove the screws to change the battery. The next method would be that the pieces snap together, so you would just have to unsnap them to be able to change the battery. The method that we choose will directly affect how waterproof the device is.

The next limitation that we have is the material and color that we are going to use. We won’t be able to use any metals or any materials that are completely opaque. We won’t be able to do this because we want the device to light up when the user is looking for it. This will be accomplished by sending a Bluetooth Low Energy signal to FAUX, once it activates the LEDs on the inside will light up. The reason that we can’t use non-translucent materials is because you would not be able to see the LEDs light up from the proper distance away. We also don’t want the protective casing to be completely clear either. We would not want the user to be able to see what is on the inside of the device.

The last limitation that we have for the protective casing is the thickness of the casing. The pieces of the case could not be too thick for two reasons: first we are using a speaker so when the user activates the device we would like him or her to be able to hear it. The speaker will be a huge help to the person that is using it because instead of just engaging one of the user's senses we will be engaging a second one. Secondly if we make the case to thick that risks the device being too heavy to comfortably put on your keys, or any place that you plan on carrying it. We want this device to be as light as possible, so the smaller we make it the less it will weigh, and the happier the consumer will be.

**3.3 Software Implementation**

The project called for both software and hardware components to satisfy the requirements for this project. In order for this project to be successful, software needed to be implemented for communication purposes. Having standalone hardware would not be sufficient enough and the goal for our project would not be satisfied. A product software implementation method is a structured approach to integrate software based services into the workflow of an organizational structure.

This project consisted of multiple separate hardware components that required them to communicate between each other in order for them to work together. The LED and speaker needed to work when the user choices them to do so and this can only be accomplished by having software implementation to allow them to interact when needed. The hardware component (LED and speaker) allowed for the action to happen whether it to be lighting the LED or sounding the speaker, but we need the action to take place when the user desires it to happen which is where software came into play.

We had no experience as a group with implementing software to hardware components so we had to conduct research in order to understand how the process works. We needed to learn how to create an application that would allow all the hardware to be able to communicate in one simple interface. All of us have taken programming classes while attending school and programming would not be an issue to learn but it’s implementing it into a mobile environment and ensuring that all pieces work fluidly. Our research for software included learning about mobile application design and how to implement hardware components into a singular environment to allow communication as a whole.

**3.3.1 Mobile Application Design**

All of the trackers mentioned in the related projects section are paired with a mobile application that is used to locate the item using the application downloaded on either a tablet computer or smartphone. This led us to believe that a mobile application is a mandatory feature for our project (we were already planning on incorporating one for FAUX).

We have multiple features to be included in our created mobile application. The main feature of our app is to locate the lost item and send this information to the user in some sort of display. The whole purpose of the project is track lost items so the location feature of the app is a must. As talked about in our previous section, we also decided to implement a speaker and LED as well in our project which we will need to also incorporate features in app. We needed to have some sort of button within the app that will turn the LED and speaker on and off. This is a feature that most of the products on the market don’t have and will set FAUX apart from the group.

One problem that we may encounter is that none of us have ever created an application for any purpose. We all have experience with programming for classes using IDEs and the command prompt but creating a mobile application is a different process that we needed to learn about. Our research conducted for mobile applications included looking at the applications for the related projects and see what they incorporated in theirs but that does not help with how we will build our own.

We needed to conduct separate research about the methods and ways an individual can create a mobile application. As beginners, we believe that we may have to learn the process by creating simple applications before we can move up to creating the application for FAUX. Our application will be complex and would require extensive knowledge of the process of mobile app design that we could only acquire from experience.

What we learned from conducting research about this process is that depending on what platform we choose, whether it is iOS or Android. The process behind creating an app is different for each of them and there will be a huge impact on which of the operating systems is chosen. Also depending on the hardware chosen for our project, the commands for programming will be different a new set of instructions will have to be learned.

We had to focus on making the app simple too because we must take into account the user. Users of our product could have knowledge of technology or could have a very poor understanding of technology. If the app is complex and hard to use, the product could be unappealing to people who are not skilled in handling technology. To create a successful mobile app, we must have a simple to use app that allows all the features to be run with minimal user requirements.

**3.3.2 Programming/SDK**

The creation of a mobile application is down to using a certain software, usually an IDE, to program certain actions to be included within the application. Throughout our studies, we learned about multiple programming languages and have experience programming in each. Programming a mobile application is not as basic as opening any IDE and using any language to complete a task. After researching about mobile app design, we learned that programming a mobile app gets more complex when introducing hardware components due to the fact that their own libraries get incorporated.

A software development kit is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform or other development platforms. SDKs are typically paired with any hardware component that requires the need for software in order to implement said component. These SDKs could result in the implementation of multiple application programming interfaces (APIs) that would require for the programmer to learn new sets of keywords that need to be used for programming with the certain component.

Along with the new features that SDKs can add on to programming, the programming language that must be used might cause more learning to be necessary. According to our research, Android applications are typically written in Java while iOS applications are written in Swift. If Android is chosen to be our operating system, this will not be an issue since we have taken multiple classes throughout our studies for Java programming but none of our group members have experience with using the Swift programming language so we would have to learn a new language in order to program for iOS. Due to the fact that we all own iPhones in our group, we are leaning on choosing iOS operating system so we will need to explore Swift tutorials to get a grasp of the language.

For SDKs, there are not really any constraints that altered our choices. SDKs are grouped with hardware components when needed. For example, Arduino has their own open-source software that makes it easy to write code and upload it to any microcontroller that supports the Arduino SDK. Their SDK is based off of Java and some other open-source software. We believe that finding a microcontroller that supports the Arduino SDK would be an ideal situation because we all have had experience with using Arduino based hardware. A few of us have also worked with the MSP430 microcontroller introduced by Texas Instruments so we are familiar with that environment.

We also had no problem with learning a completely new environment because it is very similar to learning new programming keywords. A lot of the actions will be similar to other actions we’ve seen from other environments so prior knowledge will help us learn the new environment at an accelerated pace. We focused on finding the right hardware component and took the time to learn the software designed for that hardware rather than base our decision on if we are familiar with the SDK. This ensured that our choice was not biased based on previous knowledge/experience and allowed us to choose the best hardware possible for this project without any programming concerns.

**4 Related Standards**

Standards are very important in the engineering design process because they provide a level of uniformity across all designs. It is imperative that engineers know which standards apply to their projects so that their products are easier to design for different localities. Safety is another important factor when it comes to design standards. Standards help keep people safe by allowing engineers to use designs that have been tested and proven to work, and work well. Engineering standards describe the expected quality of their related products. Products that are up to par with all involved standards are critical to the engineering process. The following is a list of standards that are related to our project.

**Standard Description**

USB Universal Serial Bus

IEEE 260 Standard letter symbols for units of measurement

IEEE 802.11 Wi-Fi

IEEE 802.15.2 Bluetooth

IEEE 802.15.6 Wireless Body Area Network (BAN)

IEEE 1074 Software Development Life Cycle

IEC 60027 Letter symbols to be used in electrical technology

IEC 60038 IEC Standard Voltages

IEC 60050 International Electro technical Vocabulary

IEC 60062 Marking codes for resistors and capacitors

IEC 60063 Preferred number series for resistors and capacitors

IEC 60076 Power transformers

IEC 60083 Plugs and socket-outlets for domestic and similar

general use standardized in member countries of IEC

IEC 60085 Electrical insulation

IEC 60086-4 Safety of lithium batteries

IEC 60287 Calculation of permissible current in cables at steady

state rating

IEC 60417 Graphical symbols for use on equipment

IEC 60445 Basic and safety principles for man-machine interface

IEC 60446 Wiring colors

IEC 60574 Audio-visual, video and television equipment and

Systems

IEC 60559 Binary floating-point arithmetic for microprocessor

Systems

IEC 60617 Graphical symbols for diagrams

IEC 60884 Plugs and socket-outlets for household and similar

Purposes

IEC 60947 Standards for low-voltage switchgear and control gear

IEC 61140 Protection against electric shock

IPC-2615 Printed board dimensions and tolerances

IPC-D-325 Documentation requirements for printed boards

IPC-2612 Sectional requirements for electronic diagramming

Documentation (Schematic and logic descriptions)

IPC-2221 Generic standard on printed board design

IPC-7351B Generic requirements for surface mount design and

Land pattern standards

IPC-6011 Generic performance specification for printed boards

IPC-6013 Specification for printed wiring, flexible and rigid-flex

RS-232 Standard for serial communication transmission of

Data

UART Universal Asynchronous Receiver/Transmitter

IP Code Degree of protection against intrusion (water)

NEMA Enclosure types

**4.1 Design Impact of Relevant Standards**

A lot of these standards should not have too much of an impact on the design of FAUX. Most of the materials that were used to create the device were already created by manufacturers. These manufactures already adhered to most of these design standards when they created their devices. Also, Bluetooth is a major standard that will be followed in this project, but Bluetooth has already been standardized within the products that we will be using. Therefore, it does not have to be looked at specifically by us. We will, however, need to be careful to adhere to the standards of board design when implementing the speaker onto the PCB. Some of the other standards such as weather proofing and software implementations are discussed in the following paragraphs.

There are also standards that must be followed when creating software that should be followed. These standards talk about the syntax of software and how software should be designed in order to make it easier for others read and understand your code. For example, how many characters should be on one line, common conventions for comments, and naming conventions for variables and classes. This is especially important in our project because we are going to be creating software for the device itself, and also creating software that will become a mobile application that the device will use. It is imperative that both of these implementations are created in a professional manner and maintained with modern coding standards.

There are also some guidelines that should be followed when creating an application for an iOS device. These guidelines aren’t necessarily standards, but we think they should still be mentioned because they do affect our designs. Apple defines the themes that they think each iOS application should embody in their iOS Human Interface Guideline that can be found online. Each application should embody the following themes: deference, clarity, and depth. Deference means that the User Interface should help the user interact with the content, but not interfere or compete with that interaction. Clarity means that the user can easily see and use what is in front of them. Specifically, they can see the text easily decipher the icons, and the functionality of the application should motivate the design. Finally, the iOS application should provide depth, meaning that it should provide visual layers that increase the user’s understanding and vitality. All of these parameters should be closely looked at and followed when we are designing our application for iOS.

Another standard that will have to be closely followed is the IP Code. This is a rating that defines the degree of protection against intrusion of water into mechanical casings and electrical enclosures. The IP Code is a code given to a device which consists of the IP indication letters followed by two numbers. The first number is a number ranging from zero to six that describes the device’s solid particle protection. Particles here mostly refer to dust with a rating of zero meaning no protection and a rating of six meaning dust tight. The second number is liquid ingress protection which can range from zero to nine. A zero in this means no protection from water while a nine means the device can survive powerful high temperature water jets being sprayed on it from multiple angles for two minutes. We want FAUX to be water resistant, meaning that it can withstand water being splashed onto it, but that it cannot necessarily withstand being submerged in water or having a water jet being sprayed onto it. This is a rating of four on the liquid ingress protection scale. The focus of this will mainly be on the protective casing surrounding the device. There has not been much discussed on how anything inside the protective outer shell will be water resistant. Therefore, it is up to the casing to provide most, if not all, of the defensive shell guarding the device from the elements.

The National Electrical Manufacturers Association (NEMA) defines types of enclosures for electrical devices. They are given a score consisting of numbers and sometimes letters that defines what that level of enclosure should be able to protect against. What we are aiming for in an enclosure for FAUX would be a level three NEMA type enclosure. This type is described as weather-resistant. This includes water resistant as well as other types of weather such as falling dirt and dust, rain, sleet and snow, and the formation of ice. We want FAUX to provide this level of protection to its components so that it does not break or malfunction due to mild weather conditions. NEMA codes are similar to IP Codes but are not directly related. However, they can be mapped to IP Codes. A three for a NEMA enclosure (what we want in FAUX) can be called an IP54. Therefore, we are using this to define what we are aiming for in IP Code. In IP Code this means that it is dust protected in that some dust may enter the enclosure, but it should not affect the satisfactory operation of the device. Also, this means that the device is resistant to the splashing of water at any angle to the device.

**5 Constraints**

**5.1 Environmental Constraints**

**5.1.1 Distance Constraints**

Distance constraints became one of the major issue for this project due to the main task implemented for FAUX. Depending on what technology used, the distance will vary, ranging from 10 meters to unlimited range. Also, location will play a factor as well. Some technologies will not be able to locate items if they are placed inside due to the technology constraints whereas others will be able to locate items inside. Also, due to the fact that we will design the project to be portable so the user can carry it around, power will play a part in the distance items can be tracked. The user must be aware of these constraints before using this certain tracker as the one to use.

If a user needs a tracker that will locate an object at any specific location no matter the distance, the user will likely need a tracker that utilizes the GPS and GSM technologies talked about in the research section. These will allow the user to pinpoint the exact location of the item and have the location sent to the user’s smartphone. The user must be wary to made known that using the GPS/GSM based project will not allow indoor locating due to the restraints on the technology. If the user needs this feature, a different product would be better suited.

If a user only needs a tracker to notify the user if the item is getting out of the desired range so the user may be forgetting the item, the user could look for Bluetooth tracking devices because the range for that technology is more suited for their situation. Bluetooth has a distance constraint from 10 – 100 meters which will be enough to satisfy the user’s needs.

Our project design will be similar to Tile and other similar products related to them where they do not need a massive range. We wanted FAUX to help users not forget their lost item and if they do, make sure they realize that before they get too far away from the vicinity. Tile has a range of 100 feet and our project will have a similar range to that. Users who need unlimited range to locate the exact location of an app no matter how far away it is will not be appealed by FAUX.

We also needed to think about how structures will affect the distance and take into account all the obstacles that would cause a signal depended project to run into any issues. In our research, it was discovered that the strength of signals is weakened when passing through buildings, walls and other manmade structures. We will need to take this into account when stating the range of our product, a lost item will not always be in line of sight or out in the open and we must account for all situations. After finishing the construction of our project, we will have to test the distance constraints in multiple situations in order to see the effect of structures. We will be able to estimate a range depending on what technology we decide to use but a more precise range will only be available once testing has been done. Users cannot rely on FAUX if the distance exceeds the maximum threshold that will be defined clearly.

**5.1.2 Weather/Other Constraints**

FAUX is a system that is comprised mostly of electronic parts regarding the hardware actually in the device itself. Electronics usually do not fair too well when mixed with many elements of nature including rain or other forms of precipitation. This means that it is important to consider what kind of wear and tear we want our system to be able to handle when designing. Our design needs to focus on keeping water and other harmful substances off of the electronics. Albeit, this system is most likely to be used on belonging that users want to keep safe, meaning that they will likely not want those belongings to get wet or damaged. If they do not want those belongings to get wet and to keep them unharmed then FAUX will not be getting wet or damaged either. This is something that can be utilized in regards to keeping the system away from the elements.

Another factor to consider is temperature. The system can overheat in extreme temperatures and not function properly or even die altogether. If the system fails, then there is no point to using it and it loses all value and security that it once provided. Heat is also detrimental to battery life. Depending on which battery is chosen in the design depends on if the battery can be changed or charged. If a battery that cannot be charged is chosen, then this battery will need to last the entirety of the system’s lifespan. Heat that is draining battery life is never good when the battery in the system is not able to be exchanged. On the opposite side of that coin is the system in extreme cold temperatures. Electronics that have been in cold temperatures and then experience warmer temperatures, can cause condensation which can cause damage. Both of these factors need to be taken into consideration when designing a shell to hold the hardware components of the system.

Bluetooth devices operate over a 2.4 GHz radio frequency. Many other technologies also use this same frequency including Wi-Fi and even microwaves. These signals as well as other Bluetooth devices can cause interference with the system. However, Bluetooth signals are generally very weak compared to other and interference between devices has proven to be relatively rare. Also, most modern Bluetooth technologies use a method called frequency hopping spread spectrum in order to switch signals between many different channels very rapidly to reduce interference between signals. Because of this, signal interference should not be much of an issue, but we thought it should at least be discussed.

Also to be considered in terms of wear and tear on the device is fall damage. It is inevitable that the device will be dropped at some point by someone. The outer shell of the system should be able to withstand dropping onto a hard surface from a moderate height. The electronics within the shell should also be designed in a way as to not shift around. Shifts of any components inside the shell of the device can cause damage to both the electronics as well as the shell itself. Components inside the device need to be flush against the inside walls of the casing, which should in turn mitigate most shifts due to fast movement of the device. The outer shell of the system should also be able to withstand impacts without losing form. What the shell is to be made of is discussed in another section of this paper, but it should at least protect the inside of the device and not cause damage in case of a drop.

**5.2 Economic Constraints**

Economic constraints can mean a variety of different things when referring to a project such as this one. The first economic constraint this section is going to cover is the available budget for the project. The project and all of its costs will be covered by all members in the team. A majority of the parts have minimal cost due to the design being on a small scale. We were fortunate enough to not have to produce a large sum of money in order to fund our project, therefore, the cost can easily be distributed among the group members.

Next, we’ll discuss the economic constraint that has to do with prices of other similar products already on the market. While the main goal of the design is to make a product that works as it was supposed to, there is always an opportunity to push our designs to market as well. If things go as planned and we create a system that not only works, but works well and can compete with products already on the market, then there is potential to turn our design into a marketable product. If this happens it is important to keep in mind the costs of creating the device and what it can be sold for. This is not to say that we are trying to create a device that will go to market, but the possibility is always there.

If the device were to go to market, all user interfaces and electronic devices would need to be user friendly. The app that is going to be designed would need to be designed in such a way that makes sense for the layman user. The device itself would need to be simple and work in a simple way with minimal human interaction. Maintenance is another thing that must be taken into consideration. If the device runs out of battery, can the battery be changed? What if the device breaks? These are questions that need to be answered so that the everyday user knows what to do with the device in these situations.

**5.3 Size Constraints**

Size is the most important factor in the design of this system. It is what can make or break the product. If a user wants to put the device on something of value, this belonging is most likely to be small. Common examples are wallets or keys. Most of the time these types of belongings are pretty small already and can fit easily in a pocket or purse. Adding the tracking device to these belongings should leave the smallest footprint on them. Ideally, the user will not be able to notice a change in the weight or shape of their belongings once the device has been added.

Size reduction is the current trend in almost all modern electronic applications. The turn of the century brought in a wave of companies all in the race to make the smallest yet most powerful devices. This was able to be done because of advances in electronic technology allowing the transistor size to shrink, which in turn led to more transistors that can fit on a chip. This progressed into more powerful chips being produced on smaller pieces of silicon. This is good for our project in that we can keep our device small, while it still being capable of many things. We will most likely be using the Light Blue Bean for our project. This device is smaller than the Bean+, but it contains everything we need to implement our designs.

Of course with a small size, FAUX could easily be misplaced. However, between the app, the speaker, and the LED, it should not be too difficult to find the thing. That is the whole point of the system in the first place.

**5.4 Ethical Constraints**

It is important for us to make clear that FAUX is not a level of security that should be solely relied upon by the user. FAUX should be viewed as a helping hand in establishing a pairing with a belonging in order to keep track of that belonging.

There is also the ethical standpoint of stealing that should be discussed. If someone were to steal something that had FAUX attached to it, the system could very well lead the user to the thief. It is important that FAUX should remain as a tool to help users keep track of their belongings. This, however, could ultimately lead some users to accuse others of stealing their stuff when their belongings turn out to be missing. These sorts of issues need to be taken very seriously and should be at the risk of the user.

**6 System Design Details**

**6.1 Hardware Components**

**6.1.1 Chosen Microcontroller**

The microcontroller is the brains of this system. It controls all the other subsystems of this project. Given that size is very important to this project we want a microcontroller that is small but also powerful enough to run the whole PCB. Because we need this we chose the Amtel ATmega 328P.

It has 23 programmable I/O pins. It has 32kBytes of flash and 2kBytes of RAM. The microcontroller operates between 1.8 and 5.5 volts. You can see the pin layout in Figure 3. For the prototyping the Arduino Mega 328P is used in conjunction with Arduino IDE for programming. This proves to be a powerful tool as libraries can be used to help make controlling certain subsystems easier.

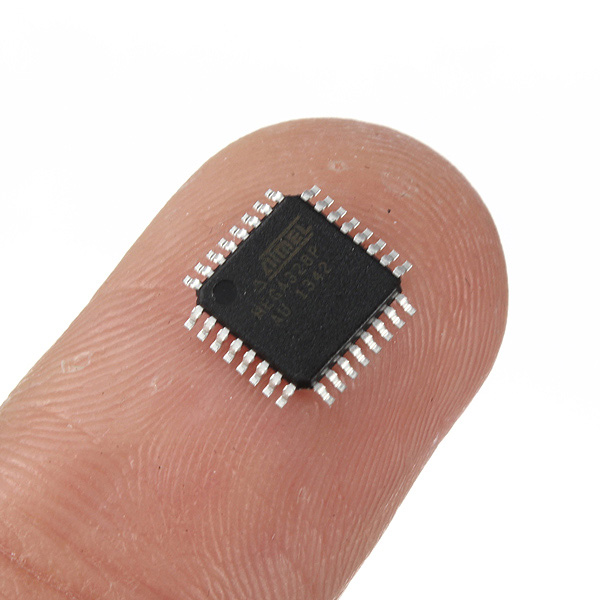


Figure **2**: Actual size of the microcontroller

The picture shown in Figure **2** shows the actual size of the Atmel ATMEGA328P. As you can see this microcontroller is very small and it did fit the requirements of our board very nicely. As you can see it has many different pins that can be programmed. We used many of these pins when creating the schematic this will be shown in Figure 3.

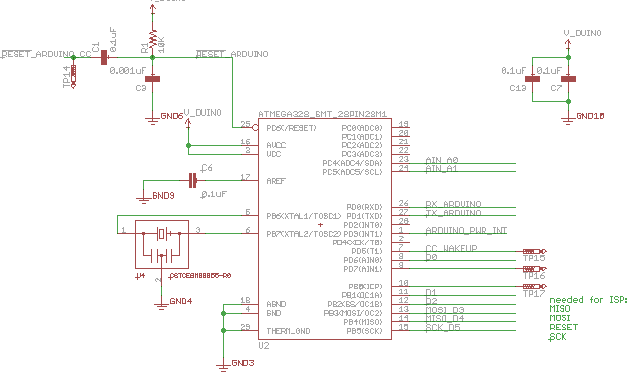


Figure **3:** PCB layout of the microcontroller

We decided to purchase this device from Punch Through. This company has great reviews from everyone that has purchased one of their devices. Punch Through was founded in 2009, Punch Through Design provides hardware and software design services. Specializing in accessory development for the iPhone, iPod, and iPad, with a proven track record of Apple approved designs. Punch Through’s portfolio includes wireless antenna design and testing, battery powered devices, iPhone app development, and embedded software design. This company is very down to earth and we can identify with this company because all of its employees are college students.

The Light Blue Bean comes with its own software program called Bean Loader that is free to download. Through this application, which is available for Macintosh and PC, you are able to wirelessly program it. The way this works is simple: first you type your code in Arduino, next you pair the Bean Loader application to Arduino, after that you can wirelessly send the program to your Light Blue Bean. The next section will go over the technical specs for Punch Through’s Light Blue Bean and reasoning for choosing this product.

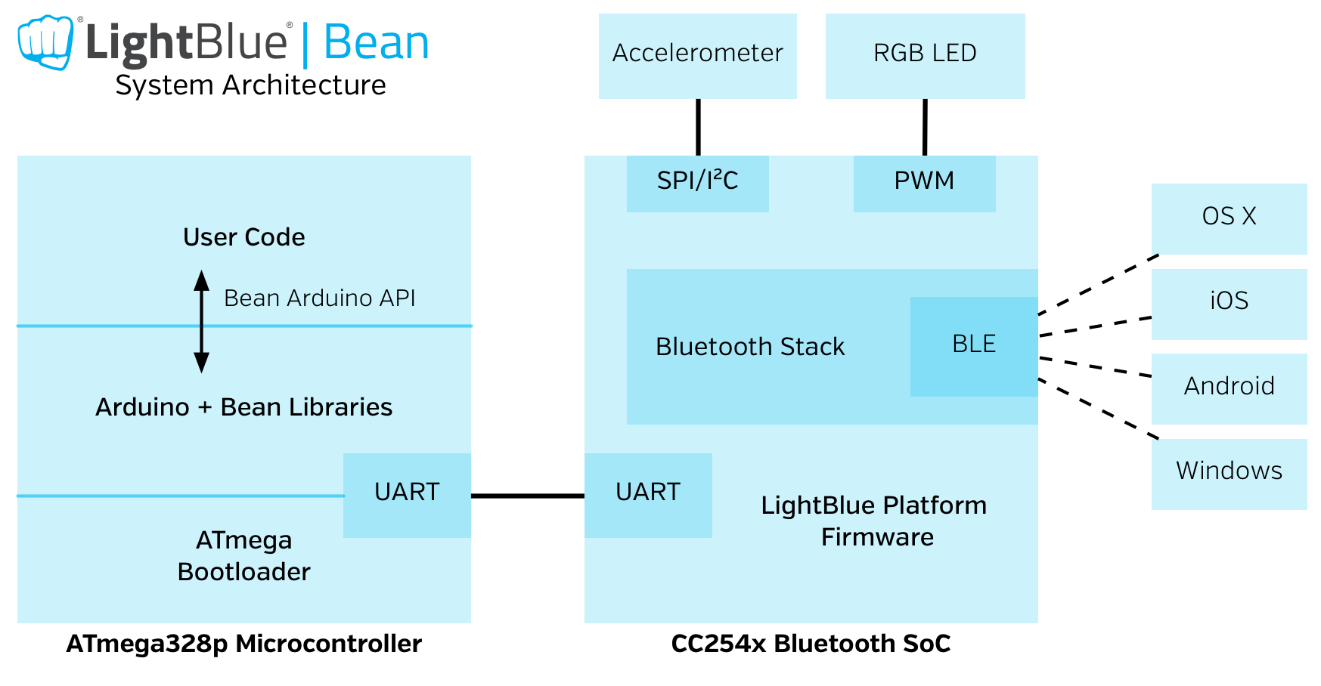


Figure **4:** System Architecture for the Light Blue Bean (with permission from Punch Through)

In Figure **4** you can see the system architecture of the Light Blue Bean. You can see how it incorporates the Arduino code along with its own Bean Libraries. It also shows that the Bluetooth Low Energy on this chip can be used in any OS X, iOS, Android, or Windows device.

**6.1.1.1 Reasoning/Purpose**

As stated previously, Punch Through’s Light Blue Bean will act as the microcontroller that will be used in FAUX. It will be the final phase in the process to find the lost object. Once the Bluetooth signal reaches the Light Blue Bean a chain reaction of things will start to happen; the on board LED will start to flash, The external LED will start to flash, and the speaker will start to make a sound. You can see the PCB layout of the Light Blue Bean in Figure **5**.

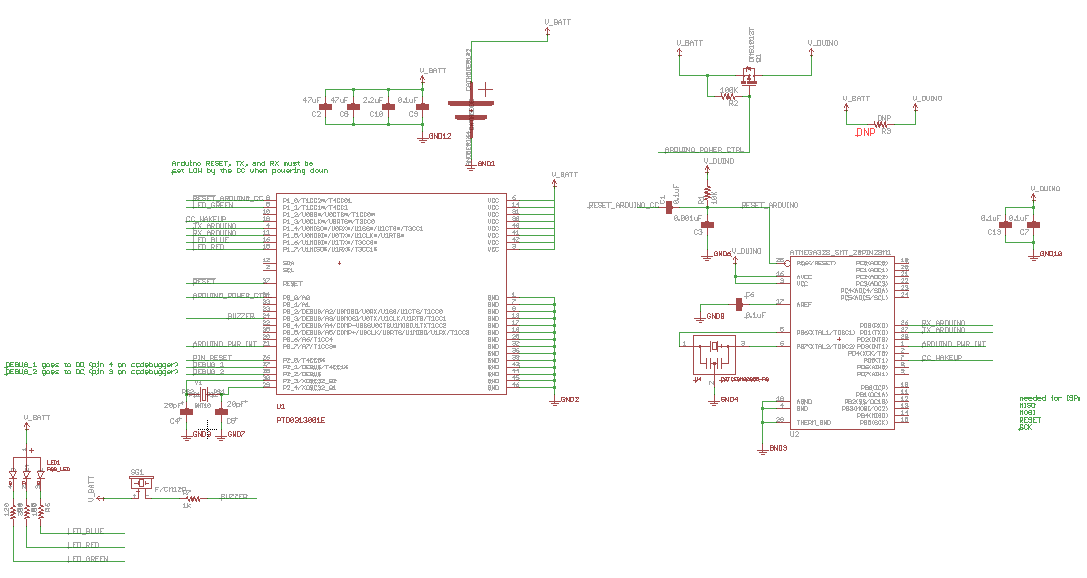


Figure **5:** PCB layout of the Light Blue Bean (with permission from Punch Through)

In Figure **5** you can observe the PCB layout of the Light Blue Bean, you can see how the on board LED is powered and how it is chosen which color you want, it feeds it through a different resistor. On the right hand side you can observe the actual layout of the chip that the microcontroller uses, the chip is an ATMEGA328. It is the 28 MLF version of this 8-bit microcontroller. The pin layout of the ATMEGA328 8-bit microcontroller can be seen in Figure **6**.

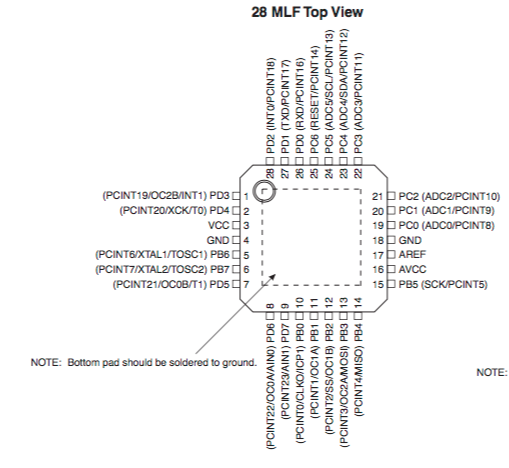


Figure **6:** Pin layout of the ATMEGA328 chip

The first thing you probably notice when you look at Figure **5** is the big square in the middle; this is the Bluetooth Low Energy Module. Some of the key features of this chip are as follows: 2.4 GHz BLE (Bluetooth Smart, Bluetooth 4.0) compliant, 8051 microcontroller with 256kB in system programmable flash and 8kB of RAM, and it has extensive development tools including IAR Embedded Workbench. This is the most important part of the Light Blue Bean, without this part it would not be able to communicate wirelessly, which means we would not be able to send the code to the microcontroller, and we would not be able to use this device to find our lost object.

The last bit of the PCB that I will be mentioning from Figure **5** is all the little circles that are at the bottom of the display. These little circles represent all of the programmable pins that are located at the bottom of the Light Blue Bean. If this microcontroller did not have these programmable pins we would not be able to use it. These pins allow is to attach outside objects onto the Light Blue Bean. As stated previously, we plan on adding a speaker onto the Light Blue Bean, so we are able to hear it when we are trying to find our lost item. We also plan on adding an extra LED on to the chip because most likely the built in LED that is already on the board will not be bright enough to see through the protective case, and even if it was it would be very dim from twenty to thirty feet away.

Figure **7** shows some of the specs of our PCB, and it also shows the connection lines on the chip.

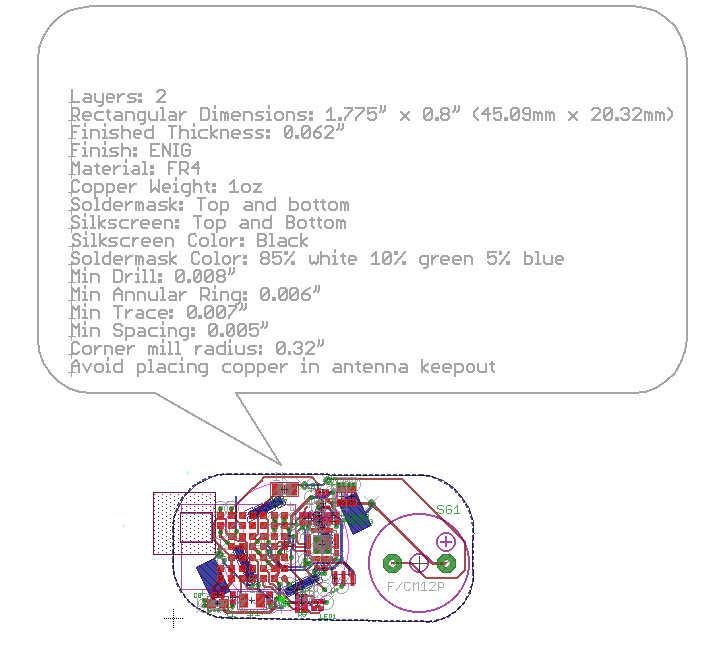


Figure **7**: Specs and wiring of our PCB

Figure **7** gives some very important information about the printed circuit board. It shows how much copper wiring is included in the board, it shows the minimum allowed spacing that is on the board, and it shows what kind of material that it is made out of. The material that it is made out of is called FR-4. The FR-4 is a grade designation assigned to glass-reinforced epoxy laminate sheets, tubes, rods and printed circuit boards. FR-4 is a composite material composed of woven fiberglass cloth with an epoxy resin binder that is flame retardant. "FR" stands for flame retardant, and denotes that safety of flammability of FR-4 is in compliance with the standard UL94V-0.

This means that our chip as a very small chance of catching on fire, which is a very good thing considering that it will be constantly on and running. The wiring that you can see in Figure **7** is very intricate, if you look at it very closely you are able to see where the different pins on the ATMEGA328 8-bit microcontroller all run to. You are able to see how everything is interconnected on to that chip, and how everything is powered and driven by that small 8-bit microcontroller.

**6.1.2 Chosen Bluetooth Module**

Our Project relies heavily on Bluetooth; we decided to use Bluetooth Low Energy (BLE) to communicate between our FAUX device and the application that we are building. The Bluetooth module that we chose is the LBM313-2540. Punch Through Design makes this Bluetooth module, and it has 2.4GHz BLE (Bluetooth 4.0) compliant. Punch Through Design isn’t a very big company, but the products that they make work very well.

This Bluetooth module has 46 different pins. It operates from 2 to 3.6 volts. This little module is 12mm by 16mm, If you don’t know how big that its, this chip is smaller then a dime. This module will control what and when a circuit will get voltage. As I mentioned before size is very important to this project. We do not want our device to be to large, that would make it harder for the user to effectively use this tracker. This is another reason that we chose this module; it is very small and fits perfectly on to our PCB design. The actual Bluetooth Low Energy module can be seen in Figure 8.

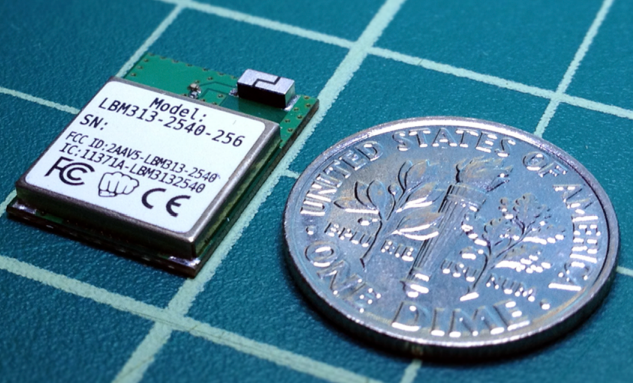


Figure 8: Size comparison of the Bluetooth module

We also had to design this for the PCB schematic that we made. We had to make it from scratch because this is not one of the standard parts that Eagle has in their inventory. From this image in Figure 9 you can see how the module is powered. You can also see where all the other plugins are. Some of the plugins that we have are the going into in BLE module are the different LEDs and the buzzer. There is also an Arduino Power plugin that allows us to program our PCB wirelessly.

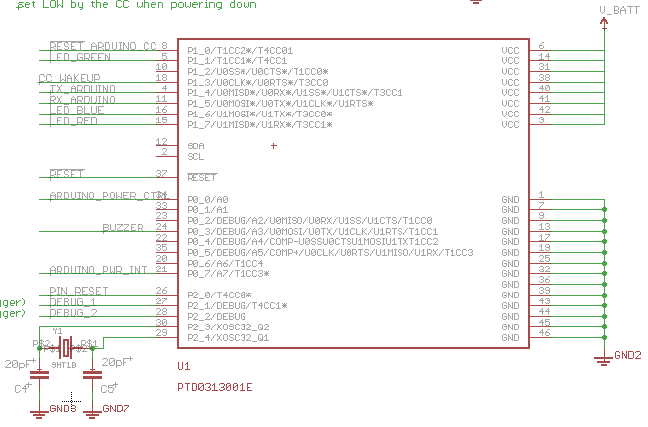


Figure 9: PCB layout of our Bluetooth module

**6.1.3 Chosen Wireless System**

As stated at the end of the research section, we decided to use Bluetooth as the means of wireless communication for FAUX. We had three options after that: Classic Bluetooth, Bluetooth Smart or iBeacon. We decided that we wanted to explore the option of iBeacon for our project for a couple of reasons. None of the related projects talked about in the beginning section use iBeacon as their means of communication so this would set our project apart from the pack due to having a unique feature. We also want to notify the user if the item goes out of range and using a beacon would make this a simple task. Due to the constant transmission, as soon as the Beacon detects the phone is out of range, it can send a notification to the user to remind the user to not forget the item.

We decided to purchase a product from Estimote Company. Estimote has been given great reviews from everyone who has purchased them. The beacons this company produces meet most of the specification that we identified at the beginning of the document and even exceeds them. As talked about in the research section, iBeacons use Bluetooth Smart technology that allows constant sending of data packets.

Since the creation of their first development package, Estimote has been expanding the horizon of their company. The chosen iBeacon we decided to go with for this project is the Estimote Stickers development kit. Figure **8**, located below,and shows exactly what the Sticker beacon looks like.



Figure **8**: Estimote Sticker Beacons (with Estimote permission)

Estimote originally only created beacons with their very own Software Development Kit. These beacons were big and bulky and would not suit the aims of our project, but recently this company has created Stickers, a beacon nonetheless but the size has been reduced so the beacons can be places on physical objects that they defined as nearables.

Estimote defines a nearable as a smart object broadcasting data about its location, motion and environment to a mobile device in range. This is ideal for our project because we need the beacon to broadcast the location and range of the misplaced item to the user’s mobile phone. Stickers work pretty much the same as beacons do, but with a few variations. Figure **9** below shows the inside printed circuit board of an Estimote Sticker.

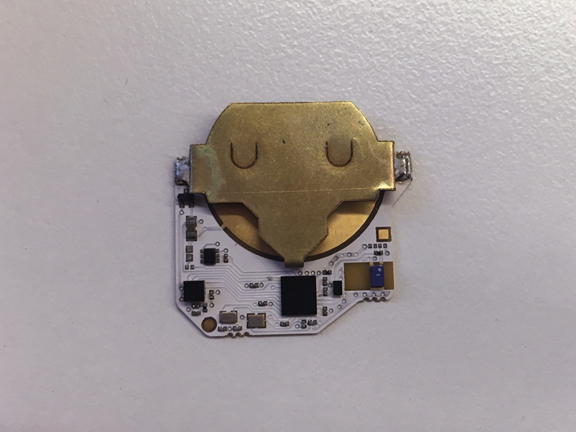


Figure **9**: Estimote Sticker Beacon PCB (with Estimote permission)

Estimote beacons have their own SDK that will require us to learn the libraries included in the software but the features of the product are too appealing for us for this to be a setback. The Stickers library is similar to the ordinary Estimote beacon but is altered slightly so we will have to focus on this. In order to learn the SDK, we will watch tutorial and browse the community portal on their website in order to better understand how to create applications with the product. The next section will go over the technical specs for Estimote Sticker beacons and reasoning for choosing this product.

**6.1.3.1 Reasoning/Purpose**

As stated previously, the Estimote Sticker beacon will act as the wireless communication between FAUX and the user’s smartphone or tablet computer. It will broadcast information about the object back to the user. There are multiple reasons why we chose the Estimote Sticker Beacon over the original Estimote Beacons. Figure **10** below is a table provided by Estimote that shows some of the key differences between the two.

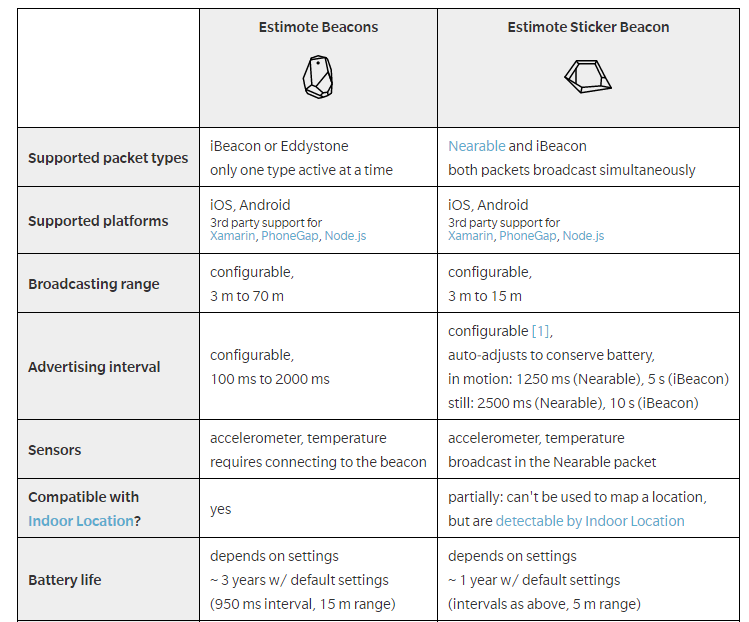
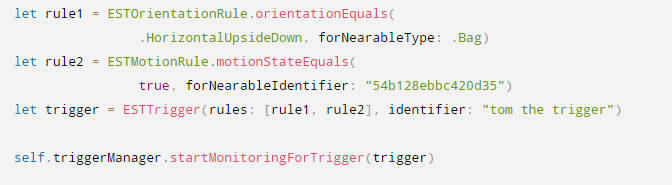


Figure **10**: Estimote Beacons vs Estimote Sticker Beacons (with Estimote permission)

The sticker beacon achieves all of the requirements that we set out for this project in the Requirements and Specifications section. Another huge improvement that was gained from the Sticker beacons was the dimensions. The Sticker beacon is just 3 mm thick and a lot smaller and less bulky than the original beacons.

Another huge advantage to using the Sticker beacon was the new SDK that was implemented with this beacon. Sticker beacons has all of the features in the original Estimote beacon SDK but with a few added modifications. The Sticker beacon introduces the Trigger keyword in its API. Trigger engine allows the user to forget all technical complexity behind monitoring and allow for commands to happen if certain criteria are met. Figure **11** below illustrates a code example from the Estimote website of how the Trigger keyword works.



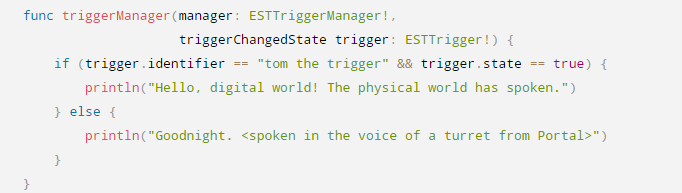


Figure **11**: Code explaining Triggers (with Estimote permission)

This code utilizes the Keyword Trigger in order to make certain actions take place. For example, the actions for this code is when you put the sticker upside down and start moving it so it is in motion, the message “Hello…” will be printed out but when the sticker beacon does not meet both of these requirements, the statement returns false and should return “Goodnight…”

These beacons have the major upside of allowing to be put on objects that will be in movement, for example like a bike or even a dog’s collar. Having this feature was a must for our project due to the fact that we do not know the state of the object at all given times so it must be prepared for motion or no motion at all. It is also easy to notify user if an object is in motion using the Trigger keyword and seeing if the object is in motion by checking the motion of the beacon. This could prevent simple theft by becoming alert of an object moving at the earliest possible time.

This beacon also focuses on monitoring and ranging. Monitoring a regain enables the app to know when a device enters or exits the range of a beacon. All of this can happen even if the user has their phone in their pocket and the phone can even be locked. Ranging returns a list of beacons in range with an estimated proximity to each. This can only occur when the app is running. If the app is running, you will be able to estimate how far away FAUX is and words will be given to you depending on the range: Immediate (within .5 m), Near (within 3 m) or Far (within 70 m). Note that these intervals are for the original beacon so the Sticker beacons will have a different set of ranges but the three words used to estimate the distance will be the same. Figure **12** to illustrate the ranging portion that was just described is shown below.

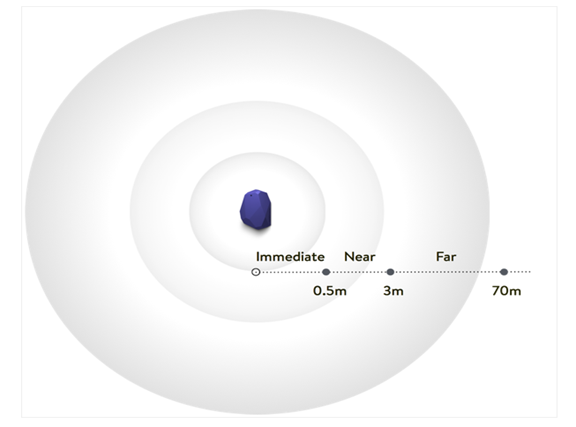


Figure **12**: Ranging for Beacons (with Estimote permission)

**6.1.4 Chosen Battery**

The battery that will be powering FAUX will be CR2032 coin cell battery. The CR2032 is lithium battery rated at 3 volts. Its diameter is 20mm and its height is 3.2mm. It weighs approximately 3 grams. Its specified temperature range is -20 degrees Celsius to 70 degrees Celsius.

Coin cell batteries are an excellent choice for small electronic devices such as calculators and watches because of their small size, simplicity, and long lasting battery life. The implementation on the PCB is also very intuitive in that it is easily replaceable, even for the everyday user. Replacement batteries can be purchased very cheaply. Single CR2032s can be bought online for less than one dollar; buying multiple batteries in a package can lead to them being even cheaper than that.

This battery needs to be able to power the PCB and all its external additions that are to be implemented such as the speaker. The following is a picture of the CR2032 battery. This picture is a generic picture for the battery since it is the typical coin battery that can be found in any store.



Figure **13**: Picture of the CR2032, the chosen battery for the project.

**6.1.5 Chosen Speaker**

Our chosen speaker for FAUX is the Multicomp MCKPRG1720-4011 PIEZO BUZZER. This speaker is specifically what we were looking for in size and efficiency for FAUX’s main goal which is to remain small in size so that it does not create a bulky feel, but also to have a loud enough sound that it could be found easily, even in a crowded atmosphere. Our biggest concern was that we did not want to sacrifice sound quality with the small size it would have and with the PIEZO BUZZER only having a 16.5mm external diameter and 4.2mm external height it fit our exact measurements.

With the PIEZO BUZZER fitting our needs in size, the next concern was the sound quality, this speaker compared to others we have considered has the best Voltage supply ratios having a Maximum range of 30V and being able to operate with a Maximum current of 3mA. Not only does this create a great sound it gives a lot of room to adjust the voltage and currents if we needed to make some modifications to the speakers output of sound once we get in into the FAUX’s programming in order to keep a strong communication with the other parts inside the FAUX.

Programming was also another major concern in selecting speakers for our project and with this speaker we are able easily program it with the other devices that will be implemented in making the FAUX run efficiently and productively for our users. Our battery is going to be putting out a 3V power supply which is a perfect match for our speaker of choice. Along with the buzzer being able to be switched on and off on command, we must also be able to program it to work alongside an LED light in the proper sequence together. Working with this Multicomp speaker we have seen the programming side of it to be a lot more stable and we are able to manipulate it in order to get it in the exact specifications we need it to be at.

Overall the PIEZO BUZZER was our choice for speakers because of not just one category, but an overall collection of impressive specifications that meet our products needs better than any other speakers we have looked in to. The low cost of the speakers will make production and assembly stay at a lower cost for us, but also maintaining a high standard in what we want it to be at. Size, efficiency, and ease of programmability were the main determining factors in choosing the PIEZO BUZZER. We are confident that we have selected all of the right parts to put in our device starting with this speaker, not only to create a pocket sized device, but it will also have the sound necessary to be able to be found by the user in any type of environment and retrieve their lost item.



Figure **14**: MCKPRG1720-4011 Piezo Buzzer, chosen speaker (with permission from Multicomp).

**6.1.5.1 Reasoning/Purpose**

Choosing a speaker was a main concern for us not only because of our exact specifications, but also we want to create a product that has great long lasting sound for the duration on our devices use. We looked into many different types of speakers such as magnetic, electrostatic, moving-iron, but ultimately we ended up choosing a PIEZO BUZZER. Many factors went into choosing this type of speaker ranging from cost of production, durability, and, consistency with the FAUX’s designed model.

The cost of this specified speaker is on the lower side of many of the PIEZO speakers, which for us will make manufacturing a lower cost for us and help create more for less. Comparing to the other types of speakers on the market we had to go with the PIEZO speaker. While price is at a lower cost for us, we made sure not to lose any type of quality with our speaker as well. The durability of a PIEZO speaker is second to none when it comes to how well they are built and the shelf life they have on them. Not only is the outer casing strong and can handle many types of embellishments, but the speaker itself can handle high levels of outputting amps. Researching these speakers we wanted to see how well they can handle extra voltage in the case that something were to malfunction. In our research we found that a PIEZO speaker can reach up to 1000 volts being output into it before there was a shutdown of the speaker. Our device is only using a 3V battery, so we are very confident that our speakers will last the entirety of the FAUX’s life without any type of decrease in the quality it offers.

The FAUX will be a very complex device and we need many components that can all come together as one unit to create a seamless transition into working as one and completing its natural function with many other factors feeding off of the same power supply. We have found that not only are PIEZO speakers easily programmable, but they are also very capable to be used intertwined with other devices such as LED lights. For the FAUX to run at our ideal efficiency we need every working part firing on all cylinders together as one. Being able to make our speaker controllable from a remote device utilizing Bluetooth was essential for the functionality of our device, which is one of the major reasons we went with the PIEZO speaker over many others.

Finding this speaker was not an easy decision for our group and many hours went into researching and comparing many different types and brands of speakers together. In the end we think we have found our ideal speaker for the FAUX, and it has everything we were looking for, ranging from price, durability, programmability, and functionality. We believe that with this speaker our device will not only work, but will be more efficient than expected and the quality will be top of the table in all aspects.

**6.1.6 Chosen LED**

During this project coming up with a device that would be useful and never been done was a challenge in itself, but creating one that anyone can use was another. After coming up with FAUX, our next step was to find a way to make sure it was accessible to all users of the device. While designing the concept for FAUX we had to make sure that anyone who may have a disability would be able to successfully use it as well as anyone else in the world. After developing the general idea of what we wanted to do, we then had to get all the working parts to alert the user to where their lost item could be located. Along with a function speaker we knew that having an LED light would be crucial for anyone who might have a hearing impairment. With the LED and speaker working together we made sure that all users would be able to get the fully function out of FAUX.

Not only was the design of FAUX important, but we also wanted to make sure that everyone was able to use this device. We have decided on two options for implementing an LED for this project. Our main source of LED lighting will be the built-in LED on the microcontroller. This LED may be small but we made the assumption that the LED lighting will only be beneficial at a close range so there is no need to have a giant LED since it will not help the user. This will also make this component easy since the only thing we will have to do is program the LED. A second option we may explore is adding an additional LED for FAUX. This will be done depending on time constraints, space left after the speaker and whether or not we decide that we need more lighting. We will include the implementation of an extra LED in our prototyping section just to simulate how we will create the LED if implemented.

Choosing this LED for our project was a very easy decision for the group, nothing came close to the bean in production value or ease of programming for us. Having selected the main base for FAUX has now opened the door for us to begin implementing our other units for the project such as the speakers and batteries and create a prototype design for FAUX to come to life.

**6.1.6.1 Reasoning/Purpose**

Finding an LED that would create a bright shiny light for our project that people would be able to see easily and in dim light situations was one of the first things we needed to complete as soon as we got our idea for what we would be trying to create. Although a bright light was something we really had to focus on we also had many other specifications we were looking for in our LED system. With having a bright enough light we wanted to be able to have a wireless connection possibility, easily be able to program, and being able to cross platform to all users would be very important for our group if we wanted have a successful product that could be used by all mobile users. With the Light Blue Bean we are confident as a group that we have found the foundation for the FAUX at a very low production cost and great starting ground for s to build our product on.

The Light Blue Bean not only has a great RGB LED that will give our users the option to choose a color they want when they active the FAUX. The bean is also able to be programmed wirelessly through Bluetooth. With having this capability it allows our group to program the FAUX from any location and we will not be constricted to requiring a computer in order to make adjustments when needed. Having this freedom will improve our response time for any type of issues our product might come across.

We understand that creating a new product is not easy to make and getting your product to the public and widely known is a very big challenge for any new invention. Our group wanted to create a mobile device and with the competition between apple and android we knew that we would have to have cross-platform device in order to reach both markets of mobile users. With the bean we now have the capability to reach all mobile users to ensure we have the biggest output for our device. The bean can be programmed with OS X, iOS, and all Android platforms which will allow us to cover all types of mobile users.

FAUX’s success is going to be based on our programming abilities and how well we are able to communicate that with our users. Using the Light Blue Bean we have given ourselves one of the best starting platforms in order to create success with our device. Having the freedom to wirelessly program and being able to use it with multiple mobile platforms are all compliments to what we really wanted to focus on which was which was creating a device that all users can use and with the bright LED lights we can ensure that anyone can see or hear it and will be able to achieve their goal, which is ultimately to retrieve their lost or stolen item.

**6.2 Software Design**

**6.2.1 Mobile Application**

The mobile application that FAUX will utilize has multiple features incorporated within for the user to choose between. Components described throughout the document including the LED and the speaker will be easily accessible via this application. There is still some questions that need to be resolved before we could begin with the creation of the app. We needed to define the key functionalities of the app and how we plan to implement them. Also we needed to take into account the best design decisions to allow the app to be simple for any user even if the user is not tech savvy.

Other questions needing to be answered include which operating system we will use. From our research, we discovered it would be impossible to write one application that would be fully compatible with both operating systems in use. This causes our group to need to make a decision into what operating system will best suit us and the project.

In the following sections, we will talk about our process of choosing what operating system to use and the functionality and purpose of the application. Once these have been presented, a general idea of what the application will be and how it will work can be imagined. Both of these will outline the model that will be used to build and develop the working application.

**6.2.1.1 Chosen Operating System**

Due to the limitations of mobile application design, we were not able to make a single app that will be compatible for both Android and Apple products. This means that we had to choose which system to focus on. In the end we chose Apple iOS operating system as the operating system to focus the creation of our app on for a numerous of reasons. One of the major reasons for this decision was the chosen beacons, Estimote Sticker. Originally, the software development kit for these beacons were only available to iOS operating systems. Recently, Estimote released the software development kit for Eddystone, an open Bluetooth Smart protocol from Google. This allowed for the Android operating system to work with these beacons. This change had little impact on our decision.

Another reason iOS was chosen was due to the technology we already owned. All three of us are owners of iPhones along with other Apple technology including iPads and MacBooks. Due to the technology already being available to us in a plethora of ways, it was an obvious choice for us to choose iOS over Android although there are some impacts that we need to account for.

As mentioned in the Programming/SDK portion of the document, iOS applications are programmed in Swift, a programming language unfamiliar to our group. As a group we decided that we will take the time to learn a new language instead of purchasing an Android device to write in a language more familiar to us. If we decided to use Java, we would have to go out and buy an Android device which would significantly increase the amount of money we would need for this project. A major task for us is to make this affordable.

Apple iOS devices are also programmed in Objective-C, which is similar to the C programming language that our group has experience with. This could be an option we explore if we decide Swift’s learning curve will be too time consuming for the timeline of the project. Similarly, the Estimote beacons SDK works with either Swift or Objective-C so the two languages under consideration will work for both of the application and the beacons.

We believed there will be a tough time implementing the communication between the beacon and the microcontroller. Due to the fact that they will more than likely need to be written in two different languages on two separate IDE’s, we needed to see if a the mobile application will allow the creation via two separate parts of code or figure a way around this problem. One solution that comes to mind was since the microcontroller we are using ATmega 328 is connected to a Bluetooth low energy module and was communicated via Bluetooth Low Energy, code can be loaded straight from the phone to the board wirelessly. This allowed saved code for certain situations to allow the user to simple complete a set of a few commands to activate the microcontroller aspect of the project. Again was intended to be used as a back-up idea if mobile application design experiences problems. Having a central mobile application to complete all the tasks at hand was the main goal to aim for and we believed that the creation of the app is very possible for FAUX even before development.

**6.2.1.2 Application Functionality and Purpose**

As stated repeatedly throughout this document, the main source of locating the valuable FAUX is attached to will be a mobile application. In order to get an idea of how our mobile application should be designed and what functions the app should be able to accomplish, we looked at apps for the current items on the market for some guidance.

Most of the apps are distance proximity locators, which is what we will do for ours as well. Basically once the user needs to find the object, the app will display the approximate distance the object is away (assuming the user is within range of the tracker) and as the user moves around trying to locate the object, the app will notify the user if they are going in the correct direction or if they should try another option. This may seem like it is inefficient and maybe a GPS location is better but we need to take into account the indoor scenarios. A major reason Bluetooth was chosen for this project was to allow for our project to work for indoor scenarios, where GPS would have no effect. If a user loses their phone in a huge auditorium or library, we wanted FAUX to be able to locate the phone for the user. The locator will still be able to work outdoors but will work identical to as it does indoors. It will find the distance between the two objects if in range and will track the user to the object. This locator will act similar to the hotter/colder game that some may be familiar with.

We will also explore the idea of tracking lost items that are out of range via GPS as well but this task may be hard to implement and will be time dependent. The way that Tile locates a lost item is that the user must mark the item as lost. This will then put all other Tiles on the lookout for the user’s Tile. If any of the Tile’s comes into contact with the user’s Tile, the Tile will discretely communicate with Tile’s cloud system and the user will be sent a message with the location of the object's whereabouts. We believe that this may be possible without choice of components. Estimote utilizes a cloud system as well so we will attempt to implement a similar way in which when an item is reported lost, it will communicate with all other Estimote beacons in an attempt to locate said object. Although this feature may be out of reach for the scope of our knowledge, this could be an additional feature to add if we decide to continue working on this project.

Along with using the application to find the object, we also needed the app to allow the user to utilize the other features of the project. We need the app to allow the user to turn the speaker on and to turn the LED on as well. We also planned to add a notification method to the app. This will act as a reminder to the user. If the user sets the object down and begins to walk away from the user, we will need to notify the user to ensure that he does not forget the object. This will be simple, simply set a designated range to trigger the action to take place. Let us assume 10 feet, now we will make the app notify the user if he steps away from the object FAUX is tracking at a distance of 10 feet or greater. We could also add other features to this like if the object gets out of this range, the LED will turn and the speaker could also turn on. Figure **15** illustrates a simple flowchart that explains how this task would be completed.

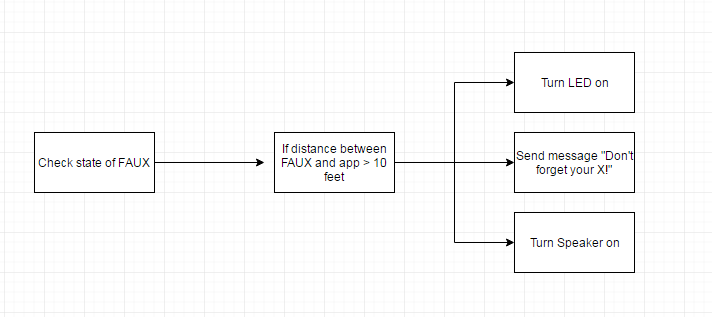


Figure **15**: Flow chart to send reminder notification to user

Another way to help the user ensure to the object will not be lost is to allow the app to test for motion of the object. If the object is in motion and is not in the user’s possession, then it is safe to say that someone else has the object. The app can’t test for simple motion though, if the user is in possession of the item and moving, even though the object is in motion, it does not imply the object is lost. To explain how this would be done in code, Figure **16** below demonstrates the process via a created flowchart. The above chart showcases how we would ensure that typical motion would not be confused with an object being stolen.

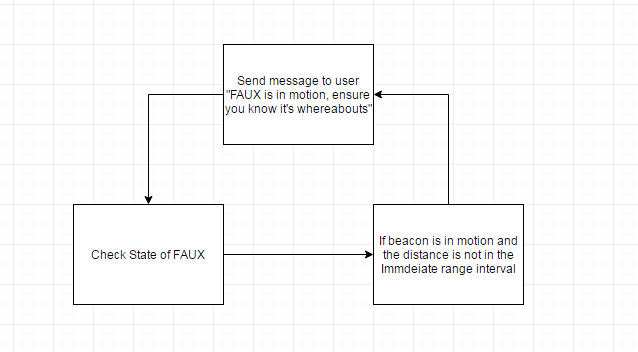


Figure **16**: Flow chart to check motion of FAUX

We decided if an object is within the immediate interval range to the user, it is safe to say that the user is in possession of the object and should rest easy. If the object is in motion and in any other range interval when being checked, we assume that the distance is far enough to say that the object could have been stolen and to inform the user to figure out why the object is in motion and attempt to locate it if necessary. Other features are be explained using flowcharts in the Software Development section of the document. To summarize, the three main functions we will need our app to be able to implement are controlling the LED and speaker, reconnect to the circuit board if the communication signal is lost and locating an object using the distance. Notifications are equally as important since it theoretically could reduce the chance an object is lost in the first place.

**6.2.2 Communication Software**

We have two components in our project that must be able to communicate simultaneously whenever the user desires. The microcontroller will be used to control the LED and the speaker for the project and the beacon will be used to transmit a signal so the user can use this signal to locate the object if needed. Our application needed to be able to utilize both of these components and give control of both of these to the user.

Both of these components have their own software development kit that will be used to program the individual components to achieve the desired task at hand. We had no prior experience of combining different components into a single application but will take the time to learn in order to create or ideal application. Due to the fact that both of the software for the components are not identical, different environments needed to be used in order to create the working part for each component. We had to follow the instructions for each of the SDKs and choose what environments are required based off the specifications presented to us.

After each component’s task has been completed in their separate environment, we had to combine the functionality of both in order to create the unified app. We worked with the LED and speaker in the microcontroller software and work with the distance locater feature in the iBeacon software. We needed to ensure that the application can utilize both components in a simple way that will not require complex instructions in order to complete the tasks for the user.

**6.2.2.1 iBeacon Software**

As previously explained, the iBeacon was used as the means of locating the object FAUX is tracking using the distance. As mentioned, we are using the Estimote Sticker beacon as our beacon of choice. These beacons broadcast data packets which includes information about the unique beacon. The features that are contained in this packet are state of motion, acceleration readings, temperature, power, voltage, firmware version and the nearable ID (unique ID assigned to individual beacons). All of the information included in each packet is broadcasted by the beacon’s generated signal.

With the information provided from the packet, Estimote has created their own unique software development kit to work with the beacons and the data received from each packet. For example, in the SDK there is a class called “ESTNearable” which allows the user to retrieve properties of the nearable. So if we needed to figure out if the beacon is in motion or if the beacon is at a standstill, we just have to look at the “isMoving” property of this class and see what value is returned when the property is checked. This class also has a property called “zone” which will indicate the distance from the device if the beacon is in range. This allowed us to achieve the distance locater that we have set out to create in the mobile application. The “zone” property works as we explained the range intervals do. It returns the zone in which the distance is in the certain range interval (immediate, near or far) and the “ESTNearbleDefinitions” class has these intervals defined.

As previously mentioned earlier in the document, the trigger keyword is an important feature of the Estimote SDK that we plan to take advantage of. Trigger is defined in the “ESTTrigger” class which will define the trigger object. Trigger can be thought of as a more complex if-statement. Trigger allows to create real life situation definition based on particular set of rules. Trigger should be accompanied with the “ESTTriggerManager” class in order to monitor the state of the trigger. Trigger defines a set of rules and should all of them be fulfilled, the state of the trigger would be yes. Actions will then occur depending on what was assigned to happen when the trigger state is yes. In order to send all of the data to the user, Estimote has the “ESTNotification” class that allows for data communication between the host app and the beacon. This allowed for communication to be simple between the application and the beacon. This class has properties that allows the user to know if you left the range of the beacon or if you have entered the range of a certain beacon.

The iBeacons can be detected even when an application is not running. The phone can detect the beacons via Bluetooth Low Energy. This helped with allowing the user to be reminded about FAUX even when the app is not running or running in the background. The user’s smartphone will always be detecting and searching for beacons to enter its proximity and will notify the user if the beacon corresponding to FAUX enters the proximity. This helped the user track down FAUX and thus finding the user’s misplaced item.

The SDK does not require a unique environment to be used. Since the SDK is just a library, we can implement the library in any IDE that allows us to program using Objective-C or Swift due to the software being written in those languages. We decided to base the development environment on whatever was easiest for designed a mobile application. Since the beacon software allowed for flexibility, we were able to focus on finding a user friendly mobile application design environment.

**6.2.2.2 Microcontroller Software**

As talked about previously in the document, we have chosen to use the ATmega 328 as our microcontroller of choice for our project. Software for this microcontroller is Arduino based (requiring us to compile on the Arduino IDE), which is highly beneficial to us due to the previous experience we have with Arduino development kits. Another plus for this microcontroller is the Bluetooth Low Energy module component which allows us to wirelessly load programs onto the microcontroller via Bluetooth communication. This software is called Bean Loader. Bean Loader is universal and will allow for any type of device to compile the program (OS X, Windows, Android, iOS). Once the user designated that the Arduino IDE is associated with the Bean Loader, the Bean Loader Cloud system will allow programs created within the IDE to be loaded onto the board with no wires needed. If the user using FAUX is skilled in programming, they could even alter FAUX if they decide they would like to change a feature straight from their smartphone.

Along with being able to use the Arduino IDE and the language incorporated with them, the PunchThrough Bean has its own software development kit that will help build native app for any platform. This would be considered a third-party library and would be what we would use to link all of our components together. As stated previously, the goal was to have the app being able to communicate with the microcontroller and the beacon at the same time whenever the user chooses to do so.

We could have also explore the option of using Bean Loader via iOS. Bean Loader has an application currently on the App Store that will allow the creation of programs. New programs can be created or already created programs can be edited at will depending on the needs of the user. The user can write their own programs straight from their own smartphone and load it to the bean from the same device. This would eliminate the IDE middle man for this process and could have some advantages if we need to adjust code on the fly. This software will be explored and depending on the usefulness may be used over the generic IDE. One problem that may arise from using the mobile Bean Loader software would be if the third-party libraries would be able to be implemented for the mobile loader. The third-party libraries, as stated above, will be the means of communication with all the components in our project so we must ensure that this has importance over simplicity.

Using this software will be complicated for us due to the fact of our lack of knowledge of using third-party libraries and implementing them together to allow separate components communicate cohesively. If we experienced problems when trying to access this software and implement it to our project, we had to rely on the community forums for this microcontroller for guidance. This may have been an issue but we believed that we could overcome any problems encountered for this software and figure out how to get the correct implementation.

**6.3 Protective Casing**

As stated previously the protecting casing is an important piece of this project. The case is so important because it will have to house all of the delicate electrical equipment. All of the competitors of FAUX have their own specific protective casing that makes them look different from the others. We are going to try and make the case in the shape of an oval, which will set us apart from the competition.

There are some limitations that we put on the protective casing though; it must be water resistant, it must be relatively easy to open to change the battery, it must be clear enough to see the LED through it, and it must be thin enough so that we can hear the speaker clearly through it. The reason that we want the protective casing to be water resistant is so that water can’t easily get inside of the case and damage the delicate hardware. Because the part of Florida that we live in is considered one of the rainiest of the entire state, we don’t want the entire device to be ruined by a little bit of water splashing up and hitting it. Water resistant does not mean that you will be able to completely submerge the entire device in water and expect it to not be damaged at all. If that were the case then our device would be considered waterproof, but out other limitations would prevent this from happening.

The protective casing must be relatively easy to open so that it will be easy to change that battery. Some of the competitors make you send the device back to them because their battery is not accessible without special tools. This causes you to have time without your tracker, and then they would have you purchase a completely new tracker to replace the one that you sent back to them. The method that we plan on using to house the Light Blue Bean is that the pieces will snap together, so you would just have to unsnap them to be able to change the battery. We will have to include a rubber O-ring in between the two pieces that snap together so that it will be harder for water to get into the case.

The next limitation that we have is the material and color that we are going to use. We won’t be able to use any metals or any materials that are completely opaque. We won’t be able to do this because we want the device to light up when the user is looking for it. This will be accomplished by sending a Bluetooth Low Energy signal to FAUX, once it activates the LEDs on the inside will light up. The reason that we can’t use non-translucent materials is because you would not be able to see the LEDs light up from the proper distance away.

The last limitation that we have for the protective casing is the thickness of the casing. The pieces of the case could not be too thick for two reasons: first we are using a speaker so when the user activates the device we would like him or her to be able to hear it. The speaker will be a huge help to the person that is using it because instead of just engaging one of the user’s senses we will be engaging a second one. Secondly if we make the case to thick that risks the device being too heavy to comfortably put on your keys, or any place that you plan on carrying it. We want this device to be as light as possible, so the smaller we make it the less it will weigh, and the happier the consumer will be.

**6.3.1 Shape and Size**

The shape and size of the protective case that we use to guard the Light Blue Bean will have to be just a little bit larger than the Bean itself. For that reason we will be making the size of the case in the same shape that the Light Blue Bean is in. We will be using a 3D printer to make our case, we are doing it this way because it is more cost efficient and we will be able to put in exactly what measurements and dimensions we want for our case. Figure **17** and Figure **18** show the top and bottom of a 3D rendering that we made to protect the Light Blue Bean.



Figure **17:** 3D rendering of the case we plan on making for the Bean (Top)

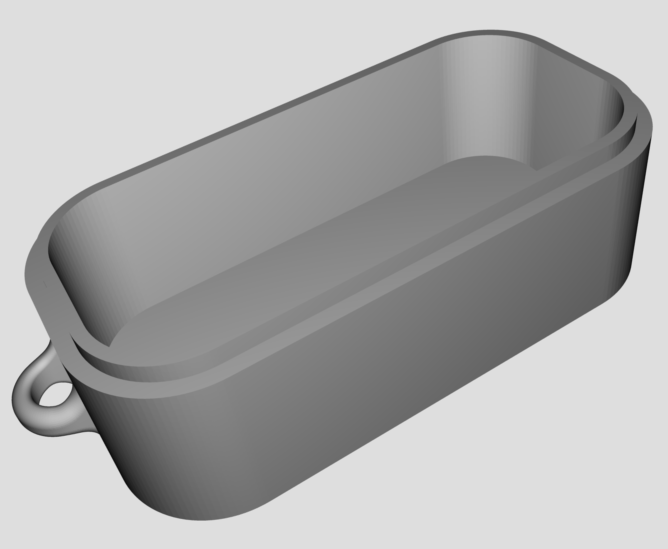


Figure **18:** 3D rendering of the case we plan on making for the Bean (Bottom)

As shown in Figure **18** this is the bottom of the case that we did 3D printing, you can see that there is a lip on the top of the case so that the lid will fit on there snuggly. That’s how the device will fit together. The dimensions of the case are slightly larger than that of the Light Blue Bean. The Bean will fit in the protective casing very snuggly. It needs to be this way so the device will not move around when it is inside of the case. We did include some foam or absorbent material in between the Light Blue Bean and the protective casing. This acts like a shock absorber, which will keep the PCB from getting damaged if the user drops it, bangs it into a wall, or other things along those lines.

Since unsnapping the clamps can easily open the case, we also made sure the battery is easily accessible, so that the user will be able to change the battery when it runs out. This will be a little bit trickier to accomplish, but we just have to play around with the design to make them more accessible. The dimensions of the Light Blue Bean are 45.7mm x 20.5mm x 8.5mm (1.8’’ x 0.8’’ x 0.3’’). We would like our case to be no longer than two inches long, one inch wide, and 0.6 inches thick.

The covers that we designed can easily be snapped on and snapped off. The way that the two covers fit together is shown in Figure **19.** This feature will make it easier for the owner of FAUX to change the battery. This protective casing for the FAUX will work very well in protecting the Light Blue Bean while maintaining the accessibility to the battery area. These are some of the size constraints that we want to put on our protective case that we are designing.

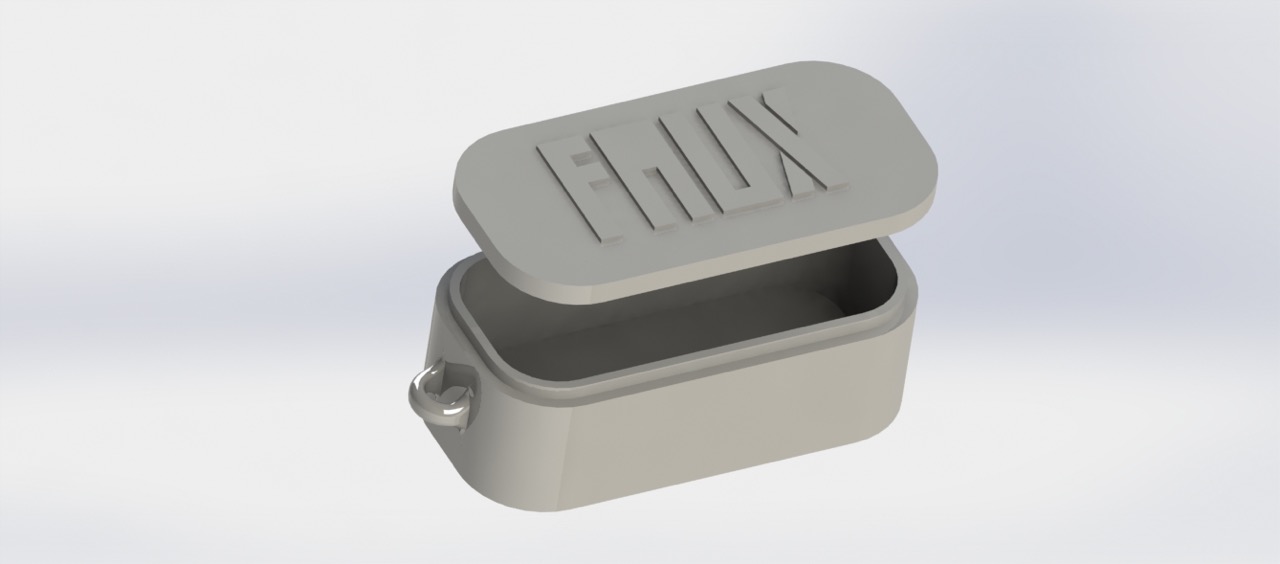


Figure **19:** The top and bottom pieces of the cover together.

**6.3.2 Weight**

The weight of this product is a very critical. We need to make sure that the device isn’t heavy at all. If we plan on the consumer putting this device on their keys then we need to make it around the weight of a standard key. The weight of the Light Blue Bean is 7.3 grams, which is equal to 0.3 of an ounce. The standard key weighs between half of an ounce to 0.8 of an ounce. If we want to stay in a feasible range we must make the protective case weight from 0.2 of an ounce to 0.4 of an ounce. The weight of this product can depend on three separate items. It depends on the thickness of the case, it depends on the density of the material that we use to make it, and it depends on the overall size of the product as a whole.

The thickness of the case is very critical because if you make the case too thick you want be able to see the LED, or hear the speaker. It will also make the device heavier than it needs to be. Our ideal case would be thin enough to see and hear everything to its maximum capacity, but not too thin because then it could break or crack much easier than we would want it to. So overall we need to print a couple test cases to find the perfect medium between too thin and to thick.

The density and the type of material that we use to make the case will affect the weight of the case as well. If we use abs plastic it will lighter than if we used any type of metal. Even if we did use metal the case would be much more stable, but it would weigh too much. We also would not be able to hear the speaker or see the LED.

The last thing that the weight of the protective case depends on is the overall size of the product. If the speaker and LED that we add on stick out too much we would have to make the case larger to accommodate to these adjustments. The larger the product is as a whole the bigger we will have to make the case. The bigger we have to make the case, the heavier the overall product will be. The heavier the product is as a whole, the less our customer will want to put it on their keychain, or use the product at all. This is how the weight will affect how we design the protective case for FAUX.

**6.3.3 Color and Material**

There are two different materials that we can use to 3D print the protective case for the Light Blue Bean. The two types of material that we can use are polylactic acid plastic (PLA plastic) and acrylonitrile butadiene styrene plastic (ABS plastic). PLA plastic is a biodegradable [thermoplastic](https://en.wikipedia.org/wiki/Thermoplastic) [aliphatic](https://en.wikipedia.org/wiki/Aliphatic) [polyester](https://en.wikipedia.org/wiki/Polyester) derived from [renewable resources](https://en.wikipedia.org/wiki/Renewable_resource), such as [corn starch](https://en.wikipedia.org/wiki/Corn_starch) (in the United States and Canada), [tapioca](https://en.wikipedia.org/wiki/Tapioca) roots, chips or starch (mostly in Asia), or [sugarcane](https://en.wikipedia.org/wiki/Sugarcane) (in the rest of the world). In 2010, PLA had the second highest consumption volume of any [bioplastic](https://en.wikipedia.org/wiki/Bioplastic) of the world. ABS plastic is a [terpolymer](https://en.wikipedia.org/wiki/Terpolymer) made by polymerizing [styrene](https://en.wikipedia.org/wiki/Styrene) and [acrylonitrile](https://en.wikipedia.org/wiki/Acrylonitrile) in the presence of [polybutadiene](https://en.wikipedia.org/wiki/Polybutadiene). The proportions can vary from 15 to 35% acrylonitrile, 5 to 30% [butadiene](https://en.wikipedia.org/wiki/Butadiene) and 40 to 60% styrene. The result is a long chain of polybutadiene crisscrossed with shorter chains of poly (styrene-co-acrylonitrile).

The [nitrile](https://en.wikipedia.org/wiki/Nitrile) groups from neighboring chains, being polar, attract each other and bind the chains together, making ABS stronger than pure [polystyrene](https://en.wikipedia.org/wiki/Polystyrene). The styrene gives the plastic a shiny, impervious surface. The polybutadiene, a [rubbery](https://en.wikipedia.org/wiki/Rubber) substance, provides [toughness](https://en.wikipedia.org/wiki/Toughness) even at low [temperatures](https://en.wikipedia.org/wiki/Temperature). There are some similarities between the two. Each requires a dry location for storage given the materials are prone to melting and warping, and furthermore, each is susceptible to moisture. Both ABS and PLA also smell while printing, as heating the thermoplastic gives off fumes.

However, there are also major differences between the two thermoplastics. ABS is going to give your projects better structural integrity and will be more suited to mechanical use given the material can better withstand the elements, but it will also require specific types of printers and printing surfaces. On the flip side, PLA will give you more precise prints and better aesthetic quality, as well as more flexibility with printing conditions if you can do without the strength and resilience of ABS.

Given this information, and the other information that we have researched we would like to use ABS plastic to 3D print our case. It will give us the right structural integrity, and it will also very durable, so it will be minimally affected by the elements outside. We also plan on using PLA plastic to make sure our case works and has no kinks before we use the ABS material to print the final design. We plan on doing it this way because generally the PLA plastic is much cheaper. In our case it is free to print our mock protective case in the Starter Lab, which is located in the Business Administration building. To 3D print our final design using ABS material we will use the 3D printer that is located in the Idea Lab, which is located in the Engineering Atrium. Given these resources I believe that it will be quite easy to create an acceptable and durable protective case for the Light Blue Bean.

The last topic that we have to tackle is how you could customize your FAUX to be a one of a kind item. You would be able to do this two different ways. The first way would be choose what color you want your FAUX to be. The color of your FAUX depends what color of ABS plastic we would use to 3D print your device. Although if you pick a darker color the LED on the inside would be very difficult to see when you activate your device.

The optimum colors that you should pick are white, light blue, light orange, or yellow. These colors are light enough that you would still be able to see the LED flashing on the inside. This would make your FAUX tracker very distinct from your friend’s FAUX tracker. The second way that you would be able to customize your FAUX tracker is by being able to engrave it. The engraving process would be easy. All we would have to do is slightly change the original design of the 3D modeled case. Putting a design on the protective case could serve as a security measure, but it could also serve as wanting a little picture on the front of your FAUX.

It could be used as a security measure because you could get your phone number engraved on it. If you get your phone number engraved on your FAUX device, and then proceed to lose it. A stranger could find it and notice that there is a phone number on it, and they would proceed to call it to return the misplaced object that you lost. The engraving could also be used to put a neat little design on the front of your FAUX tracker. If you got your FAUX 3D printed in pink ABS material you might also want to get a heart engraved on the front of it. Or maybe you get your FAUX 3D printed using yellow ABS plastic you might also want a little smiley face printed on the front of your FAUX tracker. These two things could set us apart from the competition, and it would also make the person who uses the FAUX tracker feel more special because they feel like they purchased a one of a kind product. Using these constraints and customizations I feel confident that we will be able to make a proper and durable protective case for our Light Blue Bean to be inside of.

**7 Design Summary**

**7.1 FAUX**

Now that we have talked about everything that will go into the system, we want to talk about how all of these components will work together and how everything fits in. This section will summarize what will be discussed in more detail in the following sections. FAUX is an interesting project because it has a fairly even mixture of hardware and software components that will need to be integrated together to form a functioning system.

First of all, the speaker will be attached to the microcontroller. Following this, we will need to implement the LED that is on the microcontroller and get that component working. Next, the iBeacon device will be added. The iBeacon will either be attached to the microcontroller or the outer shell casing. After this, the casing for the device will be developed, 3D printed, and then added to the system. The microcontroller will need to be programmed so that it can communicate with an iOS device. Next, the mobile app will need to be created and also programmed to communicate with the device. The app user interface will also need to be developed at this point. Development for the user to use the app to toggle the LED and speaker on the device will also need to be done around this time. Once the app and the device can communicate and there is a functioning user interface then the design phase will be over. After this, most of the work will consist of testing and putting the finishing touches on everything.

**8 Prototype Building**

**8.1 Development**

**8.1.1 Hardware Development**

As described previously we added components on to our Light Blue Bean chip. These components that we added on will go hand in hand with the software part of this project. There are programmable pins at the bottom of the Light Blue Bean. Using these pins we were able to solder the extra LED and the speaker on to the Light Blue Bean. For both of these components we will need to connect them in series with a resistor. Both of these components will be drawing from a 3-volt DC battery source. The battery source that will be powering these components is a CR2032 coin cell lithium-ion battery.

The way that the LED will connect to the battery is through the programmable pins. The way that we can simulate this circuit is by attaching a 3- volt DC battery in series with a 200Ω resistor and attaching that in series with the Light Emitting Diode (LED). Since the turn-on voltage of the LED is 0.7 volts we need a resistor that can take away 2.3 volts from the battery, that where the 200Ω resistor comes into play. This process and the readout of the voltage can be seen in Figure **20**.

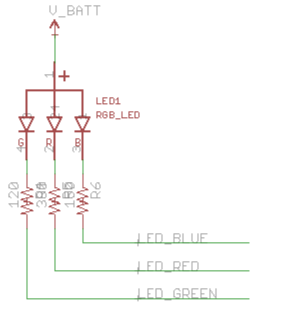


Figure **20**: Actual LED circuit

As you can see in Figure **20** we connected the 3-volt battery source to the LED and then through three different resistors. As I stated previously the LED has a turn-on voltage of 0.7 volts. From the simulation you can see that the voltage across that LED is 718.086mV. This value is just over the turn-on voltage, which means that the LED is turned on, and working. The voltage drop over the 200Ω resistor is 2.282V. This proves that this is the perfect resistor to turn on the LED. It doesn’t provide too much voltage to the LED, it supplies just enough.

The way that we plan on toggling the LED on and off is by implementing a virtual switch into the circuit. The way that that will work is when the user of FAUX presses the LED button on the interface, it will toggle the LED. The toggle happens by either sending or not sending the voltage to through the resistor to the LED. This is why we call it a virtual switch.

The way that the speaker will connect to the battery is through the programmable pins. The way that we can simulate this circuit is by attaching a 3- volt DC battery in series with a resistor and attaching that in series with the Multicomp MCKPRG1720-4011 PIEZO BUZZER. We are unsure what resistance value that we will need to put into the circuit, or if we need to put one in there at all. In the circuit that we simulated we included a resistor just in case we needed one in there. This process is shown in Figure **20**.

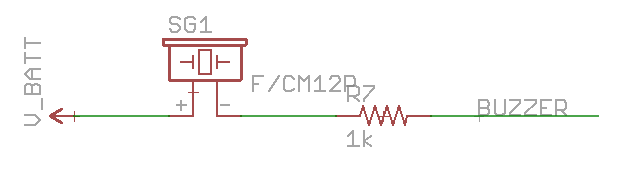


Figure **21**: Actual speaker circuit

The speaker circuit is shown in Figure **21**. It shows that the speaker will be powered by the CR2032 coin cell lithium-ion battery, which is the same battery that will power The Light Blue Bean, and the entire FAUX tracking device. You can also see that we included a resistor of unknown value that we put in series with the battery. We don’t know if we need to put that in there, so the value of the resistor could be zero, we will have to do more research on this idea. Then finally we attached a load in series with the battery and the resistor. The load represents the speaker that we plan on attaching to the programmable pins on the Light Blue Bean. We also plan on using the virtual switch to control whether or not the speaker makes noise when the FAUX tracking device is activated. The Flowchart showing how the entire hardware portion of the FAUX tracking device is shown in Figure **22**.

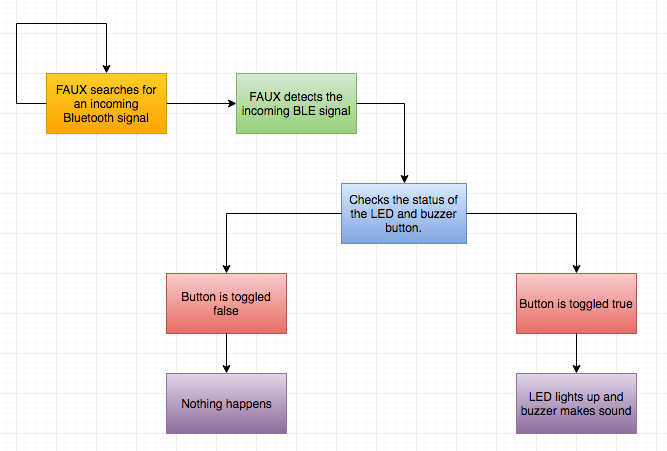


Figure **22:** Flowchart showing the entire hardware portion of the FAUX tracking device

As you can see from Figure **22** the FAUX tracking device is constantly searching for a Bluetooth signal. Once it detects an incoming Bluetooth signal a series of events starts to unfold. The device checks if it should turn the LED on and if the device should also turn the speaker on. If both of them are supposed to be turned on the Light Blue Bean will send voltage through those programmable pins, which will supply voltage to the LED and speaker circuits. If FAUX detects that the button is not pressed it will not send voltage to that particular circuit. Then it will continue to search to see of the user ends up pushing the button. This is how all of the hardware interacts and performs together to create our FAUX tracking device.

**8.1.2 Software Development**

As described previously, we needed to develop an app that will allow the user to take advantage of the features of the microcontroller as well as the features of the beacon. In order to do this, we needed to find an environment that would allow us to use the SDKs for both components. After much research and exploring, we discovered that both components prefer the use of Xcode as the integrated development environment of choice. Xcode is developed by Apple for developing software for OS X and iOS. We will attempt to do all the development using Swift because both components support this language. We have not worked with Swift before but have knowledge of the C programming language so we do not believe the learning curve will be an issue for us.

Along with discovering the use of Xcode, we also came across a program that may help us manage all our libraries. CocoaPods is a dependency manager for Swift projects, and also supports Swift if we decide to go that route instead. This allowed us to keep all our third-party libraries up to date without having to routinely check to make sure we have to the most recent version.

Xcode allowed us to combine all of our third-party libraries into one uniform environment and create a single project that will allow access of both libraries. Even though we had no previous experience with using Xcode, we all have had experience using a plethora of IDEs for previous courses so we don’t see this as a problem. Using these tools, we were able to create a working mobile application. Figure **X** shows what our user interface looks like within the application.

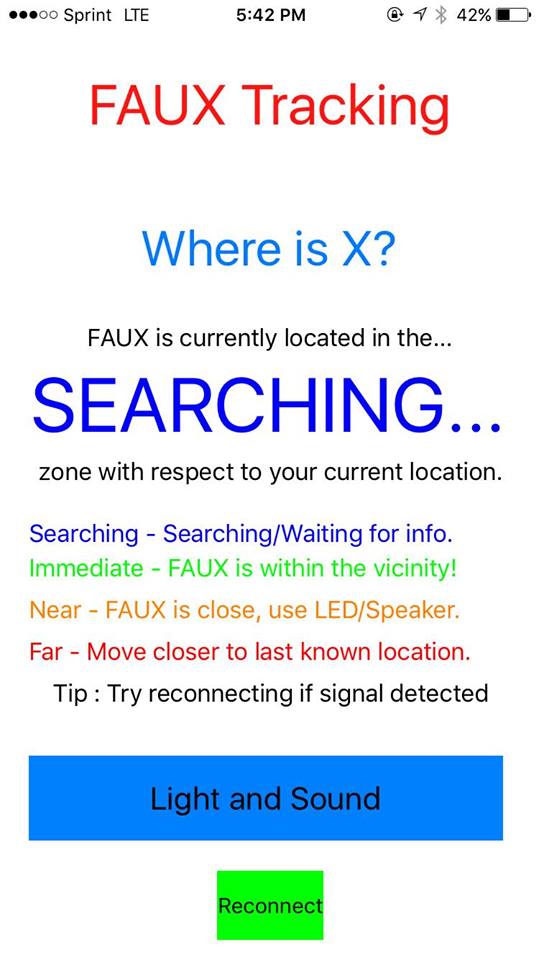


Figure **X**: FAUX user interface

Our user interface is split into three categories. First we started off with the labels, as you can see at the top, FAUX tracking is the title to the user telling them the purpose of the application. Where the “SEARCHING” label lays, is where the proximity zone will be returned to the user. Under this label we created a legend to help the user decide what to do depending on what is returned to them. This legend includes user guided messages explaining what’s returned as well as instructions on what to do next. Finally at the bottom of the interface we have our two created buttons for the components and reconnect functionality. These will be explained later in this section.

Now that all of the programs that were used to develop the application have been introduced and discussed as well as what our user interface looks like and how it is split up, we have to explain all the features that we decided needed to be implemented for the application and how we accomplished these features within the code itself. In the Application Purpose section, we described how we will send a notification to the user if the user gets out of range of FAUX and to test to see if FAUX is in motion. Since those have been described already we will not discuss them here but instead just show pictures of how these were implemented within the application. Figure **X** and Figure **Y** show what notifications look like for our application dealing with loss of range and when an object is in motion.

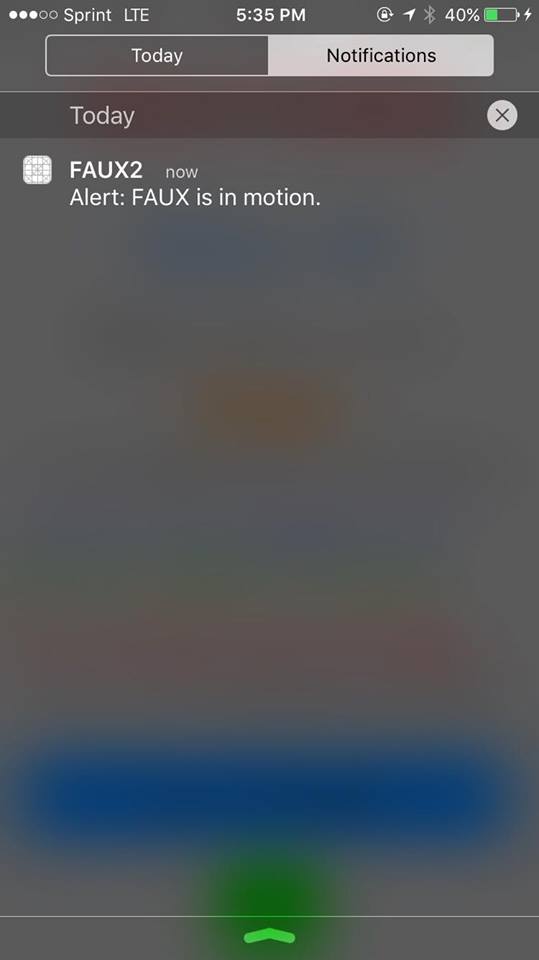


  Figure **x**: Motion mobile notification

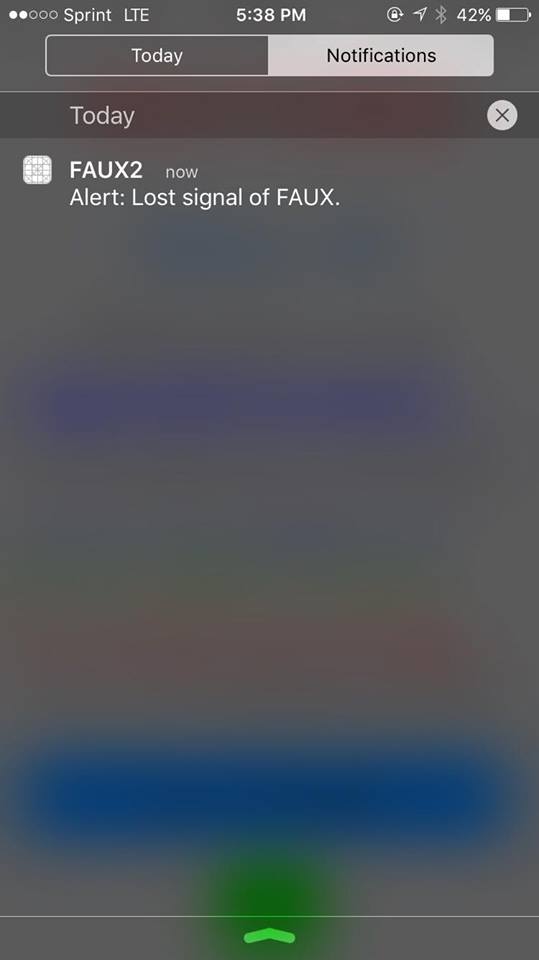


Figure **Y**: Loss of signal mobile notification

We also needed to include a notification for when the user reenters FAUX range. After the user is notified that they have lost connection to FAUX, we needed to inform the user once they have reentered the range in order for them to know when they have returned to the vicinity of FAUX. Figure **X** showcases what a notification looks like when a user reenters the range of FAUX.



Figure **X**: Entering FAUX range mobile notification

The first feature that we needed to implement in the app was the control of the LED and speaker. We would like the user to have to ability to turn the LED and speaker on and off as pleases. The best way to do this we believed was to have a simple button that has two states, yes or no, that will decide when the LED and speaker will be on or off. When the user clicks the button once, the LED and speaker state will be yes which will turn the LED on. If the button is pressed again, the state will change to no and the LED and speaker will subsequently turn off. Figure **23** below explains how we were able to accomplish this by continuously checking the state of the button and changing the state depending on the button.

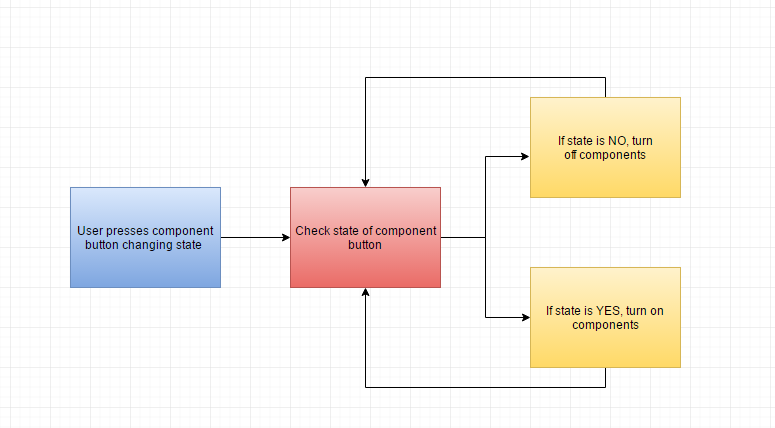


Figure **23**: Flowchart to change state of components

After conducting some initial testing of the prototype, we realized that there was times where the application would get to a distance so great away from FAUX that we would be disconnected from the board. Since the board uses Bluetooth low energy rather than the iBeacon protocol, a means of communication always has to be there for the board and application to communicate. If this communication median is broken, the user loses all functionality to the LED and speaker. Due to this, we needed to implement a reconnect button so the user could reconnect to the board whenever they get out of range of FAUX and then need to reestablish the connection. Figure **X** showcases a flowchart explaining how this is done with the button. It is similar to the hardware component button by toggling the states but instead, it simply checks to see if the application is in range of FAUX and if so, it will reconnect to the Bluetooth module.

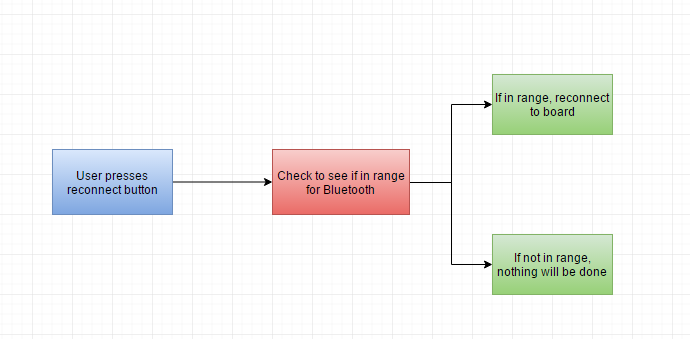


Figure **X**: Flowchart for reconnect button

Now that the two hardware as well as the reconnect functionality components have been talked about being implemented in the application via the software, the final feature being used in the application that we need developed is the actual tracking feature. As stated previously, the tracking feature will utilize the distance approximations to label the item in one of the three range intervals. We also need to inform the user if FAUX is out of range so the application cannot pick up a signal.

We had to decide what to do for the user if the beacon signal was out of range and locating the object was not going to work. We did not want to just simply say that the beacon signal was out of range because that had no positive impact in helping the user locate the object. We figured that the best thing to send to the user is a message informing the user to attempt to get to a closer location of the last known location of the object. For example, let’s say the user left FAUX in a library and realized that FAUX was missing when they arrived home. The application would tell them that the signal could not be detected and the user should navigate to the location they can most recently remember seeing the object. The user would then drive back to the library and attempt tracking FAUX again using the app and a signal should appear, showing that the application is picking up the signal FAUX is broadcasting which shows that FAUX is closer to the user than before. Figure **25** below shows an example of logic for the application of how we could accomplish the tracking feature of the mobile app development.

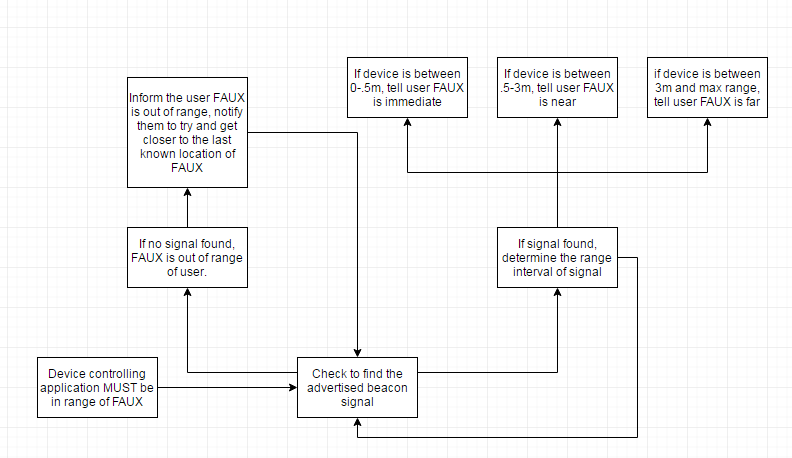


Figure **25**: Flowchart to accomplish tracking functionality

This flowchart shows the logic for all the cases that could occur. We first have to check if the signal is in range of the application. If not, inform the user that FAUX can’t be detected and tell the user to move back to the last remembered location of the object being tracked. If a signal is found, indicate what interval range the signal’s distance is within and broadcast that to the user. The state of the signal should be checked constantly so that changes in the range intervals are picked up instantly for the user. The speaker and LED features can help the user locate the app once the application reads that the signal being broadcasted is within the immediate or near range. Figure **X** and figure **Y** show two examples of our application when in two separate zones, the far zone and the near zone with the labels changing color depending on what zone.



Figure **X**: Near zone for application

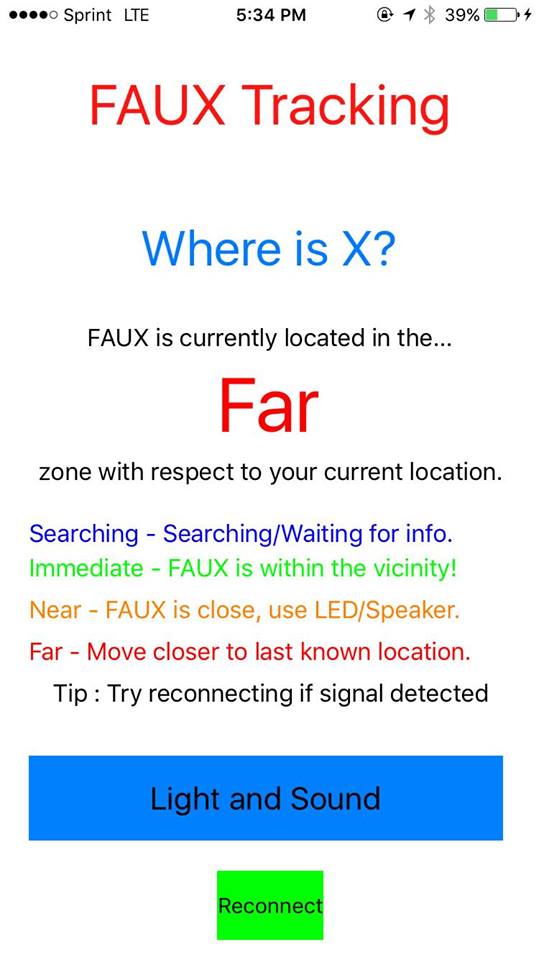


Figure **Y**: Far zone for application

**8.2 Production**

**8.2.1 Protective Case**

The protective casing for FAUX is something that we made from scratch. The protective casing that we are going to make for the FAUX tracking device was 3D printed in the Engineering Building. We will be using ABS plastic as the material for our 3D printed case. We will need to make at least one prototype case so we will be able to do multiple different tests on it.

The way that we will be designing our case is by first creating a 3D model using the Solidworks cad software. We had to determine the dimensions that we wish the protective casing to be. We input these dimensions into Solidworks so that the Light Blue Bean will fit very well, and very snugly into the protective case that we make. We did this because we don’t want the Bean to be moving around and shifting while it is inside that case. That would cause damage to the board, which would create a faulty product. Once we have coded the case in Solidworks we then converted that to a ‘.stl’ file which is needed in order to send the design to a 3D printer. We then printed the protective casing using the PLA plastic in order to perform testing on it. We used the PLA plastic for this because in general the PLA plastic is much cheaper than the ABS plastic.

We then confirmed that the dimensions are correct and it has passed all of the tests that we have set out for it, we then printed out the final case using the ABS plastic. Using that case we performed the final tests on this case to make sure it holds up to the standards that we have set for it. It then passed all of our tests we were certain that the protective case will be more than able to provide protection to the Light Blue Bean and the other electronics that we have put inside the protective casing.

**8.3 Vendors**

For the prototype, we purchased all the required components of the system from many different vendors. This section will be used to simply discuss where and from which vendors everything was purchased from. The test board that we are using is called the Light Blue Bean and it was purchased from a company called PunchThrough. The speaker (or buzzer) was purchased from a company called Multicomp. The Bluetooth capable iBeacon was purchased from a company called Estimote. All of these purchases were made from online outlets. The casing for the device will be 3D printed and will most likely be purchased from an on campus source. Everything else that is involved in the project, including an iOS device and other various components, are either already owned by a member of the group or will not have to be purchased.

**9 Prototype Testing**

**9.1 Testing Environment**

In order to make sure our FAUX tracker workers effectively indoors and outdoors performed a multitude of tests. We performed tests in two different areas: inside and outside. When we did our testing measured several different aspects of the FAUX to make sure that it is working correctly. We measured the distance away from the device when it activates and we measured how many obstacles are in between the user and the FAUX tracker.

We also measured how far away we will be able to hear the speaker of the device when it activates. Another aspect we measured is how far away the LED is visible, and if we will need to add extra LEDs to make it more visible. The conditions of testing when we are inside were quite ideal; the only things that we had to worry about were the obstacles are between the user and the FAUX tracker.

The testing we did outdoors was much trickier. The testing outside had different factors that affected the maximum range of the Bluetooth. Wind, rain, or any type of bad weather can limit the Bluetooth Low Energy range. We also needed to test the iOS application that we made. We tested this by making sure that it is user friendly, making sure the device actually responds when we prompt it to, and making sure there are no bugs or glitches in the application that make it freeze.

We also needed to test if the distance tracker that we have in the iOS application is working properly and could lead the user to the device. Another aspect that we had to test and do calculations for is how much power the device draws from the battery. We needed to know how much power that FAUX draws in order to calculate the battery life of the device.

We also needed to measure the total weight of the final product that we made. We needed to do this to make sure that the FAUX tracker will not be too heavy, or too large to carry around. Ideally the user should be able to carry it around without even noticing any difference to their keys, or the object that they put the tracker on.

The last thing we tested is the protective casing that we made for the PCB to sit inside of. We had to make sure that you are able to drop it and not have it break, and we also had to test that it is water resistant. In the following sections we will describe some testing scenarios and how realistic they are. We did this in order to prepare the FAUX tracker for use in the real world. Conditions aren’t always ideal, so it is important that we test our device very thoroughly.

**9.1.1 Indoor Testing**

As stated previously we had to test the device indoors. Half the reason that we are making this device is so that if you are in a rush to get out the door our tracker will make finding your lost keys very easy. The way that we will make this very easy is that we will have a speaker and an LED attached on the inside of the FAUX tracker. The speaker will help you find your lost item in the daytime and the nighttime. The LED will mostly be helpful when it is dark in the environment that you are looking for your device. It would be easier to find FAUX in a dark room; this would be useful for two reasons. The first reason is that it helps you save money on electricity, given that it won’t be a substantial amount, but saving a small amount is better than not saving any. The second scenario is that if the power went out you would not be able to turn the lights on, but FAUX will still be very easy to find.

One test scenario is that it is very early in the morning, and you are leaving for work, and you are trying to do so without waking anyone up in your home. Normally this would be easy, but today you misplaced your keys. Instead of going from room to room turning lights on and off in every room, flipping couch cushions, and anything else you do when you are frantically searching your house for something when you are running late. Instead of doing all of these things you could simply activate your FAUX tracker and listen where the sound is coming from, and see where the light from the LED is. One of the prompts that we would like to include in the mobile application is a silent mode, so if you really want to find your lost object as quietly you could just simply look for the glow of the LED in the dark room.

This is a very practical test scenario in which it utilizes all of the features that the FAUX tracker has to offer. The speaker makes it easy to hear what room your lost item is in, and the LED makes it easy to find your device when you reach that room. If you have the speaker on silent you would still be able to be guided to your lost item by the distance tracker that is on the iOS applications screen. Another prompt that we will include on the application screen is a button that you could press to toggle the LED on and off. This would not be useful in this test scenario, but if it were light out you would not want the LED to be on because that would waste the battery power of the FAUX device. You wouldn’t want to be changing the FAUX battery often because the coin cell lithium-ion batteries can be quite expensive.

Another test scenario that we played out is if you are in a room and your lost device is on the complete opposite side of the house, would you be able to find FAUX even though there are many obstacles and walls in the way. We needed to know this Bluetooth range so when the customer is using the device he or she would not get frustrated that their FAUX tracker is not working. It will be interesting to find out how far away we could track our device from inside the house. We need to make sure the distance readout that our iOS application gives will be accurate and not lead you in the wrong direction.

All of the features that we include on the FAUX tracker are very important as to the overall success rate of the device. All of the features that we include with the FAUX tracker; from the speaker to the LED to the distance tracking to the engraving on the protective case will all help you find your lost objects faster.

**9.1.2 Outdoor Testing**

We will also have to test this device outdoors. This where testing the tracking device starts to get much trickier. There are many more variables when we are testing the device outside. The largest variable is obviously the weather, especially in Florida where the weather can change from sunny to windy to down pouring and back to sunny all within one hour. In an ideal world our tracker would work as well outside as it does inside, but from experience we know that this is not the case. The weather takes a huge toll on the range of the Bluetooth.

The range will be much better outside when it is sunny out as compared to when it is windy or rainy out. We must test this tracking device very thoroughly in all of these weather conditions so we do not deceive the user. The first test scenario that we will perform is that you are exploring a new area that you have never been to before. You have been walking around for a while, then you decide to leave but you seem to have dropped your keys along the way. If you did not have your trusty FAUX tracking device attached to your keys you could spend a long time retracing your steps in order to find what you have lost.

If you have you FAUX tracker attached to your keys finding them would be much easier. All you would have to do is open up the application, and see how much distance is between you and the object that you have lost. Once you are close enough you will be able to turn on the speaker to use another one of your senses to guide you toward your lost item. If it were starting to get dark out when you start your search you would be able to activate the LEDs, which would make it very easy to spot once you are in range of it. We would need to perform this test in at least two different weather conditions; sunny to get when it has the ideal conditions, and windy or rainy to get the worst-case scenario. Both of these scenarios would still be better than not having any tracking device attached to your keys. If that was the case you could spend hours looking for your lost object instead of minutes. Since we live in Florida we also want to observe how FAUX behaves at the beach.

The beach could be a tricky place because of two different reasons the first reason that I stated before is the weather. The second reason is the sand and the water. We want our device to be able to repel sand, water, and dirt. We don’t want any of these to get inside of our protective case, if it does it will most definitely cause problems to the internal electronics. This would be the last thing that we want. We are going to need to perform many tests on the protective casing so that this cannot happen. We would not want to compromise our structural integrity of our case for any reason.

The designing and printing of the protective case does not fall within the electrical or computer engineering wheelhouse, but that does not mean it is any less important. If the casing has any flaws at all, the whole project will fail. For this reason we will have to test the protective casing in any and all test scenarios that we could think of. As stated previously the range of the Bluetooth Low Energy signal will be much better inside than it would be when it is outside. The weather is a key factor when trying to use a Bluetooth Low Energy signal outdoors. The best conditions you could hope for is when it is either dawn or dusk. You want these times of the day because it is not too cold, but it is also not too hot out. The problem of being in Florida when it is sunny out is that the Light Blue Bean chip could overheat and need to cool down before you could activate it.

We don’t want this to happen so we have performed simulations on this chip to determine the range of temperatures that it could deal with. The highest temperature that we simulated was 60 degrees Celsius. This is more than high enough to withstand Florida’s temperatures. After doing this we looked at the data sheet and we found out that the temperature limits are -20 degrees Celsius to 80 degrees Celsius (-4 degrees Fahrenheit to 176 degrees Fahrenheit). Performing these tests and more will be more than sufficient in testing our FAUX tracking device.

**9.1.3 Obstacles Expected**

As mentioned above there are some obstacles that we expect to encounter along the way when we are creating the FAUX tracking device. The first obstacle that we could encounter is on the software side. Creating an application in the Apple app store can be quite difficult; it requires a good bit of code. If there is something wrong with one line of code it could throw the whole application off. Generally it is hard to debug code of that nature, so this would be the first obstacle that we could encounter.

The next obstacle is on the hardware side of this project. We could have trouble with the LED and speaker. If this was the case, and we attached something wrong it could short out our board, so we would have to find another one that has the exact same design. This is a worst-case scenario, but you need to think of the worst that can happen so you have a plan for it if it actually happens.

Another obstacle that we could encounter is that the FAUX tracking device will not communicate with the mobile application well at all. This could be because of a wide variety of things. It could be because there is something wrong with the board, there is something wrong with the mobile application, or there is a problem with the Bluetooth signal. We would have to test each device individually to determine the root of the problem in this case.

The fourth obstacle does not have to do with the hardware or software; it has to do with the protective casing. As mentioned previously there could be any number of things that go wrong with the protective casing that could cause the board to short out or gradually destroy the equipment on the inside. That is why we are taking extreme precautions in doing many, many tests on the outer casing. Need to make sure that nothing can easily get into it, and we certainly to want anything to be able to fall out of it. All in all what we are trying to say here is that the largest obstacle that we have to overcome is human error. If humans were perfect, this section would be left blank, but we are not so this section is not either.

**9.2 Hardware Testing Procedures**

Before implementing the microcontroller, LED, and speaker we will have to do some testing on each of these object separately. Then we will have to put them together and see if they work as a whole. To do this we must test the devices in a variety of different environments. There will have to be tests indoors and outdoors. We will need to test in the dark and in the light, we will need to see how far away we can see the LED in both of these environments. We will need to apply these same tests to the speaker that we choose to see how far away it can be heard in and outdoors. Then we will need to test the range of the microcontroller to see how far away we can go, and it will still respond to the Bluetooth signal.

The strength of the Bluetooth Low Energy signal depends on the environment that the device is tested in. If the device is inside the only thing that will limit the range is the number of walls that it will have to pass through. If FAUX is being tested outside there are many more obstacles such as wind and rain. Since we won’t have a year to test the complete battery life we will have to measure how much power our device consumes when it is in rest mode, and we must measure how much power is dissipated when the device is activated. All of this data is very important when we are trying to implement all of these hardware and software items together.

**9.2.1 Microcontroller Tests**

Before implementing the microcontroller into FAUX and testing it as a whole, we must do some testing of the microcontroller device on its own. The microcontroller device that was purchased from Punch Through came with a built in LED and four replaceable lithium-ion coin cell batteries. Punch Through has created their own example Arduino code that we can use to make the LED run along with other things. The microcontroller that we purchased is called the Light Blue Bean, henceforth called The Bean. The Bean comes with many different programmable pins that different add-ons can be connected to.

To test this microcontroller we will need to see the reaction time and the reaction distance that it can respond to when prompted by a Bluetooth signal. The way that this will work is the same for indoor and outdoor environments but when it’s outside it will most likely work much less efficient. There will be a Bluetooth signal sent from a device if the microcontroller is within range the device will respond accordingly, if it is not within the range it will continue to scan for a signal until it picks one up, and then the device will respond accordingly. This process is shown in Figure **26**.

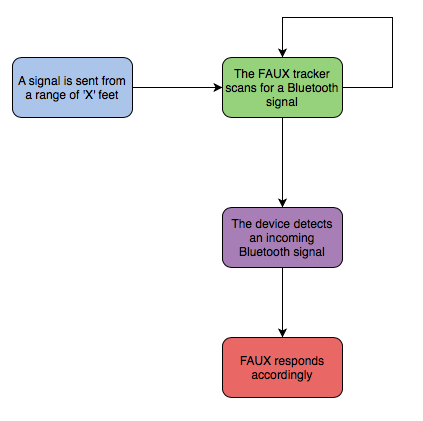


Figure **26**: Flowchart showing how we plan on testing the microcontroller

We recorded how far away from the device that we are, and how many obstacles, if any at all, are in the way. The Bluetooth signal gets weaker and weaker the farther away from the device it gets, but it also gets weaker if it has to go through any obstacles such as walls. The Bluetooth signal will be stronger when it is indoors, because the conditions inside can be controlled as opposed to when the device is outside.

When testing the device outdoors we had have to be more lenient on the distance away the device picks up the Bluetooth signal. The conditions outdoors are variable at best, especially in the environment that we will be testing it in. In Florida the weather can change in the matter of minutes, it could be sunny one minute and the next it is raining or windy. Thus the distance from the microcontroller to the device that is sending the signal will be greater when it is measured indoors compared to outdoors.

When we combine everything together we needed to test the durability of FAUX. By this we mean that we needed to make sure that it is waterproof, and if it is not waterproof it should at least be water resistant. We need to do this because we don’t want FAUX to be destroyed if it gets the slightest bit of water on it. We will also need to test if the device can be dropped from a certain height, and if it will survive multiple drops. We don’t want the device to malfunction if someone has it on his or her keys and they drop their keys.

The last test that we had to do on the microcontroller is how much it weighs when everything is put together. We can’t have a very heavy device and expect the user to carry it around with them all day. Another part that we have to test is how easily the battery pack is accessible. In many of the competitor’s products, such as Tile, you aren’t able to change the battery. This is one of the ways that we want to be different from them; we want our batteries to be easily accessible. Though doing this might affect owe water resistant the final product is.

The last thing that we had to test on the microcontroller is the battery life. The battery life of this device is very important because you don’t want your device to be dead when you lose your keys, or the item you choose to attach it to. The microcontroller is constantly on and scanning for an incoming Bluetooth signal it is constantly and consistently draining the battery. Although we are using Bluetooth Low Energy so it will take a while to completely drain a battery. We won’t have enough time to test the entire lifecycle of one battery, because it should last from six months to a year, but we have to make sure the power consumption is the lowest that it could be, and does not vary too much.

**9.2.2 LED Tests**

There are two possibilities that we had for LEDs. The first possibility is that we use the LED that is built into the microcontroller. We have to check the brightness of that LED because it needs to be able to shine through the protective casing that we put around it. The second possibility is that we connect a new LED using some of the programmable pins on the microcontroller. Doing this would guarantee that the LED was able to be see through the protective casing, but it would make the final device a little bigger. The way that we plan on testing the LED in the first scenario is that we send a signal from a device such as a phone or tablet. Once is range of the microcontroller the Bean will pick up the signal and start to flash the built in LED. We will have to put the protective casing around the microcontroller to make sure that we are able to see it.

The protective casing will have to be somewhere between transparent and opaque. We don’t want it so clear that you could see everything inside, but we don’t want it too dark that we won’t be able to see the led. If this first way doesn’t work we will have to attach another LED to the open pins on the microcontroller, and flash both of them to make sure that we could now see the LED. This process is shown in Figure **27**.

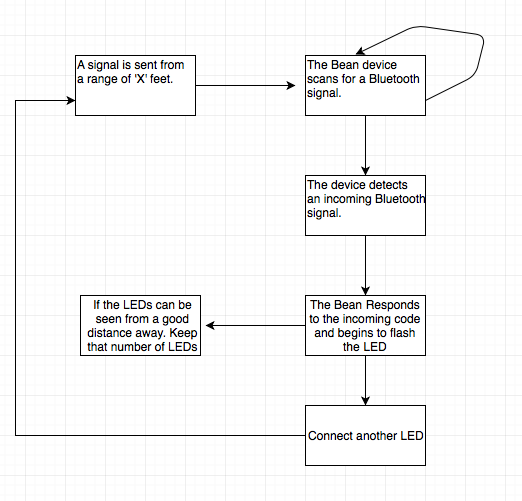


Figure **27**: Flowchart showing the process of testing the LEDs

As you can see in Figure **27** we want the LEDs to be able to be seen from a good distance away. We want this because when you lose something in the dark it is very difficult to find. Ideally we would like the LED to be able to be seen on the edge of the Bluetooth range. Once again though this will depend if the device is inside or outside. We also needed to test how much of a help, if any, the LED is when it is light out. We know for sure that the LED will help you to find something when it is dark but how much will it actually help when it is light out?

It may be more power efficient to attach a light detector to the LED so it doesn’t go off during the day, which means it won’t be wasting power by turning the LED on. We need to test to make sure that the light detector being constantly on doesn’t use up more power than the LED turning on regardless of the time of day that it is. The LED in this project is one of the key features; it is what separates us from all the other products that are on the market.

**9.2.3 Speaker Tests**

The last hardware test that we ran is a speaker test. When a device sends a signal to FAUX via Bluetooth Low Energy the device will respond by lighting up a LED and activating a speaker that will make a loud noise. The speaker is very necessary because you will not only be able to see it you will also be able to hear it. The way that we planned on testing the speaker is using a smartphone or tablet to send a Bluetooth signal, once in range the microcontroller will respond by lighting up the LED and turning the speaker on. We have to make sure that we are able to hear the speaker through the protective casing, which means we have to obtain a speaker large enough to produce a sound that is loud enough to be heard through the case. This process can be seen in Figure **28**.

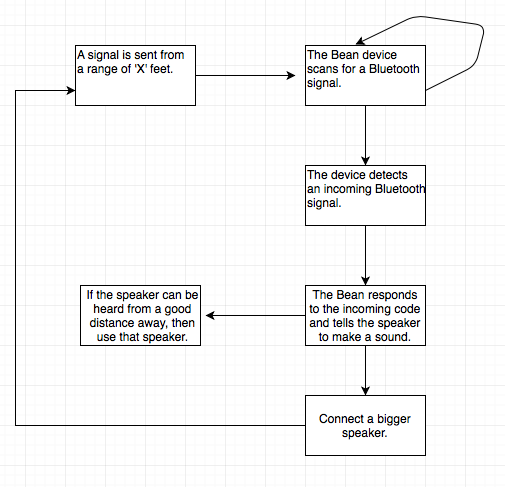


Figure **28**: Flowchart showing the process of testing the speaker

As you can see in Figure **28** we want the speaker to be able to be heard from a good distance away. We want this because when you lose something it is much easier to find something if you engage more than one of the user’s senses. Ideally we would like the speaker to be able to be heard on the edge of the Bluetooth range. Once again though this will depend if the device is inside or outside. If the device is outdoors the Bluetooth range will be decreased, which means you will would have to be closer to the device to make it activate.

The protective case that we put around it can’t be too thick or we won’t be able to hear the speaker that well. If we do make the case thick, then we would have to make the speaker bigger. If we make the speaker bigger that would make the device heavier. We need to find a good medium between thickness of the protective case, and size of the speaker. This speaker is a key feature that we will have in common with many of our leading competitors.

**9.3 Software Test Procedures**

When using any software that you have not used before, it is important to thoroughly learn the new system it before implementing it. After this, the implementations need to be tested to determine that everything that was done was done correctly and works bug free. Before implementing the iBeacon into FAUX, we plan on testing to make sure it works outside of the main system. This serves a few purposes. This will enable us to become familiar with the iBeacon system as a standalone system and allows us to learn how that system works. It will also make integrating the iBeacon system into FAUX much easier. If we know the system inside and out and know how it operates on a functional level, then implementing it into FAUX will become that much easier. Also, if something goes wrong, it will be easier to determine if the problems are arising from the iBeacon system or from the system as a whole. Understanding all parts of a system, from an individual standpoint, and how they function together, is imperative.

Learning a new programming language to use in a project is something that a few of the members of the group have already done from previous projects in other classes. I do not see this being an issue in the progress of the project. Most languages use similar syntax to other languages and learning all the ins and outs of a new one should not take very much time. Once you know how to program, most of the skills from one language can be transferred across many platforms and languages with a little patience and dedication.

**9.3.1 iBeacon SDK Tests**

Before implementing the beacon into FAUX and testing it as a whole, we decided to do some testing of the beacon as a device on its own. The package we purchased from Estimote came with 10 beacons so we had a plethora of beacons and should not worry about that. We explained that Estimote has created their own SDK for their products and we implemented them using Xcode due to the features of allowing us to use all of our third-party libraries but we needed to explore the SDK in depth and see what classes we will actually use when developing with this beacon product.

The SDK contains classes for beacons and the Sticker beacons. Again, the Sticker beacons utilizes the nearable packet so we will have to focus on classes that involve these. We will now discuss classes that we will test to see which ones are the best to use for our application

ESTNearableZone:

In this class, an enum is defined to declare the three separate range intervals: ESTNearbleZoneImmediate, ESTNearbleZoneNear, and ESTNearableZoneFar. These are the three ranges we plan to use to create the tracking aspect of our application and will attempt to use this class to help with this process. Figure **29** below will show how we plan to attempt to utilize this code within our project for our application.

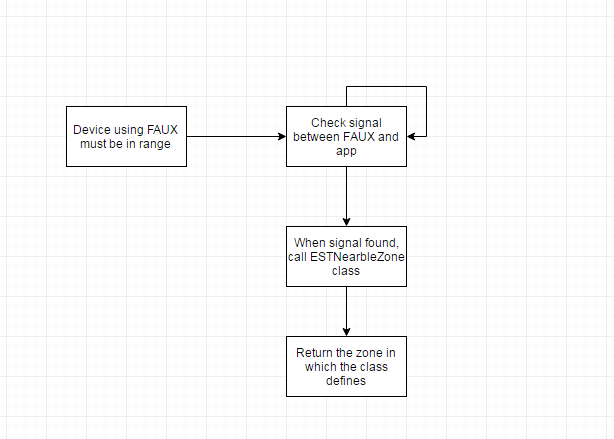


Figure **29**: Flowchart showcasing the use of ESTNearableZone class

ESTTrigger:

ESTTrigger class is similar to the context of an if-statement that any programmer is familiar within. The class will wait for a set of rules to become true and when they do, the class will carry it some sort of action depending on what the user has assigned it to do. This class must be accompanied by the ESTTriggerManager class in order to work. This other class defines the interface for defining real life rules based on triggers. One example we could implement in the application using the ESTTrigger class is the motion. We plan to notify the user if the beacon is in motion and not in the user’s possession. We already had a flowchart explaining how we would do this so one is not needed in this section to explain the thought process behind it.

ESTDeviceGeoLocation:

This class may be a bit of a reach to try and implement within our project but we believe this has some serious potential and would like to discuss it. This class allows for the programmer to assign the physical location of the device. In context, this is useless to our project because FAUX will be moving from location to location as the user moves so it does not have a set physical location but we had an idea of implementing an emergency location feature perhaps. If a stranger finds FAUX, we can inform them to connect to the device via Bluetooth (their smartphone) and click on a feature that indicates return object to user or something along those lines. Using this ESTDeviceGeoLocation class, we can define an address the user desires (like their home address) where they would like a stranger to return their object. A down flaw to this would be that if the user loses their object, their address would be available to whomever finds the device which could be life threatening in some cases. Other flaws may keep us from actually putting this feature within the application but we will still use this class in tests to see if we can work it in somehow. Figure **30** below shows how we would use this class in a scenario where a stranger finds the user’s lost item and plans to return it to the user.

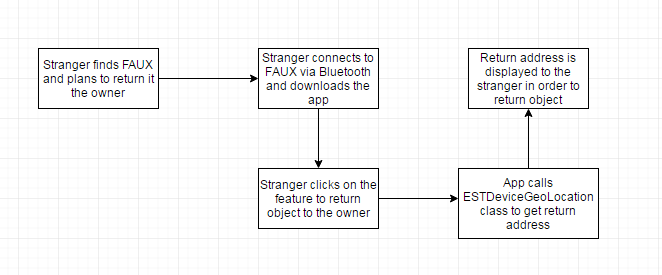


Figure **30**: Flowchart showcasing a stranger returning FAUX to the owner

If we did end up implementing this feature, some users may feel uncomfortable giving out their address for strangers to use. We understand their concerns and could make this an optional feature for those who are willing to publicize this private information. Again this feature is not a guaranteed to be implemented but would be a cool idea that would set our project apart from others.

**9.4 Mobile App Testing**

After creating the prototype of the mobile application, we needed to test it. Testing is an important part of the implementation of a system because it is where you determine that a system is functioning as the requirements have stated. The following section will outline methods to test the mobile application after it is created. The application has a set list of requirements that it should be able to perform. Meeting these requirements is important to the success of the system, therefore it is important that they are properly defined and tested.

Of course, testing the code and application becomes much easier if there is code that written solely for debugging purposes. In the prototype design of the application, there should be some sort of development environment that enhances the testing procedures. This environment should include ways to determine if integral parts of the system are functioning in a way that can be done repeatedly and rapidly, compared to what a user might have to go through when using the app. This will streamline the testing and debugging process. Also, individual requirements should be able to be tested as standalone parts of the application. Doing this will allow us to test individual parts of code to determine if something is broken. Testing multiple things at once can lead to confusion if something breaks. We will not know which parts of the code has bugs in it which will only further complicate the testing process. This should be avoided where possible. Finally, this debugging code should not interfere with the code that will be used for the final version of the system. In other words, it should be able to be taken out of the code without having any effect on the application and how it functions.

**9.4.1 Test Scenarios**

The first thing that can be tested is the user control of the LED and the speaker. These features should be able to be tested together and under the same circumstances because they are similar in their functionality. Basically, if the user is in range, then they will be able to toggle on and off the speaker or the LED. These should be able to both be on, both be off, or have one off and the other on, and vice versa. This means that there are four different test cases when discussing the LED and speaker either being off or on. Testing this seems to be pretty simple. We will simply need to have the application open, with the device in a range that is reasonably within the Bluetooth range. To test, we will use the application and the toggle switch that we will have set up to determine that the LED and speaker can be switched on and off. We will establish that all four test cases are met and that the device is responsive. Furthermore, we should test turning on and off the speaker and LED in different orders. Operating electronics in different orders and manners may have an effect on how the device operates and it is important to test all possibilities.

Different users will most definitely use the device in different ways so we need to be able to make sure that the device remains functional being used in ways that the device was even unintended for. This is not very specific when it comes to actually testing for this, but it just means that we need to create an application that will not break under any circumstance. This is extremely hard to do, therefore, we will need to make our testing thorough enough to try and find everything that can go wrong and try to use the app in every possible way, even things that the app was not specifically designed for.

Also needing to be tested is the range of the device, meaning how far the user can be away from the device when they are trying to turn on the speaker or LED. This information is important because it will help the user determine how far away from the device they can be and still be able to use these functions to help them locate their items. To test this, we will need to go near the edge of the range of the device and continue to turn on and off the LED and speaker, one at a time, until we can no longer turn them off or on. This is to determine at what range these functions can no longer be used. This will need to be done in a multitude of different environments to test for the many areas that the application and device will be used. Some different environments that can be tested locally include testing in the open outdoors, testing in a building such as a classroom, and testing in a place outdoors where there might be obstacles like trees or parts of buildings. Testing different environments is extremely important because FAUX should be designed to be used almost anywhere. Also, people using the device will of course not be confined to any certain location. The device will go wherever people’s belongings go, which is virtually anywhere. These tests should determine if there is any specific environment that FAUX does not operate well in, if there are any. We will use this information to inform users about the limits of the device and how the device is performing in different place under its best operation and maybe where it does not perform so well.

Next, and probably most important, is the testing of the tracking features of the device using Bluetooth. These tests will be the ones that take the longest as well due to there being so much to test in this section. The tracking system will use three different ranges to broadcast to the user. There are different levels to inform the user how close they are to the device. All ranges will need to be tested as well as the device being out of range. To test, we will start with the device in the first level of range, the closest one. Then, we will move the range between the device and the app to the next level range, making sure that the app is responding to the changes appropriately and at the correct distances. The distance measurements do not have to be perfectly precise in accordance to the app, but they should make sense to any user. The level switches should enable the user to find the device easier and not confuse the user by giving out information that is not correct. After this, we will need to check to make sure that if we bring the app out of range of the device, or vice versa, that the app properly displays that it cannot longer find the device. At this point the app should inform the user to return to their last known location.

Finally, we will need to check that the application responds properly to all aspects of the device. This includes the toggle switches for the LED and speaker, the different ranges for the device, and if the device is out of range. The user interface of the application should display these states of the device in a manner that makes sense to the user. When the LED is on, the app should display to the user that this component is currently on. This is the same for the speaker. The app should tell the user if the speaker is currently buzzing or if it is turned off. Testing for this should be done while testing all other aspects, because it involves actually using the app. If we are using the app, then we can determine if it is responding as intended.

**10 Administrative Content**

**10.1 Milestone Timeline**

The following table will discuss the weeks that make up the project and how our time was budgeted to meet the requirements.

|  |  |
| --- | --- |
| **Week of** | **Task Description** |
|  | **Senior Design I (Spring)** |
| 1/11/16 | Project Introduction |
| 1/18/16 | Group Formation |
| 1/25/16 | Initial Project Proposal |
| 2/1/16 | Research Bluetooth |
| 2/8/16 | Research iBeacon Device |
| 2/15/16 | Research Microcontroller |
| 2/22/26 - 2/29/16 | Continue researching and begin design document |
| 3/7/16 | Spring Break |
| 3/14/16 - 3/28/16 | Design models and continue design paper |
| 4/4/16 - 4/18/16 | Finalize paper and being ordering parts |
|  | **Senior Design II (Summer)** |
| 5/16/16 | Discuss plans/go over inventory |
| 5/23/16 - 6/6/16 | Design and test app |
| 6/13/16 - 7/4/16 | Prototype microcontroller and iBeacon device |
| 7/11/16 - 7/25/16 | Test and debug system/being final documents |
| 8/1/16 | Final Presentation/Evaluation |

Table 6: Milestone Timeline

**10.2 Budget**

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Cost Per Unit** | **Quantity** | **Total Cost** |
| iOS Device | Already Owned | 1 | $0.00 |
| Light Blue Bean | $30.00 | 1 | $30.00 |
| Casing (3D printed) | $30.00 | 1 | $30.00 |
| Speaker | $15.00 | 1 | $15.00 |
| Estimote Sticker Beacons | $10.00 | 10 | $100.00 |
| Wires | Already Owned | 2 feet | $0.00 |
| Misc. Supplies/Tools | $50.00 | NA | $50.00 |
| Total Cost |  |  | $225.00 |

Table 7: Price List

**10.3 Financing**

The budget for this project will be funded for solely by the members of the group within the project. Most of the materials needed are either not too expensive or are already owned by someone in the group. Therefore, we will not need any sort of external funding for our system.

**11 Final Project Summary / Conclusion**

To wrap up this document, we wanted to go over a few closing thoughts about what we have learned this semester, some problems that we faced or may face, and what we look forward to next semester. This semester has been very productive in terms of setting us up to build our system. Copious amounts of research, planning, and design have come together to create the foundation for our Senior Design project. We have spent many hours coming up with solutions to all the problems that we faced and with all the planning required for such an elaborate project. We think this semester's work has created an excellent basis for the implementation that we will do next semester.

With every big project there will arise problems that will have to be dealt with. The best way to deal with problems is to prevent them from happening with thorough preliminary planning. This semester has proven to be great for us and the planning of our system. Overall, our experience this semester has been relatively smooth. The biggest problems that we ran into were deciding which parts to use for different components of the system. Some of the problems that we may face next semester could arise from trying to learn to code in a new language or when actually building the device. Hopefully, everything will go as planned and we won’t run into any problems, but we are prepared for anything.

**Appendix**

**Permissions**

Estimote Permission

**Jess Anderson** (Estimote, Inc.)

Mar 25, 10:09

Hi Amish,

Thanks for reaching out! My name is Jess, I'm a Community Manager here at Estimote.

Good news! Your stickers should arrive by the end of the day today. Here is your tracking information: <https://wwwapps.ups.com/WebTracking/track?trackNums=1ZXF74620297763285&track.x=Track>

Help yourself to any photos from our website or blog! And hey, when you finish your senior project, get in touch with us. Perhaps we can feature you on the Get Inspired section of our website. Keep in touch!

Let me know if there's anything else I could assist you with, happy to help!

Take care,

Jess Anderson | Community Manager | [jess@estimote.com](mailto:jess@estimote.com)

Hello Estimote Team!

I have two questions for you today. When we talked last, I just ordered an Estimote Sticker kit and was informed that I should receive it in March. I am just checking up to see if this date is still valid or if I should be expecting them later. Also, I am using these items in a Senior project which will require documentation. I would like to use pictures from your website to help improve my document but need consensus from you to allow me to use pictures from Estimote website. So I am checking if I am allowed to use pictures on estimote.com and put them in my documentation with the proper citations. Please let me know when possible.

Regards,

Amish Kanji

Punch Through Permission

Your request (3806) has been solved. To reopen this request, reply to this email.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | Nadine Hac | **Nadine Hachouche** (Punch Through Design)  Apr 13, 14:01  Hi!  That should be good! Please share your project with us, if possible. I know the team will be excited!  Best,  Nadine |   Hello,  I am doing a project for one of my classes at the University of Central Florida, and I was wondering if it is ok if we used some of the pictures of the Light Blue Bean from your website. Please let me know as soon as you can.  Thank you,  Joshua Radicchi |

Multicomp Permission

Joel,

No problem. This email grants you permission to use images of the mentioned product in your project.

Thanks for checking.

**Christian Moist**

eCommerce Marketing Manager

[www.MCMelectronics.com](http://www.mcmelectronics.com/)

[Facebook](http://www.facebook.com/mcmelectronics) | [Twitter](http://twitter.com/mcmelectronics) | [LinkedIn](http://www.linkedin.com/company/69573)

[Our 2014 Catalog is here and it's our biggest catalog EVER! Click here to view](http://www.mcmelectronics.com/content/en-US/promo/flyer/index)

**From:** gardyasz@knights.ucf.edu [mailto:gardyasz@knights.ucf.edu]

**Sent:** Sunday, April 24, 2016 1:27 PM

**To:** customerservice@mcmelectronics.com

**Subject:** MCM - To obtain an RA or to contact customer service.- Request

**Send Email Form**

|  |  |  |
| --- | --- | --- |
| **Company Name:** |  |  |
| **Account #** |  |  |
| **First Name** |  | Joel |
| **Last Name** |  | Gardyasz |
| **Email Address** |  | gardyasz@knights.ucf.edu |
| **Phone #** |  |  |
| **Message Details** |  | I am using the MULTICOMP MCKPR-G1720-4011 PIEZO BUZZER in a Senior Design project which will require documentation. I would like to use pictures from your website to help improve my document but need consensus from you to allow me to use these images. So I am checking if I am allowed to use pictures and put them in my documentation with the proper references. If at all possible, I would like an email confirmation about the use of the picture of the buzzer in my document. Regards, Joel Gardyasz |
| **Customer Request** |  | To obtain an RA or to contact customer service. |