

# Smart Room Attendance Monitoring and Location Tracking with iBeacon Technology

A Major Qualifying Project  
Submitted to the Faculty of  
WORCESTER POLYTECHNIC INSTITUTE  
in partial fulfillment of the requirements for the  
Degree of Bachelor of Science  
in  
Electrical and Computer Engineering

By  
Savannah Redetzke, Andrew Vanner, and Raymond Otieno

Advisor: Professor Kaveh Pahlavan

Co-Advisor: Professor Jahangir Rahman

*This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see <http://www.wpi.edu/Academics/Projects>*

\*This project was sponsored by Worcester Polytechnic Institute, Electrical and Computer Engineering Department as a Major Qualifying Project (MQP) for senior undergraduate studies.

## Abstract

The objective of this project was to develop a system and a phone application using iBeacon technology to track people's attendance and location at different types of events. This includes tracking their location by using a location algorithm and receiving identifying information from each person through the use of a phone application. This information will then be sent to a server for record keeping.

The project began with extensive data collection to determine the capabilities of the iBeacon technology. We collected RSSI data based on different transmission powers and determined the correct number, placement and the transmission power to ensure complete coverage. Location data was collected using both the iBeacon & Eddystone Scanner phone application developed by Flurp Laboratories and the Least Mean Squared Algorithm. The location results for both of these was compared for accuracy using Cramer-Rao.

The next part of the project depended upon the creation of a phone application that was capable of detecting the signals from the iBeacon, collected data from the user of the application and sending this information to a server. To accomplish this, the phone application was developed for an Android Phone with Bluetooth Low Energy Capabilities. The application allows users to enter a username and password and includes the ability to terminate the monitoring capabilities of the application. The data collected by the phone application is sent to a text file stored within the offsite server Dropbox. This text file is accessed by an Excel worksheet which imports the data and quantifies it for data representation and analysis.

## Acknowledgements

The group would like to thank a few individuals and organizations for assisting us throughout this project. Professor Kaveh Pahlavan is our main advisor who helped us in coming up with ideas for this project and guided us through the entire process. He is joined by Professor Jahangir Rahman who provided us with valuable feedback on our progress, reports, and presentations. We would also like to thank Julang Ying, a WPI graduate student, for assisting us with location algorithm testing and analysis.

Finally, we would like to thank WPI and the Electrical and Computer Engineering department for providing us with the funding, locations, and resources needed to complete this project. Without all of these supporting people, this project would not be possible.

# Table of Contents

<b>1.0 Introduction</b> .....	<b>6</b>
1.1 Motivation .....	6
1.2 Project Description .....	7
1.3 Report Outline .....	8
<b>2.0 Background in Localization Using iBeacon</b> .....	<b>10</b>
2.1 Bluetooth Low Energy .....	10
2.2 iBeacon .....	12
2.3 Estimote Beacons .....	12
2.4 Performance of Location Algorithms .....	13
<b>3.0 Design and Performance Evaluation Methodology</b> .....	<b>15</b>
3.1 iBeacon Coverage Design Goal .....	16
3.2 Smartphone Application .....	18
3.3 Design of Algorithms .....	21
3.3.1 Centroid Algorithm .....	24
3.3.2 Trilateration .....	24
3.4 Performance Evaluation with Cramer Rao Lower Bound .....	25
3.4.1 Cramer Rao Lower Bound for Ranging .....	25
3.4.2 Cramer Rao Lower Bound for Localization .....	26
3.5 Server and User Interface .....	27
<b>4.0 Results and Discussion</b> .....	<b>29</b>
4.1 Algorithm Evaluations .....	29
4.2 Smartphone Application Results .....	37
4.3 Server Interface Results .....	38
<b>5.0 Conclusions and Future Work</b> .....	<b>45</b>
<b>References</b> .....	<b>46</b>
<b>Appendix A: Code for Beacon Placement in Room AK 207</b> .....	<b>49</b>
<b>Appendix B: Code for Graphic Interface Excel Sheet</b> .....	<b>56</b>
<b>Appendix C: Raw Data Collection for Measuring RSSI in Room 233</b> .....	<b>60</b>
<b>Appendix D: Conference Paper for ISENG 2017</b> .....	<b>67</b>

## Table of Figures

Figure 1.1 Project General Architecture .....	8
Figure 2.1: Frequency Organization Diagram.....	11
Figure 2.2: Two iBeacons developed by Estimote .....	12
Figure 2.3: Advertisement of UUID, Major, and Minor by 3 different Estimote devices .....	13
Figure 2.4: Localization Algorithms Summary .....	14
Figure 3.1: Initial Consideration for iBeacon Placement in Atwater Kent Room 233 .....	16
Figure 3.2: Path-Loss Color Gradient Graphic of Atwater Kent Room 233 .....	17
Figure 3.3: RSSI Color Gradient Graphic of Atwater Kent Room 233.....	18
Figure 3.4: Interaction with Identifiable Beacons .....	19
Figure 3.5 Estimote Beacon Producing Light .....	20
Figure 3.6: Reconfiguration of Beacons.....	21
Figure 3.7: Ideal Trilateration for 3 Transmitters (iBeacons).....	22
Figure 4.1: Coverage Probabilities Matlab Contour Simulation .....	30
Figure 4.2: Centroid Algorithm Showing the Intersections .....	31
Figure 4.3: Centroid Algorithm Matlab Localization Results .....	31
Figure 4.4: Simulation of Trilateration for Location 1 .....	32
Figure 4.5: Simulation of trilateration for Location 3 .....	33
Figure 4.6: Results for LMS Algorithm .....	34
Figure 4.7: Matlab Contour Simulation for the 4 iBeacons .....	35
Figure 4.8: Matlab Contour Simulation for the 5 iBeacons .....	35
Figure 4.9: Cumulative Probability of Distance Measurement Error .....	36
Figure 4.10: Developed Smartphone Application Screen .....	37
Figure 4.11: Developed Smartphone Application Message .....	38
Figure 4.12: Text File for Phone Import .....	39
Figure 4.13: Raw Data Imported with Command Button Program .....	39
Figure 4.14: Pivot Table Displaying Attendance for <i>snredetzke</i> .....	40
Figure 4.15: Pivot Table for Displaying Dates User <i>snredetzke</i> Attended Class .....	40
Figure 4.16: Bar Graph Displaying Class Attendance of <i>snredetzke</i> .....	41
Figure 4.17: Slicer Diagram of Data Imported into Excel Program .....	42
Figure 4.18: Slicer Diagram of Data Pertaining to Room 233 .....	43
Figure 4.19: Slicer Diagram of Data Pertaining to Room 233 and Class CS2102.....	44

## Table of Equations

Equation 3.1.....	17
Equation 3.2.....	22
Equation 3.3.....	23
Equation 3.4.....	24
Equation 3.5.....	25
Equation 3.6.....	25
Equation 3.7 .....	26
Equation 3.8.....	26
Equation 3.9.....	26
Equation 3.10.....	27
Equation 3.11.....	27
Equation 3.12.....	27
Equation 3.13.....	27

# 1.0 Introduction

This project began with identifying an issue which effects people in day to day life, and how we could design a system to alleviate the problem. The issue we identified was the difficulties associated with attendance monitoring and indoor localization. Once we determined the issue, our next step was to design an overall system architecture to enable us to focus our efforts on each component in turn. The introduction of our paper serves to detail both our motivation and our overall approach to solving the problem.

## 1.1 Motivation

The motivation for this project came from the difficulty of recording the number of people at any type of event or class. At WPI, the traditional ways to monitor who attends an event are to swipe your WPI Student Identification (ID) Card on a card reader, physically sign in on a piece of paper, or use a device called “clickers” that connect to the classroom computer. However, each of these traditional ways have drawbacks that would be solved with our project’s design.

The first option is swiping your WPI Student ID Card which monitors your attendance by recording an identification number that is unique to you from the card. However, it is time consuming, laborious and prone to missing people who attend an event but do not swipe their ID card. The second: physically signing in on a piece of paper is equally laborious, prone to human error and if you wished to track who went to the class or event you have to enter the information onto a computer manually. The third, the “clicker” is designed for students to answer in class questions displayed on the projector. Their attendance is monitored based on who answered the questions. However, the students who forget their clicker cannot answer the questions, therefore losing class participation points in addition to losing credit for even attending the class.

Most students, however, will always remember to have a phone in their pocket or purse reducing

the chances of a student missing out on potential class participation grade due to a failure of attendance taking. In addition, students and faculty will not have to stand in line to swipe their IDs, or pass around a piece of paper. This will not only save time to be used in more productive ways but will also increase the efficiency of the system while reducing the amount of human error.

## 1.2 Project Description

This project was organized into four main sections. These sections were: to monitor when multiple people enter a room, to transit identifying information about each person when in room, to identify the location of each person within the room and to record and organize information for each person to track attendance and location.

To complete each section of the project, we designed the basic design architecture as shown below in Figure 1.1. The design architecture is broken down into four basic components. Number one represents the multiple iBeacons required for both determining location and identifying when a person enters a room. Number two represents the smartphone that using the application that this project will design will detect the iBeacons, utilize a location algorithm to determine where in the room it is, and submit identifying information to number three, the server, at regular intervals. The server stores this information to be accessed by number four, the computer graphic interface, which is used to identify, track and record a person's username, date that they are in the room, time they are in the room, the room they are in (designated by a number) and the location in the room.



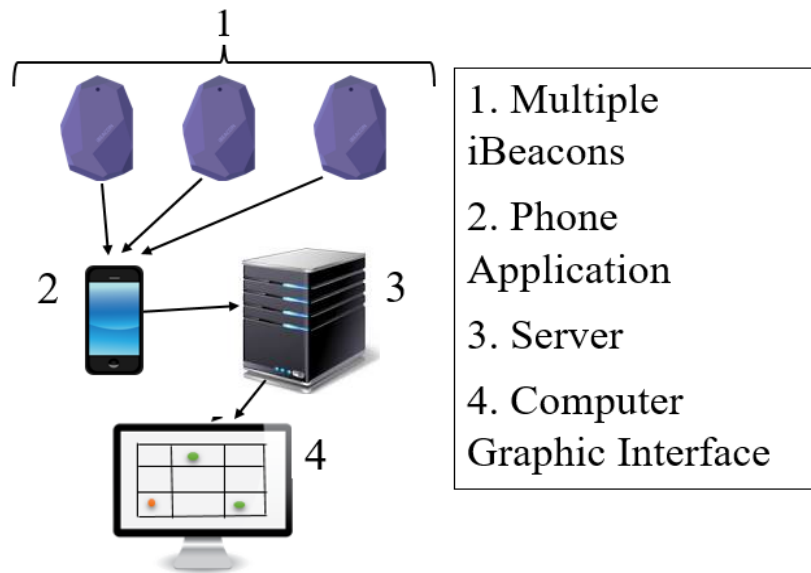


Figure 1.1: Project General Architecture Overview to Demonstrate the Block Diagram of the Project and the Lines of Communication between Different Technologies Included in the Project

### 1.3 Report Outline

This report covers the background technology that makes this project work, the methodology used in this project to achieve the desired goals, the results of the project, and a conclusion that provides ideas for future work. The Background in Localization Using iBeacon(Section 2.0) serves as an introduction to Bluetooth Low Energy as well as the iBeacon implementation of this technology. There is an explanation of Estimote Location Beacons for those that are unfamiliar with the devices. It also covers different types of location algorithms considered for this project.

The Design and Performance Evaluation Methodology (Section 3.0) covers the various components of the project such as algorithm accuracy and optimization, smartphone application development, server interface development, and physical beacon deployment strategies.

The Results and Discussion (Section 4.0) contains the results of this project to see how testing compares to theory and how the project's goals are actualized. We compare the accuracy of the algorithm implemented compared to other methods as well as the Cramer Rao Lower Bound (CRLB). This shows

how the phone application will look for users as well as how the graphic user interface works for administrators.

Finally, the Conclusion (Section 5.0) summarizes the project as a whole. In addition, we speculate on the ways this system can be used in the professional world and what changes would need to be made before this can happen.

In addition to this report we have included four appendixes. Appendix A is a compilation of the code used in Matlab to determine Beacon Placement calculations. Appendix B contains the Visual Basic code used in creating the user interface. Appendix C includes all the data collected during the project, such as RSSI, distance measurements and calculated path-loss. Appendix D contains a copy of a paper that we submitted to the 2017 International Systems Engineering Conference.

## 2.0 Background in Localization Using iBeacon

This project contains four main technologies, that when utilized together made this project possible. These technologies are: Bluetooth Low Energy (BLE), iBeacon, Estimote Beacons and Location Algorithms. BLE is a wireless network connection which is used to exchange tiny and static radio signals within short distances. It has a low power consumption which makes ideal to be used in this project because it is easy to power and does not contribute to being an obtrusive addition to any designated space. BLE is contained within our second technology, iBeacon which was designed to provide indoor geolocation by transmitting signals and allowing applications to receive the signals and information. This technology is also contained within the Estimote Beacon. It allows this project to determine RSSI values, the distance you can be from an Estimote Beacon and add information to each Estimote Beacon for it to transmit to any application with Bluetooth 4.0 or above [2]. The last section of technology that this project uses are location algorithms. These algorithms are lines of code which can use RSSI measurements and multiple Estimote Beacons to determine the exact location of someone in a designated space. Channel, or path-loss modelling was used to determine best setup for the access points and Cramer Rao Lower Bound was used to evaluate algorithm with the least variance of estimator location error [1].

### 2.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a wireless network connection used to exchange tiny and static radio signals within short distances, approximately 200 m or less depending on the device. It was introduced about half a decade ago as Bluetooth 4.0 to the M2M communication and internet of things (IOT) world although it was originally invented by Nokia in 2006 under the name “Wibree”. BLE was known by its commercial name Bluetooth Smart after being transferred to Bluetooth Special Interest Group (BSIG). What makes this technology unique, is the “Low Energy” in the name. BLE has a very

low power consumption as compared to regular Bluetooth (802.15). Based on usage and settings, BLE can last up to 3 years or more on a single battery.

The other features of BLE include using a 4 byte ARM cortex CPU embedded with accelerometer for motion detection, temperature sensor, and operates within 2400 - 2483.5 MHz on an unlicensed frequency band. It uses 3/40 available channels for advertising it's position. The three channels are 37 (2402 MHz), 38 (2426 MHz), and 39 (2480 MHz) as shown on Figure 2. 1. Broadcasting rate and width for each channel is 50 Hz and 2 MHz respectively. Figure 2.1 shows how the channels are organized within BLE. The transmission power ranges from 0 to -40dBm. The remaining channels are used for receiving the signals. BLE uses Gaussian Frequency Shift Keying Modulation and Frequency Hopping Spread Spectrum with a connection time of 6ms, 3 byte Cyclic Redundancy Check, a 4 byte integrity check, and a 16 byte Advanced Encryption Standard security.

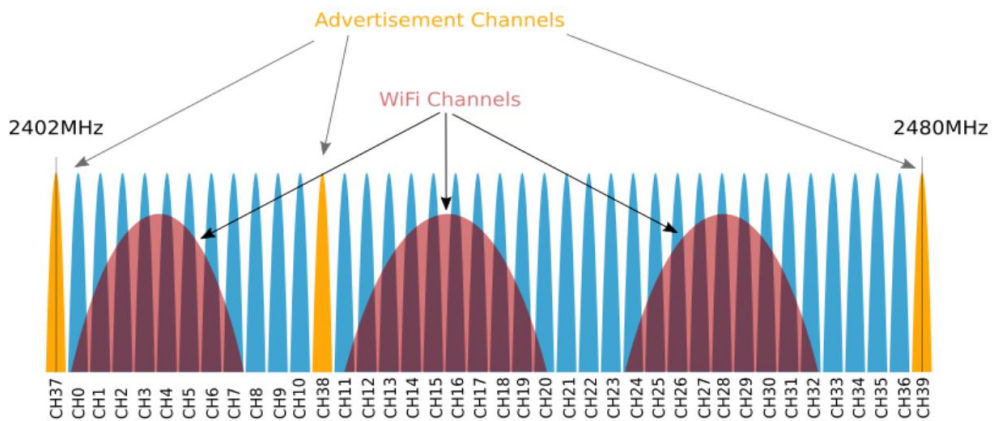


Figure 2.1: Frequency Organization Diagram of the Forty Channels used by BLE for Three of its Uses: Advertisement, Receiver and WiFi Lineup (Source argenox.com)

## 2.2 iBeacon

iBeacon is Apple's version of BLE created to provide indoor geolocation and to introduce it to

the public via Apple's iOS platform. It permits applications to listen and search for advertised radio signals. The technology is compatible with any blue tooth device with version 4.0 and above. iBeacon is used in ranging and region monitoring. The transmitter, also known as the broadcaster announces its location by sending a packet that contains Universal Unique Identifier (UUID), Major, Minor, and transmit (Tx) power. It operates in a similar way to a lighthouse, broadcasting its position and IDs however it does not receive signals.

### 2.3 Estimote Beacons

The hardware that uses both iBeacon technology and BLE that is utilized within this project is known as Estimote Beacons. These beacons are developed by Estimote Inc. and are about the size of a child's fist. Two Estimote Location Beacons are pictured below in Figure 2.2. All Estimote devices have a default UUID of B9407F30-F5F8-466E-AFF9-25556B57FE6D. UUID is a 256 (128 bits) nibble string for differentiating a large group of beacons that are related. It follows an 8-4-4-4-12 digit format.



Figure 2.2: Two iBeacons developed by Estimote utilizing BLE Technology to Transmit Information that can be Detected by BLE Compatible Devices

The Major is a 32 nibble string that is a subset of UUID. Minor is similar to Major in size except that it identifies the exact beacons. The last part of the packet is transmission power that is responsible for identifying the distance from the transmitting beacon which is measured in terms of a meter from the beacon. The Figure 2.3 below is a screenshot taken to show an iOS iBeacon application receives signals

from broadcasters.

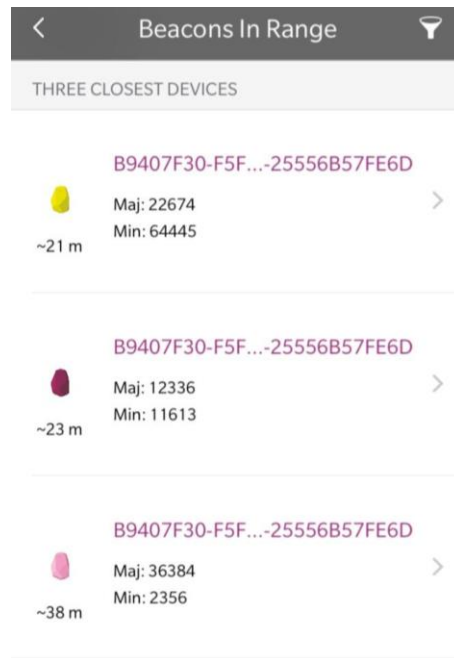


Figure 2.3: Advertisement of UUID, Major, and Minor by 3 different iBeacon devices using the Phone Application “Estimote” developed by Estimote

## 2.4 Performance of Location Algorithms

There are numerous localization algorithms that are available for use with different error ranges and accuracies. Some examples of the algorithms are fingerprinting, trilateration, triangulation, Pedestrian Dead Reckoning (PDR), proximity, and hybrid methods [35]. These location algorithms will be compared to a mathematical limit called Cramer Rao Lower Bound (CRLB). CRLB is a mathematical lower bound that represents the best accuracy for a location estimator. The closer these algorithms fall to it, the more accurate they are.

Fingerprinting uses the nearest node to estimate a distance/location with the help of access points and their corresponding RSS values[35]. The nearest node is determined by an access point with the highest RSS value. Whereas Triangulation uses three or more triangles’ intersections to estimate the

location and distance of a receiving node. It mostly focuses on the angle and one know/measured distance of the triangles being used and the other two sides determined based on the data.

Trilateration is similar to triangulation but it uses spheres to determine the distance/ location in 3D and circles in 1 and 2D configurations. This project is based on a 2D configuration.

A summary of how various localization algorithms compare with each other is depicted in Figure 2.4. These are just a few of the many available algorithms. Some of the abbreviations on the figure are Angle of Arrival (AoA), Distance of Arrival (DoA), and Received Signal Strength (RSS).

	Trilateration	Triangulation	Scene Analysis	Proximity	Dead Reckoning	Hybrid Algorithms
Measurements Techniques	RSS, ToA, TDoA	AoA, DoA	Fingerprinting	RSS	Velocity Acceleration	Hybrid
Accuracy	Medium	Medium	High	Medium	Medium	High
Time Cost	Low	Low	High	Low	Low	Medium
Distance	Low	Low	Medium	Medium	Low	High
Algorithm Type	Deterministic	Deterministic	Probabilistic	Deterministic	Deterministic	Probabilistic Deterministic
Specification	Time Based	Direction based	Range based	Range Based	Time Based	Hybrid

Figure 2.4: Localization Algorithms Summary Comparison Based on Various Measurement Techniques [35]

### 3.0 Design and Performance Evaluation Methodology

To successfully develop this product it is divided into four separate parts. Each part serves to monitor when multiple people enter a room, transit identifying information about each person when in room, identify the location of each person within the room, and record this information for each person. The four main sections are: Location Algorithms, iBeacon Coverage Design, Server and User Interface, and Smart Phone Application.

The first part of this project, monitoring when multiple people enter the room, falls into the section of iBeacon Coverage Design. iBeacon Coverage Design is focused on determining the range of each of the iBeacons, as well as their placement so they can transmit over an entire designated space. The second part of this project, transiting identifying information about each person in the room, falls under the purview of the Smartphone Application. This section concentrates on working with the Android operating system, Dropbox Server and the Estimote iBeacon Application Program Interface (API) to both enable the application user to submit their information and send it to the Dropbox Server for data collection and processing. The third part of this project, identify the location of each person uses the section, is Location Algorithms. The location algorithms section concentrates on determining a feasible location algorithm using RSSI values, determining its accuracy compared to other location algorithms and implementing the chosen algorithm into the Phone Application. The last part of this project, recording the information sent by the Application for each person falls into the Server and User Interface task. This task is focused on developing an Excel program on a Dropbox server which imports the data deposited by the phone into a text file, and organizes the information to a user friendly display.



### 3.1 iBeacon Coverage Design Goal

This section was accomplished by configuring beacon locations in a room and checking the RSSI values at multiple points in the room. Essentially, RSSI measurements were taken at multiple points with multiple configurations and multiple dBm transmission levels. The ideal result is the most amount of coverage with the lowest power setting to ensure any person in the room can be tracked and the battery in the beacon will last as long as possible.

We collected our data from Room 233 in Atwater Kent. Our first step was to collect data using five beacons in the room at the different available transmission levels that the beacon could be set at. These transmission levels were: -12 dBm, -4 dBm, and 4 dBm. The second data collection tested different selections of iBeacons in different locations. Ten possible locations for iBeacons were selected and the RSSI for each iBeacon in each location in the room was measured and graphed. In Figure 3.1, the testing locations of the iBeacons are shown.

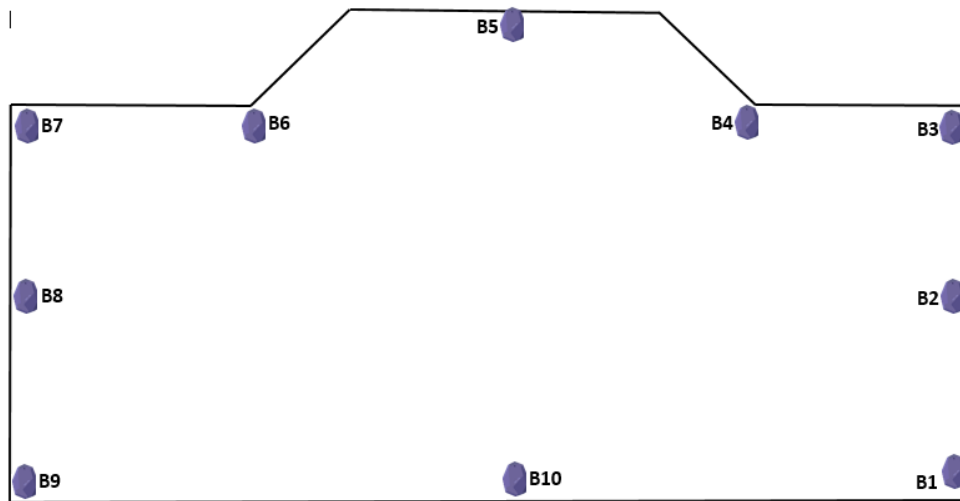


Figure 3.1: Initial Consideration for iBeacon Placement in Atwater Kent Room 233 to Ensure the Complete Coverage of the Room

From these measurements of the multiple iBeacons, we determined that 5 iBeacons arranged in

the following configuration would be sufficient in providing coverage to the entire room. To test our determination, we placed the iBeacons in their respective locations and calculated the path-loss model and the RSSI measurements at seventeen points. The path-loss model is shown below in Figure 3.2 and the RSSI measurements are shown in Figure 3.3. In determining the path loss model, we used the IEEE model path loss equation given by:

$$L_o = 10\log(G_t) + 10\log(G_r) + 20\log(\lambda/(4\pi)) \tag{3.1}$$

$$L_p = L_o + 10\alpha_1\log(d/PB) + 10\alpha_2\log(d/dPB) \quad d < dPB$$

The assumptions made in this equation were that the gain is 0, the room is a typical office environment and alpha  $a = 2$ . We also assumed, based on a phone’s sensitivity to RSSI, that an acceptable path loss was 85.

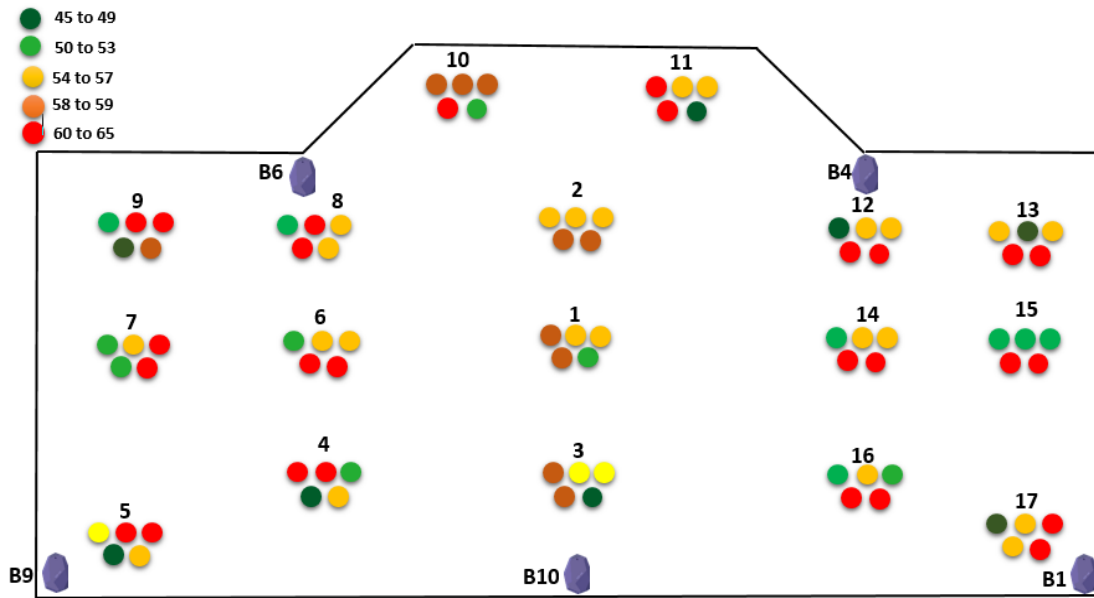


Figure 3.2: Path-Loss Color Gradient Graphic of Atwater Kent Room 233 Using Chosen Beacon Configuration to Provide Sufficient Room Coverage

This conclusion was supported by our data measurements of RSSI, which is shown in Figure 3.3

as a gradient of colors based on the measurement at each of points in the room. The dark green means the signal from the iBeacons was the strongest, and as you can see the majority of the room is well covered, with only one point (8) displaying a RSSI of above -95dbm. However, we still have 4 additional readings from the other iBeacons which will enable the Phone Application and Location Algorithm to successfully determine both if a person is in the room and their location at point 8, despite the one questionable reading.

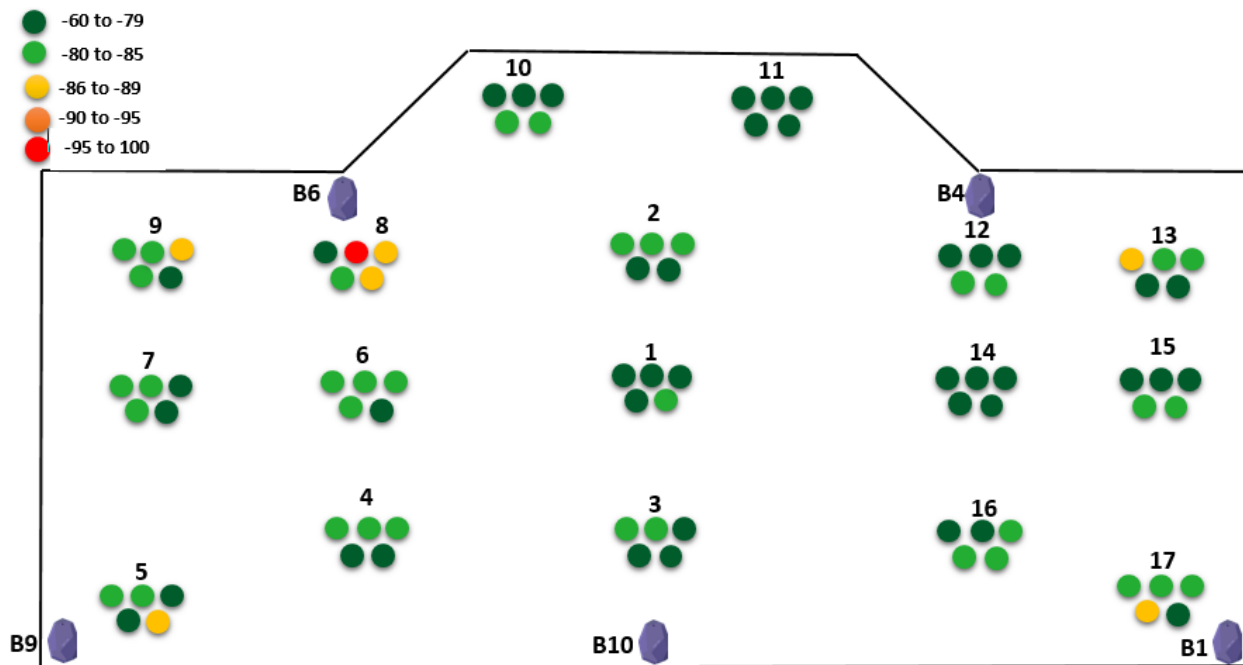


Figure 3.3: RSSI Color Gradient Graphic of Atwater Kent Room 233 Using Chosen Beacon Configuration to Provide Sufficient Room Coverage

### 3.2 Smartphone Application

The application for interacting with the beacons is developed for Android. This mobile OS allows programmers to develop for free and easily be downloaded onto other phones without having to go through an official app store or modifying the phone. To assist in application development, the group is using the Estimote app which can be found on the Google Play Store. Figure 3.4 is a screenshot of the application locating two nearby beacons.

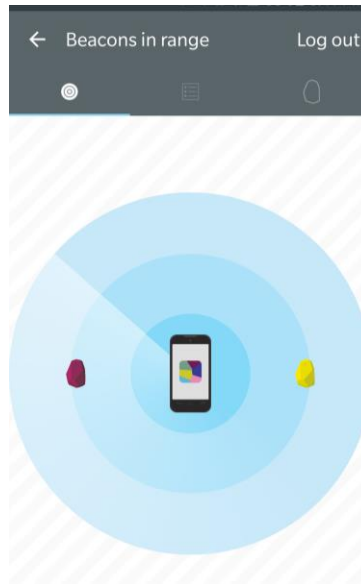


Figure 3.4: Estimote Application Demonstrating that Two Unique and Identifiable Beacons are Currently in Range of the BLE Capable Device

This application has some functions, but serves as a demonstration of Estimote iBeacon technology rather than a full-fledged application. It allows the user to check distance from any individual beacon and shows what kind of text, images, or data can be sent from beacons. The end goal of the application is to have the smartphone use the location algorithm being developed to determine location. It will then send location, user ID, and a time signature to a server. This will be repeatedly sent at a certain interval to keep track of any users who leave a room.

Since the real accuracy of the Estimote Location Beacon is reported to be forty to fifty meters, nine devices were ordered. A joint group email was made to establish ownership of the beacons so data can be read and the settings of the beacons can be changed. By default, the beacons are named after their color. In these sets, there are two named “lemon” (the yellow devices), “beetroot” (maroon devices), and “candy” (pink devices). As a basic form of interaction with the devices the group renamed two of the devices.

It is important to note that only the Estimote account associated with the email address

[mqpie@wpi.edu](mailto:mqpie@wpi.edu) can rename the devices. In this case, the name “lemon” was changed to “Sunny”. As this happened, the physical device began to emit a white light, which is pictured here in Figure 3.5.



Figure 3.5: Estimote Beacon Producing Light to Notify User that the Beacon is Being Reconfigured by Owner

Only the beacon that is being modified is emitting a light. The maroon beacon right behind it can still be detected by the phone but does not receive data for a name to be changed. This is evidence that each beacon can be interacted with individually even if there are multiple in range. To confirm these changes work correctly, the “beetroot” beacon was renamed to “Beats”. Finally, both beacons were checked for names and UUID as seen in Figure 3.6.

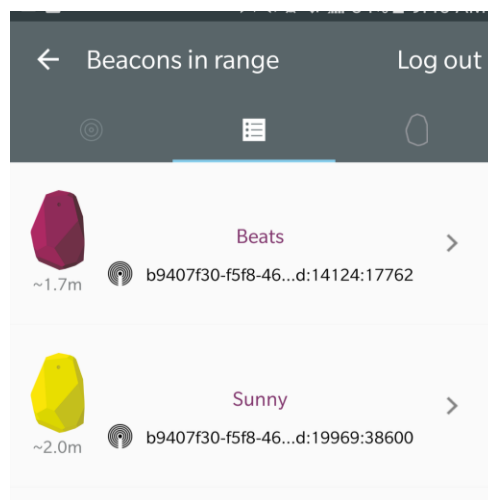


Figure 3.6: Interaction with Beacons Demonstrating the Reconfiguration of the Name of the Beacon was Successful

This figure is very important for a number of reasons; it shows that we can permanently make changes to the programming of the beacons, it shows that we can receive the signal strength to determine distance, and it shows the UUID of each device can be read by the phone. Any other phone will now be able to interact with the newly named beacons even without the owner Estimote account. Once we fully understood how to interact with the beacons using a phone application, we developed our own.

The application was developed for Android smartphone devices using Android Studio. It is compatible with any device using Bluetooth 4.0 and newer. This includes all smartphones running Android 4.3 (released July 24, 2013) and newer.

### 3.3 Design of Algorithms

The process of localization is done via Trilateration. Trilateration, which is very similar to ranging, involves the use of known transmitter locations and Received Signal Strength information to estimate the distance of the receiver. There are two major forms of trilateration; Ideal and Real. Ideal Trilateration, as shown in Figure 3.7 involves finding the distance from at least three points. These distances are the radii of three circles which represent all possible locations that the receiver could be located. The radii is determined by the RSS data as a function of path loss. The exact location of the receiver is then calculated using the intersection of the three circles based on the equation of a circle which mimic the distance formula. The square root of d in equation 3.4 is the same as radius of a circle.

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = d_1^2 \dots (x_n - x_0)^2 + (y_n - y_0)^2 = d_n^2 \quad (3.2)$$

d - distance

n – nth access point,

$x_n, y_n$  - coordinates for  $i^{\text{th}}$  location,

$x_0, y_0$  – coordinates for reference point, and  $i = 1, 2, 3, \dots$

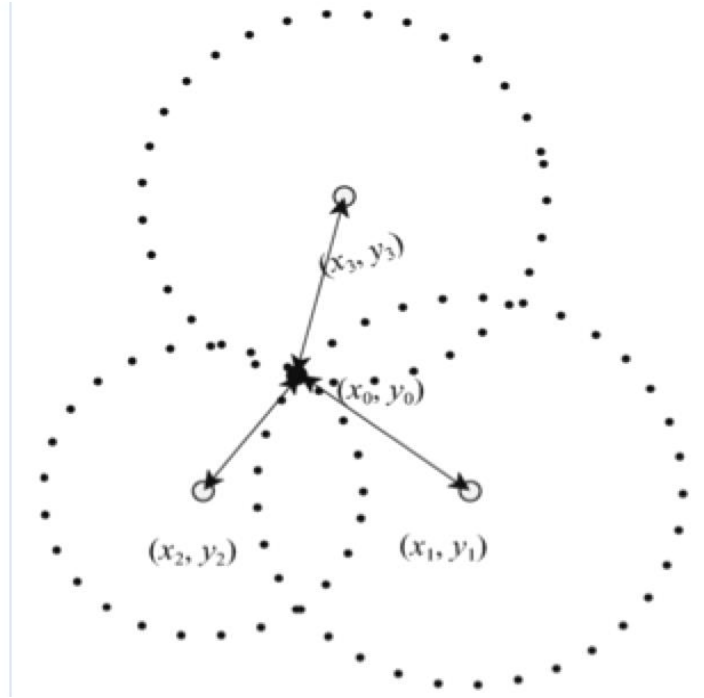


Figure 3.7: Ideal Trilateration for 3 Transmitters (iBeacons) with intersection  $(x_0, y_0)$  as the Location of the Receiver

The team was able to collect more than 300 points on RSS, distance, and power for the calculation of path-loss, variance, and Distance Measurement Error (DME). This information was used to develop graphs to compare the following algorithms to the Cramer Rao Lower Bound (CRLB) for accuracy: Trilateration, Centroid and Least Square.

The first step in comparing location algorithms was to accurately calculate path-loss modeling which is shown in Equation 3.2.

$$C = 3 \cdot 10^8 \text{ m/s}$$

$$F = 2.4 \cdot 10^9 \text{ Hz}$$

$$\lambda = C/f = 1/8 = 0.125$$

$$L_0 = -20 \log (\lambda / (4\pi)) = 40.5 \text{ dB}$$

$$L_p = L_0 + 10\alpha \log (d/d_0) \quad \alpha = 2.5 \quad (3.3)$$

Sigma used was from a typical environment type C that is approximately 5 according to Principles of Wireless Access and Localization [1]. The calculated attenuation for the project's path-loss model was 2.5 although the expected value is between 1.6 and 1.8 for an indoor location in a line of sight setup. This is due to interference from noise, wifi, other transmitting devices and even shadow fading.  $C$  is the speed of light in a vacuum. The  $F$  is the frequency of the transmitting node.  $L_0$  is path loss for a 1 meter distance while  $L_p$  is the path-loss with respect to a particular location based on RSS.

Trilateration alone is not enough to produce an accurate estimation of location. Because of this, other algorithms were considered and evaluated to provide better estimations. The Centroid and Least Square Algorithm were compared to ranging and CRLB.

### 3.3.1 Centroid Algorithm

In Centroid Localization, each transmitter advertises its location and RSS to the receiver, which forms a polygon. Based on this polygon, the receiver uses the average of the center as its approximated location. This method is prone to error depending on the number of nodes it receives information from. The higher the number of nodes, the more the accuracy in estimating the location of the receiver.

Centroid algorithm assumes a beacon range or coverage is in the form of a circle. It is dependent on information from the transmitters (iBeacons) and it uses the node's location information. The nodes that centroid is dependent on are always fixed in the setup environment no matter how the beacons are distributed or set up in a room, centroid will give the estimated location based on the mean or the average of the distances between the beacons. Despite the fact that it is not very accurate, it is still one of the simplest algorithms to work with and also has modified versions like weighted centroid to improve accuracy.



$$m U_i(x,y) = \sum_{j=1}^m B_j(x,y) \quad (3.4)$$

$U_i(x,y)$  : unknown position of receiver

$B_j(x,y)$  : Known position of iBeacon.

m: number of transmitters

### 3.3.2 LMS Algorithm

Least Square is an iterative algorithm that provides a method of estimating x and y as shown below. The algorithm iterates based on guessed value to calculate the position of the unknown node (receiver). The fewer the iterations, the closer the guessed value. Iteratively the error in the solution reduces as the iterations increase since new error is added to old error until the solution stops changing as demonstrated in Equation 3.5.

Suppose  $[ f_1(x,y) f_2(x,y) \dots f_N(x,y) ]^T$

If jacobian matrix  $\mathbf{J} = \begin{bmatrix} \frac{\partial f_1(x,y)}{\partial x} & \frac{\partial f_1(x,y)}{\partial y} \\ \dots & \dots \\ \frac{\partial f_N(x,y)}{\partial x} & \frac{\partial f_N(x,y)}{\partial y} \end{bmatrix}$  (3.5)

An estimate of the solution picked  $\mathbf{U} = [x^*, y^*]$ . The error in the solution is determined as  $\mathbf{E} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{F}$  evaluated at estimate  $\mathbf{U}$ . Thus the new solution is  $\mathbf{U} + \mathbf{E}$  [1]

### 3.4 Performance Evaluation with Cramer Rao Lower Bound

Cramer Rao Lower Bound is the lowest value of variance of an estimator. This bound is used to compare the theoretical minimal DME using RSS with results of different algorithms finding the location using the RSS from deployed Beacons. It is used to determine how far off the developed Android

Application DMEs are by comparing them to the CRLB.

### 3.4.1 Cramer Rao Lower Bound for Ranging

The CRLB is calculated using the inverse of Fisher Information Matrix. For a single beacon and a single RSS reading, we observe,  $\mathbf{O}$ , which is the RSS given by Equation 3.6.

$$\mathbf{O} = P_r = P_0 - 10\alpha \log d + X \quad (3.6)$$

Equation 3.6 was used to relate the observation  $\mathbf{O}$  (the received power  $P_r$ ) to probability distribution function of the observation given the distance:

$$f(\mathbf{O}/d) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)e^{-\frac{(P_r - P_0 + 10\alpha \log d)^2}{2\sigma^2}} \quad (3.7)$$

The probability distribution function (Equation 3.7) of the observation point was used to find to find the Fisher matrix, which is shown in Equation 3.8. CRLB is shown on equation 3.9 as the inverse of the Fisher equation. It is also the variance of estimate ( $\sigma^2$ )

$$-E \left[ \frac{\partial^2 \ln f(\mathbf{O}/d)}{\partial d^2} \right] = E \left[ \frac{\partial \ln f(\mathbf{O}/d)}{\partial d} \right]^2 = \left[ \frac{10\alpha}{(\ln 10)\sigma d} \right]^2 \quad (3.8)$$

$$CRLB = \mathbf{F}^{-1} = \sigma^2 = \left[ \frac{(\ln 10)\sigma d}{10\alpha} \right]^2 \quad (3.9)$$

$$i = 1, 2, 3, \dots, N$$

$\mathbf{O}$  is the observation being measured,  $\alpha$  represents the parameter being measured.

$f(\mathbf{O}/d)$  Represents probability distribution function of the observation based on the observation and the parameter. A single location has zero or no Gaussian noise,  $x = 0$ . For multiple points,  $\mathbf{O}$  changes as shown below and the Fisher equation is multiplied by number of observations ( $N$ ).  $\sigma^2$  is variance while standard deviation is the square root of variance[1].

This equation can be extended for multiple points (N). Since the N depends on the number of observations, Fisher equation is repeated with a change in the observation point O for the desired number of points. Fisher equation is then multiplied by N to match the number of N observations.

### 3.4.2 Cramer Rao Lower Bound for Localization

Localization in a 2D environment required the use of CRLB for localization. This variance estimator is normally used on locations with RSS values and coordinate information of the transmitters (iBeacons). The location of the reference points  $(x_i, y_i)$  and the access points  $(x, y)$  are used to calculate and estimate the location of the received power  $dP$  as depicted in Equation 3.10 [1].

$$dP_i(x, y) = -\frac{10\alpha_i}{\ln 10\alpha} \left( \frac{x-x_i}{r_i^2} dx + \frac{y-y_i}{r_i^2} dy \right) \quad i = 1, \dots, N, \quad (3.10)$$

The range estimate  $dr$  was then used to  $dP$  in vector form as shown in Equation 3.11.

$$\begin{aligned} d\mathbf{P} &= \mathbf{H}dr & dr &= (\mathbf{H}^T\mathbf{H})^{-1} \mathbf{H}^T d\mathbf{P} \\ d\mathbf{P} &= \begin{bmatrix} dP_1 \\ dP_2 \\ dP_N \end{bmatrix} & dr &= \begin{bmatrix} dx \\ dy \end{bmatrix} & \mathbf{H} &= \begin{bmatrix} \frac{x-x_1}{r_1^2} & \frac{y-y_1}{r_1^2} \\ \vdots & \vdots \\ \frac{x-x_N}{r_N^2} & \frac{y-y_N}{r_N^2} \end{bmatrix} \end{aligned} \quad (3.11)$$

The covariance of the location estimate was used due to the shadow fading being a zero Gaussian random variable as shown in Equations 3.12 and 3.13.

$$cov(dP_i, dP_j) = \begin{cases} \sigma^2, & i = j \\ 0, & i \neq j \end{cases} \quad i, j = 1, 2, \dots, N \quad (3.12)$$

$$cov(dr) = \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \quad (3.13)$$

$$\sigma_r = \sqrt{\sigma_x^2 + \sigma_y^2}$$

### 3.5 Server and User Interface

To accomplish the tasks of collecting and managing the data provided by the Phone Application, this project uses a DropBox Server and an Excel program to input the data and organize into a user friendly graphic interface. The Graphic Interface is located on the DropBox Server for access from any computer with the correct username and passcode.

To accomplish the task of creating a server and user interface, multiple server possibilities were entertained. The first was to use Amazon Web Services which offers a broad set of possible services such as global computation, storage, database, application and deployment services. Amazon Web Services would allow the group to make a website that was accessible from any computer and was graphically pleasing and easy to use. However, this service was found to be beyond the scope of this project due to a lack of experience in the coding needed to create a website. In addition, to successfully launch a website, there is no option but to pay monthly for it. The project also hit the roadblock that there was a lack of availability to successfully link a Phone Application to the website. Though it is possible to do this, it requires more background knowledge than what our team possesses. Due to these difficulties an alternative was discovered for proof of concept. Instead of using Amazon Web Services, this project is utilizing the free service known as DropBox. DropBox is set up on your computer just like a folder on your hard drive, except the files on this folder automatically sync online to any other computer or mobile device linked to that account. To fulfill the proof of concept for this project, the Phone Application will be sending its information to a text file on a designated DropBox folder. In this DropBox folder, there is an Excel File which automatically imports the information from this text file and sorts it by username, date, time and class. To automatically import information, the excel file uses a combination of macro's and Visual Basic, the code for which is provided in Appendix A. It imports the data and then uses a combination of graphs, pivot tables and slicers to represent the data.

## 4.0 Results and Discussion

Overall the methodology for each component of our system was successful in producing results. The first set of results was iBeacon Coverage Design. After collecting data we determined that in AK 233, five beacons transmitting at a power of -4 dBm would provide the coverage required. The rest of this results section covers algorithm evaluations, smartphone application, and server interface results.

### 4.1 Algorithm Evaluations

A simulation of four Beacons in a room was run in Matlab as seen in Figure 4.1. It shows the coverage probabilities of each Beacon in relation to changing distance. The yellow is the strongest and the closest coverage of the iBeacon while dark blue is the lowest strength. The white part is due to noise and surrounding interference including shadow fading. It is due to 90% shadow fading margin which is determined by the coverage possibility based on distance from transmitter (iBeacon). The coverage is calculated using path loss formula (Shown in Equation 3.2) except that a fade margin (X) is added to it. Adding a fading margin to the estimate increases the probability of coverage on operating with minimum signal strength. Shadow fading causes variations in average RSS and decrease in power as the distance between transmitter and receiver increases.

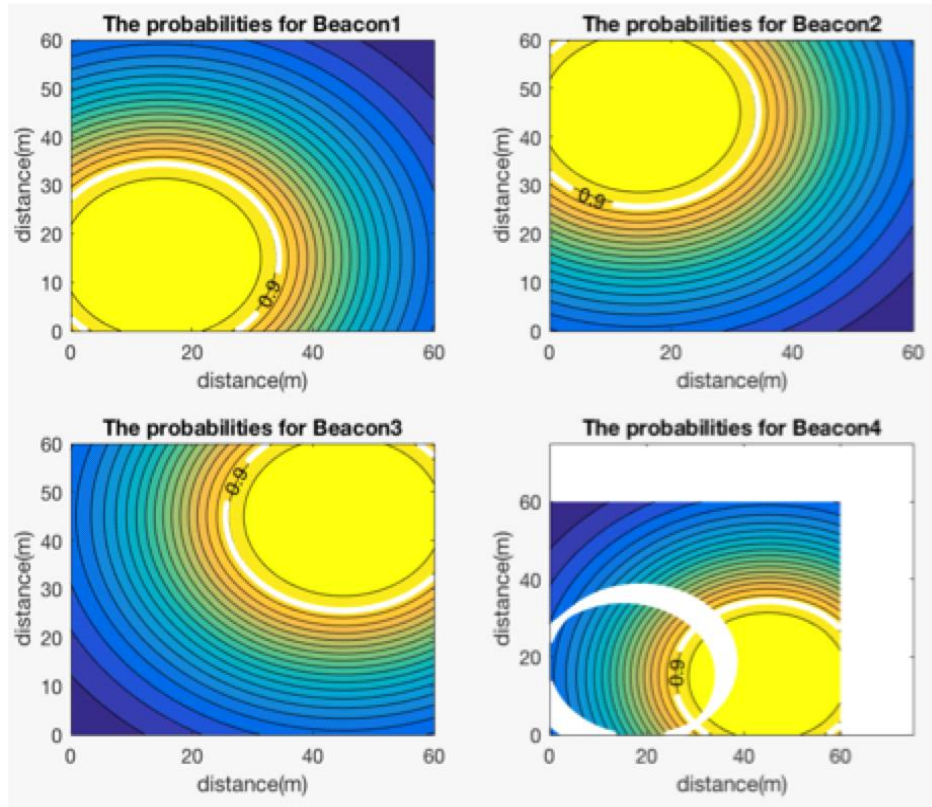


Figure 4.1: Coverage Probabilities Matlab Contour Simulation for 4 iBeacons based on Distance from Transmitter

As a result of this coverage, we were able to test different algorithms and determine which should be used to give the best accuracy. The first algorithm that we tested was the Centroid Algorithm as shown in Figure 4.3. Using Matlab we ran the Centroid Algorithm using 4 Beacons in room 207. We found that no matter where the Beacons were placed in the room, the algorithm will always give you the location in center of the room as displayed by Figure 4.2. This causes this algorithm to have a large error, and therefore unreliable for our project.

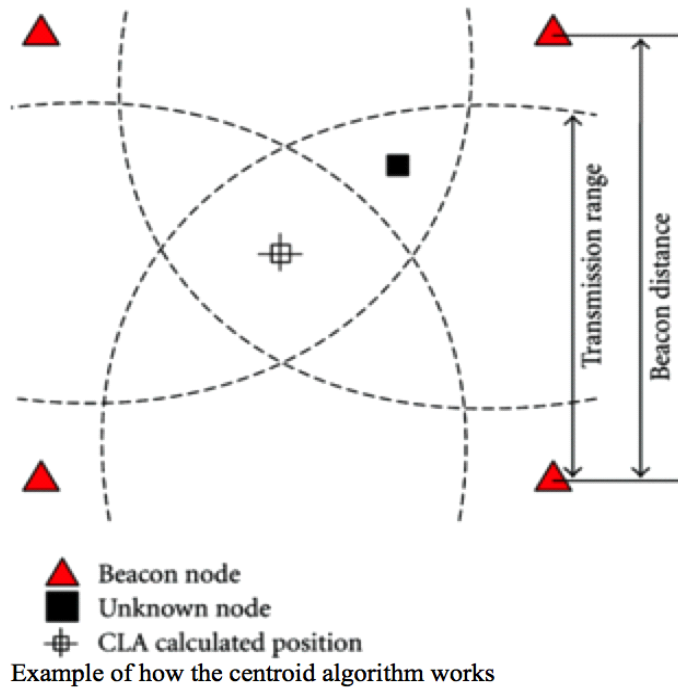


Figure 4.2: Centroid Algorithm Showing the Intersections of the beacons' coverage and the mean of the distances between them

```

Command Window

>> Centroidtest ([1.8,2.4;1.8,3.7;2.3,2.5])

ans =

    1.9667    2.8667

fx >>
  
```

Figure 4.3: Centroid Algorithm Matlab Localization Results with the Calculated Coordinates of the Receiver.

The second algorithm that we tested was the Trilateration Algorithm. Using data collected from room 207 in Atwater Kent we ran a Matlab simulation of RSS ranging localization (Trilateration) as shown in the Figure 4.4 and Figure 4.5. The closer the intersection of the circles, the better the accuracy of the algorithm. In some cases the circles did not intersect, which makes it prone to a larger error when approximating the location of a receiver. Having more transmitting devices, thus more coverage and

intersections of the circles increases the accuracy of the algorithm. The locations, indicated by a star, are of a receiver based on RSS from the surrounding access points. This algorithm ended up being better than the Centroid Algorithm however still unreliable especially when the circles do not intersect.

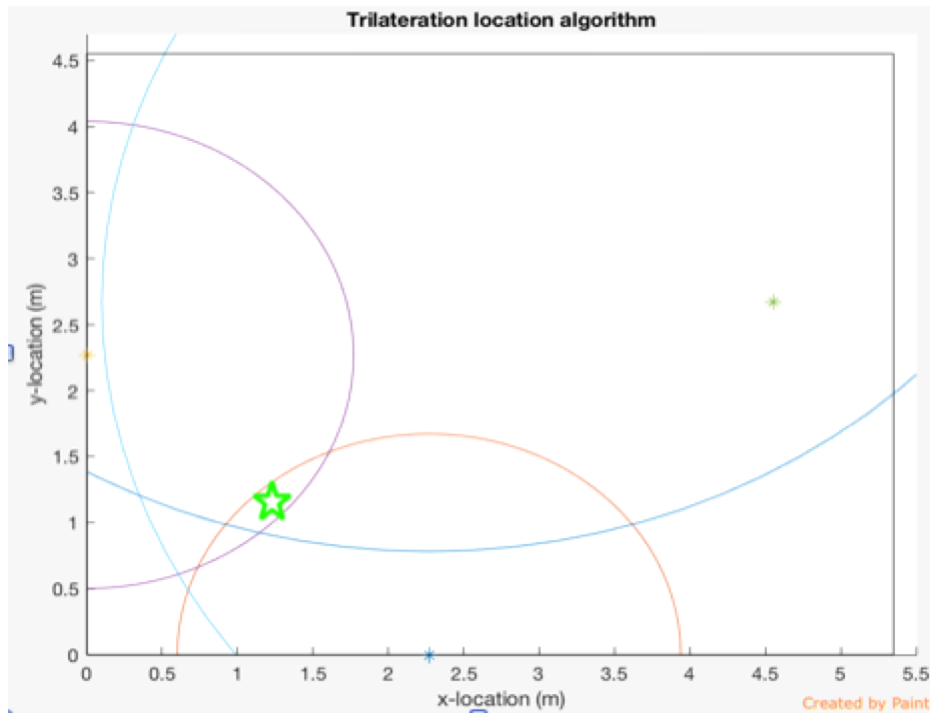


Figure 4.4: Simulation of Trilateration for Location 1 with 4 circles intersecting at approximately (1.13, 1.12)



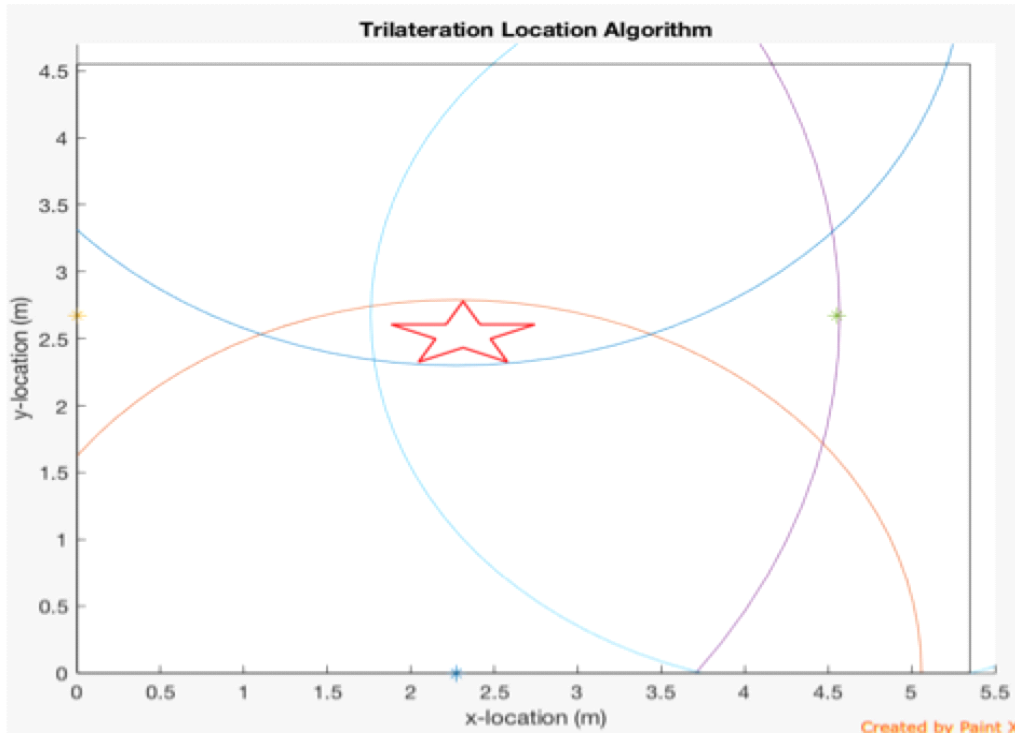
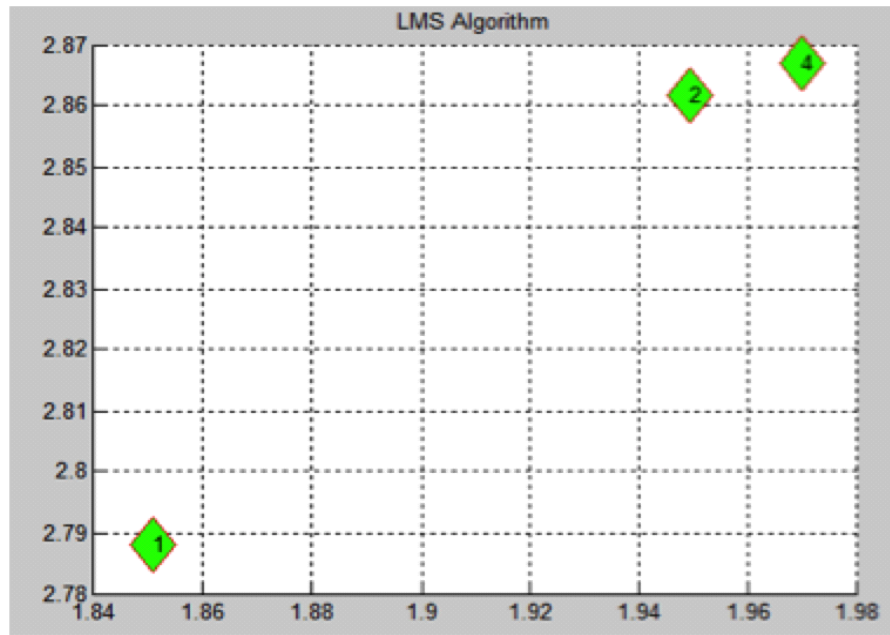


Figure 4.5: Simulation of Trilateration for Location 3 with 4 Circles Intersecting at Approximately (2.45, 2.52)

The final Location Algorithm we tested was the Least Mean Square. Figure 4.6 shows the results obtained from simulating Least Mean Square in Matlab. Point 3 is invisible because it is very close to where the algorithm estimates as the location of the receiver (point 4). The numbers in the diamond shapes are the iteration numbers. In this case, the coordinates of the receiving node is approximately 2.865 for y and 1.965 for x. This algorithm is by far the most accurate with a final error percentage of 0.069%. This error is well within the acceptable threshold of 10%, which makes it a good fit for this project.

### Least Mean Square Algorithm (for location 3)



Results for LMS algorithm

Figure 4.6: Results for LMS Algorithm based on iteration of guessed location value. The largest number is the calculated location of the receiver.

To analyze and compare the location algorithms, we used the CRLB as shown in Figures 4.7 and 4.8. The graphs display the accuracy of location determination based on distance from four Beacons (4.7) and five Beacons(4.8). The locations where the beacons were located based on the path-loss modelling figures used to determine best positions for the access points. The yellow and green parts of the figures have closer contours which represents lower Cramer Rao values with better coverage while the dark blue parts have higher CRLB values and spaced out contours. The further apart the contours, the further a receiver is from the transmitter, and the lower the coverage[3].

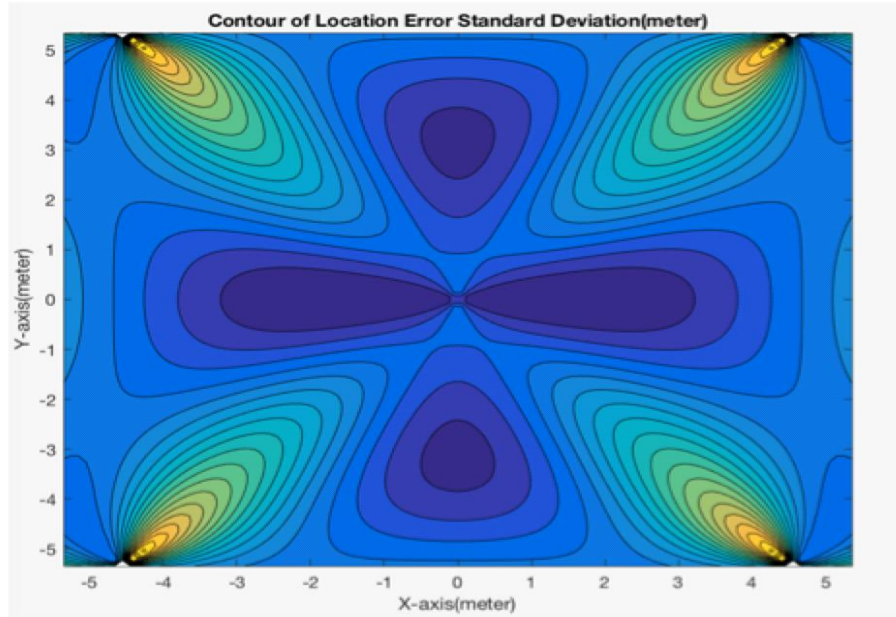


Figure 4.7: Matlab Contour Simulation for the 4 iBeacons estimate location error Installed in Room AK 207

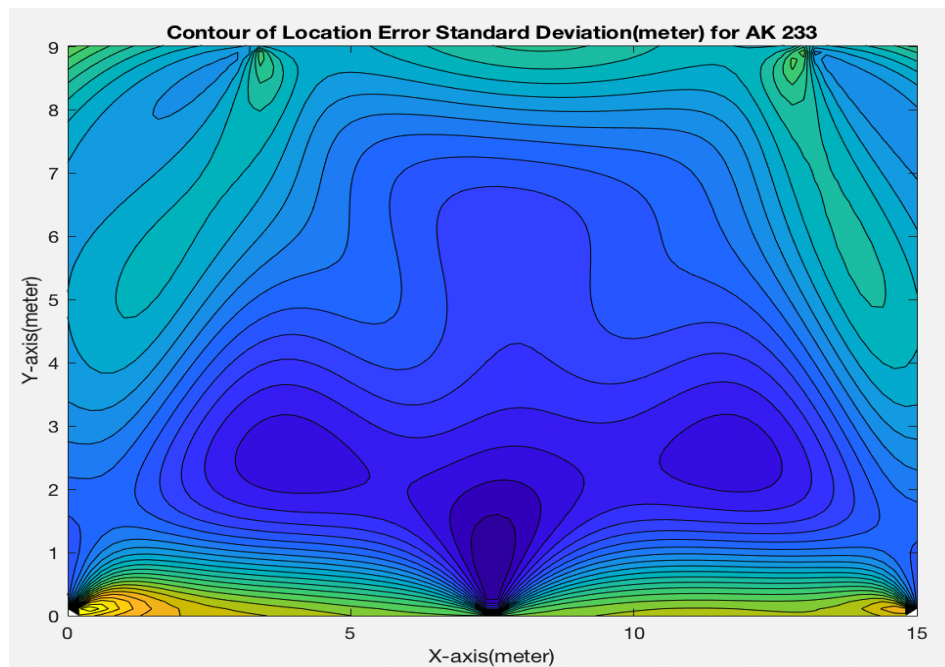


Figure 4.8: Matlab Contour Simulation for the 4 iBeacons estimate location error Installed in Room AK 233

Using Matlab, our final step in Location Algorithm analysis was to compare the CRLB to the different algorithms as shown in Figure 4.9. The algorithm that is closest to CRLB is the Least Square Algorithm, meaning that it is by far the most accurate of the algorithms we analyzed. Therefore this is the algorithm we implemented in the phone application. The ranging data in the graph was calculated by the difference between actual measured error and application distance given by the iBeacon Scanner Application developed by Flurp [36], which was used to obtain RSS values. This gives the cumulative probability as a function of distance error, represented as “Data” in the graph. Cumulative probability is the percentage chance that a variable falls within a set range. In short, a higher cumulative probability per amount of DME is better.

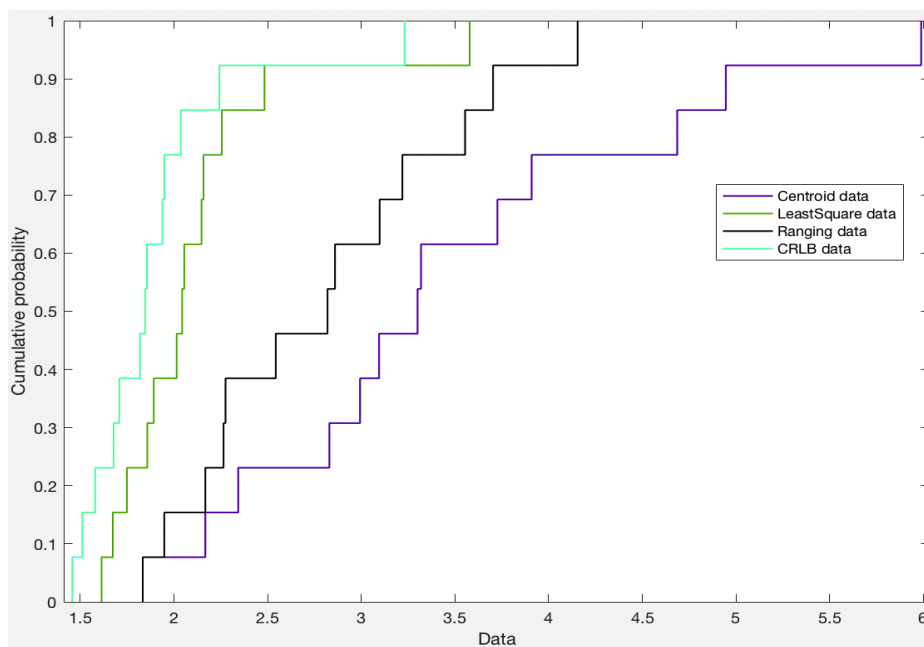


Figure 4.9: Cumulative Probability of Distance Measurement Error of LMS, Centroid Algorithm, and Ranging Cramer Rao Lower Bound Model.

When the phone application runs this on multiple sets of Beacons, a user’s location is determined with a percent error of 0.069%. Typically, an error of less than 10% is considered acceptable so this result is extremely good. This algorithm was successfully implemented on the developed smartphone

application.

## 4.2 Smartphone Application Results

When the user opens the application on their phone they are prompted with a screen as shown in Figure 4.10. This figure shows where the user can input their name and password or ID number. They also have the option to enable or disable when they are being monitored to protect their privacy. Once they accept being monitored, they can keep the app open or allow it to run in the background.

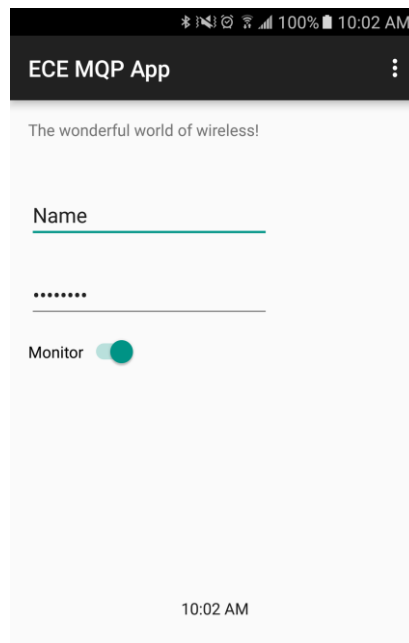


Figure 4.10: Developed Smartphone Application Screen Including Fields for Name, Password and Monitoring Switch

Once the device picks up the signal from the iBeacons, a message will display to the user. Figure 4.11 is an example of one such message.

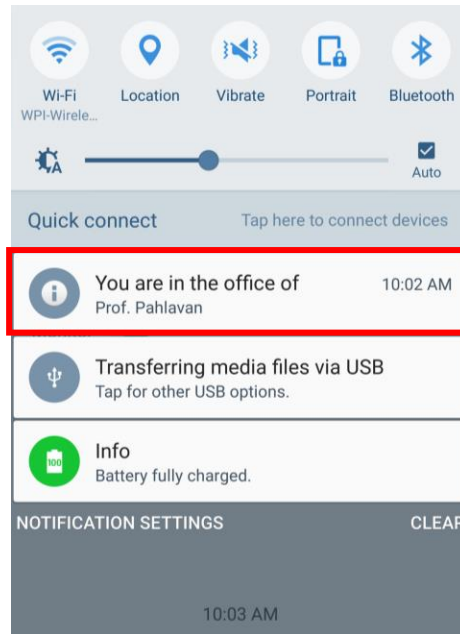


Figure 4.11: Developed Smartphone Application Message Confirming that the User’s Attendance has been Successfully Accounted for as Displayed in Red Rectangle

In this example, the user is informed that they are in a professor’s office. This is a confirmation that the user is being tracked. The message can be customized depending on which iBeacons they are near. This application then sends the name, time, and location to a text file located on the Dropbox server.

### 4.3 Server Interface Results

Next, the excel file imported this information and created pivot tables, pivot graphs and slicers to represent the data. The text file created is shown below in Figure 4.12. The image shows the username, date, time, room number and simulated location coordinates as they would be imported into the excel file.

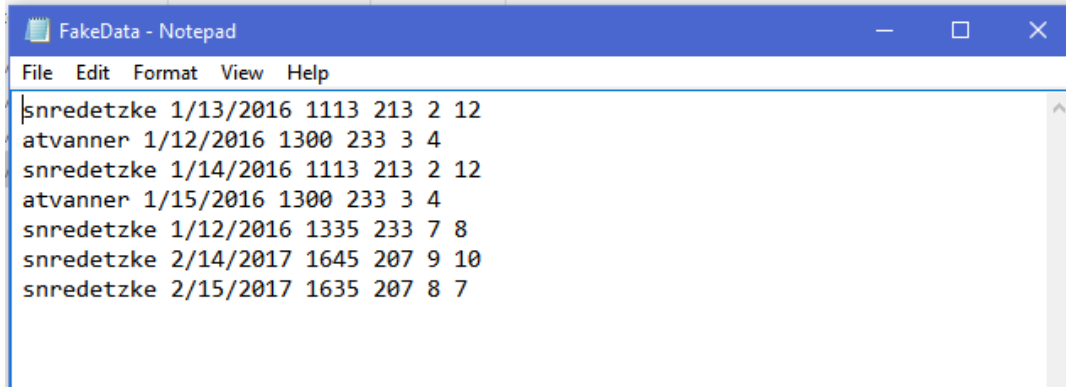


Figure 4.12: Text File Located on Drop Box Server Used for Information Depository for Data Sent from Developed Smart Phone Application

To automatically import this data into the excel program, a macro is used which records set of instructions to perform a particular task that is summed up in a single instruction. This macro is called in a command button program which is displayed as a button on the main sheet. The macro is called Refresh and the code is display in Appendix B into the visual basic command button. After inputting the data, a program is run that compares the times, dates and room number against a database of the known classes that are included in our a separate page of the excel program. The raw data is shown below in Figure 4.13. Also note that the button to the right is the command button referenced earlier. It both refreshes the data inputted by the phone as well as determines what class each person is in.

User	Date	Time	Room	X Axis	Y Axis	Date	Class
snredetzke	1/13/2016	1113	213	2	12	1/13/2016	ME1800
atvanner	1/12/2016	1300	233	3	4	1/12/2016	ECE2310
snredetzke	1/14/2016	1113	213	2	12	1/14/2016	ME1800
atvanner	1/15/2016	1300	233	3	4	1/15/2016	ECE2310
snredetzke	1/12/2016	1335	233	7	8	1/12/2016	ECE2310
snredetzke	3/14/2017	1645	207	9	10	3/14/2017	CS2102
snredetzke	5/15/2017	1635	207	8	7	5/15/2017	CS2102

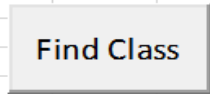


Figure 4.13: Raw Data Imported from Text File Located on Drop Box Server with Command Button Program Using Visual Basic

The next step to designing this graphic is to display this information in an easy to understand table.

To do this we used a pivot table which allows us to select the data we want to see, and to filter out what we don't. The pivot table, is shown below in Figure 4.14. This table is a way to display how many times a user goes to each of their classes' total. As shown in the figure, you can select which user you want to see, in this case it is on selection *snredetzke*. The data is shown in the "Count of Date" row, which counts how many times you attended each class, and at the end it shows the "Grand Total" which totals how many times you went to all the classes combined.

User	snredetzke			
Class	ME1800 ECE2310 CS2102 Grand Total			
Count of Date	1	2	2	5

Figure 4.14: Pivot Table Displaying Attendance for user *snredetzke* Displaying the Number of Days they Attended Class

You can also add in the actual dates into the rows to display the days on which they went to class, as shown below in Figure 4.15. As displayed in the figure, you can also see that how many classes on any particular date each student attended, based on chosen person in the user filter.

User	snredetzke			
Attendance	Class			
Date	ME1800	ECE2310	CS2102	Grand Total
1/12/2016			1	1
1/13/2016	1	1		2
1/14/2016		1		1
3/14/2017			1	1
Grand Total	1	2	2	5

Figure 4.15: Pivot Table for Displaying the Individual Dates that User *snredetzke* Attended Class in Addition to the Grand Total of Classes Attended

However, tables can be hard to read, so there are two other ways this excel program can display the data for easy comprehension. The first is in graph form. The graph is linked to the pivot table above.



It changes based on the information you select for the table to display. For Figure 4.15, data, the table displays the information in bar form as shown in Figure 4.16. This graph displays the user *snredetzke* and which classes they attended, how many times. The user *snredetzke* attended ECE 2310 a total of two times for instance, as displayed by the orange bar.

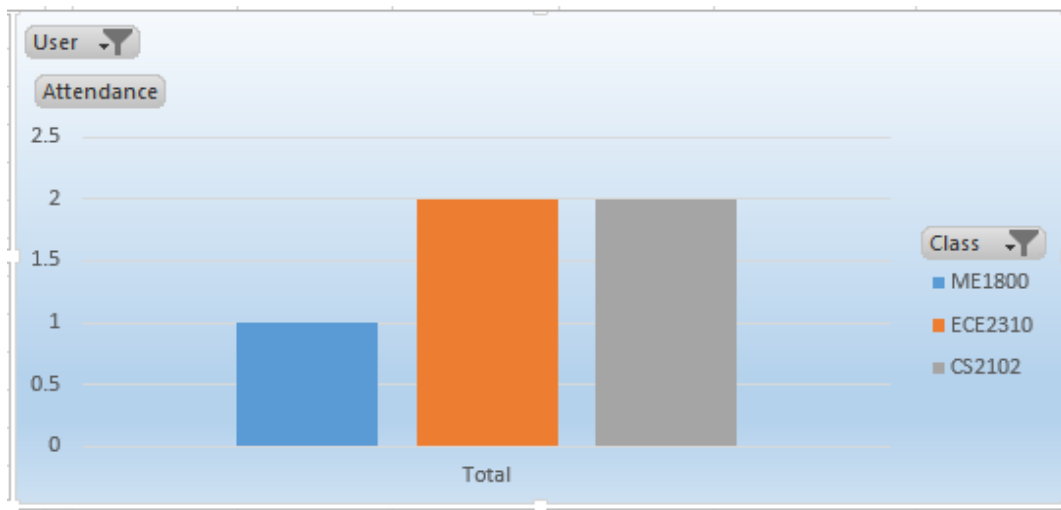


Figure 4.16: Bar Graph Displaying Class Attendance of *snredetzke* Using Pivot Table Referenced in Figure 4.15

The other way to display this data is through what is called a slicer, as shown below in Figure 4.17. It takes the data and allows you to sort it by user, room number, time and class. For instance, if you wanted to determine who had class in a particular room at all, select what room you want and the other sections will update to reflect that data that it has on that particular room. As you can see, all the data is reflected in four main boxes, class, user, time and room.

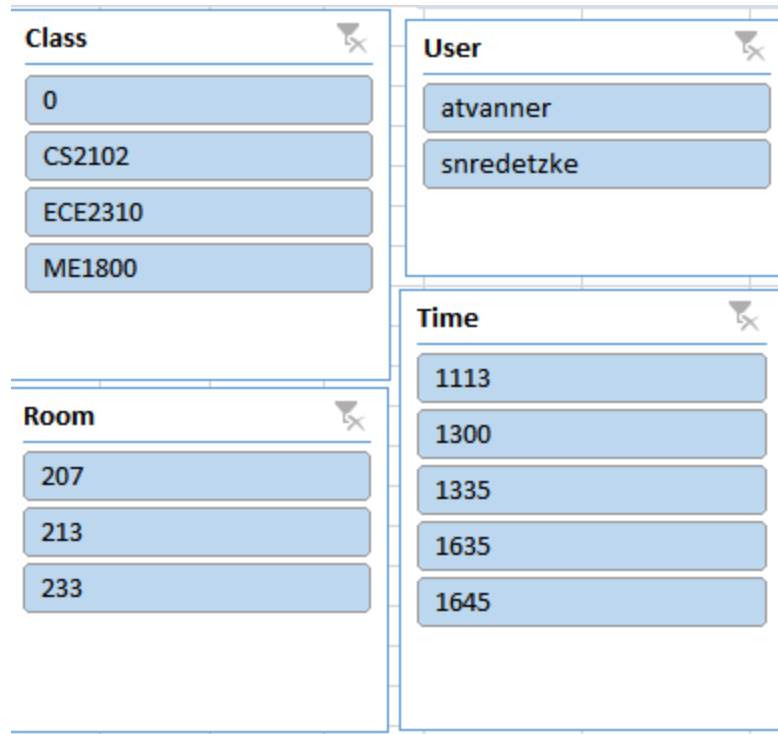


Figure 4.17: Imported Data Organized in Slicer Form Representing the Class, User, Time and Room with the Potential to Filter the Information

To select information, you need only click on one of the pieces of data. The data that pertains to the information you selected is stays highlighted blue, while the data that doesn't pertain to it is removed from the slicer, as shown in Figure 4.18.

Class	User	Room	Time
CS2102	atvanner	207	1300
ECE2310	snredetzke	213	1335
ME1800		233	

Figure 4.18: Imported Data Organized in Slicer Form Representing the Data only from the Selected Room, 233

Selecting the room number tells you that both users, *atvanner* and *snredetzke*, took multiple classes in that room, if you are looking for something even more specific you can select a class, or time to narrow the range even further. For instance if you select a specific class such as CS2102 as shown in Figure 4.19 it removes information that doesn't meet both the class and room criteria. The information you are left with tells you that only user *snredetzke* takes that class in room 233, and the last time she was recorded as being there was at 1335, and 1:35.

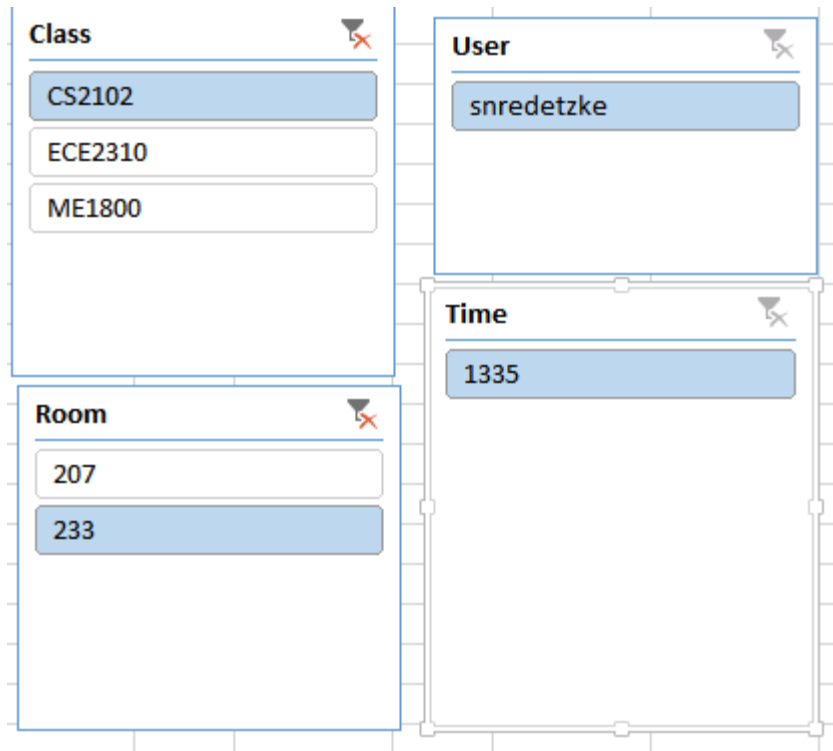


Figure 4.19: Imported Data Organized in Slicer Form Representing the Data only from the Selected Room, 233 and Class CS2102

## 5.0 Conclusions and Future Work

In the end this project met the design specifications set forth. We took the problem of Attendance Monitoring and created a new system to improve on the problem's lack of efficiency. After researching location algorithms and BLE technology, we created a system that uses multiple components working together. Using a combination of BLE beacons, a smartphone application, and a computer server, the entire process of attendance monitoring is complete. The beacons were set up in a way that offers maximum coverage of a room. In addition, the configuration considered power usage and expected battery life to make it more appealing to expense-savvy businesses. The least means square algorithm implemented provides a very accurate estimation of location. It is the closest to the CRLB of the given algorithms and offers an extremely small distance measurement error of 0.69%. The smartphone application is straightforward and easy for new users to understand. It is able to run this algorithm and send the data to a server, which gives certain users access to information they never had before.

There are always ways products can be improved and this project is no exception. To bring it to market there would be a real private server that has upkeep costs. In addition, the data being sent would be encrypted. The computer server application would be a website that takes a username and password so the right person could access the data anywhere. Also, more clients could be reached if the phone application was also built for iOS devices.

This project has potential for future use in business and academia. A college professor could check which students come to class the most and even where they sit and see if that can affect their grades. In addition, faculty could see if certain classrooms lead to higher or lower amounts of attendance. Overall, this method is faster than manually swiping students in or having an attendance sheet while providing more data.

## REFERENCES

1. K. Pahlavan and P. Krishnamurthy, Principles of Wireless Access and Localization, Wiley, November 2013
2. Yi Sun, Olawole Tunde-Lukan, and Teng Weng, ECE, iBeacon Applications and Hybrid Wi-Fi Localization , MQP
3. Yang, Yang, "Application Performance Evaluation for ." In Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2016 IEEE International Multidisciplinary Conference on, pp. 1-5. IEEE, 2016.
4. Nayef Alsindi, Ind HYPERLINK "<http://www.wpi.edu/Pubs/ETD/Available/etd-042308-115256/>"oor Cooperative Localization for Ultra Wideband Wireless Sensor Networks, 2008
5. "Sign In: Registered Users." *International Journal of Distributed Sensor Networks*. N.p., n.d. Web. 13 Apr. 2017. <http://journals.sagepub.com/doi/full/10.1155/2012/829253>
6. "What Is Ibeacon? A Guide To Ibeacons". 2014. *Ibeacon.Com Insider*. <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>.
7. "A BLE Advertising Primer." *Argenox Technologies*. N.p., n.d. Web. 13 Apr. 2017. [HYPERLINK "http://www.argenox.com/a-ble-advertising-primer/"](http://www.argenox.com/a-ble-advertising-primer/)/ HYPERLINK "<http://www.argenox.com/a-ble-advertising-primer/>"www.argenox.com/a-ble-advertising-primer/
8. "Estimote." *What Is IBeacon? - Estimote Developer* N.p, n.d. Web.13 Apr. 2017. <http://developer.estimote.com/ibeacon/>
9. "What is IBeacon? A Guide to IBeacons." *IBeacon.com Insider*. N.p, 12 June 2014. Web 13 Apr. 2017. <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>
10. Li, Dong, and Jiancheng Wang. "Research of Indoor Local Positioning Based on Bluetooth Technology." *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing* (2009) n. pag. Web <https://pdfs.semanti> HYPERLINK "<https://pdfs.semanticscholar.org/2961/435e73a2348075602d3e89d24ce20429f74d.pdf>"cscholar.org/2961/435e73a2348075602d3e89d24ce20429f74d.pdf
11. Lewars, Giselle and Minh Truong. 2016. *Performance Study Of Mtx Motion Tracker Technology For Indoor Geolocation*. Ebook. 1st ed. Worcester. <http://web.wpi.edu/Pubs/E-project/Available/E-project-043009-162614/unrestricted/MQPLewarsTruong.pdf>.
12. Frenzel, Lou. 2016. "What'S The Difference Between Bluetooth Low Energy And ANT?". *Electronicdesign.Com*.
13. Zeynep Turgut, Gulsum Zeynep Gurkas Aydin, Ahmet Sertbas, Indoor Localization Techniques for Smart Building Environment, *Procedia Computer Science*
14. Lixiong Tan, Fei Luo and Kai Liu, "Weighted centroid location algorithm in Wireless Sensor Network," *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2011)*, Shanghai, 2011, pp. 414-418.
15. Kwang-Yul Kim, Yoan Shin, "Test node-based WCL algorithm for wireless sensor networks", *Ubiquitous and Future Networks (ICUFN) 2016 Eighth International Conference*

16. "A BLE Advertising Primer · Argenox Technologies". 2016. *Argenox.Com*.  
<http://www.argenox.com/a-ble-advertising-primer/>.
17. "Beecon". 2016. *Beaconsandwich.Com*. [http://www.beaconsandwich.com/what-](http://www.beaconsandwich.com/what-is-ibeacon.html) HYPERLINK  
"<http://www.beaconsandwich.com/what-is-ibeacon.html>"is-ibeacon.html.
18. "Bluetooth Vs. Bluetooth Low Energy". 2015. *Link Labs*. [http://www.link-labs.com/bluetooth-](http://www.link-labs.com/bluetooth-vs-bluetooth-low-energy/)  
vs-bluetooth-low-energy/.
19. Contreras, David, Mario Castro, and David Sanchez de la Torre. 2016. "Performance Evaluation Of Bluetooth Low Energy In Indoor Positioning Systems". *Emerging Telecommunications Technologies*. <http://onlinelibrary.wiley.com/doi/10.1002/ett.2864/epdf>.
20. Frenzel, Lou. 2016. "What's The Difference Between Bluetooth Low Energy And ANT?". *Electronicdesign.Com*. [http://electronicdesign.com/mobile/what-s-difference-between-bluetooth-](http://electronicdesign.com/mobile/what-s-difference-between-bluetooth-low-energy-and-ant)  
low-energy-and-ant.
21. "Based Micro-Location with Particle Filtering" *Global Communications Conference* , 2015  
*IEEE* 2016
22. Alexandre Alapetite, John Paulin Hansen "Dynamic bluetooth beacons for people with disabilities" *Internet of Things (WF- IoT), 2016 IEEE 3<sup>rd</sup> World Forum*, 2017
23. Naoki Honma, Kazuki Ishii and Yoshitaka Tsunekawa "DOD-based localization technique using RSSI of indoor beacons" *Antennas and Propagation (ISAP), 2015 International Symposium* , 2016
24. A. Coma, L. Fontana, A. A. Nacci "Occupancy detection via iBeacon on Android devices for smart building management" *Design, Automation and Test in Europe Conference & Exhibition (DATE), 2015* , 2015
25. Zhouchi Li, Yang Yang, Kaveh Pahlavan, "Using iBeacon for newborns localization in hospitals", *Medical Information and Communication Technology (ISMICT) 2016 10th International Symposium on*, pp. 1-5, 2016, ISSN 2326-8301
26. Zhiheng Zhao, Ji Fang, George Q Huang, Mengdi Zhang, "iBeacon enabled indoor positioning for warehouse management" *Computational and Business Intelligence (ISCBI), 2016 4<sup>th</sup> International Symposium on*, 2016
27. Yingying Guo, Yatao Li, Yan Sun, "Accurate Indoor Localization based on Crowd Sensing," *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, ISSN 2376-6506
28. Louay Bassbouss, Gorkem Guclu, Stephan Steglich, "Towards a remote launch mechanism of TV companion applications using iBeacon," *Consumer Electronics (GCCE) 2014 IEEE 3<sup>rd</sup> Global Conference on*, 2014
29. Yang Yang, Zhouchi Li, Kaveh Pahlavan, "Using iBeacon for intelligent in-room presence detection", *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA) 2016 IEEE International Multi-Disciplinary Conference on*, pp. 187-191, 2016, ISSN 2379-1675.
30. Miller. Steven, *The Method of Least Squares*
31. "Gale - Enter Product Login". *Go.galegroup.com*. N.p., 2017. Web. 25 Apr. 2017.  
<http://go.galegroup.com>

32. iBeacon and Eddystone Scanner created by Flurp Laboratories  
<https://play.google.com/store/apps/details?id=de.flurp.beaconscanner.app&hl=en>



## Appendix A: Code for Beacon Placement in Room AK207

```
%%Plot for AK 207 with corresponding beacon placement
%%and distance from receiver (phone)

% function h = circle(x,y,r)
close all;          clear all; clc;
ch = 1;% to continue entering information
while(ch~=2) % choice 2 is for quitting
rectangle('Position',[0 0 5.35 4.55])
axis([0 5.5 0 4.7])

%*****
% x = [0,0,2,2.5,4,4.5,6.5,6.5];%*
% y = [0,2,2,2.5,2.5,2,2,0];    %* Modified for Room
% fill(x,y,'k');                %* AK233 & any other
% axis([0 6.5 0 2.5])           %*
%*****

hold on
th = 0:pi/50:2*pi;
fprintf('What is the radius of the circle\n');
r = input('Give radius size\n');
fprintf('Where do you want to draw circle\n');
x = input('Give x-location\n');
y = input('Give y-location\n');
xunit = r * cos(th) + x;
yunit = r * sin(th) + y;
plot([x], [y], '*', 'MarkerSize', 7);
h = plot(xunit, yunit);
title('Trilateration Location Algorithm')
xlabel('x-location (m)'); ylabel('y-location (m)');
hold off
fprintf('Do you want to plot another circle\n');
ch = input('Enter the choice\n');
end
```

## CODE FOR PATHLOSS & BEACON PROBABILITY

```
clear all;close all;clc;
%% basic path loss model
alpha1=6.7.0; % power gradient
lpmax=85.5; % max path loss in dB
f=2.4e9; % transmitting frequency
c=3e8; % speed of light
```

```

lamda=c/f; % wave length
L0=40.04599702; % 1st meter path loss
pace=0.5;
reliablity = 0.9;
%Distance=40;
t = 1; o = 1;
for Distance = 30:1:75;
x1=Distance/4;y1=Distance/4;
x2=Distance/4;y2=3*Distance/4;
x3=3*Distance/4;y3=3*Distance/4;
x4=3*Distance/4;y4=Distance/4;
x=0:pace:Distance;y=0:pace:Distance;
L1=length(x);
r1=zeros(L1,L1);
r2=zeros(L1,L1);
r3=zeros(L1,L1);
r4=zeros(L1,L1);
for i=1:L1
    for j=1:L1
        r1(i,j)=sqrt((x(i)-x1)^2+(y(j)-y1)^2);
        r2(i,j)=sqrt((x(i)-x2)^2+(y(j)-y2)^2);
        r3(i,j)=sqrt((x(i)-x3)^2+(y(j)-y3)^2);
        r4(i,j)=sqrt((x(i)-x4)^2+(y(j)-y4)^2);
    end
end
lp1=L0+max(10*alpha1*log10(r1),-L0);
lp2=L0+max(10*alpha1*log10(r2),-L0);
lp3=L0+max(10*alpha1*log10(r3),-L0);
lp4=L0+max(10*alpha1*log10(r4),-L0);
alpha2=-4.28;
beta=0.9372;
gamma0=5.31;
sigma1=alpha2*exp(-beta*r1)+gamma0;
sigma2=alpha2*exp(-beta*r2)+gamma0;
sigma3=alpha2*exp(-beta*r3)+gamma0;
sigma4=alpha2*exp(-beta*r4)+gamma0;
for p=1:L1
    for q=1:L1
        pc1(p,q)=1-0.5*erfc((lpmax-lp1(p,q))/sqrt(2)/sigma1(p,q));
        pc2(p,q)=1-0.5*erfc((lpmax-lp2(p,q))/sqrt(2)/sigma2(p,q));
        pc3(p,q)=1-0.5*erfc((lpmax-lp3(p,q))/sqrt(2)/sigma3(p,q));
        pc4(p,q)=1-0.5*erfc((lpmax-lp4(p,q))/sqrt(2)/sigma4(p,q));
    end
end
[c1,h1] = contour(x,y,pc1','w','LevelList',[reliablity],'linewidth',3);
R1 = sqrt((c1(1,2)-x1)^2+(c1(2,2)-y1)^2);
if Distance == 60;
    subplot(2,2,1);
[X1,Y1]=contourf(x,y,pc1',20);

```

```

xlabel('distance(m)'),ylabel('distance(m)'),title('The probabilities for Beacon1');
hold on;
[c1,h1] = contour(x,y,pc1','w','LevelList',[reliablity],'linewidth',3);
R1 = sqrt((c1(1,2)-x1)^2+(c1(2,2)-y1)^2);
clabel(c1,h1);
subplot(2,2,2);
[X2,Y2]=contourf(x,y,pc2',20);
xlabel('distance(m)'),ylabel('distance(m)'),title('The probabilities for Beacon2');
hold on;
[c2,h2] = contour(x,y,pc2','w','LevelList',[reliablity],'linewidth',3);
R2 = sqrt((c2(1,2)-x2)^2+(c2(2,2)-y2)^2);
clabel(c2,h2);
subplot(2,2,3);
[X3,Y3]=contourf(x,y,pc3',20);
xlabel('distance(m)'),ylabel('distance(m)'),title('The probabilities for Beacon3');
hold on;
[c3,h3] = contour(x,y,pc3','w','LevelList',[reliablity],'linewidth',3);
R3 = sqrt((c3(1,2)-x3)^2+(c3(2,2)-y3)^2);
clabel(c3,h3);
subplot(2,2,4);
[X4,Y4]=contourf(x,y,pc4',20);
xlabel('distance(m)'),ylabel('distance(m)'),title('The probabilities for Beacon4');
hold on;
[c4,h4] = contour(x,y,pc4','w','LevelList',[reliablity],'linewidth',3);
R4 = sqrt((c4(1,2)-x4)^2+(c4(2,2)-y4)^2);
clabel(c4,h4);
end

end

```

## CODE FOR CALCULATING CRLB H MATRIX

```

Rx=[1.5241,1.3398; 2.6797,1.3398;4.0200,1.3398;4.0200,3.0480; 2.6797,3.0480;
1.5240,3.0480;1.5240,4.5720; 2.6797,4.5720; 4.0200,4.5720];
Tx=[4.0200,3.0480]; % Coordinates for transmitter
bp=5; % break point
var=5; % variance
for i=1:1:4] r(i)=sqrt((Tx(1)-Rx(i,1)).^2+(Tx(2)-Rx(i,1)).^2);
    if r(i)<5;
a1=6.7; % alpha in free space
P(i)=-10*a1/log(10)*((Tx(1)-Rx(i,1))/(r(i).^2));
Q(i)=-10*a1/log(10)*((Tx(2)-Rx(i,2))/(r(i).^2));
H(i,1)=P(i); %H matrix left column
H(i,2)=Q(i); %H matrix right column
end
if r(i)>=5;
a2=4.5; % alpha in obstructed room

```

```

P(i)=-10*a2/log(10)*((Tx(1)-Rx(i,1))/(r(i).^2));
Q(i)=-10*a2/log(10)*((SD(2)-Rx(i,2))/(r(i).^2));
H(i,1)=P(i);
H(i,2)=Q(i);
end
end
CRLB=var^2*inv(H'*H)
plot(CRLB);

```

CODE FOR CONTOUR LOCATION ERROR

```

close all; clear all; clc;
APx(1) = 4.55;
APy(1) = 5.35;
APx(2) = 4.55;
APy(2) = -5.35;
APx(3) = -4.55;
APy(3) = -5.35;
APx(4) = -4.55;
APy(4) = 5.35;
APx(5) = 0;
APy(5) = 0;
SD = 5;
NUM = 5;
mx = -5.35:0.1:5.35;
my = -5.35:0.1:5.35;
nxy = length(mx);
for yi = 1:nxy
for xi = 1:nxy

for i1 = 1:NUM
alpha(i1) = 6.7;
r(i1,xi,yi) = sqrt((mx(xi)-APx(i1))^2+(my(yi)-APy(i1))^2);
H1(i1,xi,yi) = -10*alpha(i1)/(log(10))*(mx(xi)-APx(i1))/r(i1,xi,yi)^2;
H2(i1,xi,yi) = -10*alpha(i1)/(log(10))*(my(yi)-APy(i1))/r(i1,xi,yi)^2;
end
H(:,:,xi,yi) = [H1(:,xi,yi) H2(:,xi,yi)];
Covv(:,:,xi,yi) = SD^2*(H(:,:,xi,yi)'*H(:,:,xi,yi))^(-1);
SDr(xi,yi) = sqrt(Covv(1,1,xi,yi)+Covv(2,2,xi,yi));
end
end
SDr = SDr';
contourf(mx,my,SDr,20);
xlabel('X-axis(meter)');
ylabel('Y-axis(meter)');
title('Contour of Location Error Standard Deviation(meter)');

```

## CODE FOR LEAST MEAN SQUARE

```
%function [final_x,final_y] = LMSforMQP
known_references = [1.8,2.4;1.8,3.7;2.3,2.5];
initial_guess = [1.93, 2.454];
distances = [0.49669,0.850,0.4933];
if size(known_references,2) ~= 2

error('location of known reference points should be entered as Nx2 matrix');

end

figure(1);
hold on
grid on
i=1;
temp_location(i,:) = initial_guess ;
temp_error = 0 ;

for j = 1 : size(known_references,1)
    temp_error = temp_error + abs((known_references(j,1) - temp_location(i,1))^2
+ (known_references(j,2) - temp_location(i,2))^2 - distances(j)^2) ;
end

estimated_error = temp_error ;
% new_matrix = [ ];
while norm(estimated_error) > 1e-2 %iterative process for LS algorithm

    for j = 1 : size(known_references,1) %Jacobian has been calculated in
advance
        jacobian_matrix(j,:) = -2*(known_references(j,:) - temp_location(i,:)) ;
%partial derivative is i.e. -2(x1-x)
        f(j) = (known_references(j,1) - temp_location(i,1))^2 +
(known_references(j,2) - temp_location(i,2))^2 - distances(j)^2
    end

    estimated_error = -inv(jacobian_matrix' * jacobian_matrix) *
(jacobian_matrix') * f' ; %update the U and E

    temp_location(i+1,:) = temp_location(i,:) + estimated_error' ;
%    current_point = [temp_location(i+1,1),temp_location(i+1,2)];
%    new_matrix = [ new_matrix; current_point];

plot(temp_location(i+1,1),temp_location(i+1,2),'rd','MarkerFaceColor','g','MarkerSi
```

```

ze',20) ; % plot

    text(temp_location(i+1,1), temp_location(i+1,2) , num2str(i));
    drawnow;
    i = i + 1;

end

final_x = temp_location(i,1) ;
disp(final_x);

final_y = temp_location(i,2) ;
disp(final_y);
title('LMS Algorithm')

```

#### CODE FOR CENTROID ALGORITHM

```

function center = Centroidtest(points)

mass = [];
if nargin==1
    %Array of points
    pts = points;
    %pts = [2 2;6 1;6 5;2 4];
elseif nargin==2
    % either POINTS+MASS or PX+PY
    var = varargin{1};
    if size(var, 2)>1
        % arguments are POINTS, and MASS
        pts = var;
        mass = varargin{2};
    else
        % arguments are PX and PY
        pts = [var varargin{2}];
    end
elseif nargin==3
    % arguments are PX, PY, and MASS
    pts = [varargin{1} varargin{2}];
    mass = varargin{3};
end

%% compute centroid

if isempty(mass)

```

```
% no weight
center = mean(pts);

else
% format mass to have sum equal to 1, and column format
mass = mass(:)/sum(mass(:));

% compute weighted centroid
center = sum(bsxfun(@times, pts, mass), 1);
% equivalent to:
% center = sum(pts .* mass(:, ones(1, size(pts, 2))));
end
```

## Appendix B: Code for Graphic Interface Excel Sheet

```
Private Sub CommandButton1_Click()
Refresh
Refresh2
Worksheets("DATA").Range("A:F").Copy Worksheets("Working with Data").Range("A:F")
Range("B1:B300").Copy Range("G1:G300")
Date_to_Day

Dim i As Integer
Dim j As Integer
Dim lngCount As Long

lngCount = Application.WorksheetFunction.CountA(Columns(1))
classCount = Application.WorksheetFunction.CountA(Worksheets("Class
List").Columns(1))
Range("H2:H300") = 0

For i = 2 To lngCount
    For j = 2 To classCount
        If CInt(Cells(i, 3).Value) >= CInt(Worksheets("Class List").Cells(j,
2).Value) And CInt(Cells(i, 3).Value) <= CInt(Worksheets("Class List").Cells(j,
3).Value) And CInt(Cells(i, 4).Value) = CInt(Worksheets("Class List").Cells(j,
4).Value) And StrComp(WeekdayName(Weekday(Cells(i, 7)), False), (Worksheets("Class
List").Cells(j, 5))) Then Worksheets("Class List").Cells(j, 1).Copy Cells(i, 8)
        Next j
    Next i

Next i

End Sub

Sub Refresh2()
'
' Refresh2 Macro
'
'
    Sheets("Graphs").Select
    Range("A4").Select
    ActiveSheet.PivotTables("PivotTable").Name = "PivotTable"
    Range("C4").Select
End Sub

Sub Date_to_Day()
'
' Date_to_Day Macro
'
```



```

'
Columns("G:G").Select
Selection.NumberFormat = "[$-F800]dddd, mmmm dd, yyyy"
Selection.NumberFormat = "dddd"
End Sub
Sub Refresh()
'
' Refresh Macro
'
'
'
ActiveWorkbook.RefreshAll
End Sub
Sub Refresh_Table()
'
' Refresh_Table Macro
'
'
'
Sheets("Graphs").Select
ActiveSheet.PivotTables("PivotTable1").PivotCache.Refresh
End Sub

Sub Collect_Data()
'
' Collect_Data Macro
'
'
'
Sheets("DATA").Select
Range("A:A,E:E,F:F").Select
Range("F1").Activate
Selection.Copy
Range("A:A,E:E,F:F,D:D").Select
Range("D1").Activate
Application.CutCopyMode = False
Selection.Copy
Sheets("TABLE ").Select
Columns("E:H").Select
ActiveSheet.Paste
Range("J10").Select
Columns("F:F").ColumnWidth = 8.57
Columns("G:G").ColumnWidth = 5.29
Columns("H:H").ColumnWidth = 8.29
Range("J1").Select
End Sub

```

```

Sub Sort_List()
'
' Sort_List Macro
'
'
'
Range("A2:F9").Select
ActiveWorkbook.Worksheets("DATA").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("DATA").Sort.SortFields.Add Key:=Range("A2:A9"), _
    SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("DATA").Sort
    .SetRange Range("A1:F9")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
End Sub
Sub Application_Users_List()
'
' Application_Users_List Macro
'
'
'
Sheets("DATA").Select
Columns("L:L").Select
Selection.Copy
Sheets("DISPLAY").Select
Columns("E:E").Select
ActiveSheet.Paste
Selection.ColumnWidth = 15.86
Range("E1").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "Application Users"
Range("E2").Select
ActiveWindow.ScrollRow = 2
Range("E2:E21").Select
ActiveWorkbook.Worksheets("DISPLAY").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("DISPLAY").Sort.SortFields.Add Key:=Range("E2:E9") _
    , SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("DISPLAY").Sort
    .SetRange Range("E1:E9")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With

```

```
ActiveWindow.ScrollRow = 1
Range("F10").Select
End Sub
```

### Appendix C: Raw Data Collection for Measuring RSSI in Room 233.

This data is based on multiple beacons in the room, labeled as B1 through B10.

B1	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	331.1665	-87	-87	-76
2	369.4578	-92	-91	-88
3	305.8996	-97	-91	-97
4	451.4818	-94	-85	-87
5	521.0388	-97	-87	-85
6	468.9696	-95	-89	-79
7	539.9733	-96	-93	-77
8	496.7497	-98	-90	-85
9	564.2685	-101	-91	-93
10	500.9942	-98	-87	-81
11	390.5375	-93	-85	-77
12	265.2213	-98	-86	-75
13	233.9447	-91	-85	-69
14	208.6013	-94	-81	-83
15	164.2416	-93	-87	-75
16	165.5832	-94	-82	-70
17	82.7916	-81	-83	-68

B2	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	297	-83	-82	-75
2	305.8996	-82	-80	-77
3	305.8996	-93	-86	-81
4	451.4818	-93	-90	-75
5	531.2368	-92	-87	-90
6	445.5	-96	-84	-75
7	519.75	-94	-87	-81
8	451.4818	-90	-84	-77
9	524.8566	-92	-89	-92
10	421.1637	-97	-90	-77
11	280.8973	-82	-74	-80
12	165.5833	-91	-84	-69
13	108.6537	-80	-73	-65
14	148.5	-86	-77	-63
15	74.25	-72	-70	-65
16	165.5833	-95	-96	-76
17	132.6106	-80	-77	-70

B3	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	331.1665	-84	-88	-73
2	305.8996	-93	-89	-77
3	369.4578	-91	-87	-78
4	496.7498	-91	-88	-81
5	579.5414	-101	-96	-87
6	468.9696	-97	-92	-91
7	540.002	-97	-94	-86
8	451.4818	-93	-92	-82
9	524.8863	-103	-98	-78
10	382.9976	-102	-87	-82
11	219.5713	-95	-90	-90
12	165.5833	-85	-92	-71
13	108.6537	-85	-77	-79
14	208.6013	-91	-81	-71
15	164.2416	-93	-91	-75
16	265.2213	-97	-93	-73
17	266.9105	-98	-88	-88

B4	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	284.6019	-83	-93	-75
2	254.7579	-96	-83	-97
3	328.3688	-98	-84	-83
4	449.8292	-90	-87	-79
5	532.4995	-96	-88	-80
6	418.9493	-96	-85	-81
7	489.1726	-97	-95	-84
8	399.2766	-99	-97	-88
9	472.4332	-96	-92	-89
10	330.2059	-102	-94	-83
11	167.2924	-96	-78	-76
12	120.357	-95	-85	-69
13	78.15449	-87	-75	-71
14	174.8785	-92	-84	-78
15	148.0332	-93	-91	-82
16	239.6045	-90	-88	-82
17	257.2542	-90	-88	-77

B10	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	146.5	-87	-82	-74
2	219.75	-96	-85	-72
3	73.25	-89	-72	-74
4	165.5833	-91	-88	-84
5	225.7113	-89	-79	-88
6	208.6013	-85	-83	-80
7	266.5829	-93	-80	-77
8	265.2213	-97	-83	-76
9	312.8806	-98	-92	-75
10	335.3719	-87	-76	-70
11	334.1442	-93	-81	-66
12	265.2213	-97	-83	-76
13	308.6594	-103	-85	-72
14	208.6013	-91	-83	-70
15	266.608	-95	-82	-83
16	165.5833	-100	-87	-68
17	225.7409	-92	-93	-76

B5	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	209.5	-98	-83	-81
2	136.25	-92	-81	-73
3	282.75	-94	-88	-93
4	319.3741	-99	-85	-76
5	389.3643	-98	-89	-83
6	256.7927	-91	-81	-79
7	305.7686	-95	-94	-79
8	201.5349	-96	-83	-80
9	261.0905	-100	-89	-83
10	90.3678	-90	-88	-80
11	85.69912	-88	-81	-69
12	201.5349	-102	-86	-79
13	256.0169	-91	-81	-81
14	256.7927	-95	-87	-83
15	305.7905	-87	-78	-90
16	319.3741	-98	-82	-88
17	389.3815	-94	-91	-77

B6	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	288.9001	-91	-86	-74
2	259.5507	-93	-89	-82
3	332.101	-95	-91	-84
4	241.6409	-88	-79	-75
5	257.7154	-92	-93	-82
6	177.6584	-91	-77	-89
7	148.8332	-86	-72	-72
8	124.3616	-85	-76	-70
9	77.81147	-82	-87	-62
10	167.2924	-99	-85	-68
11	330.2059	-93	-92	-76
12	404.1928	-96	-96	-92
13	471.475	-99	-88	-84
14	423.6372	-97	-90	-82
15	493.974	-93	-87	-85
16	454.1985	-94	-85	-85
17	536.9136	-97	-87	-79

B7	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	331.1665	-93	-87	-83
2	305.8996	-92	-85	-94
3	369.4578	-91	-85	-74
4	265.2213	-95	-93	-81
5	266.9105	-96	-93	-87
6	208.6013	-87	-88	-81
7	164.2416	-94	-84	-84
8	165.5833	-94	-99	-85
9	104.322	-90	-94	-75
10	214.6242	-92	-96	-78
11	378.0147	-95	-92	-86
12	451.4818	-99	-90	-88
13	518.9457	-94	-92	-98
14	468.9696	-92	-95	-96
15	540.0021	-95	-89	-84
16	496.7498	-95	-91	-81
17	579.5414	-95	-95	-94

B8	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	297	-94	-89	-85
2	305.8996	-92	-78	-78
3	305.8996	-92	-85	-83
4	165.5833	-86	-87	-72
5	132.6274	-93	-78	-66
6	148.5	-83	-73	-84
7	74.28	-89	-62	-77
8	165.5833	-87	-84	-75
9	104.322	-89	-70	-74
10	277.0475	-88	-80	-83
11	416.6376	-93	-78	-69
12	451.4818	-99	-85	-81
13	518.9457	-99	-82	-89
14	445.5	-88	-79	-83
15	519.75	-85	-69	-77
16	451.4818	-96	-81	-82
17	531.2368	-96	-82	-82



B9	Power	-12	-4	4
Point	Distance from iBeacon	RSSI	RSSI	RSSI
1	331.1665	-93	-91	-80
2	369.4578	-98	-92	-86
3	305.8996	-90	-93	-75
4	165.5833	-91	-86	-81
5	82.81853	-99	-86	-67
6	208.6013	-87	-82	-84
7	164.2552	-91	-74	-78
8	265.2213	-94	-90	-80
9	231.9646	-95	-83	-83
10	387.7777	-99	-88	-84
11	497.1953	-96	-86	-85
12	496.7498	-97	-90	-81
13	558.7747	-98	-91	-87
14	468.9696	-95	-95	-93
15	540.0021	-99	-91	-90
16	451.4818	-99	-98	-87
17	521.0388	-93	-92	-82

The pathloss for room 233, using 5 of the 10 Beacons is shown below.

Assumptions		B1		B10		B4		B9		B6		
Transmitted Power	-12	Beats	7FE6D	Pinkie	E6D55	Lemon	EF7471	Bigfoot	4F3E3	Sunny	BE2A4	
Receiver Sensitivity	-95	Lp	Distance	Lp	Distance	Lp	Distance	Lp	Distance	Lp	Distance	
Lo	40.045997	1	58.5436	8.411629	51.45942	3.7211	57.22743	7.228889	57.35762	7.338062	58.5436	8.411629
Lambda	0.125	2	59.49397	9.384228	54.98125	5.58165	56.26522	6.470849	56.42712	6.592588	59.49397	9.384228
Frequency	2400000000	3	57.85425	7.76985	45.43882	1.86055	58.46991	8.340567	58.56807	8.435365	57.85425	7.76985
Alpha	2	4	61.23548	11.46764	52.523	4.205815	61.20362	11.42566	55.80608	6.137678	52.523	4.205815
		5	62.48007	13.23439	55.21374	5.733067	62.66906	13.52549	56.36548	6.54597	46.50522	2.103591
		6	61.56496	11.911	54.52901	5.298473	60.5859	10.64131	53.13439	4.512523	54.52901	5.298473
	0.0254	7	62.79012	13.71532	56.65932	6.771206	61.93191	12.42498	51.59667	3.780363	52.45305	4.172082
		8	62.06542	12.61744	56.61484	6.73662	60.16815	10.14163	50.0364	3.158785	56.61484	6.73662
		9	63.17239	14.33242	58.05024	7.947167	61.62948	11.9998	45.96354	1.976411	55.45111	5.891901
		10	62.13933	12.72525	58.65321	8.518447	58.51837	8.38723	52.61219	4.249226	59.91433	9.849553
		11	59.97593	9.919653	58.62135	8.487262	52.61219	4.249226	58.51837	8.38723	62.07321	12.62876
		12	56.61484	6.736621	56.61484	6.73662	49.7521	3.057068	60.27444	10.2665	62.06542	12.61744
		13	55.52494	5.942195	57.93226	7.839949	46.00175	1.985124	61.61184	11.97546	63.08741	14.19288
		14	54.52901	5.298473	54.52901	5.298473	52.9974	4.441915	60.68255	10.76039	61.56557	11.91183
		15	52.45233	4.171737	56.66013	6.771843	51.54985	3.760042	62.01675	12.54694	62.79058	13.71605
		16	52.523	4.205813	52.523	4.205815	55.73257	6.085954	61.28759	11.53664	61.23548	11.46764
		17	46.5024	2.102907	55.21488	5.733819	56.34992	6.534256	62.74076	13.63761	62.48007	13.23439

The data below was collected from 207 while measuring RSSI from the Beacons. It includes the RSSI from each Beacon, how far away the iBeacon & Eddystone Scanner phone application said we were from the beacon, how far the actual distance was and the pathloss.

power -4									
	B1 Measured Application PathLoss					B2 Measured Application PathLoss			
	RSSI	Distance	Distance			RSSI	Distance	Distance	
1	-85	1.67241	5.49	44.51285213	1	-91	1.76942	5.79	45.00261566
2	-98	2.7668	5.77	48.88555236	2	-68	1.33985	2.21	42.58712063
3	-81	4.07858	4.32	52.25617673	3	-75	1.89515	2.35	45.59886882
4	-81	4.09158	3.61	52.28381796	4	-88	3.32967	5.98	50.49402088
5	-93	2.78594	7.07	48.9454322	5	-70	3.048	1.71	49.72629627
6	-76	1.83351	3.83	45.31166268	6	-73	3.25975	2.38	50.3096829
7	-78	2.96504	3.02	49.48660815	7	-73	4.71581	1.91	53.517123
8	-82	3.69455	8.76	51.39722799	8	-76	4.572	2.19	53.24812145
9	-78	4.62452	3.75	53.34733026	9	-68	4.76441	2.85	53.60617957
	B3 Measured Application Distance Error					B4 Measured Application Distance Error			
	RSSI	Distance	Distance	Error		RSSI	Distance	Distance	Error
1	-80	3.95038	3.45	51.9787745	1	-77	1.76942	3.05	45.00261566
2	-80	2.84183	3.38	49.11795891	2	-75	1.33985	1.92	42.58712063
3	-81	1.63988	3.25	44.34223841	3	-80	1.89515	2.91	45.59886882
4	-81	1.54099	3.99	43.80199343	4	-78	3.32967	2.64	50.49402088
5	-82	2.78603	3.54	48.94571279	5	-83	3.048	3.55	49.72629627
6	-84	3.91036	5.37	51.89033186	6	-79	3.25975	3.84	50.3096829
7	-86	4.46498	6.41	53.04238738	7	-75	4.71581	2.8	53.517123
8	-75	3.5223	2.52	50.98252388	8	-84	4.572	4.45	53.24812145
9	-72	2.64949	2.66	48.50924271	9	-84	4.76441	4.69	53.60617957

# Smart Room Attendance Monitoring and Location Tracking with iBeacon Technology

Andrew Vanner, Savannah Redetzke, Raymond Otieno, Jahangir Rahman, Julang Ying, and Kaveh Pahlavan  
Center for Wireless Information Network Studies (CWINS)  
Worcester Polytechnic Institute (WPI), Worcester, MA, 01609  
Email: {snredetzke, atvanner, rhotieno, jrahman, jying, kaveh} @wpi.edu

**Abstract**—Using the recently developed technology, iBeacon was developed by Apple Inc and uses Low Energy Bluetooth (BLE) which can be detected by phones that have the software Bluetooth 4.0 or above. In this project we utilized this technology and designed an attendance and location monitoring system at in-room events. We have developed a system to receive the necessary information from multiple iBeacons, derived the optimal placement, number and transmission power of each iBeacon, developed a phone application to send information such as user name, time, and location, to a server for data analysis and created an excel program to represent the data. In addition, we compared multiple location algorithms using Received Signal Strength indicator RSSI data to determine the most accurate localization for this system and created simulation results of Cramer-Rao low bound (CRLB) estimation of the location error model. We conclude our work by discussing the potential of further developing this system.

**Keywords**— *iBeacon; Attendance; in-room; localization; phone application*

## Introduction

When large groups of people meet for events, conferences, and college classes, one of the largest logistically challenging problems is both taking attendance of who is present and keeping track of where they are. Current solutions to this challenge are both error prone and time consuming, such as registration tables and sign in sheets. This project has developed a new solution that will enable coordinators of events to have an automated system to monitor the attendance and where people are inside a room.

In this paper, we utilized two main technologies. These technologies are: Bluetooth Low Energy (BLE) for iBeacon, Estimote Location Beacons and Location Algorithms. BLE is a wireless network connection which is used to exchange tiny and static radio signals within short distances. It has a low power consumption which makes ideal to be used in this project because it is easy to power and does not contribute to being an obtrusive addition to any designated space. BLE is also considered to be a crucial addition to the Internet of Things [6] BLE technology is contained within our second technology, iBeacon. iBeacon broadcasts unique signals that can be received by BLE enabled devices such as phones with Bluetooth 4.0 standard or above. iBeacon has been used in many location-based applications [8], such as intelligent in-room presence detection systems[1], indoor positioning systems[2][3], predicting interactions inside crowds based on iBeacon proximity detection [4][9], and the enabling of communication between companion applications and television [5]. In addition, the Beacon has been considered in systems for assisting people with disabilities to interact with an Internet of Things made possible through Beacons. [7]



Figure 1: Estimote Location Beacons using BLE for promimity broadcasting for smart devices

Their properties can be customized to alter power output and improve longevity. These Location Beacons, and corresponding phone application, can be used to determine RSSI values, the distance you are from an Estimote Beacon, and add information to each Estimote Beacon for it to

\*This project was sponsored by Worcester Polytechnic Institute, Electrical and Computer Engineering Department as a Major Qualifying Project (MQP) for senior undergraduate studies.

transmit to a mobile device. However, though the Estimote phone application was useful in determining RSSI values for location algorithm analysis, we used Android Studio to develop our own unique phone application so we could customize our system. Using our own phone application allowed us to extrapolate information about different location algorithms as well as run simulations to determine which algorithms would be the most accurate. The simulations included Cramer-Rao lower bound and how other algorithms compare to it based on distance measurement error. In addition to working with different location algorithms, we maximized the accuracy of the location algorithm by creating path loss models for different numbers of Beacons in various points of a large lecture hall at different power outputs. Using this we were able to optimize power output as well as coverage for the entire room. We then used this design and tested various location algorithms, before settling on the Least Mean Square algorithm.

This algorithm is then run by the phone application to determine location in a room. The user is then able to input their personal information, which gets sent to an online server. At this point, a computer program can read this data and organize it for an administrator. This completes the attendance monitoring process.

The rest of our paper is organized into three major sections. In section II we summarize our system architecture and how the pieces of our technology work together. In section III we describe our performance analysis, including path loss modeling, location algorithm accuracy and Beacon placement. In section IV we discuss the results, conclusion and the potential of this project.

## System Architecture

The design architecture is broken down into 4 basic components as shown below in Figure 2. Number 1 represents the multiple Beacons required for both determining location and identifying when a person enters a room. Number 2 represents the smartphone that using the application that this project designed to detect the Beacons, utilize a location algorithm to determine where in the room it is, and submit identifying information to Number 3, the server, at regular intervals. The server stores this information to be accessed by Number 4, the computer graphic interface.

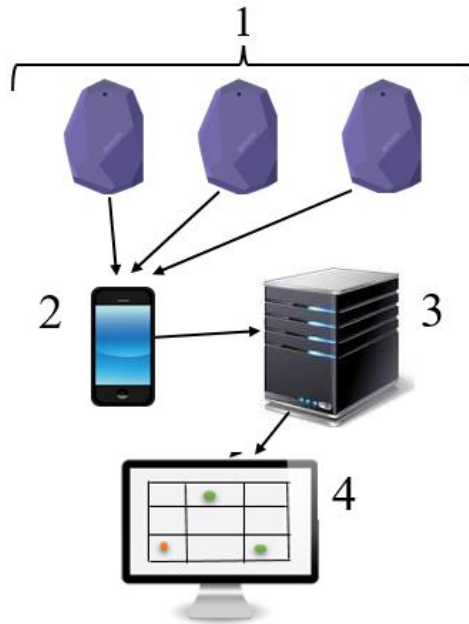


Figure 2: Project General Architecture Overview to demonstrate the general block diagram of the project and the lines of communication between different technologies included in the project

For our proof on concept, we chose a lecture hall that was 15 by 7.5 meters large. It seats approximately 50 students. We determined that 5 Beacons would be sufficient to cover the entire area. When a phone comes into range of at least one of the Beacons located in the room, our phone application downloaded on to a smart phone detects the signal that the Beacon is producing. It then provides a prompt to the user of the smart phone to to inset their name and some kind of ID or password. They also have the option to allow themselves to be monitored. The application can determine which room the user is in based on the unique identification numbers of the Beacons, demonstrated in Figure 3.

```

beaconManager.checkForBeacons {
    while (MonitoringAllowed){
        if (UUID == CertainNumber)
            display(MessageForRoom)
        }
    }
}

```

Figure 3: Application Code. Using a beaconManager class, the application checks the UUID of a Beacon to determine what region the user is in. It will then display a message for the room.

The phone application also runs the location algorithm for the room based on the Least Mean Square It returns coordinates describing where in the room a person is located based on a predetermined coordinate system. This application sends the user a message when their location has been determined. This prompt screen is shown in Figure 4.

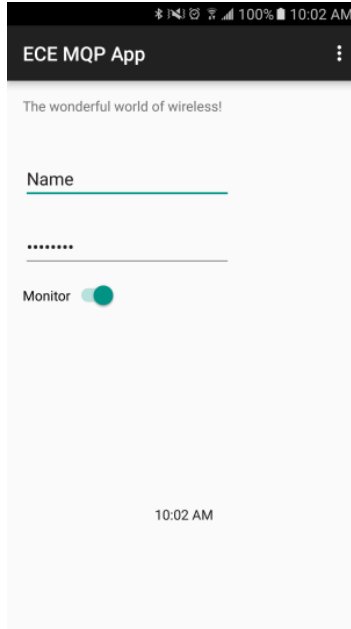


Figure 4: Phone Application User Information Input Screen which includes name, password and a switch to determine if user wants to be monitored or not.

Once the username, password and the location information have been determined, the phone application then sends the username, date, time, password, location and room number to an offsite server. For our proof of concept, we have utilized the Dropbox server system and set up a text file that allows the phone application to update it periodically. From this text file, we utilized an excel program to import the data into a excel list. From this list we extract meaningful information, such as what class they are in based on a database of classes in the same excel file, the time, date and room number. The excel file is also able to track how many users went to a class on a particular day by using pivot tables which are interactive so you can narrow down the information you are looking for. The pivot table is shown in Figure 5.

User	snredetzke				
Attendance	Class				
Date	ME1800	ECE2310	CS2102	Grand Total	
1/12/2016			1	1	
1/13/2016	1	1		2	
1/14/2016		1		1	
3/14/2017			1	1	
<b>Grand Total</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>5</b>	

Figure 5: Excell Pivot Table in Graphic User Interface which shows the username *snredetzke* and the date of their attendance as well as what classes they went to and how many times

Once a pivot table has been created, the excel program can be used to create graphs of the information, as shown in Figure 6.

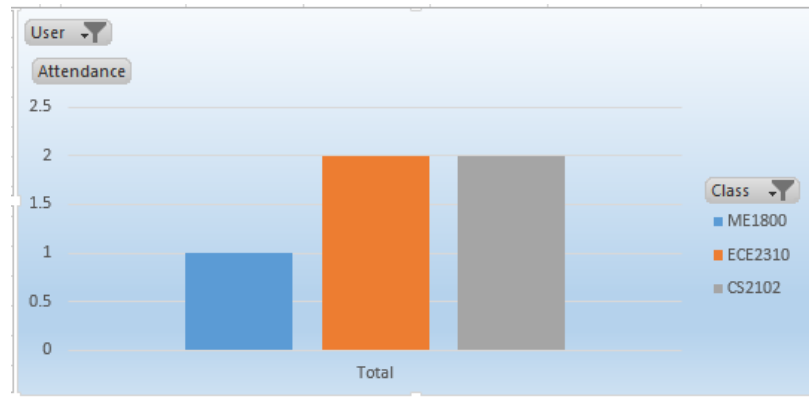


Figure 6: Pivot Table Graph as shown in Graphic User Interface which displays on a bar graph the number of times a user has attended each of the different classes in the database

## Performance Analysis

In this section of our report, we display our system of data collection and analysis. We display our RSSI and path-loss model for the main lecture hall we used for our proof of concept. We then compare the different location algorithms, that could be used and why we chose the least mean square algorithm.

### RSSI and Path Loss Model

Using the Beacons produced by Estimote, we were able to collect RSSI values at different power levels to determine the most effective and efficient way to place the Beacons in a room to ensure complete coverage. This involved multiple configurations in which Beacons should be placed around the walls while taking RSSI measurements from multiple points covering the entire room and comparing the values to a calculated pathloss. The ideal configuration would have the overall strongest RSSI values in every location in the room with minimal weak points. From these measurements of the multiple iBeacons, we determined that 5 Beacons arranged in the following configuration would be sufficient in providing coverage to the entire room, as shown in Figure 7.



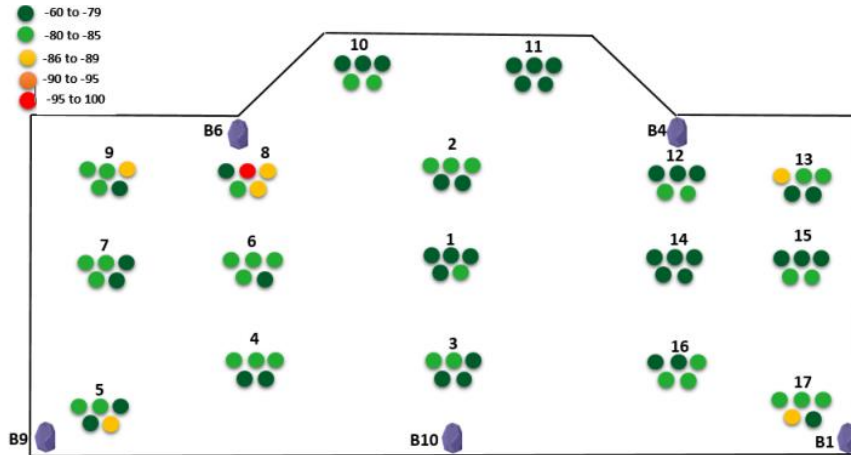


Figure 7: RSSI Room Model. In this model, every point in this room has a secure wireless connection to at least four Beacons at a time.

To test our determination, we placed the Beacons in their respective locations and calculated the path-loss model and the RSSI measurements at 17 points. The path-loss model is shown below in Figure 15 and the RSSI measurements are shown in Figure 16. In determining the path loss model, we used the IEEE model path loss equation as shown below.

$$L_o = 10\log(G_t) + 10\log(G_r) + 20\log\left(\frac{\lambda}{4\pi}\right) \quad [1]$$

$$L_p = L_o + 10\alpha_1 \log(dPB) + 10\alpha_2 \log(d/dPB) \quad [2]$$

dPB- reference distance

d- measured distance

The assumptions made in equation 1 were that the gain (G) is 0, the room is a typical office environment and alpha  $\alpha = 2.5$ . We also assumed, based on our phone's sensitivity to RSSI, that an acceptable path loss was 85. This was calculated using equation 2. As shown in Diagram 8, the path loss never rises above 63.09.

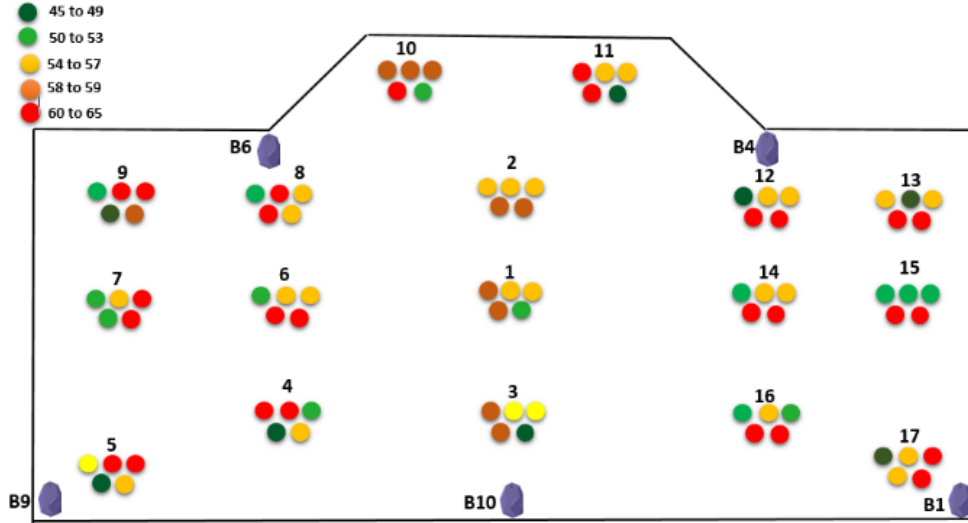


Figure 8: Manual path-loss for different locations throughout the room in a gradient color scheme

## Location Algorithm Selection and Analysis

Using the data we collected from the RSSI, distance, and power, and the path-loss. We were able to calculate the variance, and Distance Measured Error (DME). This information was used to develop graphs that compare the results to a set model for accuracy. The model used as mentioned was Cramer Rao Lower Bound (CRLB). CRLB is the lowest value of the estimator. In this paper it's used to compare the theoretical least distance measurement error while applying RSS values from each beacon as shown in Equation 2.

It used to determine how far off the android app measurements are by comparing it to the Distance measured error (DME). The closer an algorithm graph is to CRLB, the better the accuracy.

$$\sigma_p^2 = \sqrt{\sigma_x^2 + \sigma_y^2} \quad [3]$$

$$CRLB = \sqrt{\sigma_p^2} = \sqrt[4]{\sigma_x^2 + \sigma_y^2} \quad [4]$$

Cramer Rao, equation 4 is similar to the square root of equation 3 which represents the variance of points with x and y as coordinates.

We used this process to test three algorithms: trilateration, Least Means Squared (LMS) and the Centroid Algorithm. Different algorithms have different errors and this determines their reliability based on accuracy in locating unknown position. Figure 9 shows how the three chosen algorithms rank based on distance measurement error. LMS tend to have a better accuracy as compared to the other two. It stands out with a location error of about 0.69%. Centroid proves to be the least reliable of the three.

This is because it highly relies on the location of access points which are always fixed in this setup environment hence it will just give the center as location every time it is used.

Cumulative probability, CDF is used to make sure a randomly chosen value falls within specified range

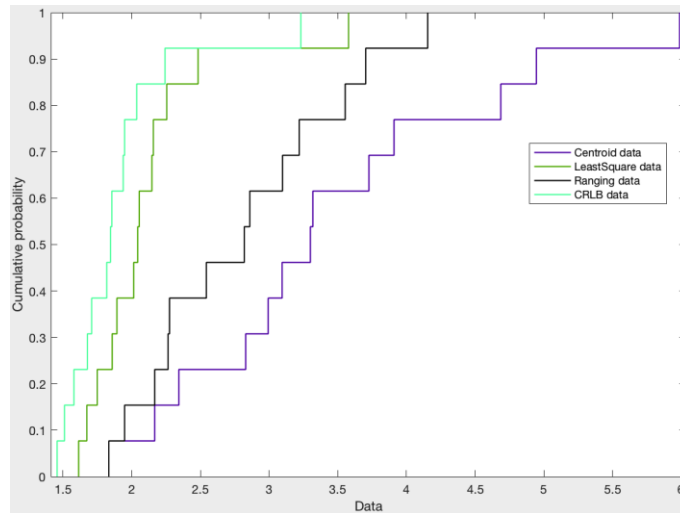


Figure 9: Comparison of LMS, Centroid Algorithm, and Estimate/Ranging using Cummulative Probability of Distance Error

$$\left(x_n - \frac{1}{n} \sum_{i=1}^n x_i\right)x_0 + \left(y_n - \frac{1}{n} \sum_{i=1}^n y_i\right)y_0 =$$

$$\frac{1}{2}(x_n^2 + y_n^2 - d_n^2 - \frac{1}{n} \sum_{i=1}^n (x_i^2 + y_i^2 - d_i^2)) \quad [5]$$

Least square algorithm in its linear format used in estimating the location of unknown receiver.

$d$  – distance

$n$  – number of access points

$x_i, y_i$  - coordinates for  $i^{\text{th}}$  access point

$x_0, y_0$  – coordinates for reference point

$i = 1, 2, 3 \dots\dots$

Least square algorithm in Equation 5 is the closest to CRLB model. As discussed on previous page, its location error is 0.69%.

# CONCLUSIONS

Upon completing our proof of concept system, we have concluded that this system could be improved upon and be implemented for events and classrooms. One important feature of the system is that every room is modular. That is, multiple rooms can have Estimote Beacons installed to provide locations for different rooms across different buildings. This has huge potential for college campuses. Professors and administrators can check who is attending their classes and how often students attend on average. In addition, events with Estimote Beacons can have people check in with their devices and you can easily see when people arrived, how long they stayed, and where in the room they are most likely to be in. Future developments for this project include testing it in larger rooms and taking our excel program data to be accessed via an online system for ease of use. The location algorithm could be manipulated to add a third dimension to the location tracking, allowing it to be a three dimensional tracking system. This design can be used to impact the way we take attendance.

## REFERENCES

- [1]. [1] Faheem Zafari, Ioannis Papapanagiotou "Enhancing iBeacon Based Micro-Location with Particle Filtering" *Global Communications Conference* , 2015 IEEE 2016
- [2]. Alexandre Alapetite, John Paulin Hansen "Dynamic bluetooth beacons for people with disabilities" *Internet of Things (WF- IoT), 2016 IEEE 3<sup>rd</sup> World Forum*, 2017
- [3]. Naoki Honma, Kazuki Ishii and Yoshitaka Tsunekawa "DOD-based localization technique using RSSI of indoor beacons" *Antennas and Propagation (ISAP), 2015 International Symposium* , 2016
- [4]. A. Coma, L. Fontana, A. A. Nacci "Occupancy detection via iBeacon on Android devices for smart building management" *Design, Automation and Test in Europe Conference & Exhibition (DATE), 2015* , 2015
- [5]. Zhouchi Li, Yang Yang, Kaveh Pahlavan, "Using iBeacon for newborns localization in hospitals", *Medical Information and Communication Technology (ISMICT) 2016 10th International Symposium on*, pp. 1-5, 2016, ISSN 2326-8301
- [6]. Zhiheng Zhao, Ji Fang, George Q Huang, Mengdi Zhang, "iBeacon enabled indoor positioning for warehouse management" *Computational and Business Intelligence (ISCBI), 2016 4<sup>th</sup> International Symposium on*, 2016
- [7]. Yingying Guo, Yatao Li, Yan Sun, "Accurate Indoor Localization based on Crowd Sensing," *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, ISSN 2376-6506
- [8]. Louay Bassbouss, Gorkem Guclu, Stephan Steglich, "Towards a remote launch mechanism of TV companion applications using iBeacon," *Consumer Electronics (GCCE) 2014 IEEE 3<sup>rd</sup> Global Conference on*, 2014
- [9]. Yang Yang, Zhouchi Li, Kaveh Pahlavan, "Using iBeacon for intelligent in-room presence detection", *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA) 2016 IEEE International Multi-Disciplinary Conference on*, pp. 187-191, 2016, ISSN 2379-1675.
- [10]. Miller. Steven, The Method of Least Squares