Media Engineering and Technology Faculty
German University in Cairo
Embedded Interactive Systems Laboratory
Universität Passau

Indoor Localization Using Bluetooth Low Energy Beacons and WIFI Fingerprinting

Bachelor Thesis

Author:	Omar Essam
Supervisors:	Prof. Matthias Kranz
	Dr. Amr Elmogy
	Dr. Gerold Hőlzl
Submission Date:	31 August, 2016

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

Omar Essam
31 August, 2016

Acknowledgments

I would like to thank my supervisor Dr. Gerold Hőlzl,for his unfailing support and advice throughout my period here, always giving me advice, yet letting me depend on myself He always let me know that my voice and point of view were heard and valid. I wouldn't be able to achieve so much without his advice and support pushing me beyond my limits.

My experience here was far beyond great. I learned so much, and was given the best tools to work with, and all of this is thanks to a chance bestowed upon me by Professor Matthias Kranz, so I would like to thank him for this amazing opportunity.

I would also like to thank Professor Slim Abdennadher, for inspiring me to join the Media Engineering and technology department in the German University in Cairo, and for showing' me throughout the years the beauty of Computer Science, and giving me advise to do my bachelor thesis abroad.

Finally I couldn't be more thankful to my parents, who never stopped believing in me and supporting me through all the years and obstacles of my education especially living away to finish this thesis. They always taught me to shoot beyond being mediocre but to strive to be the best! I could never have accomplished any of this without them.

Abstract

This paper presents a self learning WIFI fingerprinting indoor localization system that uses an easy to set up and deploy RSSI indoor localization system based on bluetooth low energy(BLE) beacons as its teacher, showing that system's capabilities and limitations, and also comparing it with other indoor localization technologies, showing the aspects this system is better at than current technologies.

The experiment used to test the localization system shows that the system has a lot of potential as it competes with current technologies and even has better accuracy, 0.5m - 0.77m, and better stability, even though it has some limitations that limits it from showing its full potential

Contents

1. Introduction

1.1. Motivation

A good indoor localization system is a crucial step for the evolution of our smart services, as a good indoor localization system could provide useful information to improve existing smart products such as let a NEST's thermostat[18] know when there is no one in the room to lower the temperature to save energy or even let it know how many people are in a room so that it adjusts the temperature accordingly, or let a Samsung's smart fridge[20] send you a list of your food's shopping list when you are leaving your home. It can also introduce new services such as a smart evacuation system, where the evacuation system guides each person to the closest exit, preventing any clustering of people at a specific exit, or a museum navigation system to easily find that piece of art you desperately want to see or even a search functionality for all your belongings, that whenever you want to find one of them such as your keys you simply search for it using your mobile device and it starts navigating you to them.

But what's a **good** indoor localization system? A good indoor localization system must have specific properties:-

- **Easy setup process:** The system must have an easy setup process that does not take a long time, nor require an unintuitive, or complex setup process
- **High Accuracy:** The system must be within a reasonable accuracy (1 meter for example)
- **High precision:** It also must give consistent readings and does not jump all over the place
- **Reasonable price:** The system should not be too expensive that's it's only accessible to multi-billionaire companies, but it should be within the price range of most people / companies

1.2. Goals

Unfortunately, none of the available indoor localization solutions satisfy all of the above properties of a good indoor localization system, so the goal of this paper is to implement and test an indoor localization system that is using Bluetooth Low Energy (BLE) beacons, and also use that system to teach, then test a WIFI fingerprinting system, showing both systems' capabilities and limitations.

2. Related Work

Localization is a well established research topic with many implementation approaches that include but are not limited to Global Positioning System (GPS) , fingerprinting based approaches, Angle of Arrival (AOA) based approaches, and Received Signal Strength Indicator (RSSI) based approaches.

2.1. GPS

GPS is the most used positioning system in the world with Google Maps having more than one billion monthly active users [6]. It was originally designed for military applications in the 1960s [16], but it was not introduced to the public until 1983, when a Korean Air Lines plane entered the Soviet airspace after a navigation error and was shot down, resulting in the death of all 269 passengers. Because of that tragic incident the president of the united states at that time, Ronald Reagan, ordered the United States military to make GPS available for the public use, so that this tragedy could be avoided in the future [19].

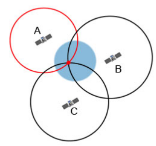

Figure 2.1.: Three satellites used for lateration

The idea behind GPS is rather simple. GPS Tracking works by using four satellites and a receiver, a mobile device for example. By knowing the positions of the four satellites and calculating the distance from each satellite to the receiver. tri-latereration [17] is used to get the position of the receiver using three of the satellites and the fourth satellite is used for confirmation. For example if there are four Satellites A, B, C, and D; A, B, and C are used to get the position of the receiver and then the fourth satellite, D, is used alongside A, and C to get a second position, and If the two positions do not match, another combination of the satellites is used. This process is repeated until a consistent result is obtained.

Even though GPS is highly reliable with an accuracy of 3.5 meters[7], has a total of 30 satellites at its disposal, and is widely supported by Google, it can not be used for indoor localization, due to the fact that its tri-lateration positioning technique requires a direct line of sight between the receiver and the satellites to calculate the distance between them.

2.2. Fingerprinting Based Approaches

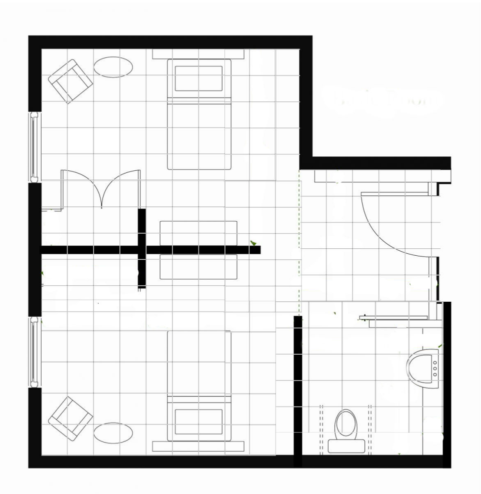

Figure 2.2.: Gathered RSSI signals of an estimote 1.5 m away from device for 10 minutes

Another approach that is used for indoor localization is fingerprinting. The idea behind that approach is very easy to Grasp: It has two stages: an offline stage, and an online stage. The offline stage is for training the system. A room is divided into a grid as shown in figure2.2, and then a person stands in each place of the grid and tells the system his/her position. The system then gathers the fingerprint such as the RSSI signals vector of all available access points, and saves them. Ecah place has more one RSSI vector for a higher chance of detection later.

When the system has gathered enough locations, the offline stage is finally over, and the on-line stage begins. In the online stage, whenever a person is standing in a place in the room, the system gathers the RSSI vector of all access points, just like the offline stage, but instead of saving the RSSI vector, it instead tries to match that vector with one of the saved RSSI vectors in the database. If a match is found then the position is detected and is shown on the map.

Although, as explained, a fingerprinting based approach is relatively simple to understand and implement, and it is also fairly accurate with a median accuracy of 0.6m[24]C, It is not user-friendly at all, as it takes a long time to deploy, as it requires to repeat the same process for each single cell of the grid, which takes an extremely long amount of time.

2.3. AOA Based Approaches

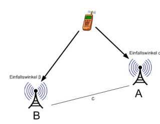

Figure 2.3.: Angle of Arrival Example

The next approach is AOA; this approach is fairly complex. It works by getting the angle of the direct path between the receiver and two access points as shown in the figure, and also knowing the distance between the two access points. Upon knowing these two pieces of information, the position of the device is calculated using triangulation[10].

In spite of this approach being really accurate with the best known implementation having a median accuracy of

0.4m[8][23][13], there are some major problems that emerge.
First, to get the two angles shown in the figure, a special kind of expensive and not that common access points needed, which have either six or more antennas, or 3 or more rotating antennas, as getting the angles requires measuring the phase shift of the incoming signals [14].

2.4. RSSI Based Approaches

This approach is also quite simple. this works similarly to GPS, in that there are three or more transmitters, access points , and a receiver, such as a mobile device. By knowing the position of each transmitter, and measuring the distance between each transmitter and the receiver, tri-lateration can be used again to get the position of the receiver quite easily.

To get the distance between the transmitter and the receiver. the transmitter sends an RSSI signal to the receiver, and according to the strength of the signal, the distance is calculated

This approach would work quite well, except that due to the interference of the environment, the signal sent from the transmitter fluctuates heavily, so it can only achieve a median accuracy of 2-4 meters[1][2][13].

2.5. Conclusion

It is obvious that there is no one good approach, as every approach has its limitation: GPS can not be used in an indoor environment, fingerprinting has a tedious set up process, AOA is complicated and requires special kind of hardware, and finally RSSI requires signal filtering due to environment interference. That is why, the best way is to use two approaches together, that one approach makes up for the limitations of the other.

3. Research Expirement

3.1. Approach used

The indoor localization experiment tested in this paper is going to be a mixture of two of the approaches mentioned in the previous section, which are RSSI and fingerprinting based approaches. The idea is that by setting up an RSSI based approach and leave it working for a period of time, it would work as the WIFI fingerprinting system's offline stage that whenever the RSSI system registers a person's location in a room, the system would also gather the WIFI fingerprints and saves them along with the location it registered. Once it gathered enough fingerprints for each location available in the room, a faster, more reliable fingerprinting system would be ready to be used and the beacons can be removed, and used somewhere else. This approach has many advantages: first it removes the offline stage for a fingerprinting system, by having an RSSI system working out of the box, and it also reduces the cost of an RSSI based approach as a person, would not need to buy the beacons and keep them forever, but he/she would only rent those beacons for a specific amount of time till the WIFI fingerprinting system has gathered enough fingerprints, and is ready to be used.

3.2. How to use the App

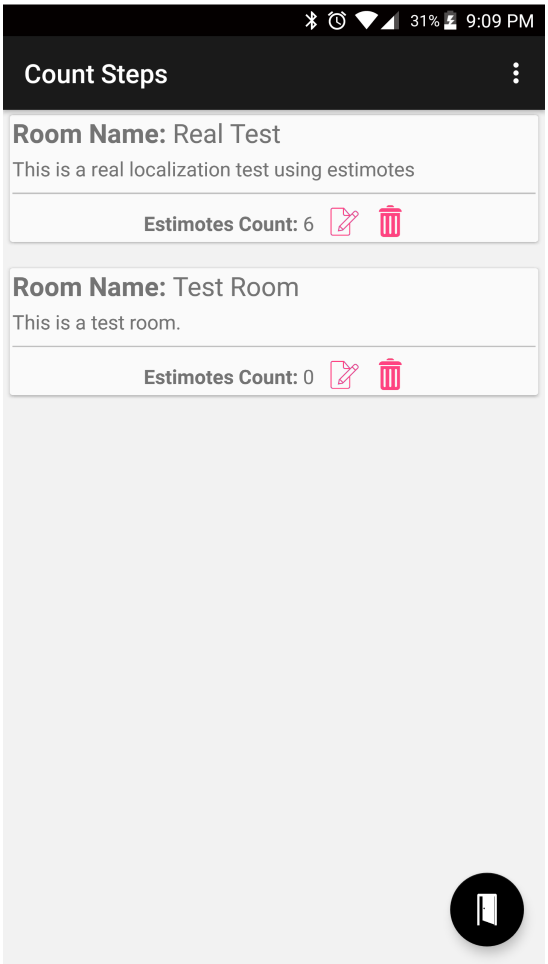

Figure 3.1.: Rooms List shown at start of the app

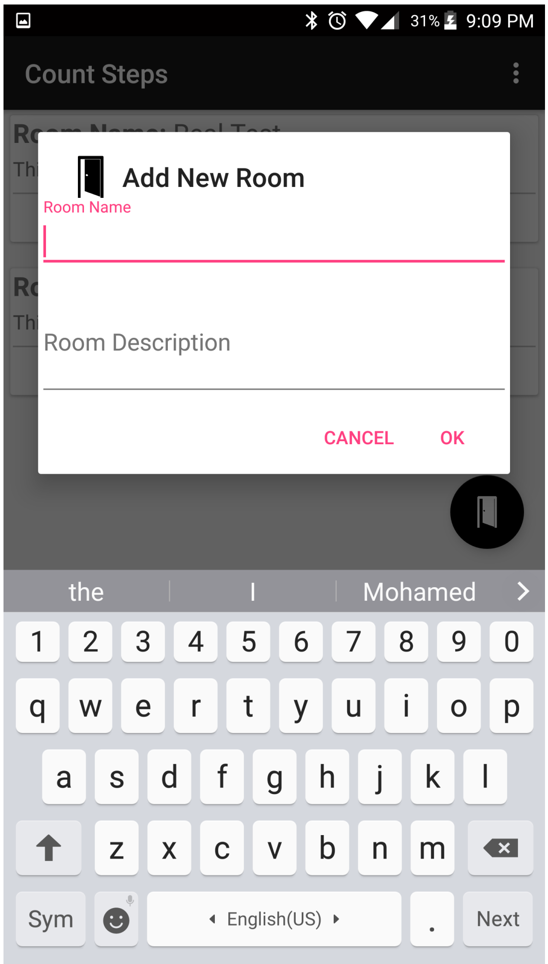

Figure 3.2.: Popup presented when creating a new room

When the user first opens the application, he/she would be presented with the view shown in figure 3.1 which includes a list of the rooms available, in addition to some options, which are editing a room, deleting a room, creating a new room, and of course selecting a room. To create a new room, the user should click on the floating action button at the right bottom corner of the view which will open a pop up as shown in figure 3.2 where the user would insert the name of the room, as well as, a description for it.

Figure 3.3.: Estimote Calibration

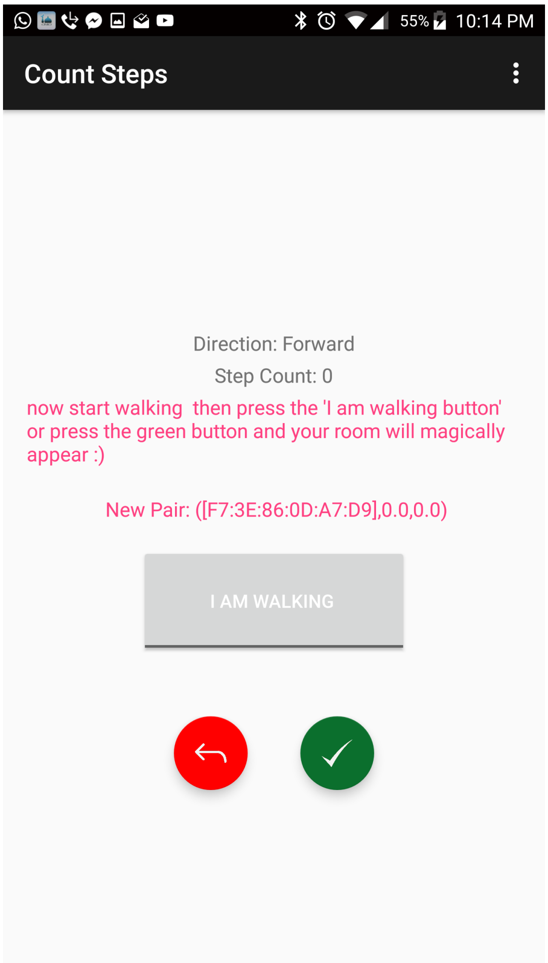

Figure 3.4.: start walking button

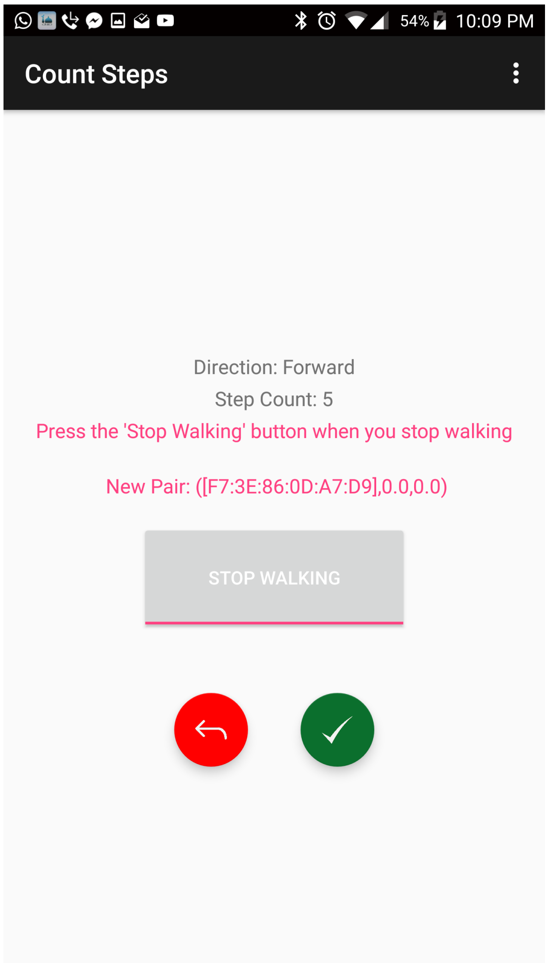

Figure 3.5.: Stop walking button

Figure 3.6.: The process of creating a new room

Once the user inserts the information and presses "ok", he/she will be directed to a step by step walk through to create the room. The user should first place a beacon, from Estimote inc [3], at the corner of each wall. Then he/she should stand in on of the corners of the room, and put the very close to the first estimote, and press the calibrate button shown in figure 3.3. Once the estimote is calibrated, the user should press the "I AM WALKING" button shown in figure 3.4, and start walking along the wall till he/she reaches the other end of the wall. when the user reaches the end of the wall, he/she should press the stop button shown in figure 3.5 and repeat the calibration process for the next button. Then the user should repeat the process for all the beacons, and then press the green button, and he/she would be redirected to the rooms' list again, with the new room added to the list.

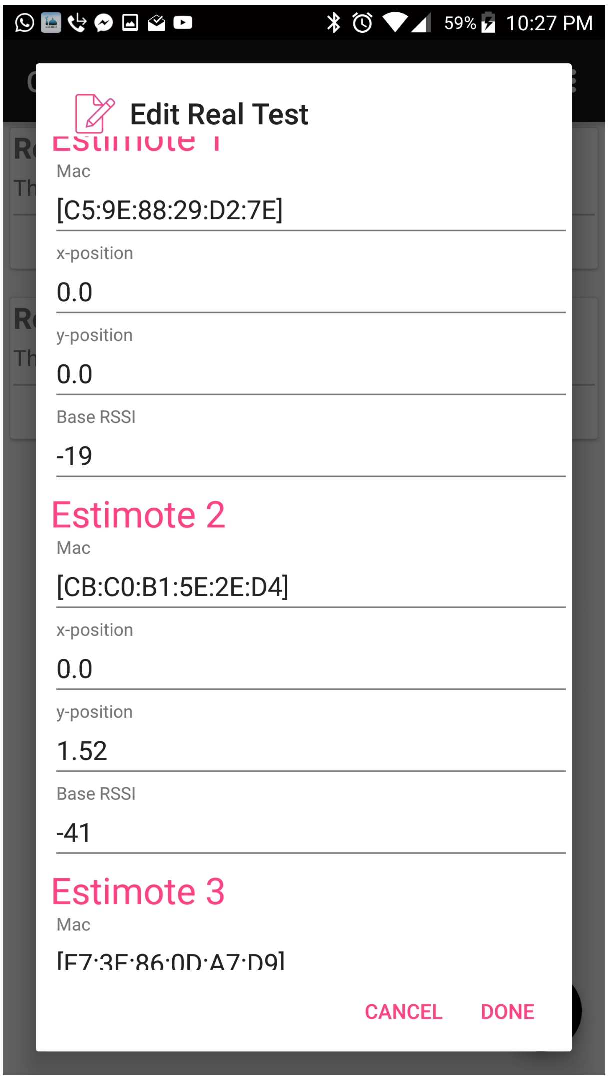

Figure 3.7.: Pop up for editting a room's estimotes information

The user can edit the room, if he/she is not satisfied by the room's dimensions and wishes to adjust them; or if one of the estimotes is broken and should be replaced by pressing the edit button next to a room's estimote count shown in figure 3.1. When the user presses the button another pop-up would appear shown in figure 3.7 which include each estimote's mac address for replacing an estimote, and each estimotes x and y position for adjusting the room's dimension. The application would work as follows, all a person would do is put an estimote at the edge of each wall, then he/she would stand in a room's corner, and walk around the room stopping at each beacon to register it once the person has waled around the whole room, he/she would press the done button and the room would simply be drawn.

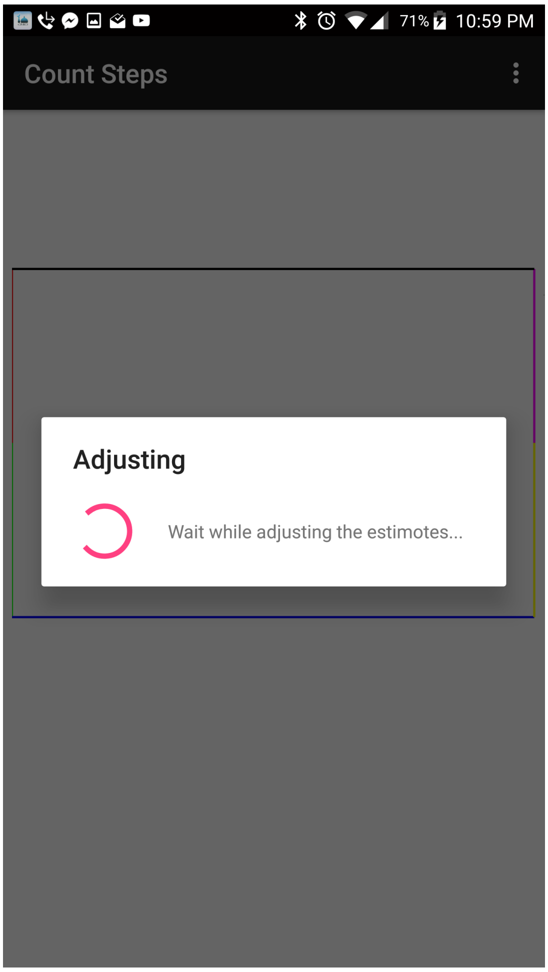

Figure 3.8.: Loader indicating the adjustment of estimotes

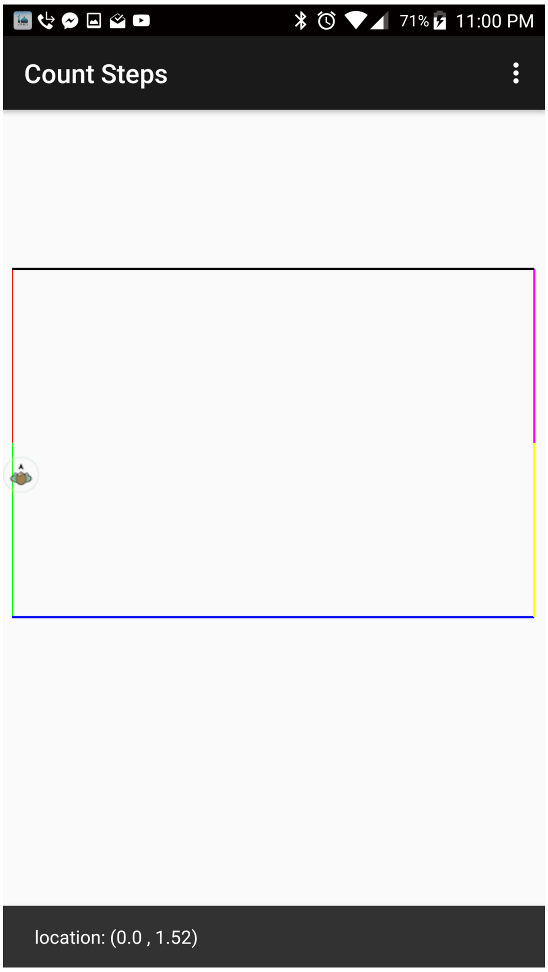

Figure 3.9.: A room drawing, with an avatar indicating position

To start using localization, the user should first select the room from the list. Then he/she would be redirected to a drawing of the room with a loader telling the user to wait for adjustment as shown in figure 3.8. While waiting the user should stand in a room corner and place the phone near an estimote. Then Once the loader is dismissed, an avatar would appear on the drawing, indicating the position of the user as shown in figure 3.9, and whenever the user takes a step the avatar would change its position, indicating the new location of the user

3.3. App Implementation

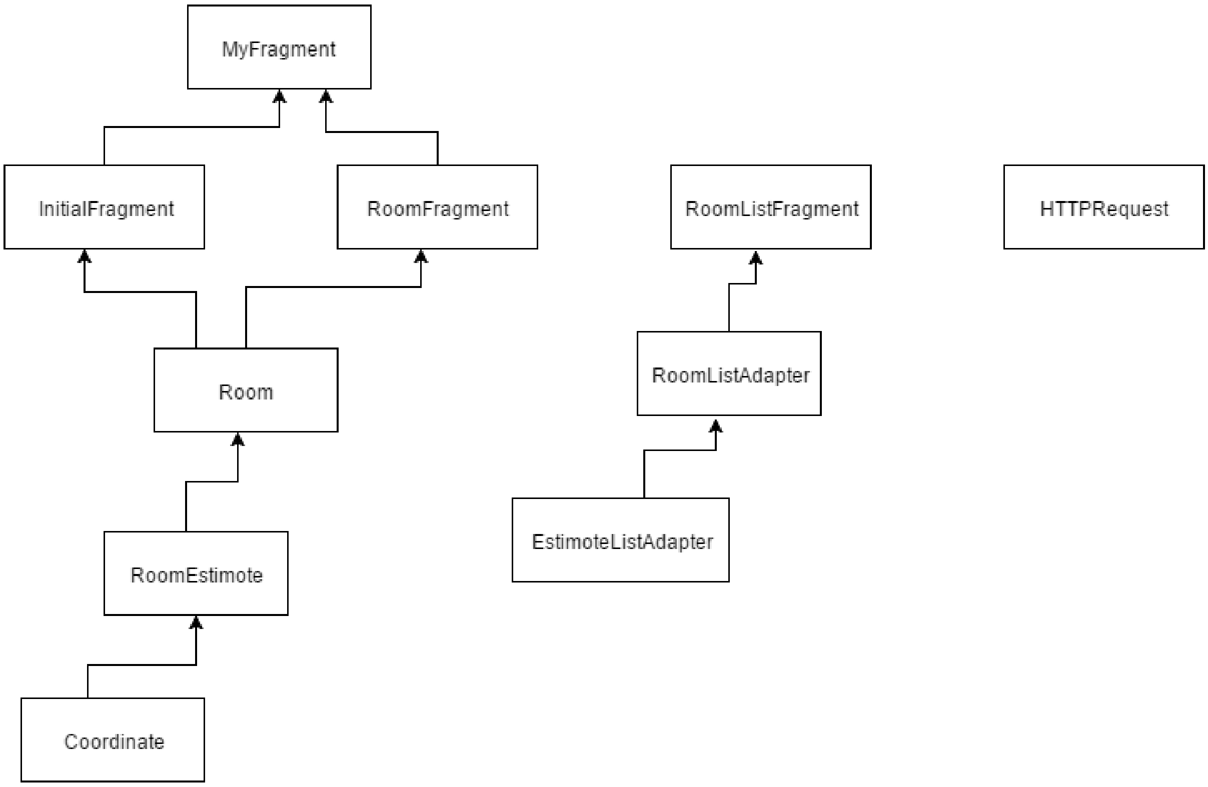

Figure 3.10.: A diagram showing the app classes hierarchy

The app is divided into three main fragments, and a separate class for all Hypertext Transfer Protocol Secure (HTTP) requests as shown in figure 3.10. The RoomListFragment is the fragment responsible for getting all the rooms from the database and organizing them in the list shown in the previous section, figure 3.1. The InitialFragment is the fragment responsible for guiding the user when he/she is creating a new room, and also sending the new room created to HTTPRequest class to save it in the database. Last but not least, the RoomFragment is the fragment responsible for handling localization and asking the HTTPRequest class for the selected room's info to draw it. All the app data are saved on a remote Rails[21] server and the HTTPRequest class is the class that handles all interactions between the app and the server.

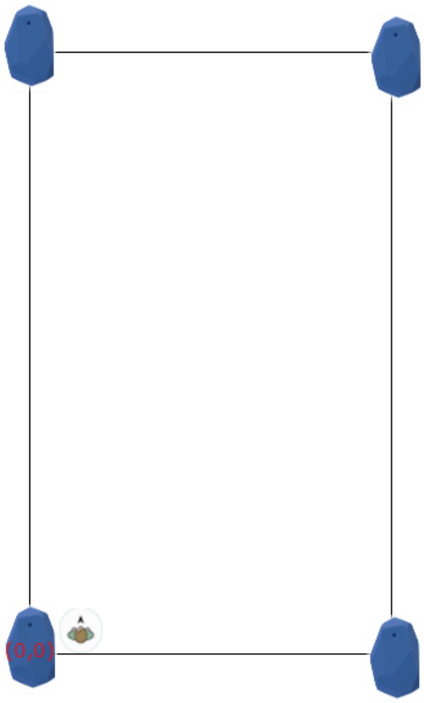

Figure 3.11.: First Estimote Position

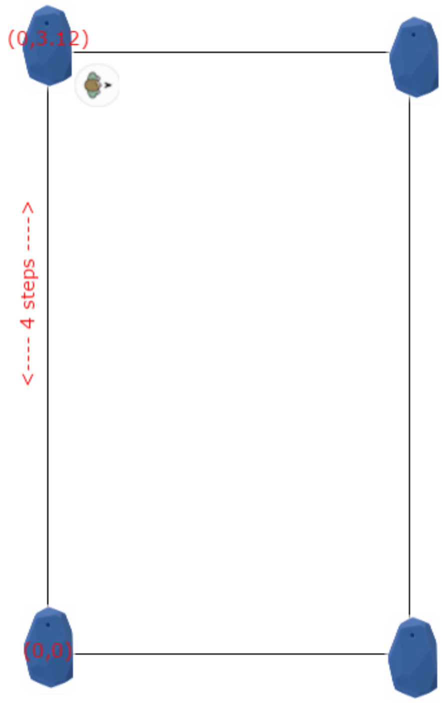

Figure 3.12.: Second Estimote Position

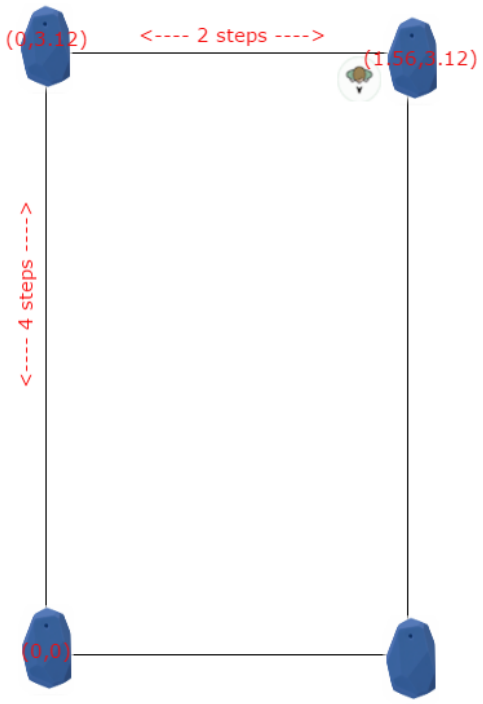

Figure 3.13.: Third Estimote Position

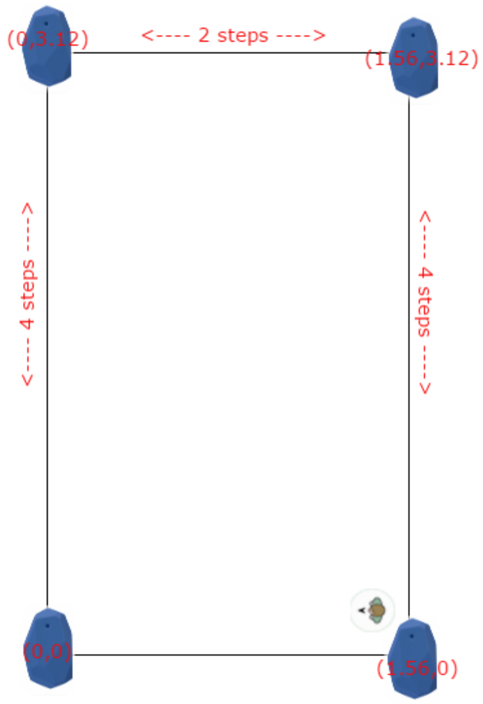

Figure 3.14.: Fourth Estimote Position

To draw a room, the app needs to know an x and y coordinate for each estimote, and then connects them together. The estimote at the corner a user starts in is given the position (0,0), and then using a pedometer and a walking direction detector, all the other estimotes' positions are calculated. This is done by using the pedometer to get the width of each wall by calculating the number of steps needed to walk from a wall's first edge to the wall's other edge, and then multiply this number of steps by a person's average step length (0.78 m)[21], and using The walking direction detector to know whether to add or subtract the width of a wall to the either the x or y coordinate of the previous position. For example, in figure 3.11, a person is standing in the bottom left corner of a room. so the estimote he is standing next to is given position (0,0). Then as shown in figure 3.12 he walked 4 steps to reach the next estimote, so by multiplying 4 by 0.78 and adding it to the y-axis of the previous position, (0,0), the app gets the position of the next estimote which is (0,3.12). Then in figure 3.13, when he turns 2 steps to the right towards the next estimote, it's given position (1.56,3.12), by adding 1.56 to the x-axis of the previous position. Finally, in figure 3.14, when he walks another 4 steps backwards towards the last estimote, it is given position (1.56,0) by subtracting 3.12m from the position of previous estimote, (1.56, 0).

The pedometer is implemented using the phone's gravity sensor [5]. Whenever the event.values[1], which is the acceleration minus the gravitational acceleration on the y-axis increases or decreases by 0.95 or more (the value of 0.95 was chosen based on trial and error); the number of steps increases by one. The direction detection ,on the other hand, is implemented using the phone's gyroscope [4]. The app sets the walking direction at first to be forward. Then it detects a user's turn by checking the change in the event.values[2], which is the z-axis. If a change of 1 or more is detected, then the user has turned left, and if a change of -1 or more is detected, then the user has turned right. Directions are calculated as follows: if a user is walking in forward direction then turns left, then the walking direction is now left, and if the user turns right then the walking direction is now right; if a user is walking in the left direction and turns left, then the direction is now backwards, and if the user turns right, the direction is now forwards; if a user is walking in the right direction, and then turns left, then the direction is now forward, and if the user turns right, the direction is now backwards; finally, if a user is walking in backwards direction, and then turns left, the direction is now right, and if the user turns right, the direction is now left. Each direction stands for a change in the previous coordinate as shown in figures 3.11 to 3.14: forward is positive change in the y-axis, backwards is negative change in y-axis, left is negative change in x-axis, and right is positive change in x-axis.

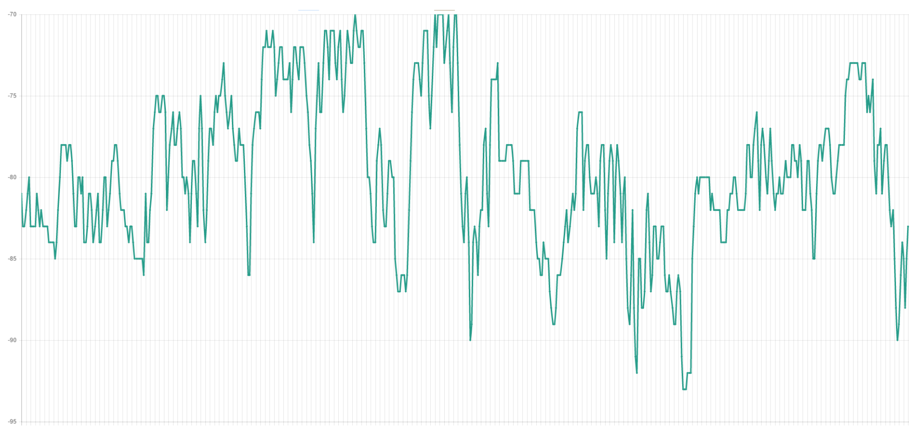

Figure 3.15.: Gathered RSSI signals of an estimote 1.5 m away from device for 10 minutes

One of the biggest problem encountered during the implementation was that the estimotes' RSSI signals fluctuated too much to get a distance estimation between the mobile device and the estimote as shown in figure 3.15, so filtering techniques had to be used to get a more stable RSSI signal.

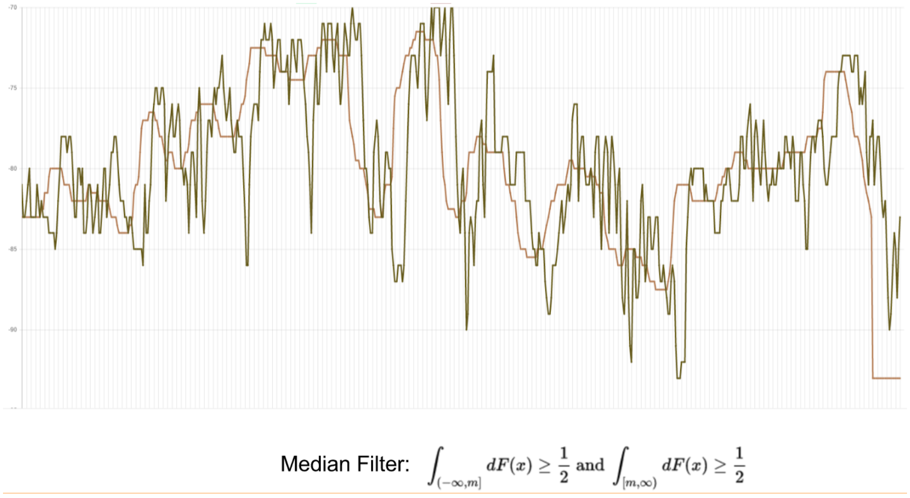

Figure 3.16.: Median filter for Gathered RSSI signals of an estimote 1.5 m away from device for 10 minutes

The first filtering technique tried was a very simple filter, which just calculates the median of every twenty values, and plots that instead. There was some improvement, shown in figure 3.16, but it still was not good enough to be used.

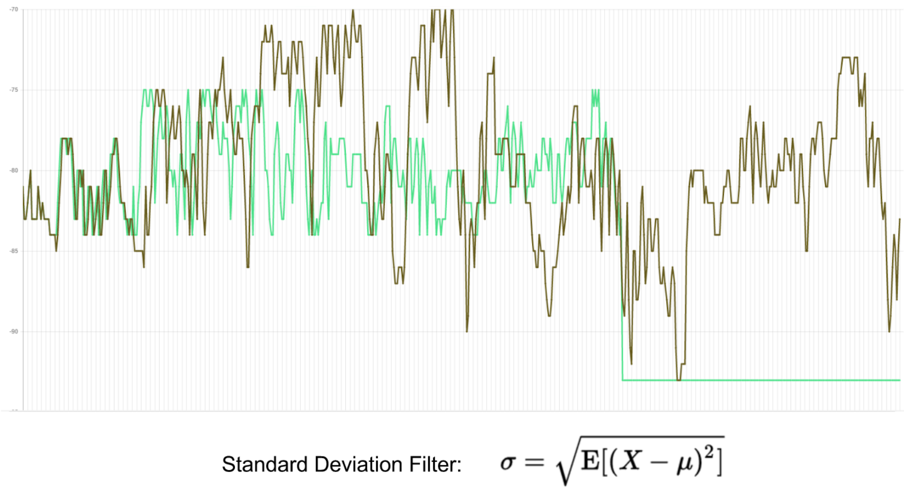

Figure 3.17.: Standard deviation filter for Gathered RSSI signals of an estimote 1.5 m away from device for 10 minutes

The next thing tried was a little bit more complicated filter which uses the standard deviation[15] and the average of all values, and then removes the values that the absolute value of the difference between them and the average is higher than the standard deviation. The results shown in figure 3.17 showed a big improvement, but it still was not satisfying.

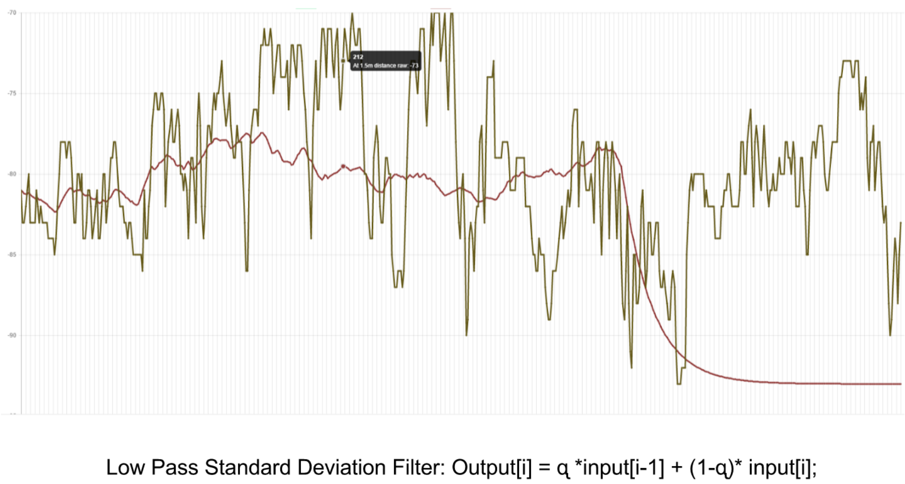

Figure 3.18.: Low pass standard deviation filter for Gathered RSSI signals of an estimote 1.5 m away from device for 10 minutes

A low pass filter is a filter that is used to smooth a curve by changing the current value to a percentage, α, of the previous value, added to 1 - α of the current value, so that sudden increases or decreases are smoothed; Therefor, by applying a low pass filter to the standard deviation filter, shown above, was a vast improvement as shown in figure 3.18, and the results were very satisfying. Other filters worth mentioning are Kalman filter [22], and particle filter [12], but they both are computationally expensive, and the Kalman filter is also very complicated and hard to understand.

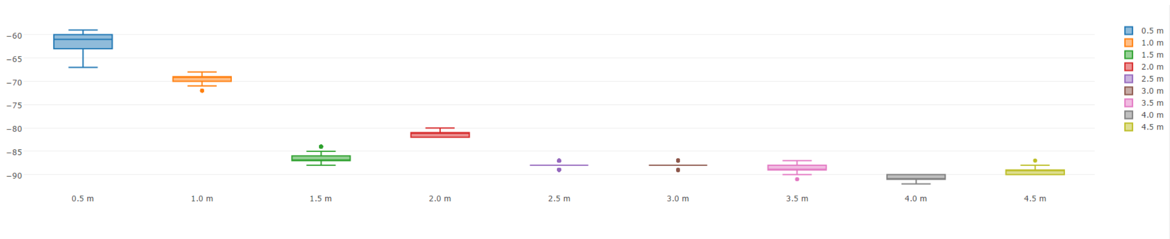

Figure 3.19.: Distance experiment done outside with minimal influence

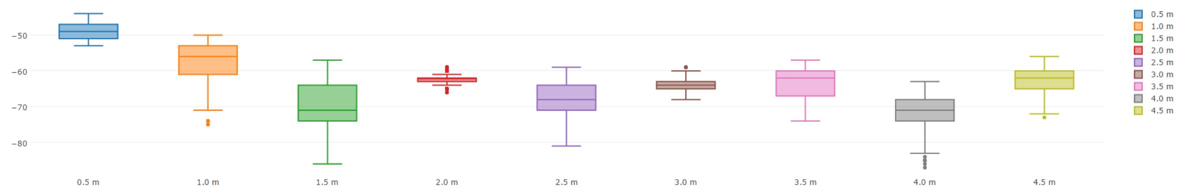

Figure 3.20.: Distance experiment done inside with normal influence

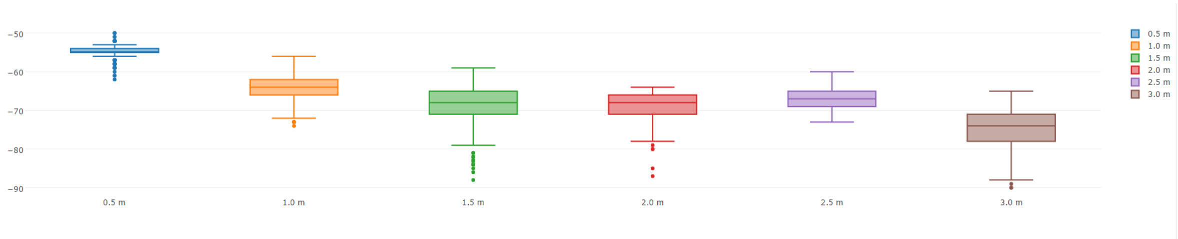

Figure 3.21.: Distance experiment done inside with high influence

To calculate the distance between an estimote and a phone, an experiment was made by placing the phone at distances of 0.5m to 4.5m with 0.5m intervals and gathering the RSSI values at each distance for 10 minutes, and then filtering them, using the Low Pass standard Deviation Filter mentioned above. This experiment was repeated for different environments:

- **Figure 3.19** in a park with almost no external influence at all
- **Figure 3.20** inside a room with some Access points around
- **Figure 3.21** inside a lab with very high ferromagnetic influence

The three experiments show that the distance between an estimote and a phone can be measured up until 1.5m with 0.5m accuracy, so the phone uses the info above and the base RSSI value calculated in the setup process to know how far is a phone from the estimote, by returning either 0.5m, 1.0m, 1.5m, or 3m, if the distance is too far to measure, based on the difference between the base RSSI and the current estimote RSSI value so from the above experiments the following formula was deduced

$$distance(estimote) = \begin{cases} 0.0, & \text{if } baserssi - rssi(estimote) \leq 5 \\ 0.5, & \text{if } baserssi - rssi(estimote) \leq 13 \\ 1.0, & \text{if } baserssi - rssi(estimote) \leq 25 \\ 1.5, & \text{if } baserssi - rssi(estimote) \leq 30 \\ 3.0, & \text{otherwise} \end{cases}$$

For this formula to work, the app needs to find the closest three estimotes to the user. It does that by setting a start position when the user selects a room to localize his/her position in it, and places the phone near one of the estimotes as mentioned in previous section; the start position is set to the same position of that estimote. Then whenever the app detects that the user has moved, using the pedometer, an approximate position is calculated based on the direction the user moved in, and the number of steps he/she walked, by adding the distance walked to the previous position. This approximate position is used to know the nearest three estimotes to the user, by calculating the distance between this approximate position and all the estimotes, and then the three estimotes with the shortest distances are the closest. Then by getting the RSSi values of each estimote of those three , and using the results from the above experiment the distance between the phone and each of the estimotes is calculated. After that, to get the user's position, these distances are used along with the position of each of the three estimotes, to tri-laterate the user's position. However if the tri-lateration results in a coordinate that is outside the room's boundaries due to fluctuations in RSSI signal, the approximate position is used instead.

The app also continuously gathers the RSSI values of all available access points in a room, and whenever a position is calculated using the estimotes. This position along with current RSSI values of all available access points are sent to the Rails server, as a WIFI fingerprint for that room, so that once enough fingerprints have been gathered the fingerprinting offline stage would be over, and the estimotes could be removed from the room, and WIFI fingerprinting could be activated by changing the value of the boolean "stillLearning" from true to false. Once this boolean value is changed the app would only gather available access points' RSSI values and check them with the server and the position corresponding to that RSSI vector would be returned to the app from the server and used as the user's current location.

Duration: in minutes
Start date: date and time
Environment: friendly, not friendly or outside
Distance: in meters
Hardware: Estimote or WIFI
rssis = [array of data]

Table 3.1.: Data set format

All data gathered can be found in this link here. The data are organized as shown in table 3.1

3.4. Limitations

Unfortunately there are some limitations, that prevent the app from running as intended. First of all as, the experiments shown in figures 3.19 to figure 3.21 show that only distances from 0.5m to 1.5m can be detected therefore if less than three estimotes are within that distance, the position is calculated based on the pedometer, and direction detector alone. Also because the direction detector is relative and not absolute, a user should start the localization process facing the same direction as he was when he was setting up the room, the relative forward direction.

Some other lesser limitations include the fact that the phone's WIFI manager scans every four seconds only, and the pedometer sometimes misses a step or add a step if the user is walking too fast. Also the step counter is limited to only four directions and it would be better if they were eight or even more. Also, when first selecting a room for localization, the user have to place the phone above an estimote till the adjusting process is over then he can move wherever he/she wants. Finally having a fixed step length of width 0.78m is not convenient as a person's step length differs according to a person's height.

3.5. Results and Accuracy Achieved

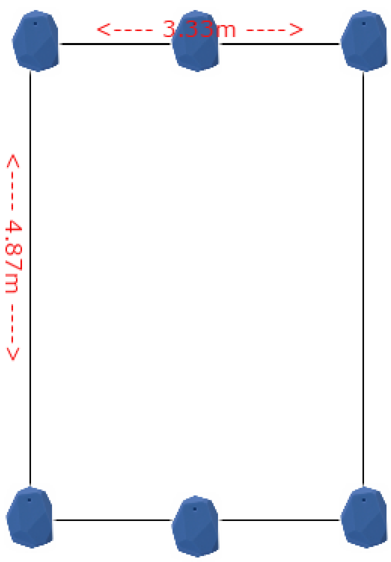

Figure 3.22.: The room used for the experiment

The experiment was done in a room with dimensions 4.87m x 3.33m, with 6 estimotes, one at the edge of each wall and the other each of them is placed in the center of the two opposing walls of length 3.33m as shown in figure 3.22. Drawing the room using the app resulted in a room of dimensions 4.56m x 3.04m, which is an accuracy of 92.75%. During the experiment , while still in the RSSI stage, when the distance between the user and three of the estimotes were 1.5m or less, the best accuracy achieved was 0.5m; however, when walking towards the middle of the room where the distance between the user and the estimotes were more than 1.5m, only the approximate location, calculated using the pedometer and direction detector, is used; therefore the accuracy was 0.77m, and only if the user is walking in one of the 4 detectable directions by the direction detector. In addition to the fact that the pedometer would sometimes skip and count a step without the user walking, and upon that happening the user had to go back to one of the esimotes and put the phone close to one of the estimotes to register its position and start over.

Unfortunately the fingerprinting was not tested thoroughly, but the test that was made was only within 1.5m of the beacons, so the fingerprints saved were from the locations found by the estimotes only, and not by the pedometer, so the accuracy was same as the estimotes at 0.5m; however, during using fingerprinting, the avatar was much more stable and there was absolutely no jumping at all, nor the need to go back to an estimote to register a position. It was working out of the box.

3.6. Comparison with Estimote's Own App

Estimote inc, the company that provided the beacons for the experiment, is one of the leading companies in the indoor localization business, so this section is a comparison between the software implemented in this paper and Estimote's own software.

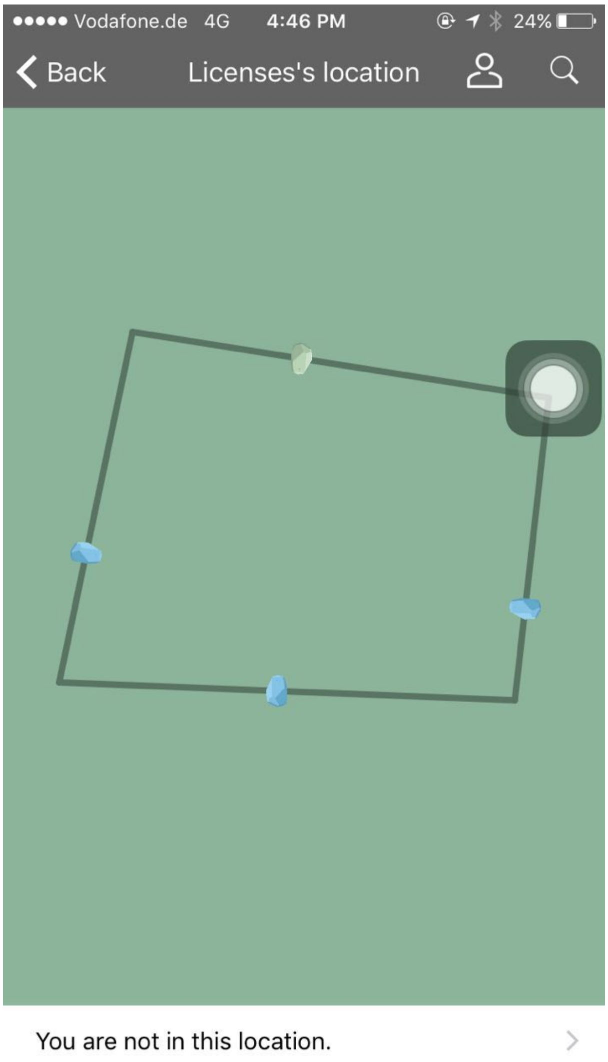

Figure 3.23.: Estimote's app saying that the user is not in the room even though the user is standing right in the middle of the room

First of all Estimote indoor localization api is ios only however the application presented is android which has a smart phone OS market share of 86% [9]. Also during the experiment done in previous section, the set up process took forty seconds, but during testing the estimotes app, the set up process took more than 3 minutes after failing to do the process twice due to taking a long time for the mobile to recognize the estimotes. Moreover, the application implemented always gives the user a reading but Estimote's app sometimes notify a user that he/she is not in the room, as shown in figure 3.23, event though he/she could be standing in the middle of the room. Also the app can behave abnormally and the avatar that is supposed to show the user his/her position would jump all over the map, but the implemented app showed much less jumping during the learning process and no jumping at all during the fingerprinting process. According to Estimote, the accuracy of their indoor localization in a small rectangular room is 1.5m on average [11]; however, the app implemented showed an accuracy of 0.77m as a worst case.

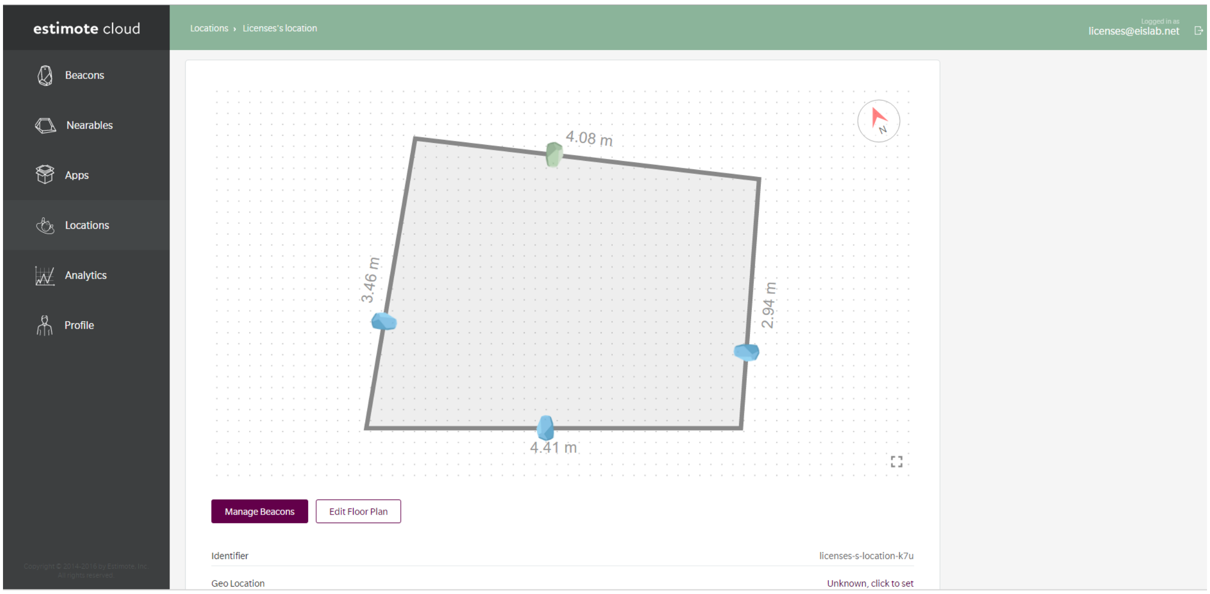

Figure 3.24.: Estimote Cloud Options

On the other hand Estimote's app does have some benefits over the app implemented. First of all it has a better UI interface, and a little bit more features, such as the ability to access the locations create from their website and also the ability to alter the estimotes transmission rate, as shown in figure 3.24. Also, as mentioned before, the app implemented has the limitation of only using the estimotes when the user is within a 1.5m distance from three estimotes or more, and if not it uses the pedometer, which is less accurate and limited to only four directions; however, Estimote's app does not have that limitation and is capable of using the estimotes for indoor localization inside the whole room.

Category	Estimote App	Implemented App
OS	ios	Android
Time to set up	3 minutes	40 seconds
Reliability	Notifies a person that he/she is not in the room sometimes	always gives a reading
Consistency	avatar jumps all over the place a lot	avatar stays in same place
Accuracy	1.5m	0.5-0.77m
Directions	all	4
UI	better	still good
Web Interface	Yes	No

Table 3.2.: Comparison between Estimote App and Implemented App

3.7. Experiment Conclusion

All in all, the implemented app's approach which is using an RSSI based system with beacons as an offline stage for a fingerprinting based approach system looks very promising, and even though it has some limitations, it was still able to compete with the current technologies available on the market such as Estimote indoor location, and come on top with better accuracy, 0.5m - 0.77m, and more stability, and it also achieved all 4 features of a good indoor localization system:

- A quick set up process of 40 seconds
- Accuracy of 0.5m - 0.77m
- High precision with almost no jumping at all
- Last but not least a relatively low price as the estimotes would not be bought but just rented for a period of time till the fingerprinting process is complete.

It still also has the capability to improve and be one of the best indoor localization solutions available on the market, with some modifications, and improvements.

4. Future Work

4.1. Direction Detection Re-implementation

The app can be improved in many ways, but the most important thing that needs improvement is the direction detections, as direction detection is relative for now and not absolute, meaning it needs to be given a default initial position and detects the rest of the directions according to that initial direction, which limits the user in that he/she needs to be standing in the same direction, forward, initially when they are starting the app. Also if the gyroscopes misses up once, the only way to restore the correct direction is by rebooting the app. That's why an absolute direction implementation would be better like using the compass for example.

4.2. Another Teacher

Another thing that would be worth trying is to try another approach, such as an AOA based approach, as a teacher for the fingerprinting based approach system instead of RSSI system to bypass the limitations of the RSSI system and see whether the fingerprinting system would be better than the one that is taught by the RSSI based system.

Appendix

A. Lists

BLE	Bluetooth Low Energy
GPS	Global Positioning System
RSSI	Received Signal Strength Indicator
AOA	Angle of Arrival
HTTP	Hypertext Transfer Protocol Secure

List of Figures

Bibliography

[1] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. *INFOCOM*, 2000.

[2] K. Chintalapudi, A. A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. *MobiSys*, 2005.

[3] Estimote Company. Estimote indoor location. http://estimote.com/indoor/, 2014.

[4] develope.android.com. Anroid gyroscope. https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GYROSCOPE, 2009.

[5] develope.android.com. Anroid gravity sensor. https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_GRAVITY, 2010.

[6] gpsbusinessnewsmote Company. Google maps reaches 1 billion active users. http://www.gpsbusinessnews.com/Google-Maps-1-Billion-Monthly-Users_a4964.html, 2014.

[7] gps.gov. Official u.s. government information about the global positioning system (gps) and related topics. http://www.gps.gov/systems/gps/performance/accuracy, 2014.

[8] R. Henniges. Current approches of wifi positioning. *TU-Berlin*, 2012.

[9] idc.com. Smartphone os market share. http://www.idc.com/prodserv/smartphone-os-market-share.jsp, 2015.

[10] imaginationplay.com. Triangulation method. http://www.imaginationplay.com.au/short_website_pics/how-to/TRIANGULATION%20METHOD%20EXPLAINED.pdf.

[11] Estimtote inc. Estimote indoor localization accuracy. http://blog.estimote.com/post/118294444205/super-simple-and-accurate-indoor-positioning-with, 2015.

[12] H. Kaijen, D. Henry, and M. Jason. Particle filters and their applications. *Cognitive Robotics*, 2005.

[13] M. kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi-research. *Stanford University*, 2015.

[14] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. *MobiCom*, 2014.

[15] mathisfun.com. Standard deviation explanation. http://www.mathsisfun.com/data/standard-deviation.html.

[16] mio. History of gps. http://www.mio.com/technology-history-of-gps.htm, 2014.

[17] mio. Tri-lalteration explanation. http://www.mio.com/technology-trilateration.htm, 2014.

[18] Nest. Nest's thermostat. https://nest.com/thermostat/meet-nest-thermostat/, 2011.

[19] Pingdom. Everything you ever wanted to know about gps. http://royal.pingdom.com/2010/03/23/everything-you-ever-wanted-to-know-about-gps, 2015.

[20] Smasung. Samsung smart refrigirator. http://www.samsung.com/us/explore/family-hub-refrigerator/, 2016.

[21] walkingwithattitude.com. A person's average step length. https://www.walkingwithattitude.com/articles/features/how-to-measure-stride-or-step-length-for-your-pedome, 2016.

[22] B. Wouter. Kalman filters explained: Removing noise from rssi signals. *https://wouterbulten.n*, 2015.

[23] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. *NSDI*, 2013.

[24] M. Youssef and A. Agrawala. The horus wlan location determination system. *MobiSys*, 2005.