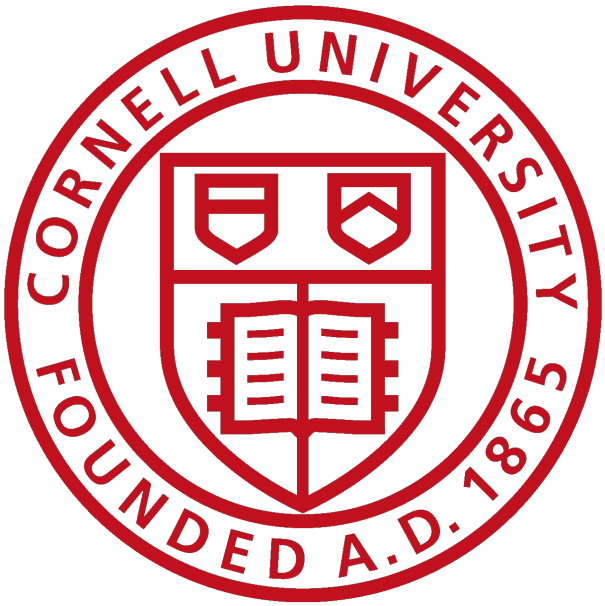
SMARTPHONE-BASED BUILDING OCCUPANCY TRACKING

A Design Project Report

Presented to the School of Electrical and Computer Engineering of

Cornell University in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering, Electrical and Computer Engineering



Submitted By

Yang Yang, Zecheng Xu

MEng Field Advisor: David Albonesi

Degree Date: January 2015

Acknowledgements

We would like to thank Professor David Albonesi who served as the faculty supervisor for our design project. We joined the group with little background required for this project, but Dave believed us and always encouraged us to move forward.

Thanks for giving us the opportunity to participate in this project. Our weekly meetings were critical brainstorming sessions to define the problem accurately and provide great opportunities to gain insight into other research related to the project. Dave has always been kind enough to help in every possible way right from the project definition and the technical issues to the project equipment and poster revisions.

Abstract

Master of Engineering Program

School of Electrical and Computer Engineering

Cornell University

Design Project Report

Project Title: Smartphone-based Building Occupancy Tracking

Author: Yang Yang, Zecheng Xu

Abstract: The project is to create an integrated system for real-time tracking of occupant movement inside buildings and adjust HVAC and lighting systems dynamically to reduce energy usage. Android applications were developed last year, which communicate their received signal strength of visible access points to backend server that uses this information to calculate the real time occupant locations and then pass locations information to the HVAC control. We have further developed android application and server applications to improve energy efficiency and tracking accuracy, and to deploy and test the system. Improvements have been performed such as developing a better localization algorithm via cooperation between INS navigation and WIFI positioning, multi-floor localization, multi-threaded server and optimizing communication schema.

Executive Summary

Smartphone App: The Android smartphones periodically scan for all the Wi-Fi Access Points it can see from a given location along with the relative signal strengths and send that information to the backend server via a TCP Socket. Along with this information the Phone also adds its own information such as MAC address and timestamp to the message to help server uniquely identify whom it is tracking.

Backend Server: The backend server is multithreaded so that it can serve lots of clients at the same time. The server receives multiple such messages from different clients, parses the message to extract different fields. The server has the building map in terms of where individual Access Points are located and then it determines the distance from each of these Access Points by extracting the Received Signal Strength Indicator (RSSI). Then the server determines the occupant location by using the weighted centroid localization algorithm. It's also able to determine use's location on multiple floors by finding the floor number of the access point that has the greatest signal strength value. This process is then clustered over multiple clients to obtain the real time occupancy information within buildings and updates user's position dynamically on the map.

Team Coordination

Zecheng Xu: Client-side, Android smartphone application

Yang Yang: Server-side, multi-floor localization, workflow optimization

Work together: weighted centroid localization algorithm, multi-threaded server

Table of Contents

1 INTRODUCTION 6
1.1 MOTIVATION 6
1.2 INDOOR LOCALIZATION 6
1.3 SYSTEM COMPONENTS 7
2 DESIGN PROBLEM AND SYSTEM REQUIREMENTS 9
2.1 CLIENT-SIDE 9
2.2 SERVER-SIDE 10
3 RANGE OF SOLUTIONS 12
3.1 CLIENT-SIDE 12
3.2 SERVER-SIDE 13
4 DESIGN AND IMPLEMENTATION 15
4.1 CLIENT-SIDE 16
4.2 SERVER-SIDE 17
5 TEST RESULTS 20
6 CONCLUSION AND FUTURE WORK 20
REFERENCE 22
APPENDIX 23

Chapter 1 Introduction

1.1 Motivation

We live in a world of smartphones and not so smart buildings. Commercial and residential buildings alone consume 40% of the total energy in the United States and out of this nearly half of energy consumption by the buildings is due to the HVAC (Heating, Ventilation and Air Conditioning) Systems used for heating and conditioning the buildings. As a result, buildings present a huge sustainability problem. Current HVAC systems treat occupancy in a very coarse manner. Buildings do make use of motion sensors to achieve occupancy related information, but this is more like an on/off switch and for better sustainable buildings we need a fine grained approach to how HVAC systems handle occupancy.

We see future HVAC Systems being really smart about deciding the areas within the building which it should control. We believe that in future the HVAC systems automatically “know” where users are within the building and condition accordingly.

So in our project we ask a few important questions and find ways to solve them. How to locate occupants within building accurately? How can the buildings use this information to achieve optimal setting and save energy in the long run?

1.2 Indoor Localization

In our project, occupancy is one of the most important factors. To determine occupancy, we can either use hardware sensors or software programs. Currently we

have occupancy sensors installed in Duffield Hall, however, occupancy sensors could not satisfy the granularity for the occupancy detection and could not tell the difference between various numbers of occupants. Besides, GPS is usually not suitable to use in an indoor environment, since microwaves will be attenuated and scattered by roofs, walls and other objects. However, nowadays every people have a smartphone, so we will take advantage of this to develop a more economical software solution. The major consumer benefit of indoor positioning is the expansion of location-aware mobile computing technique. Applications benefiting from indoor localization include augmented reality, school campus, guided tours of museums and so on.

1.3 System Components

There are three components in this project.

Smartphone App: The Android smartphones periodically scan for all the Wi-Fi Access Points it can see from a given location along with the relative signal strengths and send that information to the backend server via a TCP Socket. Along with this information the Phone also adds its own information such as MAC address and timestamp to the message to help server uniquely identify whom it is tracking.

Backend Server: The backend server is multithreaded so that it can serve lots of clients at the same time. The server receives multiple such messages from different clients, parses the message to extract different fields. The server has the building map in terms of where individual Access Points are located and then it determines

the distance from each of these Access Points by extracting the Received Signal Strength Indicator (RSSI). Then the server determines the occupant location by using the weighted centroid localization algorithm. It's also able to determine use's location on multiple floors by finding the floor number of the access point that has the greatest signal strength value. This process is then clustered over multiple clients to obtain the real time occupancy information within buildings and updates user's position dynamically on the map.

HVAC System Control: The control algorithm used in HVAC system is called Model Predictive Control (MPC), which takes into account occupancy information, weather condition in its objective function and outputs an optimal set point temperature. The energy usage of buildings will be simulated using software to obtain the results and also help compare the energy savings obtained by MPC with and without occupancy information.

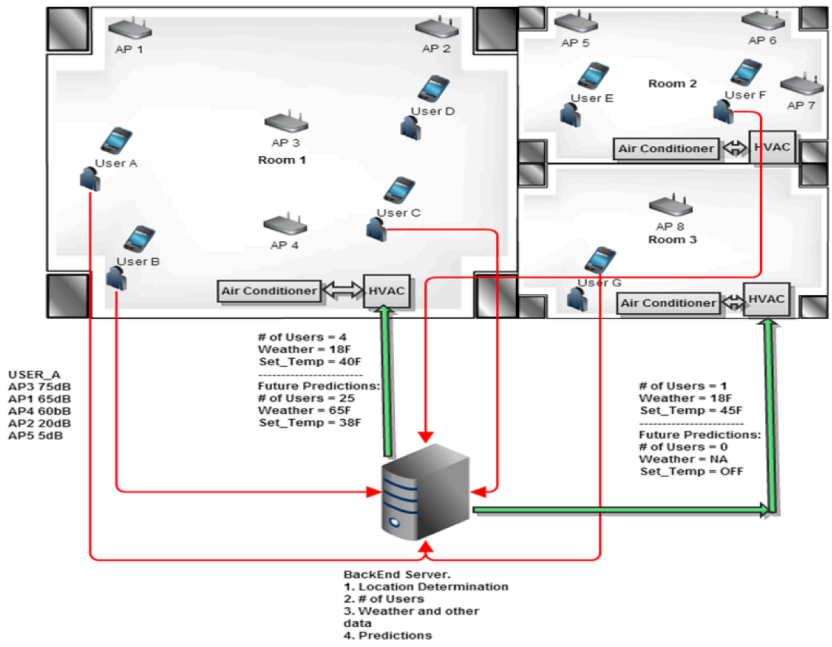


Figure 1 Infrastructure of Project Setup

Chapter 2 Design Problem and System Requirements

2.1 Client-Side

2.1.1 Design Problem

The client-side development process is based on the MEng project last year, we modify and made some improvement:

(1) Better tradeoff between energy efficiency and instant localization

Using the implementation finished last year, we found that although it can reduce energy consumption, it sometimes does not send information to server while user is moving and we should hold the phone and shake hand to make the process begin. In our implementation, we need to modify the trigger of event or test to find a better threshold to ensure localization process starts when user start moving.

(2) Modify server IP in App

In the implementation last year, we have to modify a variant “serverIP” in code and reinstall the app once the IP of server is changed. Since the IP address always changes in LAN, it’s a tedious process. We should make it convenient by adding a function to modify the information of server the phone would communicate with.

2.1.2 System Requirements

Applications in phone are different from those in computers due to many constraints of phones including limited batteries and relative limited computation ability. In order to make the whole system work well, client app should satisfies the following:

- send information to server when and only when it is moving
- read sensor data in an acceptable frequency
- little computation in client side
- application should run as a background service to work even when screen is off

2.2 Server-Side

2.2.1 Design Problem

The server-side development process is based on the MEng project last year.

We have made several improvements, as following:

(1) Improve Localization Algorithm

In the work performed by MEng students last year, weighted algorithm was used. The average error is 6.7m with standard deviation of 4.6m. However, accuracy suffered at some locations, especially at corners of hallways and border of zones divided artificially. We have adopted an alternative algorithm to determine location more accurately, which is called Weighted Centroid Algorithm.

(2) Multi-Floor Localization

The MEng students last year focused on single floor localization. It's of vital importance to expand it to multiple floors. We managed to implement multiple floors localization by finding the access point that has the greatest signal strength value.

(3) Server Workflow Optimization

We have made several optimization of the performance of the server, such as data structures and algorithm optimization and optimization of I/O operations to make it able to respond to clients as quickly as possible rather than wasting time in unnecessary operations. The optimization is important to meet the requirements of a real time application.

(4) Multi-threaded Server

We have improved the server to be multi-threaded so that it can serve lots of clients at the same time, thus minimizing the delay between clients and server.

2.2.2 System Requirements

Because this is a real time system, the server should be able to respond as quickly as possible. Besides, the accuracy of the localization algorithm is the key for this indoor localization system. The application requirements are as the following:

- The server should be able to serve multiple clients at the same time and process the information captured by multiple users.
- The server should be able to calculate users' positions as quickly as possible and as accurate as possible, and then return a three dimensional location for every occupant inside the building.
- The server should minimize the delay as much as possible.
- The server should be able to adaptable to multiple floors and different buildings.
- The server should be power-efficient.

Chapter 3 Range of Solutions

3.1 Client-Side

3.1.1 Tradeoff between energy efficiency and instant localization

3.1.1.1 Send information with certain interval

The method is very easy to implement, what we need to do is simply periodically collect WIFI information it can see and send it to server. There are some advantage using this method: no need to access sensor data via system call and nearly no computation in client side. However, a very severe problem is that it send information to server not only when user is moving but also stationary. It's obvious that it drain battery quickly due to frequent communication.

3.1.1.2 Set appropriate threshold to send information

The condition to decide whether to send information to server is the state of user: moving/stationary. The most convenient and relatively accurate way to determine the state is by using sensor data. Accelerometer is commonly used in this scenario. We need to set a period to read acceleration data, and then using a simple algorithm to judge whether it is moving. If moving, send information to server, do nothing otherwise. It needs some computation in client side using this method, however it ensures no message sent while stationary.

3.1.2 Add a module to modify IP of server it communicate with

The implementation last year tried to do this part but not work as expected. We modify the code to pass *serverIP* between “Activity” in Android app. Since we are not using a public IP, internal IP in LAN always changes, it’s the best way to get rid of the reinstall process.

3.2 Server-Side

3.2.1 Localization Algorithm

3.2.1.1 Radio Signal Strength Indications (RSSI) and fingerprint with digital signal processing technique

The only available information for WIFI equipment is the signal strength received from each access point. Radio Signal Strength Indications (RSSI) can be translated into distances by means of theoretical or empirical radio propagation models.

The main approach for the estimation of location making use of RSSI values is fingerprinting techniques. In these techniques, the mobile terminal estimates its location through best matching between the measured radio signals and those corresponding to locations previously registered in the radio map. This process consists of two stages:

1) Training phase, also called offline phase, in which a radio map of the area in study is built. RSSI values from different beacons are recorded at different locations and optimized with digital signal processing technique.

2) Online phase, in which the mobile terminal infers its location through best

matching between the radio signals being received and those previously recorded in the radio map.

3.2.1.2 Localization algorithm – Weighted Centroid Algorithm

Localization algorithms employed in this case generally make use of deterministic or probabilistic techniques. The MENG students last year used a weighted algorithm to calculate the location of the mobile, which has an average error of 6.7 meters with standard deviation of 4.6 meters. As far as we're concerned, weighted centroid algorithm is a better algorithm to use, which selects several nearest neighbors, forming a polygon, and the centroid of the polygon will be considered as the estimated location.

3.2.2 Multi-Floor Localization

The MEng students last year focused on single floor localization. It's of vital importance to expand it to multiple floors.

One solution is to implement multiple floors localization by finding the access point that has the greatest signal strength value.

Another solution is to measure the variance of the signal strength from different access points to determine the user's direction. If the signal strength is getting smaller, the user is walking away from that access point.

3.2.3 Server Workflow Optimization

When the server boots, it needs to load several files and create helper functions and

data structures to help calculate position. Optimization is important because in a real time environment, the delay should be as little as possible. Optimization can be performed, such as when to read/write files, how to improve the calculation process, try to increase the respond speed and use better data structures and algorithms.

3.2.4 Multi-threaded Server

A multi-threaded server is desired to serve multiple clients at the same time, thus minimizing the delay between clients and server.

Chapter 4 Design and Implementation

4.1 Client-Side

4.1.1 Sleeping Algorithm

We choose to implement sleep algorithm by setting an appropriate threshold. The reason is that people are sitting or staying in the same place when indoor at most of the time. Energy efficiency is guaranteed using this method, whereas simply sending information periodically may drain battery very quickly.

First off, there should be an outer loop with certain interval. Only after the interval, we access accelerometer data and determine whether to send message using lock mechanism. We set the interval to be 1000-5000ms. In each iteration, thread waits until it is notified. After that, message was sent and the thread sleeps for a duration of interval.

We create a sensor event listener to listen to changes of accelerometer data. Every time acceleration is updated, we compare it to previous acceleration data and using a low-cut filter to determine the state of user. While it is moving, notify the send-message loop.

4.1.2 Modify Server IP in App

In order to achieve this, we only need to pass the argument between main activity and the activity responsible for modifying server IP. In background service to send message to server, we use the *serverIP* argument in main activity, in this way the

process is finished.

4.2 Server-Side

4.2.1 Localization Algorithm

We have adopted weighted centroid algorithm. Weighted centroid algorithm is a simple, cost-efficient, and battery-conserving technique to get accurate location information of mobile devices. To implement this algorithm, the necessary inputs are the physical locations of access points and the signal strengths. According to the following table, weighted centroid algorithm and advanced trilateration algorithm give the smallest average errors, however, weighted centroid algorithm is much faster in terms of computation compared to advanced trilateration. The less time it takes for computation, the less power it costs to run this algorithm. Therefore, weighted centroid algorithm is chosen in this project due to its accuracy and power efficiency.

Performance indicator	Weighted Centroid	Advanced Trilateration	Advanced Trilateration (optimized)	Local Signal Strength Gradient
Average error	1.97 m	3.87 m	2 m	2.52 m
Other performance indicators				
Median error	1.60 m	3.01 m	1.62 m	1.91 m
Maximum error	4.02 m	21.97 m	4.7 m	5.87 m
Average time to calculate one BSSID[7]	~0.2 ms	~4.7 s	~25 ms	~42 s

Table 1 Performances comparison of several algorithms

In essence, different weights are assigned to the Access Point depending on the Received Signal Strength Indicator (RSSI) values. The calculations use the following equations:

$$\text{Signal power} = 10^{\frac{RSSI}{20}}$$

$$\text{Weight} = (Signal\ power)^{g} \quad \text{where g = 1.3 according to section 4.1}$$

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i^2 * x_i}{\sum_{i=1}^{N} w_i^2}, \bar{y} = \frac{\sum_{i=1}^{N} w_i^2 * y_i}{\sum_{i=1}^{N} w_i^2}, \bar{z} = \frac{\sum_{i=1}^{N} w_i^2 * z_i}{\sum_{i=1}^{N} w_i^2}$$

For N Access Points captured, ω_i is the weight of i^{th} signal strength, x_i, y_i, z_i are the X, Y, Z coordinates of i^{th} Access Point, and, $\bar{x}$, $\bar{y}$, $\bar{z}$ are the predicted location for the current occupant. The following image shows the whole picture of this algorithm:

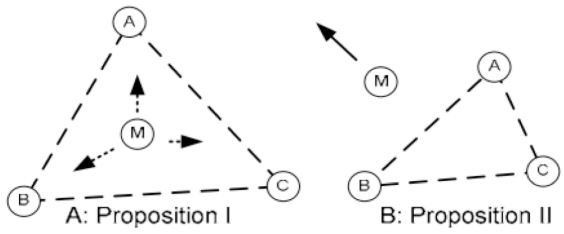


Figure 2 Weighted Centroid Algorithm

Besides, after calculating the predicted location, we also check several locations around the predicted location, and use some formulas to calculate the possibility of each location. Finally, we pick up the location that has the greatest possibility, thus improving the accuracy of the algorithm.

We gave up the fingerprint technique finally because there are not enough access points in Phillips Hall on each floor and the access points are also too far away from each other. The fingerprint technique is not so useful as we expected but slows

down the computation speed.

4.2.2 Multi-Floor Localization

We used the first solution, which is to implement multiple floors localization by finding the access point that has the greatest signal strength value. This solution works well in most cases, but might be interfered by walls and doors. However, the second solution, which is to measure the variance of the signal strength from different access points to determine the user's direction, also suffers from the interference by walls and doors and is also difficult to implement because the signal strengths from different access points are too close to each other, and the variance of the signal strengths value is too small. Compared to the second solution, the first solution is easier to implement and more reliable.

4.2.3 Server Workflow Optimization

We have made several optimization of the performance of the server. When the server boots, it first does some initial setup, such as read files and store the information in an dictionary to be used later, create helper data structures to convert signal strength to RSSI, read building map and store information about the access points in dictionaries, and then sets up a TCP listener. These optimizations make it able to respond to clients as quickly as possible rather than wasting time in unnecessary operations. The optimization is important to meet the requirements of a real time application.

Chapter 5 Test Results

The application system have been tested on the second, third and fourth floor of Phillips Hall with the android phone. It's able to determine which floor the user is on correctly, and the localization accuracy has been improved. From our test, the new technique has a mean error of less than 2 meters under ideal circumstances and is very accurate if the user is close to the access point, which leads to a 300% improvement compared to the original technique. We have also tested multiple clients. The server is able to accept different clients at the same time and finish the calculation within desired time period.

Chapter 6 Conclusion and future work

By improving indoor localization algorithm, we could greatly improve the accuracy of indoor positioning and reduce the power consumption of this technique. Besides, we also made the system useful in a three-dimensional environment, which is a great improvement in multiple floors. By optimizing communication schema and workflow, the server becomes more energy efficient and more tracking accurate.

In the future, further improvements of the accuracy can be made by using the sensors of the smartphone itself as much as possible and combine it with WIFI localization. The system can also be expanded to other buildings. Then finally, we would be able to create an integrated system for real-time tracking of occupant movement inside buildings and adjust HVAC and lighting systems dynamically to reduce energy usage.

Reference

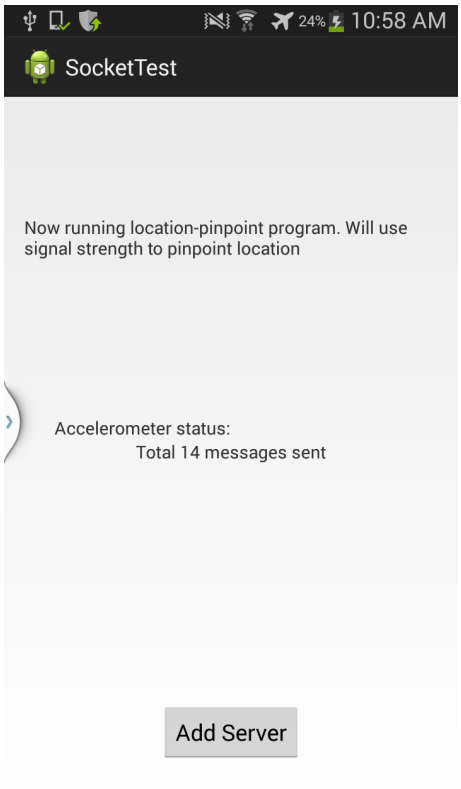
[1] Advanced Integration of WiFi and Inertial Navigation Systems for Indoor Mobile Positioning, Frédéric Evennou and Francois Marx, *Division R&D, TECH/IDEA, France Telecom, 38243 Meylan, France*

[2] Precise Indoor Localization Using Smart Phones, Eladio Martin, Oriol Vinyals, Gerald Friedland, Ruzena Bajcsy, *MM'10,* October 25–29, 2010, Firenze, Italy. Copyright 2010 ACM 978-1-60558-933-6/10/10

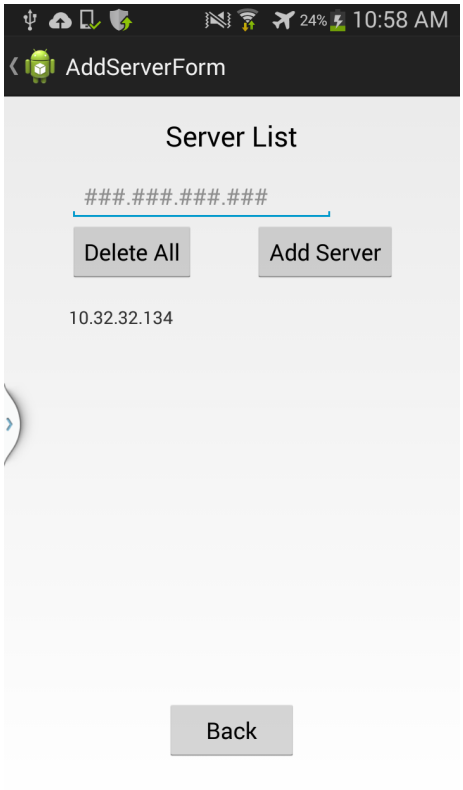
[3] Development and Evaluation of a Combined WLAN & Inertial Indoor Pedestrian Positioning System, Korbinian Frank, Bernhard Krach, Noel Catterall and Patrick Robertson, *German Aerospace Center (DLR), Institute of Communications and Navigation, Oberpfaffenhofen, Germany EADS Military Air Systems, Manching, Germany, HW Communications Ltd., Lancaster, England*

Appendix

Client



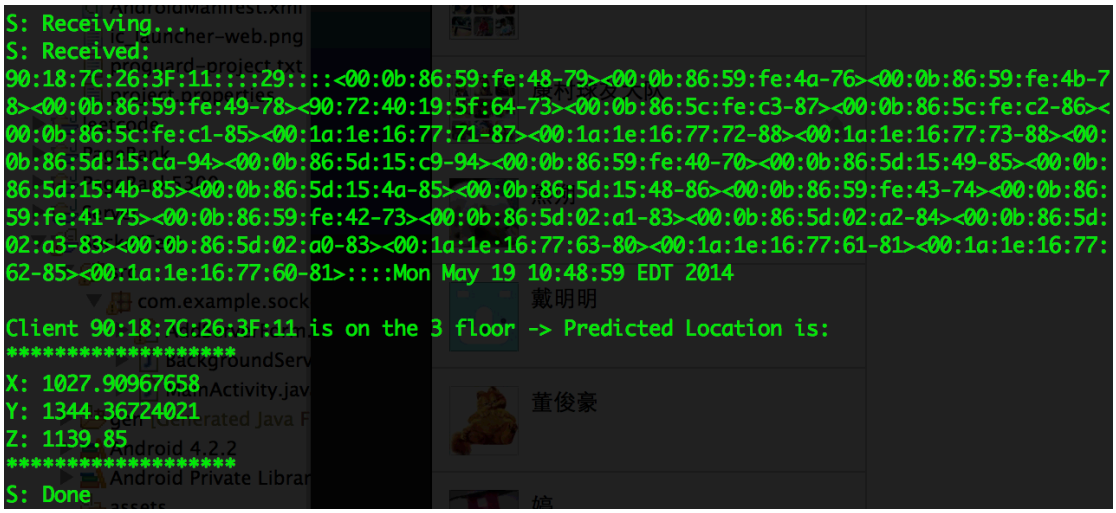
main activity



add-server activity

Server

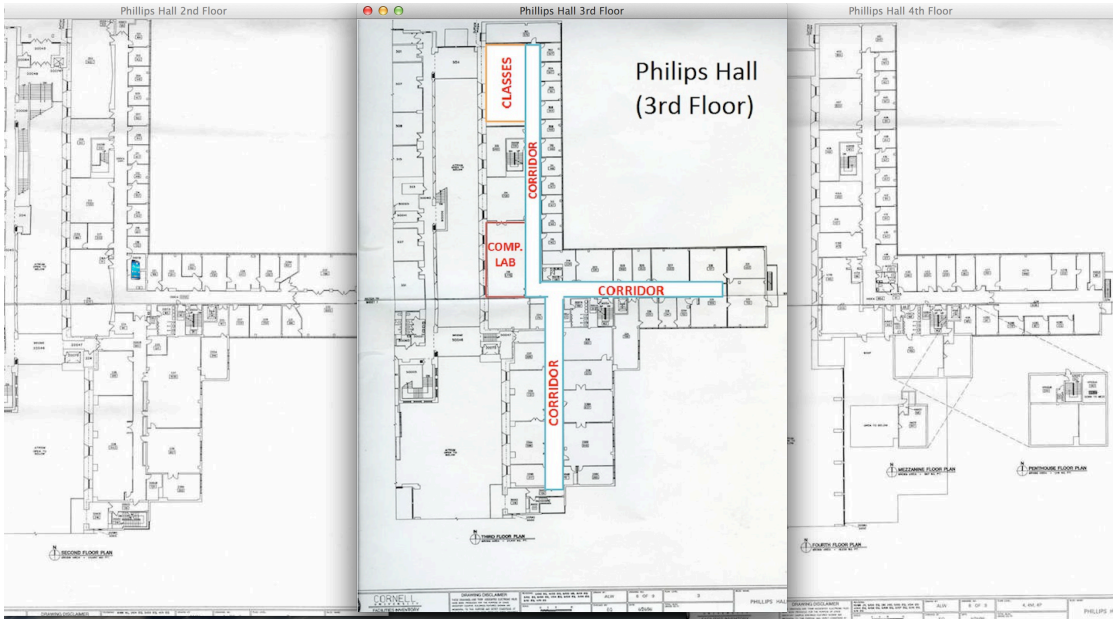
server receives message from client and computes location:



Test Result in Phillips Hall

After getting the position of the user, the server is able to show the position on the map.

User is on the 2nd floor.



User is on the 3rd floor.