

Team Number: May1714

Adviser: Jay Kim

Team Members/Roles:

Ming Wu (team leader)

Xiayang Sun (communication leader)

Chenliang Xu (webmaster)

Xin Ying (key concept holder)

Sam Jones (key concept holder)

Lyle Bishop (key concept holder)

Team Email: [ecpemay1714@gmail.com](mailto:ecpemay1714@gmail.com)

Team Website: <http://may1714.sd.ece.iastate.edu>

Revised: 11/8/2016

# *Android App based on iBeacon*

## DESIGN DOCUMENT

# Contents

## 1 Introduction

### 1.1 Project statement

### 1.2 Purpose

### 1.3 Goals

## 2 Deliverables

## 3 Design

### 3.1 System specifications

#### 3.1.1 Non-functional

#### 3.1.2 Functional

### 3.2 PROPOSED DESIGN/METHOD

### 3.3 DESIGN ANALYSIS

## 4 Testing/Development

### 4.1 INTERFACE specifications

### 4.2 Hardware/software

### 4.2 Process

## 5 Results

## 6 Conclusions

## 7 References

## 8 Appendices

# 1 Introduction

## 1.1 PROJECT STATEMENT

This project is based on iBeacon, which is a protocol developed by Apple. IBeacon is a class of Bluetooth low energy devices that broadcast their identifier to nearby portable electronic devices. This technology enables smartphones, tablets and other devices to perform actions. In our project, we will focus on smartphone applications.

## 1.2 PURPOSE

iBeacon uses Bluetooth low energy proximity sensing to transmit a universally unique identifier picked up by a compatible application or operation system. It allows us to program for more functions. The most important thing is saving energy.

## 1.3 GOALS

We would develop an applications on android phones to communicate with iBeacon. The major functions we would implement are silencing phones and sending announcements posted by authorized users.

# 2 Deliverables

Hardware:

1. We will measure the accurate distance from each iBeacon to electronic devices.
2. Build connections between iBeacon and android application.

Software:

1. We will develop login page for different users. (i.e. for professors to post announcements, they are required to register. Others will login as guest)
2. Design clear and functional User Interface.
3. Work on functions:
4. Send announcements from each authorized professors.
5. Silence phones in specific areas (i.e., classroom)

## 3 Design

### 3.1 SYSTEM SPECIFICATIONS

System Specifications:

**Compatible Platforms:** Android

**Bluetooth Beacon:** ibeacon by Social Retail

**Database:** Iowas State University Servers

#### 3.1.1 Non-functional

For this system to be successful it must meet the following non-functional requirements.

1. Usable. This system must be easy to use and learn.
2. Reliable. This system must perform as intended without problems.
3. Satisfactory. This system must satisfy user or solve the problem it was intended to solve.

#### 3.1.2 Functional

For this system to be successful it must also meet the following functional requirements.

1. Detect and translate Ibeacon Bluetooth broadcast.
2. Send and Receive data from a database server.
3. Display information on the screen.
4. Correctly flow between UI screens.
5. Maintain security by using passwords.

### 3.2 PROPOSED DESIGN/METHOD

Our team has decided to create an application that connects professors and students through a Bluetooth iBeacon device. Our application will be able to detect a professor's iBeacon and pull information or commands from a database about their class.

The application will open up to a welcome screen where the user indicates whether they are a student or a professor.

If they are students, they will be directed to a student home screen where they can toggle between detected classes and view information or commands for each one.

If they are a professor, they are directed to a professor home screen where they have the option to upload a new announcement or command for a detected class. Before the announcement or command is sent to the database, the professor must enter the password related to the class he is trying to upload content too.

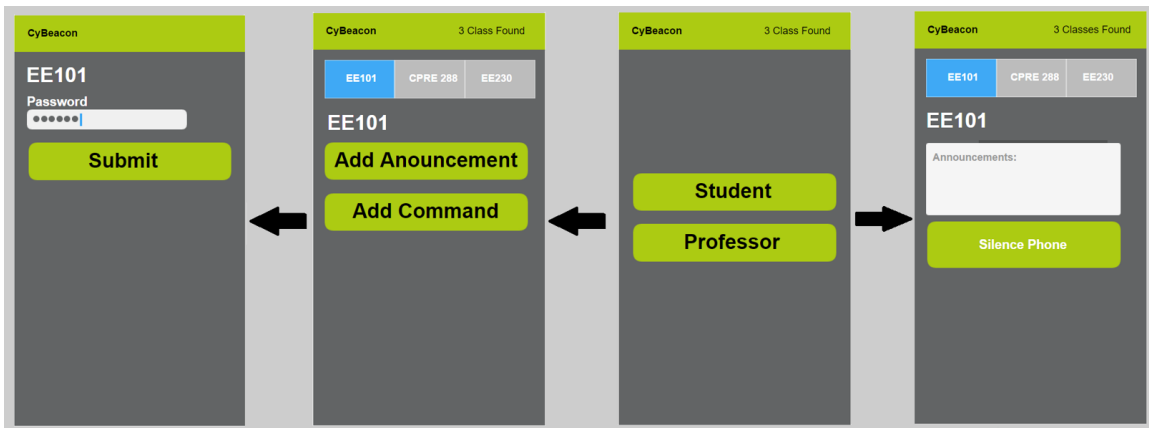
Overall this system streamlines the way students and teachers can communicate into a fast, easy-to-use application.

### 3.3 DESIGN ANALYSIS

Our team has moved forward with this project by determining the basic functionality and beginning the application development process.

We have begun to practice and familiarize ourselves with Android Studio and android application programming since most members of our team have minimal android programming experience.

We have also begun User Interface development. We have created a non-functional prototype of the screen layouts as seen in the diagram below.



*figure 1, user interface screens.*

## 4 Testing/Development

### 4.1 INTERFACE SPECIFICATIONS

Hardware: iBeacon device, Android devices

iBeacon is the key device of our project, every function we came up with is relied on iBeacon. Although iBeacon is developed by Apple, for convenience purpose, our advisor suggested us to develop an application on Android platform.

Software: Android studio, MySQL, Geekbench

Android studio is the compiler and simulator we choose to develop our application. MySQL and Geekbench are both good software for dealing with server and database.

### 4.2 HARDWARE/SOFTWARE

For early test, we will just use the simulator that Android studio provides to check the correctness of UI, connection between client, server and database.

After this app get installed on an Android device, we will use multiple android devices to test the broadcasting announcement functions and group silencing phone functions.

For the iBeacon itself, we will use ruler to measure the hard restriction like maximum range for the iBeacon to broadcast and we will find a software (TODO) to detect the signal of iBeacon as well.

### 4.2 PROCESS

Group silencing phone:

We will turn on the programmed iBeacon, and let people carry multiple Android devices with science mode off enter the broadcast range of iBeacon. Then we will check if the device is completely in silent mode (if we have the permission), or if there are a pop up notification telling user to silencing your device.

Broadcasting announcement:

We will have the programmed iBeacon ready, and similarly let people carry multiple Android devices enter the broadcast range of iBeacon. We need to check if all the devices have correct information displayed. Moreover, we need to test in

situation that there are 2 differently programmed iBeacons in field, and see if the Android device display two different information.

Silence + broadcasting:

This test is the combined version of above two, basically we want to see if we turn on silencing function and broadcasting function at the same time, the devices will perform corresponding functions, i.e. devices are automatically silenced or being reminded you have to silence your phone, and the information page is correctly displayed.

## 5 Results

### 5.1 SOFTWARE

The software test results will be corrected for each error that is shown by the simulator that Android Studio uses. The correctness of the software will be checked and corrected as each part is created to ensure errors are pinpointed immediately.

### 5.2 HARDWARE

When the application is downloaded to an Android device, functionality can be checked. If this functionality is not correct; the errors can be corrected through the software. The functionality will also have to be working with multiple Android devices. If this does not occur, the programming of the database, server, or connection can be corrected.

The range of the iBeacon is to be set to specific distances for each individual application. The measurements of the iBeacon range can assist in knowing the size of rooms or areas the application is possible for and also if the range must be reduced to prevent interference to areas outside the designated application area.

### 5.3 PROCESS

The process of silencing phones, will help allow us to know if the software is interfacing with the hardware properly. If this does not happen, we may have to alter the design to create a pop-up alert to silence an individual's phone. This will be known when we find out if permissions are granted by the user or phone to allow access to silence the phone.

The process of broadcasting announcements can give insight if the server is working properly with the application. Once this is tested adjustments can be made to more specify the announcements to the individual or class.



## 6 Conclusions

Apple's iBeacon was chosen as a bluetooth device, due to its low energy proximity sensing, in transmitting a unique identifier code to be picked up by a software application on any device within proximity to perform certain actions. This serves as a platform for professors and students to communicate with convenience within a classroom or proximity. The software application will be developed based on Android platform for our project due to convenience. Android studio will be used as the compiler and simulator to develop our application. MySQL and Geekbench are good options as a software to deal with the servers and databases needed in this project. Later, multiple testings on broadcasting and silencing functions will be done. Then, measurements of the broadcast signal from the iBeacon will be taken and analyzed to determine the broadcast range.

## 7 References

"CapTech Webinar: iBeacon Demystified." *YouTube*, uploaded by CapTech, 3 Mar. 2014, [www.youtube.com/watch?v=oIGeQqEGhx4](http://www.youtube.com/watch?v=oIGeQqEGhx4).

Joshua Sullivan. *Apps That Know Where You Are: Our Experimentation with Apple's iBeacon Technology*. Nerderly, n.d., <https://www.nerdery.com/blog/nerdery-labs-ibeacon-experiments/>. Accessed 8 Nov. 2016.

Kurt McIntire. *Crafting and Launching Successful Beacon Apps*. Vektor Digital, 16 Nov. 2015, [www.vektordigital.com/2015/11/16/crafting-and-launching-successful-beacon-apps/](http://www.vektordigital.com/2015/11/16/crafting-and-launching-successful-beacon-apps/). Accessed 8 Nov. 2016.

Shane Russell. *A Semi-Technical Lowdown on Working with iBeacons*. ThoughtWorks, 25 Feb. 2014, [www.thoughtworks.com/insights/blog/semi-technical-lowdown-working-ibeacons](http://www.thoughtworks.com/insights/blog/semi-technical-lowdown-working-ibeacons). Accessed 8 Nov. 2016.

Shubhi Mittal. *Li-Fi vs Wi-Fi vs iBeacon (BLE) Technology*. beaconstac, 10 May 2016, <http://blog.beaconstac.com/2016/05/li-fi-vs-wi-fi-vs-ibeacon-ble-technology>. Accessed 8 Nov. 2016.

*The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs (2015)*. Aislelabs, 4 May 2015, [www.aislelabs.com/reports/beacon-guide/](http://www.aislelabs.com/reports/beacon-guide/). Accessed 8 Nov. 2016.

"What can't beacons do?" *YouTube*, uploaded by Ask A Dev, 30 Apr. 2014, [www.youtube.com/watch?v=YGkczwyzyDk](http://www.youtube.com/watch?v=YGkczwyzyDk).

## 8 Appendices

Getting Started with iBeacon: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>