May 17-14 Final Report: CyBeacon

Client/Advisor: Jaeyoun Kim

Team members: Sam Jones
Xin Ying Loh
Ming Wu
Chenliang Xu
Xiayang Sun

Purpose

Sound and vibration from mobile devices can be distracting in classrooms. Therefore, providing a better class environment and convenient service for both college students and professors, was the inspiration of this android application. Smartphone application is the main focus of this project.

Existing iBeacon technology will be used for this project in transmitting signals to mobile devices. The developed smartphone application, CyBeacon will then detect iBeacon signals and perform certain functions on the phone and application.

The primary function is to have mobile devices automatically silenced in classrooms. Second function implemented is easy access to daily course materials. Additional functions can be easily added onto the CyBeacon application.

Definitions

Abbreviation	Description
BLE	Bluetooth low energy
UUID	Universally unique identifier
UI	User interface
URL	Web address/link
APP	Application
PHP	Hypertext Preprocessor
MVP	Model View Presenter

Project Design

Intended Users and Uses

The intended users for this CyBeacon app are mainly students and professors of Iowa . However, CyBeacon's concept may be implemented for other universities as well. Additional users may include any private corporation or events that would like to utilize this concept.

Uses of CyBeacon are as listed below**:**

- Automatically silences phone to improve any existing educational environment
- Quick upload and access to daily materials
- Improves student attendance
- Additional functions can be easily implemented into the app

Overview

Since our design is based on iBeacon technology, we have named our application CyBeacon. iBeacon is a class of BLE that broadcasts a unique UUID. In classrooms, CyBeacon will scan for iBeacon signals and display available courses associated with iBeacons detected. At the same time, mobile devices will automatically be silenced after detecting any iBeacon signals. Users will select options displayed on the screen and the application will interact with database to display relevant information or to perform certain tasks. Students can select relevant courses and access materials uploaded by professors.

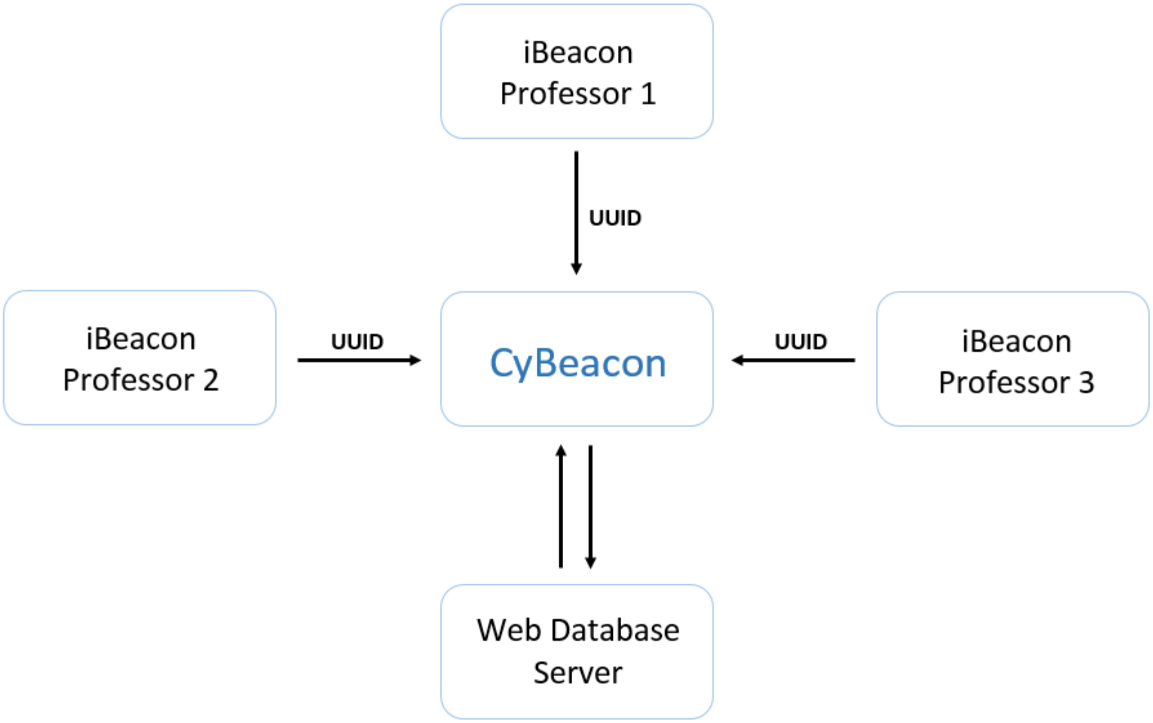


Figure 1. Concept block diagram

System Specifications

Compatible Platforms:	Android
Bluetooth Beacon:________	iBeacon by Social Retail
Database: ____________	Iowa State University servers

Functional

- Maintain security by creating login accounts (username & password)
- Detect and translate an iBeacon broadcast
- Silencing mobile devices when approaching iBeacon broadcasting range
- Display list of courses and its corresponding class lectures
- Store and request data from the database server
- Course material pop ups
- Intuitive flow of user interface

Non-functional

- Usability: easy to learn and use.

- Reliability: performance as intended without problems.
- Security: only authorized users (logging in with username and password) can upload information for courses.
- Capacity: Unlimited number of users can use Cybeacon simultaneously.

Implementation Details

We implemented our design using the android studio environment and developer tools. We used a variety of different packages and classes to achieve each functionality.

We implemented the bluetooth functionality using the android bluetooth class and functions in order to scan and analyze iBeacon devices in the environment.

We utilized the android clock system to create repeated scans without user input to silence a device in range of an iBeacon while the user is not using the app.

We used a server and database provided by ISU to realize the register, login functions for instructors to use and the announcing function which is similar to “Blackboard” for students to check out after they catch the iBeacon signal.

There are two tables in our database named “userinfo” and “website”. The userinfo table stores the username, password, first name and last name in plaintext, assuming we do not need to consider security issue for our App. The website table is the primary table, it stores the UUID, class name, class number, and the website link to the document for each class. We locate the website link by inputting three variables: UUID, class name and class number, which should be obtained from the Android application.

The server contains all the PHP files such as “dbConnect.php”, “getCourseList.php”, “getWebisite.php”, “login.php” and “register.php” that we used to establish connection between Android side and database. Our request methods to database are “GET”, “POST” and “UPDATE” to get access, insert and update the database entries. For the “GET” request, we return the result in volley string array, and with volley library installed in Android Studio, we are able to parse the strings.

The third-party frame used in this project: Okhttp

Project structure description:
This project is based on MVP structure. Model layer is used to Encapsulation interface data and provide method to get data. Presenter layer is used to handle activity logic. It

also has view and model layers reference. Developer could provide public interface when visiting model and view layers in this class. The complex logic is written here. View layer is to handle screen logic, for instance, showing or hiding UI, progress loading and communication with UI. All layers, which are model, view and presenter have their own function. This will reduce module coupling, which made code logic clear and convenient for maintaining.

Project multidex description:
This project dex is based on screen function: register, login, main page and course. Presenter layer is inheritance of father's class: BasePresenter, which has the reference of view layer. View layer is for screen communication. It is inheritance of IBaseView, which has only one mathod: showErrorMessage to handle error logics. In order to provide convenience for developers, encapsulate repeated code into activity and provide initialized presenter and view method. This will reduce redundant and repeated code.

Project view description:
All lists used in this project are from RecyclerView and Adapter of RecyclerView, which are inheritance of BaseCommonAdapter. This class encapsulate some repeated code. Developers only need holder for UI setting in convert, such as setText. If there is no method for use, developers could also expand ViewHoder or get view ID through holder. If there are different request for various types items, developers could inherit BaseCommonMultiItemTypeAdater and set different item proxy to handle UI. The XZRecyclerView in our porject is inheritance of RecyclerView, which could refresh head, load more view, pull up loading and pull down refresh.

Testing Process/Results

In order to test the CyBeacon application during and after development, we utilized a number of different tools.

During development we used both the device emulator and a Nexus 7 to test and verify code functionality. Different parts/functions were coded and tested separately to ensure correctness before implementing them one by one into the app.

Using a smartphone application we were able to simulate an iBeacon device signal and use it to verify that CyBeacon can detect and analyze iBeacon devices. We used multiple smartphone with different UUID, Major, and minor values to verify that CyBeacon can detect multiple iBeacons in the environment.

We verified the phone silencing functionality by playing music on the Nexus 7 and observing the tablet being silenced after turning on a known iBeacon device nearby.

We also made full use of android testing and debug environment. Using this frame, we could precisely control this application. Before launching our program, we created simulative system objects, such as Context; triggering UI events; checking application status during executing. Junit is also helpful trying to implement two main functions and connections.

In order to test the database and PHP files separately from the application, we used a browser to test the "GET" request directly by using the URL that used to access PHP file and with the input to the required variables specified after the question mark. We can then check if the volley array it returned matches the database. In addition, we made a testing Android app with to test the connection between Android side and server. We verified the "POST/UPDATE" method worked by cross checking the results seen with the database.

Appendix 1: Operation Manual

The Cybeacon app is mainly designed for two groups of users: students and professors. Due to different purposes of using the Cybeacon app, the app provides two activities, which are for student use and for professor use respectively. Every user will choose one of the two activities in the main page of UI depending on whether he/she is a student or a professor. The Cybeacon app can fulfill two functions: muting android devices and searching for course material. In this appendix, there will be a step-by-step instruction on how to use the Cybeacon app in both activities. For better understanding of this instruction, the appendix includes several screenshots of the UI.

- **Automatic Silencing Function**

  The Cybeacon app utilizes the android clock system, which can constantly scan for iBeacon signal in a specific range while the app is not in use. For the silencing function, the Cybeacon will not distinguish the two groups of users, all cellphones with the

Cybeacon app installed will be silenced when in range of an iBeacon device. Both professor and student users don't need do anything to silence their cellphones. Once their cellphones are in the range of an iBeacon device, the Cybeacon app will detect the iBeacon signal and execute the silencing function to mute their cellphones.

- **Course material Function**

  Users will see the main page when they open the Cybeacon app. As we can see in Figure 1, there are two buttons "Student" and "Professor" in the main page. Each button stands for one group of users.

  <u>Student User</u>

  If you are a student, you will click on the "Student" button in the main page. Then you will enter the course list page, which is shown in Figure 2. All the courses will be displayed in the course list page. In our testing example, there are 10 courses (Course 0-10) in this page and each button represents a specific course. When you click on any one course button in course list page, you will go into the class list page (Figure 3). You can open a website which contains course materials by clicking on a specific class.

  <u>Professor User</u>

  If you are a professor, you are supposed to click on the "Professor" button in the main page. Then a login page will appear in the screen. As you can see in Figure 4, you will be required to type in the username and password before going to next page. If you are a new user to the Cybeacon app, you can simply register an account by clicking on the small "register" button which is located below the "LOGIN" button. In the register page (Figure 5), you will be asked to type in your last name, first name, username, and password. Once you succeed in logging in, you will enter a page (Figure 6) where three buttons are shown. You can choose to send message, add command, or add announcement by clicking on one specific button.

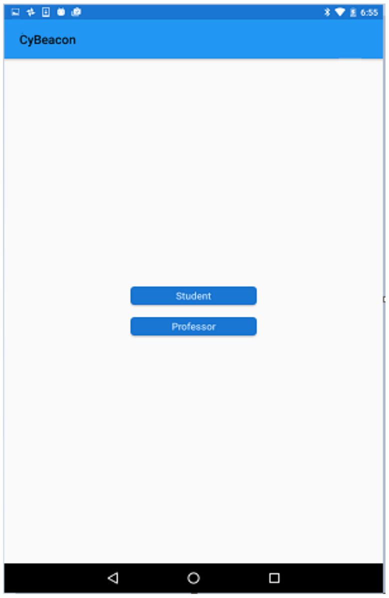


Figure 1

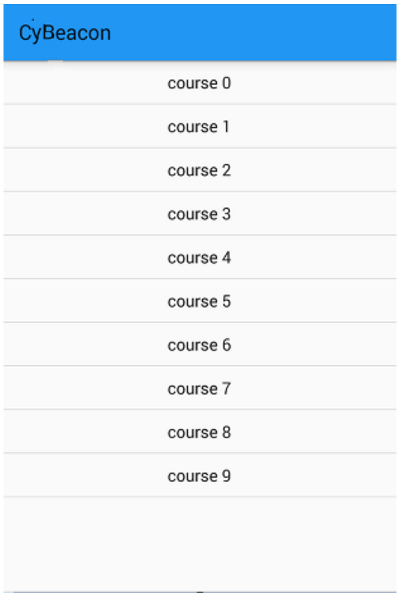


Figure 2

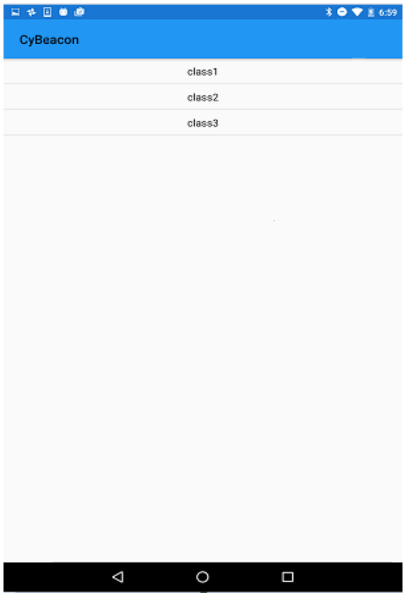


Figure 3

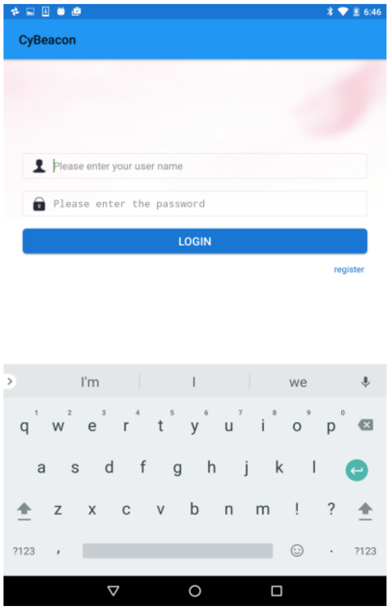


Figure 4

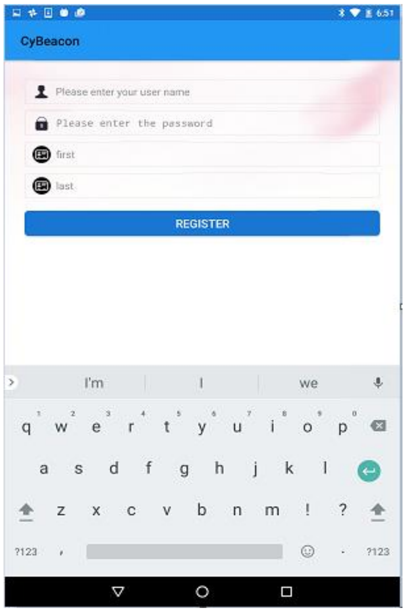


Figure 5

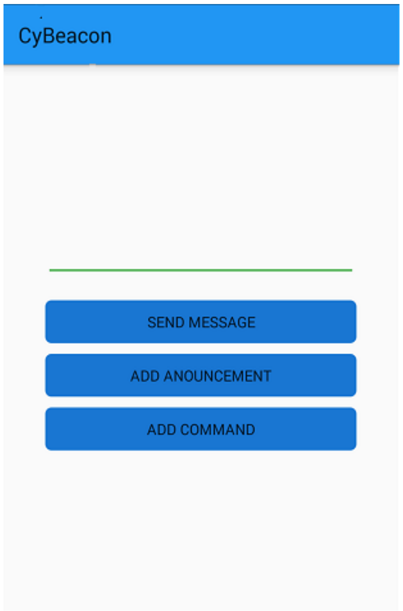
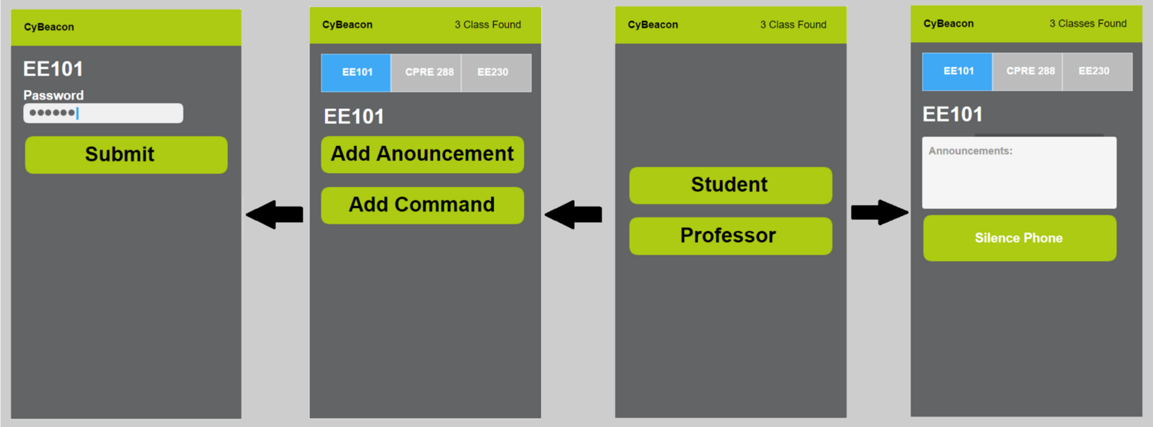


Figure 6

Appendix 2: Previous Design Versions



This is previous design version, the main page has two activities for different authorities, which are students and professors. For students, they don't have further activities, instead, on the top of right corner, there will be displayed the number of available course in specific range. If users choose professor, this app will link them to the "function page". Because we designed to implement two major functions: muting phones and searching for available courses, this app will allow professors to add command, which is muting phones. However, after discussing details, we modified this app in several ways:

1. For main page, there will be students and professors. In order to give function permission for the professors, it is necessary for them to register and login. While for the students, consider the large population, we will default as guest to login.
2. For students, we modified the app to use a "refresh" function to search and display all available courses in the range of bluetooth areas, which are the number of iBeacons.
3. For each available course, there will be a URL link that will take students to the webpage professors assigned.
4. Instead of allowing professors to manually click on the "Silence phone" command, the silencing function was modified to perform automatically in presence of an iBeacon signal.

Appendix 3: Other

I learned a lot about android application development and bluetooth during this project.
-sam