



The Chinese University of Hong Kong

Department of Computer Science and Engineering

LYU1403

Exhibition Navigation with iBeacon

Supervised by Professor Michael LYU

NG Ping Him 1155029312

YEUNG Tak Wing 1155009050

Table of Contents

- Introduction 1
 - Background 1
 - Introduction to Bluetooth..... 2
 - What is indoor exhibition and navigation system? 3
 - Why Bluetooth Low Energy (BLE)? 4
 - Customized marketing 5
 - Objective 6
- Requirement Analysis 7
 - Services 8
 - Kontakt.io..... 8
 - IndoorAtlas..... 8
 - Proximity Kit by Radius Networks 9
 - Case studies.....10
 - City Gallery10
 - Macy’s14
 - LINE Maps for Indoor14
 - Project Scope and Stepwise Refinement.....27
- Technology29
 - Bluetooth Low Energy30
 - Generic Attribute Profile30
 - Speed31
 - Frequency Band.....31
 - Setup31
 - Range31
 - Security.....31
 - Topology32
 - Power consumption32
 - Bluetooth stack32
 - Services and Characteristics.....32
 - Descriptors.....33
 - Central and Peripheral Profiles33

GATT servers and clients.....	33
Cost	34
iBeacon	36
Modes	36
Editing an iBeacon	36
Advertising packet.....	36
TxPower, RSSI	38
Working mechanism	39
Android.....	40
Permissions	40
Setup	41
BluetoothAdapter	41
Start Bluetooth	41
Connect to a BLE device.....	43
Read BLE device	44
Notifications	44
Close connection	44
Third Party Library	45
Indoor Positioning	47
Overview	47
Deployment and Calibration	48
Trilateration	49
Quality of Measured RSSI	51
Cell-based Region Detection	56
Design	60
Android.....	61
Interface to be shown to visitors.....	61
Interface to be shown for internal testing.....	65
Web.....	67
Database	67
CMS.....	68
REST API	78
JSON	80

Documentation.....	82
Implementation	84
Android.....	84
Web.....	115
Limitations.....	154
No background scans.....	154
Beacon Distances, Noises and Interferences	154
Floor and ceiling, neither good	155
Not to be deployed on the floor	155
Not to be deployed on ceilings	155
Neither too close, nor too far	156
Too close	156
Too far	156
Background noises	156
More is less	157
Not on the same wavelength	157
Different scanning devices.....	158
Publicly available IDs.....	158
Preview of Future Works.....	159
Multiple user CMS	159
Extend API	160
More accurate positioning determination	160
Acknowledgement.....	161
Bibliography	162

Introduction

Background

With the advancement of technology, the accuracy of outdoor positioning and navigation systems has increased rapidly these years. For instance, the development of GPS is mature. According to the United States Government, the accuracy of civilian GPS is up to 7.8 metres and has a 95% confidence [1]. However, there are far fewer people developing indoor positioning and navigation systems and the pace of the development is far slower than GPS. We want to be one of the pioneers of developing indoor positioning systems, which have potential uses in marketing and exhibition navigation in near future. Hence, we have chosen this topic for our final year project.

Introduction to Bluetooth

Bluetooth can easily be found in many mobile phones, laptops and sensors nowadays.

The ease of pairing devices and low battery consumption have made Bluetooth more and more popular.

Bluetooth is currently regulated by Bluetooth Special Interest Group (SIG) and uses frequencies band between 2.4 GHz and 2.485 GHz. Bluetooth is claimed to have a maximum range of 60 metres; while in reality, the range might change due to different environment, connections nearby and obstacles in the area. To prevent interferences between different connections nearby, Bluetooth uses a frequency hopping spectrum which jumps from one frequency to another at 1 MHz or 2 MHz intervals, making the frequency spectrum contains 79 channels or 40 channels respectively. Up to today, there are already four versions of Bluetooth - from version 1.0 to 4.0, specified by the SIG. Bluetooth Low Energy (BLE), which is being used in our research project, is specified in version 4.0, released as a subset of Bluetooth. In 2011, the SIG used “Bluetooth Smart” as the marketing name for this new specification. In the following article, Bluetooth Low Energy and Bluetooth Smart are considered the same technology.

BLE uses even less power than previous versions of Bluetooth. Because of its low power consumption, a BLE chip can work by using a silver oxide button cell which is commonly found in calculators. Furthermore, one battery is enough for the chip to work for about 1 year without replacement. In December 2013, the SIG updated their specification from version 4.0 to 4.1. New features of Bluetooth version 4.1 include dual mode and limited discovery mode.

What is indoor exhibition and navigation system?

Indoor exhibition and navigation system provides relevant information on exhibits and turn by turn navigation routes from any source to a destination. It can also be configured to provide customized marketing information to customers, instead of massive advertisement without a target.

An indoor exhibition and navigation system involves two major parts. The first part is indoor positioning, which determines the users' context and the relevant information to be shown. In our project, BLE and mobile devices are chosen to fulfill this task. The second part is data storage, which describes the details of the exhibition and provides the information. An online database backend, from which the mobile device retrieves data, will be deployed for this task.

The requirement of an indoor exhibition navigation system will be further discussed in the section "Requirement Analysis".

Why Bluetooth Low Energy (BLE)?

GPS does not work very well indoors; while WiFi is not designed to provide information on navigation; the range of NFC too short for accurate position determination; the range of cell towers is too large that it covers up to 20 miles [2]; and RFID requires an antenna and a special reader that is not readily available on consumer mobile devices [3].

Therefore, a better alternative solution must be found. We later discovered Bluetooth Low Energy (BLE) is built for information systems that works well both indoors and outdoors and compatible to the major mobile platforms and devices today. Even more, BLE uses little energy - implying that the maintenance needs not to be done very frequently. Therefore, we have decided to use BLE as our indoor exhibition system.

Customized marketing

Most of the advertising nowadays, including TV ads, spams and printed leaflets, do not have a specific target. These advertising messages are often ignored by the majority of the customers and hence results in high promotion costs and poor marketing outcome. BLE exhibition and navigation can provide customized information to a specific group of customers. For instance, shop owners can setup a database storing customers' shopping behavior and preferences on the products they like. When a customer visits the store again, relevant information such as customized recommendations can be provided to them. These data collected by the database not only enhances in shop experience to shoppers, but also provide precious analytics to shop owners to fine-tune their marketing strategies.

Objective

Through this project, we hope to:

- provide a more accurate indoor navigation system as the market lacks this kind of information system.
- offer a brand new marketing and advertising experience to customers and better customer analytics to shop owners.

Requirement Analysis

The deployment of beacons for exhibition navigation systems is still not very popular today. There are not many companies providing easy beacon deployment services.

However, we can still look into some examples and hope to provide some inspirations to our project.

Services

Kontakt.io

Kontakt.io provides an API and SDK on the cloud along with beacons setup and management services [4]. It also has a global CDN to deliver fast data queries to customers. In contrast to ordinary beacons, the company claims their beacons are smart that they are suitable for a wide range of applications. Their beacons have a 32-bit ARM CPU, 256KB of RAM and a Bluetooth chip, which supports a range up to 70 meters. Even more, Kontakt.io beacons are IP-57 verified - the casing protects the component from water splash and dust. However, the application developed on these beacons can only invoke a web page activity when they are detected - which limits the extensibility of the application. Even worse, we have to buy the subscription plan if we want to use their beacons, which greatly increases the cost of development and maintenance.

IndoorAtlas

IndoorAtlas, a startup based in Finland, claims to provide a far more accurate solution for indoor positioning systems [5]. The technology that the company uses is the detection of the disturbances of the Earth's magnetic field by the architecture. This method has been verified by a journal article, "Global Indoor Self-localization Based on the Ambient Magnetic Field" [6], published in October 2009 by one of the founders of the company.

IndoorAtlas emphasizes that no additional hardware is required for setting up the location navigation system, as the technology only uses the magnetometer, which can be found in most smartphones today. Furthermore, there is no cost in using the SDK,

API and CMS provided by the company. Only large-scale deployment schemes require a fee to use their service. The online CMS requires no codes to operate - only dragging and dropping of indoor layout map and entering of some basic information are needed. Yet, this method for location estimation is unusable in outdoor exhibition navigations as the interruption of Earth's magnetic field by outdoor objects are weak, thus, the recognition of patterns of disturbances of fields would be more difficult.

Proximity Kit by Radius Networks

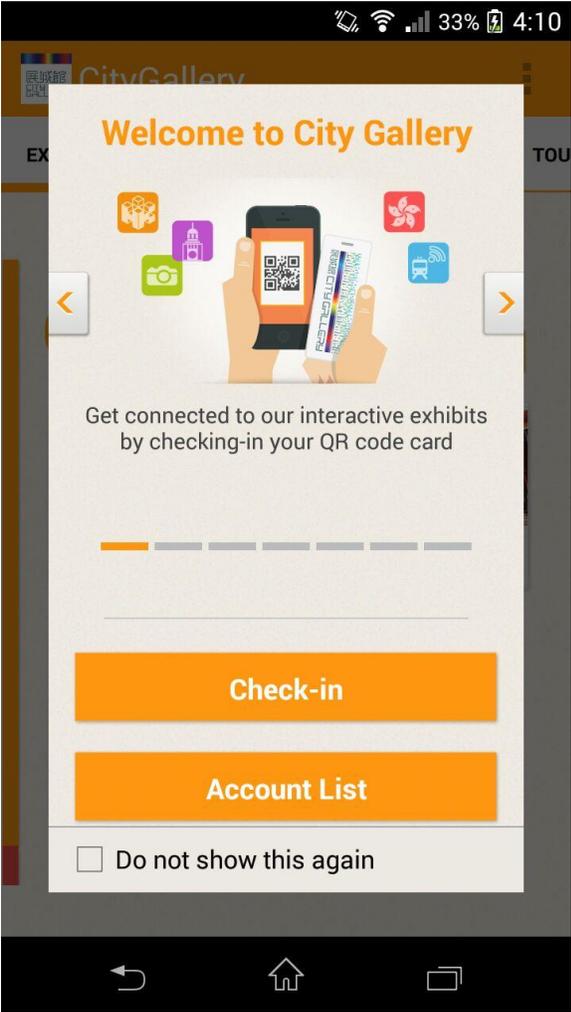
Though Proximity Kit does not bundle beacons with their subscription plans, the company sells AltBeacon standard beacons (another product by Radius Networks) and peripherals [7]. They sell subscription plans that provide an online CMS and analytics. Moreover, like Kontakt.io, Proximity Kit has its own SDK. The company states that their advantage over other competitors is that hard-coding of coordinates of the beacons is no longer required. However, the estimation of location by Proximity Kit on Android devices is not accurate as there is a bug in averaging the signal strengths in the library that they used [8]. Moreover, different Android phone models have drastically different Bluetooth signal strength response. Yet, the library only has the calibration data of a few models.

Case studies

City Gallery

The City Gallery, which is formerly called the Hong Kong Planning and Infrastructure Exhibition Gallery, located at City Hall, Central, makes use of indoor exhibition location technology to implement audio guided tours [9]. The Gallery provides applications for iOS and Android for the exhibition with almost the same feature sets. Features like exhibition overview, tour guide and latest news are included in the application [10] [11].

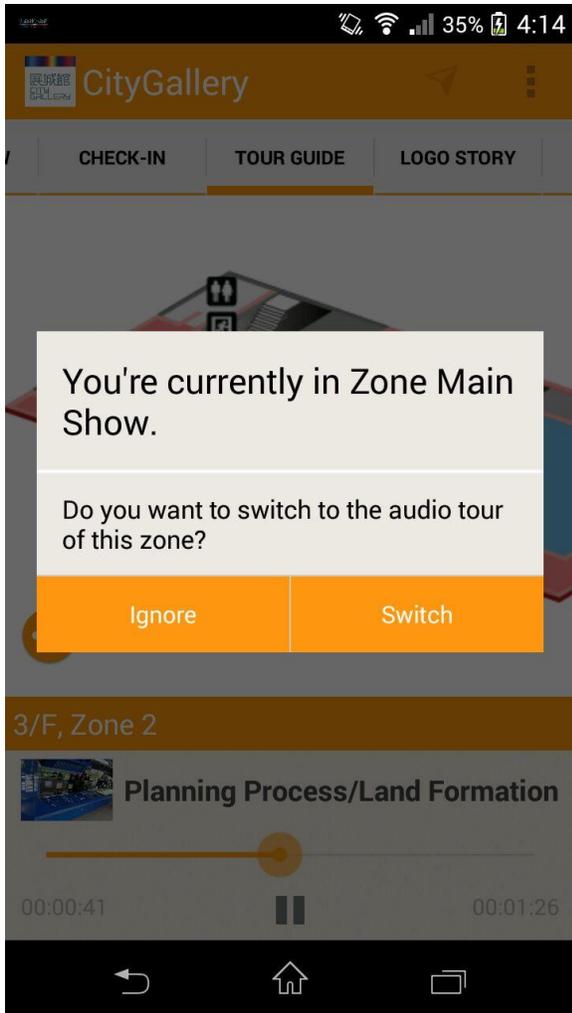
The navigation system there makes use of ultrasound and WiFi, instead of Bluetooth, to estimate the position of the user. The system is able to identify the exhibition area in which the user is located at a high accuracy. The audio track corresponding to the area will be played accordingly. Though the technologies involved in the Gallery are not in the scope of our project, the content and user interface of the mobile application provide good examples for us to develop our application.



Welcome screen of City Gallery's application



Floor plan and audio tour



Automatic location detection

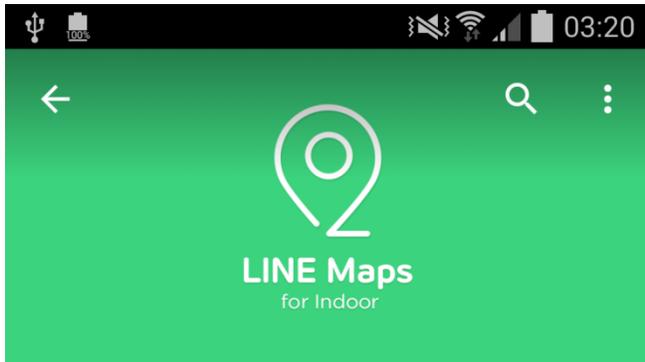
Macy's

Macy's owns a big chain of department stores in the United States. Two years ago in November 2012, the department store giant had already deployed Bluetooth beacons in its biggest store in New York City and updated its iOS application to test out the new indoor navigation technology [12]. The company partnered with Meridian [13], now owned by Aruba Networks, a WiFi solutions company, to roll out the beacons installation and application deployment project. The iOS application brings in-store turn-by-turn navigation and product information to shoppers [14]. Recently, an Android application by Macy's without turn-by-turn navigation is also released on Google Play [15]. However, both Meridian and Macy's did not provide further information on the technical details on the mobile application. It is believed that Macy's uses BLE technology to bring customers to the products they want.

LINE Maps for Indoor

A new indoor mobile map application, which targets shopping malls and department stores, was released in November 2014 by LINE. This application provides turn-by-turn navigation for multi-storey buildings, shops and facilities searching, and search results filtering according to shop categories [16] [17]. It also provides useful information like locations of washrooms, AEDs, entrances and exits, directions to the nearest train stations and shopping mall opening hours. Its pannable 3-dimensional map layout with short text descriptions makes following the route easier than plain text instructions. However, this new application is quite unstable and have unexpectedly quit for thrice during our test. The technology used for indoor location determination by it is WiFi and we doubt the accuracy of the estimated location. LINE Maps for Indoor is currently available on Android for a few malls and department stores in Tokyo and Kanagawa

only. We hope that this promising application can be used in popular locations all around the world to promote the use of indoor exhibition navigation.



 **LINE Maps for Indoor**
LINE Corporation

[UNINSTALL](#) [OPEN](#)

 **10**
Downloads

 **4.3**
★★★★★
70 

 **Travel & Local**

 **Similar**

No more wandering around confusing indoor areas.

[READ MORE](#)



[LINE Maps for Indoor](#)

Shibuya



Bunkamura
Complex Cultural Facility
Hours 10:00 - 22:30



LUMINE MAN SHIBUYA
Shopping Mall
Hours 11:00 - 21:00



SHIBUYA109
Shopping Mall
Hours 10:00 - 21:00



Shibuya Hikarie
Shopping Mall
Hours 10:00 - 23:30

[List of malls that provide indoor maps](#)



Satei Zenkashoin

5F | Teahouse



Shibuya Hikarie

5F

Indoor map showing shop names



From LAWSON

To Satei Zenkashoin

65m, 4Min.

Go 6 floors

 Start Guide

11F

Go to Floor 7 using
an Express Elevator

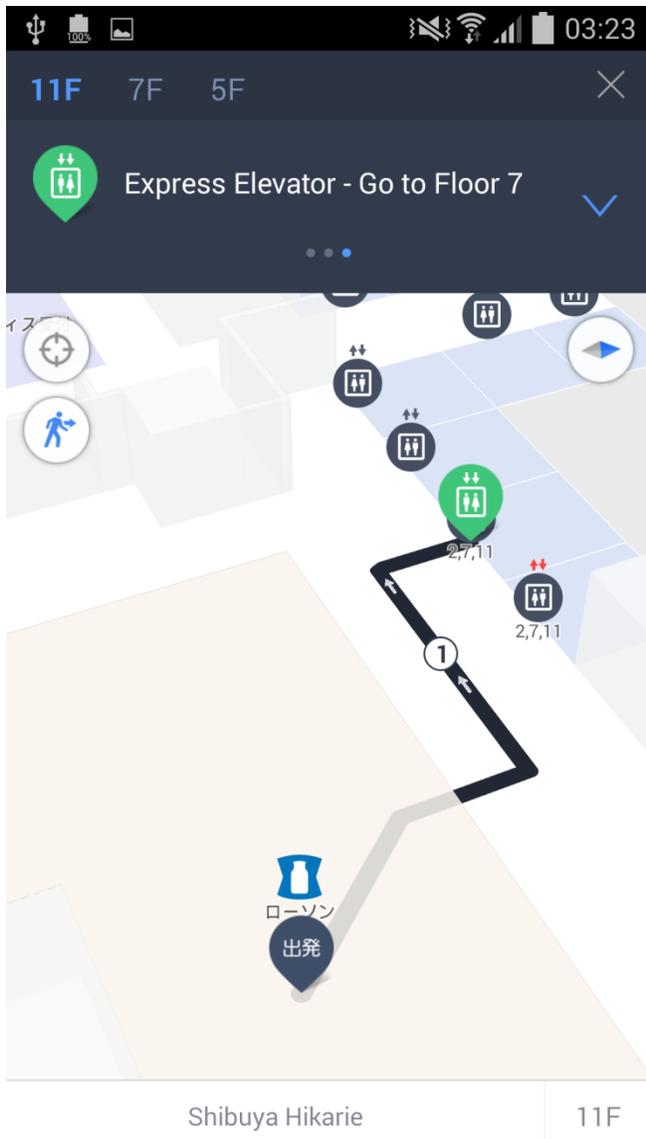
1m, 1Min.



Go using an Express Elevator

7F

Route overview



3D Indoor turn-by-turn navigation



3D turn-by-turn navigation showing destination



Shibuya Hikarie

Shibuya, Shibuya-ku, Tokyo
2-21-1

Phone Number

03-5468-5892

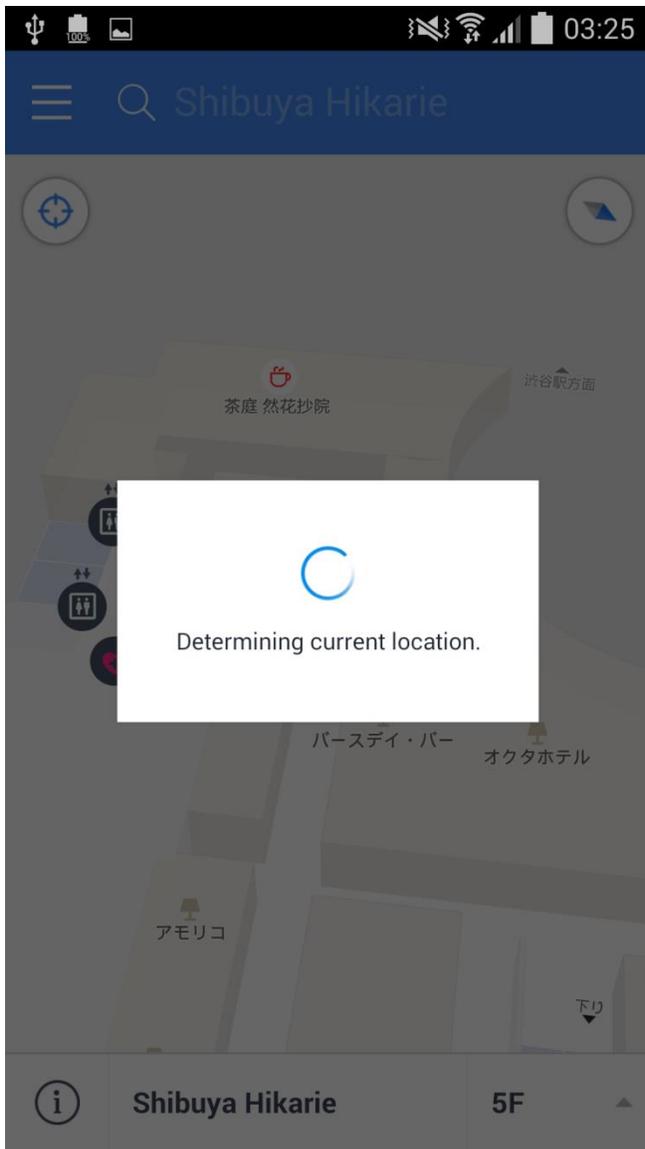


Hours

[ShinQs](#)

Every day 10:00 ~ 23:30

[Mall information](#)



Automatic location determination

Category

 Exit & Entrance (7)

 Toilet (36)

 Smoking Area (4)

 Facility (267) ▼

 Convenience Store (1)

 Cafe, Dessert (13) ▼

 Food (49) ▼

 Fashion (21)

 Fashion Accessory (33)

 Shopping (184) ▼

 Repair (1)

[Shops sorted according to categories](#)

The following table summarizes the exhibition navigation systems and services mentioned above.

Features	IndoorAltas	City Gallery	Macy's	LINE Maps for Indoor
Technology	Magnetic field	WiFi and ultrasound	Bluetooth Low Energy	WiFi
Calibration	Required	-	-	-
Ease of use	Difficult	Easy	Easy	Very easy
Provides custom information	Information can be customized	Customized information	Customized information	Not much, provided by shopping malls in Kanto, Japan only
Other relevant information	In-store advertisements , needs to be set up manually	Exhibits information	In-store advertisements	Toilets, AEDs, entrances and exits and train and bus stations
Accuracy of location determination	Very high	High	Low, according to user reviews	Rough estimation
Turn-by-turn navigation	Yes, needs to be set up manually	No	Yes	Yes, supports multi-storey buildings and method of travelling (by escalators or elevators)
Cost of deployment	Second lowest among four	High, tailor-made	High, tailor-made	Free to a very small fee

Apart from IndoorAtlas, City Gallery's, Macys' and LINE's application cannot be customized as they are all tailor-made for specific customers, making editing information in the application difficult. Only IndoorAtlas' service requires a set up and calibration for detection of the magnetic field. All of the applications can provide relevant information, for example, advertisements and directions to exits, to customers.

Out of our expectation, Macy's application performed the worst among four navigation systems in indoor location determination. According to reviews on App Store, the navigation of Macy's application is disastrous. IndoorAtlas' service performed surprisingly well as we can consider the earth's magnetic field should be constant at all times and ultrasound used by City Gallery should be less easier to be diffracted. As expected, LINE's map mobile application can only provide a rough estimation of a user location as it uses the signal strength of WiFi only.

Except City Gallery's application, all the other three provide turn-by-turn navigation. However, the navigation provided by IndoorAtlas has to be manually set up. LINE Maps for Indoor provides excellent navigation as the application supports multi-storey buildings and multiple buildings. It also provides the best route or customized routes to users.

Finally, the cost for deployment and maintenance is the lowest if we use LINE's application for navigation. However, a small fee might have to be paid to LINE for allowing the application to show the information of the mall or exhibition. As City Gallery's and Macys' application is tailor-made for the exhibitions and stores, we can expect that the cost for deployment is very high.

From the comparison of some of the popular exhibition navigation systems and services above, LINE Maps seems to have more advantages than the other three and we suggest that we can refer to the successful results of LINE's application as our direction of application or API development.

Project Scope and Stepwise Refinement

From the above analysis, we have determined that our project should include numerous parts. A client-side mobile application will be developed for indoor navigation. It will be responsible for scanning beacons, determining user's location and display of relevant information. All information will be retrieved from the server. The app should be able to identify different users for personalized navigation and marketing.

On the server side, a database will be deployed. The database will store all information related to the deployed beacons, the exhibition layout, and the exhibits information and user analytics. On top of the database, a content management system (CMS) will be developed. The CMS enables easy modification to the database and eliminates the need to update the app due to changes in exhibition content.

Apart from these software components, two design questions will have to be solved.

The first one is how an indoor environment layout can be represented in the database.

The second one is how the beacons should be deployed.

The flow of exhibition navigation involves the following six steps:

1. When a user enters a venue, the application automatically downloads the floor plan and information related to the venue.
2. The phone continuously scans for beacons in the area that the user is located. The scan is done in background, no user interference is required. However, no location can be determined at this stage.
3. The list of scanned beacons is saved temporarily in the mobile application. This list provides a lot of useful information, for example, beacon UUID, major, minor and their RSSI (emission power), for estimating the location of the user in the later stages.
4. The application uses beacon IDs, RSSIs (power levels) with our REST API to identify the user's location.
5. After the location is determined, a request is sent to the server to display the exhibit(s) nearby. The page is displayed to the user by using WebView, which allows dynamic content to be updated frequently and efficiently without adding loads of extra codes to the mobile application. Multimedia content will be supported.
6. Finally, the requested page gathers information from the database, which stores the exhibits details, and is displayed on the screen of the mobile device. If the user moves away from one exhibit to another, the page automatically changes content to another exhibit.

Technology

The following sections further analyses the choices of technology we have made, namely BLE and the Android mobile platform.

Bluetooth Low Energy

Bluetooth Low Energy is a subset of traditional Bluetooth. However, it uses far less energy than conventional Bluetooth. BLE can be used to transmit small packages within mobile devices, sensors and wearable devices.

Generic Attribute Profile

The Generic Attribute Profile (GATT) makes definitions on how Bluetooth devices communicate with each other [18]. There are four GATT profiles defined in the Bluetooth specification, but we will only cover two of them, which Android supports as of November 2014, peripheral and central. Similar to onion routing, GATT profiles nest information inside themselves. The outermost layer is usually called Service, indicating the name and type of function that the Bluetooth device provides. The next layer usually nests a number of Characteristics - the detailed features that the devices give. Under Characteristics, there are numerous name-value pairs defining the custom preferences of each character. For instance, some custom preferences are the properties, value and descriptors of the beacon.

As mentioned above, the two profiles that Android supports by now are Central and Peripheral. The Bluetooth network adopts a star topology approach to connect devices nearby (to be discussed a subsequent section). The beacon to be scanned holds a Peripheral profile, also called the GATT server, while the phone or tablet looking for beacons hold a Central profile, also known as the GATT client.

Profiles do not exist in peripherals, to be exact, peripherals only hold a list of services defined by manufacturers. For example, the Heart Rate Profile, only mashes up the

Heart Rate Service with Device Information Service. The complete list of profiles can be found on Bluetooth's official website [19].

Speed

Though Bluetooth Low Energy has a maximum data transfer speed of 1 Mbps, it is not designed to transfer larger amount of data. The technology supports only small packets - from 8 octets up to 27 octets only.

Frequency Band

To reduce interferences with other Bluetooth devices nearby, Bluetooth 'hops' a frequency of 2MHz per step, while 1MHz for Bluetooth versions below 4.0, using Adaptive Frequency-Hopping. From 2.402GHz to 2.480GHz, Bluetooth 4.0 provides 40 channels and the hopping frequency is up to 1600Hz [20] [21].

Setup

Bluetooth Low Energy has a minimum setup time of 3 ms, which is 2 ms more than NFC for connection initialization.

Range

The maximum range for Bluetooth devices is up to 100 metres. However, we place Bluetooth beacons far closer than this maximum range, for example, 2 to 10 metres, for more accurate position estimation.

Security

Bluetooth chips encrypt data using AES-128 before sending it. The encryption mode uses CCM for authentication before encryption to provide extra security. However, we do not have to take care of the encryption process for now because the data we are

sending are not highly sensitive - publicly available information like UUID, major and minor.

Topology

Bluetooth Low Energy adopts a star topology for connections. This is possible because each packet for each connected node uses a 32-bit address, which allows a large number of Bluetooth devices communication with the central hub. Again, this is out of our scope of our project for now because there are far more slots for connections than we need.

Power consumption

The new Bluetooth specification used in BLE uses less power than conventional Bluetooth - the official website states that one coin battery (for example silver oxide battery used in watches and calculators) can be used for at least one year. The type of battery used, and thus the lifetime, still depends on the model of beacon.

Bluetooth stack

Apart from the conventional Bluetooth stack used in versions prior to version 4.0, BLE introduces a new low energy stack which allows manufacturers and developers explore new functions using BLE.

Services and Characteristics

Services and Characteristics are both defined in the UUID, Universally Unique Identifier. UUID can have a length of 16-bit or another custom length, as specified by the manufacturer. Services and Characteristics are just substrings of UUID. For example, the Heart Rate Service is assigned with the UUID with substring of 0x180D; Glucose

Service with 0x1808. Through reading these values of services and characteristics in UUID, client can understand what the Bluetooth device does.

To be concise, characteristics are abstract classes to be implemented in the beacon hardware and services are sets of characteristic(s).

Descriptors

To allow debugging easier, we set short texts, as known as descriptors, to describe what the characteristic(s) do(es). These descriptors, like comments within codes, are optional.

Central and Peripheral Profiles

To connect two devices using BLE, we need one device taking up the central profile and another one taking the peripheral profile. Both central and both peripheral will not work.

To be concise, the central one is the scanner and the peripheral one is the broadcaster.

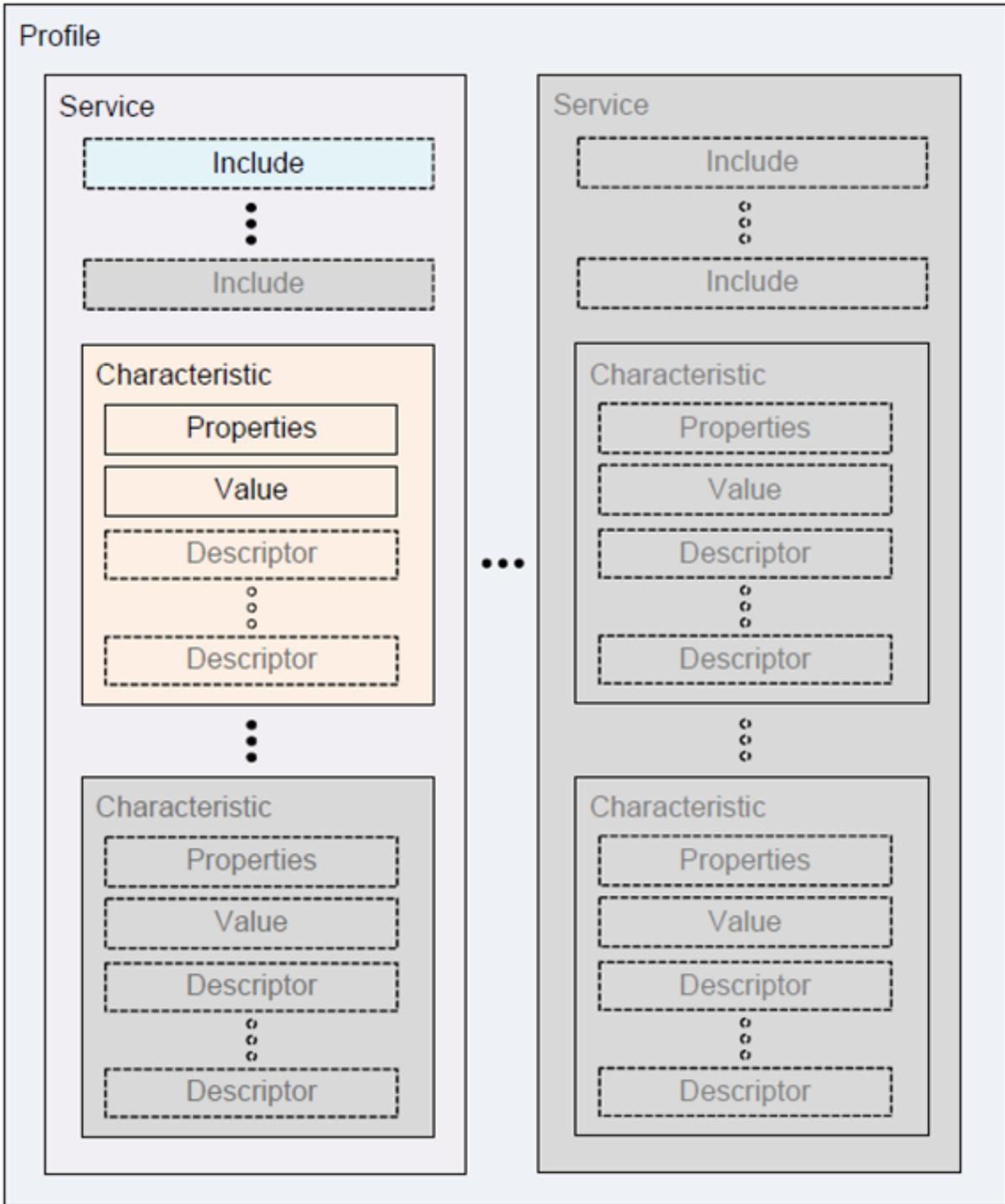
GATT servers and clients

After two devices are connected, they have to choose a role to take up - either a server or client. It is crucial to understand that mobile devices are not always necessarily to be GATT clients and beacons are not always GATT servers - it depends on their communication. The example in the same section above states that the Android device is the client and the beacons are the server. This is true when the beacons are sending their metadata, for example, UUID, major and minor, to the phone. In another case, an Android phone can also act as a server - the phone sends alerts to a BLE smartwatch.

To be concise, the functions of BLE devices do not limit them to be servers or clients - their connections between them determine which device takes up the server and client role.

Cost

Cost is one of the major obstacles that slows down the development of indoor positioning by Bluetooth beacons. As of November 2014, one beacon costs at least HKD\$40 (¥30) on Taobao [22]. Comparing to NFC tags, which costs only for one dollar, Bluetooth beacons are far more expensive. It is estimated that the deployment of indoor navigation system in a museum uses at least hundreds of beacons. Not to mention the application development, testing and maintenance cost, the high cost of installation of indoor exhibition navigation systems scares people away from it.



The profile hierarchy

iBeacon

iBeacon is a subset of Bluetooth Low Energy specified by Apple. It works similarly as other ordinary Bluetooth beacons - both emit radio waves and can be connected by mobile phones. However, there are some slight differences between iBeacons and other Bluetooth beacons. For example, their advertising packets (data that the beacons emit) differentiate them from each other.

Modes

The most simple and popular mode that iBeacons use is advertising. The beacons simply broadcast signals to the environment nearby. We call these signals as advertisement. We can use iOS devices running iOS version 7.0 or above to spoof a beacon that runs the advertisement mode. However, in Android, this feature is not available until version 5.0, Lollipop.

Editing an iBeacon

App Store does not allow applications to edit iBeacon details using wildcard UUID (usually specified by manufacturer; one manufacturer, one UUID). In other words, you can only modify the data of an iBeacon using the official application that is made by the manufacturer; you can write your own app to modify the beacon, upload it to GitHub, or your blog, but your application cannot be sold on Apple's App Store.

Advertising packet

Advertising packet refers to part of the data that is broadcast by a beacon. As mentioned in Introduction of this section, iBeacon has a slightly different specification from other BLE devices, as defined by Apple. The content of the advertising packet of an iBeacon is 28 bytes long. The fields are listed in the following table.

Byte offset	Field content	Remarks
0	AD length	Must be 0x1A
1	AD type	Manufacturer-specific
2-3	Company Identifier	0x004C for Apple
4-5	Beacon Code	Must be 0x0215, which represents iBeacon advertisement
6-21	UUID	Usually same for all beacons owned by the same organization
22-23	Major	Used to distinguish a larger area, e.g. buildings or floor
24-25	Minor	Used to distinguish beacons in an area
26-27	TxPower	2's complement of calibrated TxPower

Different beacon manufacturers may use different number of bytes in a beacon signal to send data, for example, a hyperlink in a beacon signal. To let our application to decode the beacon signal properly, we have to tell our application which field(s) in the beacon signal stand(s) for what meaning. For example, a beacon that is produced by manufacturer A uses fields #3, 4 and 5 in the beacon to transmit hyperlinks. To decode the hyperlink that is sent by the beacon, we have to extract data from fields #3, 4 and 5;

not #6 or 7, not other fields. If we specify the layout wrongly (e.g. using fields #5, 6 and 7 to decode the hyperlink), we will experience results similar to Mojibake (e.g. using Big5-HKSCS to decode Unicode text), as a result, seeing scrambled data.

The problem of fragmentation of these beacon layouts here arises - different beacon manufacturers are very probably use their own standards, and different beacon layouts. Deploying beacons in an exhibition hall with beacons that are produced by different manufacturers, it is very likely that the cost of maintenance and installation is higher than only using beacons by one single manufacturer, as the beacons behave differently. It is suggested that to use one brand of beacon only when deploying in reality.

The Android Beacon Library supports scanning for beacons with different packet layout. To add capability for scanning iBeacons, the layout of iBeacon should be added with the following expression:

```
m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24
```

TxPower, RSSI

TxPower is the standard level of power emission by a Bluetooth beacon exactly 1 metre apart from the point of detection. Like other properties of a beacon, TxPower can be calibrated (modified) by measuring the signal strength of a beacon.

iBeacons broadcast its TxPower along with its advertising packets. By decoding the advertising packets that iBeacons emit, we can easily obtain the standard power level of that beacon. By analyzing the measured signal strength and obtaining the TxPower of a beacon, we can estimate the distance between the user and the beacon.

Working mechanism

Unlike GPS, which broadcasts a timestamp, iBeacons only broadcast its ID (the UUID, major and minor triplets). We cannot determine one's location solely by the IDs emitted by beacons, mathematics should be involved to get a user's location. In other words, at least 3 beacons have to be deployed in an area to find out the location of a user that is in. After getting the signal strength of each beacon, we can use the preset coordinates of each of the beacons to calculate the position by the signal strengths of beacon. Details will be discussed in the section of Trilateration.

Android

Starting from API 18 (version 4.3), Android supports Bluetooth Low Energy. However, Android only supports Central and Peripheral profiles as of mid-November 2014.

Android devices hold a Central profile when they are continuously looking for beacons nearby [23].

Permissions

Like other hardware sensors or communication parts, for instance, NFC, vibrator and GPS, using BLE requires declaration of permission in the Android Manifest XML file.

Apart from the declaration of usage of Bluetooth, declaring the use of Bluetooth Admin is also required and the declarations should be the child elements of manifest, in other words, the permission declarations should be written between the manifest tags. Code snippets are displayed below:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission
android:name="android.permission.BLUETOOTH_ADMIN" />
```

If everything is successful, the permissions required by the application should show up in the application information page after compilation. After everything is set up in the manifest XML, we can start coding our app.

Setup

Before starting any Bluetooth activities, we can check if Bluetooth Low Energy is available on the phone. We can be sure that BLE is available and turned on by simply adding the `<uses-feature>` tags in the Manifest XML file. If BLE is not supported, we can either quit the application or prevent any Bluetooth scanning actions.

BluetoothAdapter

All Bluetooth actions require a `BluetoothAdapter`, including conventional Bluetooth scans and BLE, no exceptions. As of API 18 (version 4.3), `BluetoothManager` is proposed. We can get an instance by the following codes:

```
final BluetoothManager myBluetoothManager = (BluetoothManager)
getSystemService(Context.BLUETOOTH_SERVICE);

myBluetoothAdapter = myBluetoothManager.getAdapter();
```

Start Bluetooth

By running `startLeScan()`, we can look for BLE devices nearby. However, we should not infinitely look for beacons as the scan drains battery. Once the target device is located, we should stop scanning and restart only when the user manually request for another scan. We can scan for beacons by adding the following codes to our program:

```

private Handler myHandler;

/* scans for 10000ms */
private static final long SCAN_INTERVAL = 10000;

/* ... */

myHandler.postDelayed(new Runnable() {

    @Override

    public void run() {

        myBluetoothAdapter.stopLeScan(myLeScanCallback);

    }

}, SCAN_INTERVAL);

myBluetoothAdapter.startLeScan(myLeScanCallback);

```

The code above scans for all kinds of BLE devices. To only look up for particular devices, we can add one more parameter, a UUID objects array, to the method - call `startLeScan(UUID[], myLeScanCallback)` **alternatively**.

Connect to a BLE device

Before starting any activities between the phone and BLE devices, we have to connect them first. We can simply use `connectGatt()`, which accepts three variables, to get the job done. The first parameter accepts a Context; the second parameter tells the method to automatically connect once that device is visible; and the last parameter accepts a `BluetoothGattCallback`.

```
myBluetoothGatt = device.connectGatt(this, false,  
myGattCallback);
```

The `BluetoothGattCallback` object provides a wealth of helpful information, for example, GATT actions. By using `onConnectionStateChange()`, we can perform actions once the BLE peripheral is connect or disconnected. Moreover, by overriding `onServiceDiscovered()` and `onCharacteristicsRead()`, the Android device will be able to read available services on the beacon and run specific codes based on beacon characteristics, in other words, functions that the beacon can perform. Functions that a specific type of beacon can perform depend on their profiles. The list of profiles is available on the Bluetooth official website [24].

Read BLE device

Once connected to a BLE device, we can read and write values from and to it. However, the value that can be read and written all depend on the features available on the BLE device. In other words, the values are device-specific and we have to check the specifications on the beacon manufacturer's website for details. For example, using `getUuid()` can read the UUID from the beacon.

```
uuid = gattService.getUuid().toString();
```

Notifications

Notifications here should not be mixed up with the notifications sent when beacons are in range - we use `onCharacteristicChanged()` to keep track if there is any modifications on the characteristics on the BLE device.

Close connection

When all BLE activities are done, simply call `close()` end the Bluetooth scans.

```
myBluetoothGatt.close();
```

Third Party Library

Although the official Android API provides methods to scan for beacons nearby, the API is not easy to use. For example, to scan a beacon, first we have to initiate the `startLeScan` method to start scanning. Next, we have to connect to a GATT Server (a visible beacon). Then, we broadcast updates from the beacon to the app and receive the updates through a listener. Finally, we can read the beacon attributes, such as UUID. Furthermore, to receive notifications from GATT servers, another method must be initiated. With this large amount of work for just obtaining little information from a beacon, we have chosen not to use the official Android API. Instead, we have picked some third-party open source beacon libraries for our application.

The table below shows the similarities and differences of two of the libraries that we have looked into.

	Estimote SDK	Android Beacon Library
Developer	Estimote	Radius Networks
Platform	Android, iOS	Android
Standard	iBeacon	AltBeacon
Types of beacons supports	Estimote iBeacon	Any
Get basic information from beacon (e.g. UUID, major, minor)	Yes	Yes
Get signal strength	RSSI	RSSI and TX Power
Estimate distance	No	Yes (normalized over 20 s interval)
Background scan	Partly, relies on iOS' location services	Yes
Price	Free	Free

From the comparisons above, we have adopted the Android Beacon Library, developed by Radius Networks, to develop our Android application. The Android Beacon Library supports more types of beacons, is able to get signal strength of beacon and allows background scans.

Indoor Positioning

Overview

Determining user's position is an important part of our project. Without the location, it is impossible to determine what exhibits is the user close to. Under the iBeacon standard, the TxPower of a beacon is broadcast within the advertising packet. The TxPower of an iBeacon is defined as the received signal strength indicator (RSSI, measured in dB) measured at a distance of 1 m. By comparing the strength of the iBeacon signal received by the phone with the TxPower of the iBeacon, it is possible to estimate the distance from the beacon to the receiver.

Traditionally, in an indoor exhibition navigation system, each object of interest (an exhibit, a product shelf etc.) is associated with one beacon. The object that the user is closest to can simply be determined by picking the detected iBeacon with the strongest signal strength. This deployment scheme is programmatically simple but inflexible.

Whenever the layout of the exhibition changes, the location of the beacons must also be moved. Next, the exhibits database must be altered. Moreover, as the number of exhibits increases, the number of beacons to be purchased also increases.

To enhance flexibility and reduce deployment cost, it is suggested to deploy a fixed grid of beacons in the room and locate users mathematically. We have conducted several tests to investigate how this can be achieved. First, we attempted to use the distances from three nearby beacons to perform trilateration. As the result was unsatisfactory, we switched to a cell-based positioning method. To understand the best deployment strategy, we also did an experiment to analyse the characteristics of the beacon signals. The details of these tests are discussed in the following sections.

Deployment and Calibration

Before any of the tests, we first install the beacons at their designated places. Then, the indoor relative 2D coordinates of each beacon is measured with a rule and entered in our database. We used a Samsung Galaxy Note 3 Android phone to perform all the tests. When the phone detects a beacon, the indoor coordinates of it will be retrieved from the database for subsequent calculations.

After each beacon is deployed, its TxPower is calibrated using the test phone. The phone is placed stably at 1 metre distance from the beacon. Using the iBeacon Locate app developed by Radius Networks, the average RSSI of the beacon at 1 m over a 30 seconds interval can be measured. The beacon then has its TxPower value changed to the measured average RSSI using the eBeacon app, and restarted. This calibration process caters for the differences among individual beacons, the deployment environment of each beacon, and the differences among the receivers (the phone), which are not the factors that our tests shall look into.

Trilateration

Trilateration refers to positioning using distances from known points. In the 2D case, it requires at least three beacons to estimate one's location. However, the major difference from triangulation is that trilateration does not require angles to be calculated.

First, we measure the RSSI of all the visible beacons nearby. The RSSI of each beacon is temporarily stored in the mobile device's memory. Only the three beacons with the greatest (less negative) RSSI, in other words, the three nearest beacons, are used for calculations; while the information of all other visible beacons are discarded. We only use three sets of data from the three nearest beacons because we believe the interference and noise by the background acting on the beacons are less severe than farther ones. Using their calibrated signal strengths and hence the estimated distance to each beacon can be determined. We use the three estimated distances as the three radii of the three chosen beacons. Theoretically, beacons are point sources of signals. If there is no or only little interferences or noises, the signals should be transmitted radially outwards as a sphere. By calculating the intersection point of these spheres, the location can be calculated.

We have tested this method in a 12 m × 6 m room with some large obstacles imitating the real environment of a small exhibition room. Four beacons were installed on the roof in a rectangular shape to reduce the blocking effect of obstacles. The beacons were 3.6 m to 7.2 m apart. Since all beacons are deployed at the same height, the problem can be reduced to 2D by taking care of the height difference between the receiver and the roof. Let r_i , h_i and k_i be the height-adjusted distances from the phone to the i^{th} beacon,

the x-coordinate and y-coordinate of the beacons respectively. The user location (x,y) can be found by solving the following linear system:

$$\begin{cases} (x - h_1)^2 + (y - k_1)^2 = r_1^2 \\ (x - h_2)^2 + (y - k_2)^2 = r_2^2 \\ (x - h_3)^2 + (y - k_3)^2 = r_3^2 \end{cases}$$

Rearranging the equations, a linear system of two unknowns can be obtained.

$$\begin{cases} (-2h_1 - 2h_2)x + (-2k_1 - 2k_2)y + h_1^2 - h_2^2 + k_1^2 - k_2^2 = r_1^2 - r_2^2 \\ (-2h_2 - 2h_3)x + (-2k_2 - 2k_3)y + h_2^2 - h_3^2 + k_2^2 - k_3^2 = r_2^2 - r_3^2 \end{cases}$$

In ideal situation, the three measurements should be accurate and intersect at a single point, which is the location needed. Yet, in reality, there are lots of obstacles in the exhibition hall blocking radio signals emitted by the beacons. The mobile device might over-estimate the distance to the beacon due to reduced strength of radio waves received. The calculation formula can still give an estimated result even if the measurements are inconsistent.

The testing result indicates indoor trilateration using iBeacons is unsatisfactory. The fluctuation of location is huge. The signal of each beacon fluctuates greatly. We believe this is caused by the reflection of radio waves by the large objects, thus the beacon signals travel in multiple paths. The effect becomes very significant when there are large exhibits being placed in a comparably small room. When the fluctuation of three beacons adds up, the location is still extremely unstable.

As a remedy, we have tried using normalized distances, averaged over a short interval, to stabilize the signal. Further, we have used the signals from all four beacons to calculate four sets of coordinates and take its average. The results showed no improvement. Whenever the signal of one beacon fluctuates, the calculated location

also fluctuates in all directions. We thus concluded that trilateration using iBeacons is impractical.

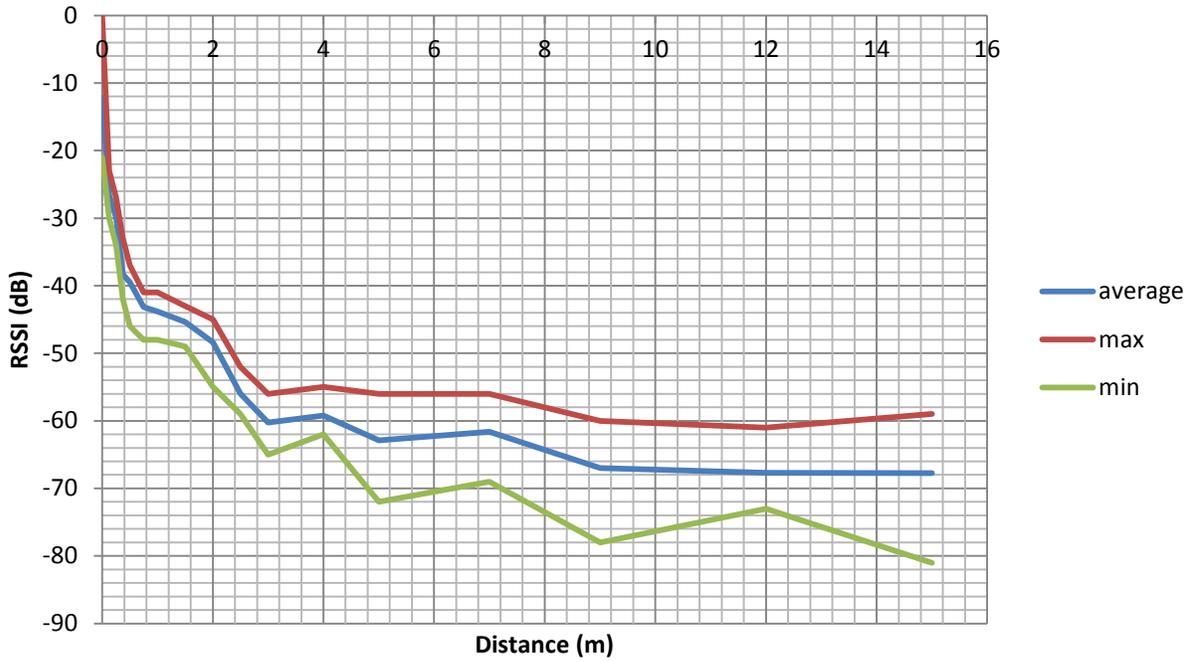
Quality of Measured RSSI

In Apple's implementation of BLE, any detected beacon only has its proximity available. The proximity has three possible values: Immediate, Near and Far. This design is probably due to the reliability of beacon signal and limited resolution in beacon distance. To evaluate the trustworthiness of the measured beacon signal strength, an experiment has been done to measure the fluctuation in RSSI measured by the phone.

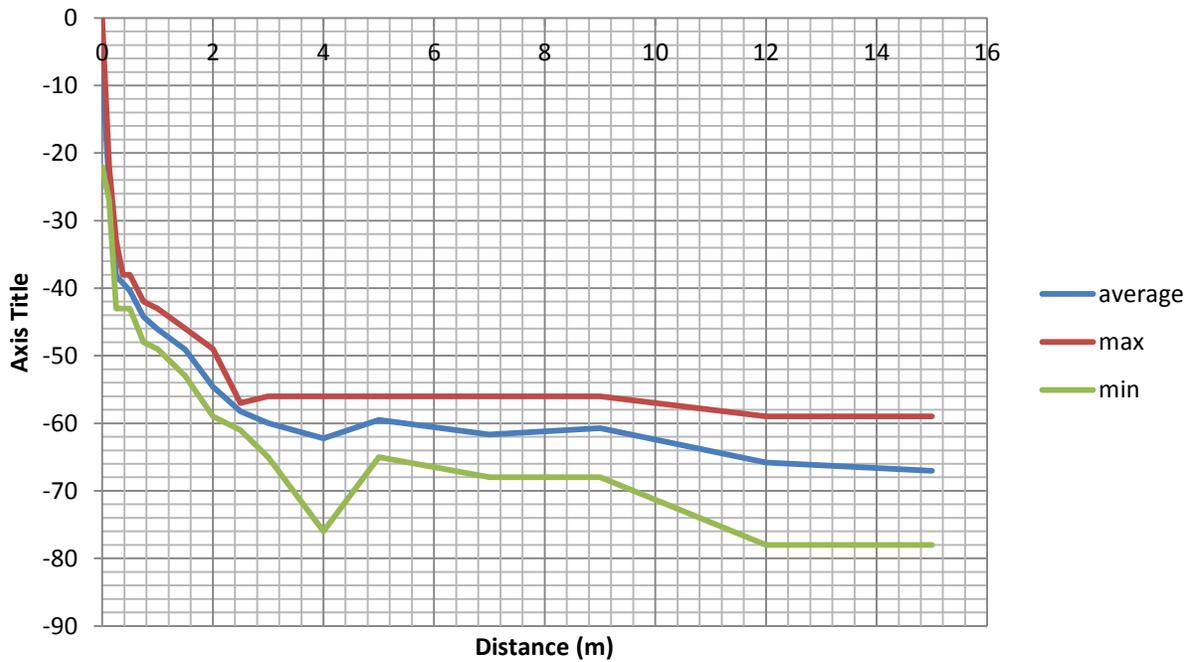
We used the same test phone, Samsung Galaxy Note 2, to measure the RSSI of beacons. A custom mobile application was developed to scan for beacons and report the ID and RSSI of the beacons to a web page, which serves as a data logger. The test beacons were deployed on a wall in an open area and calibrated. Recording of RSSI was done at various points. At each point, 100 continuous samples of RSSI were taken. Yet, only the middle 60 samples were used for analysis to make sure the signal is stable. Two beacons of the same model, manufactured by Bytereal, will be used for the experiments.

The experiment involves measurement of two beacons at various distances. The experiment aims at measuring the fluctuation in RSSI and average RSSI, thus determine the range of distances that can be more reliably determined using measured RSSI. The measurements were taken at 0m, 0.125m, 0.25m, 0.375m, 0.5m, 0.75m, 1m, 1.5m, 2m, 2.5m, 3m, 4m, 5m, 7m, 9m, 12m and 15m apart from the beacon. The following graphs indicate the results of the measurement.

Range of RSSI against Distance of Beacon #1



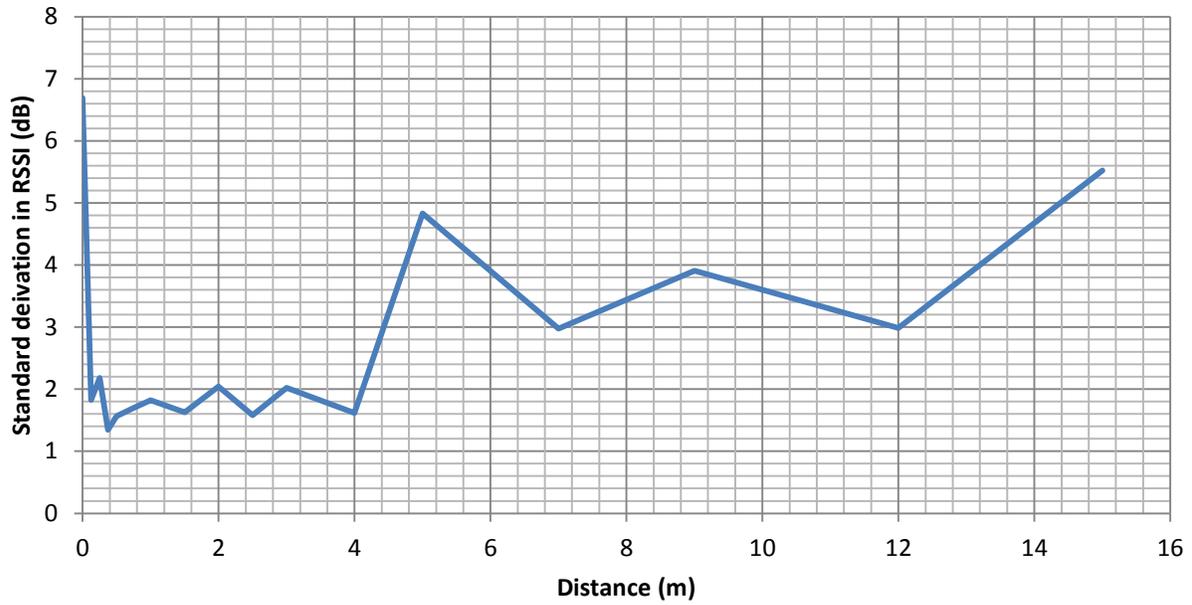
Range of RSSI against Distance of Beacon #2



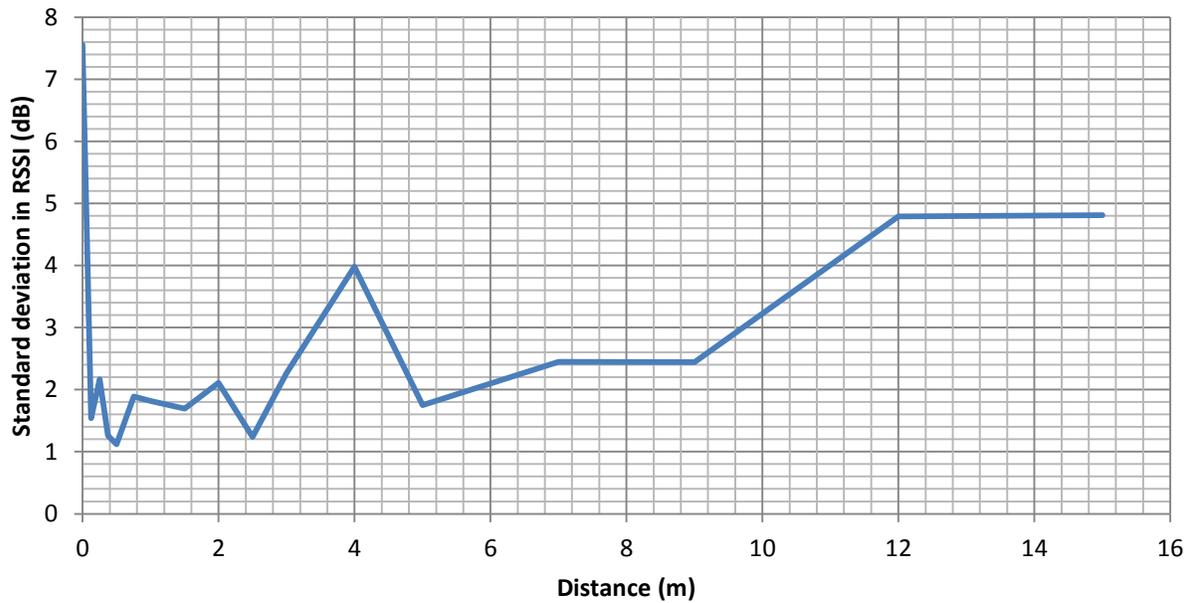
From the above graphs, it can be seen that the average, maximum and minimum RSSI decreases with increasing distance. Yet, the drop trend flats out after 4m. The result is correct because RSSI is measured in logarithm scale. We believe the minor rises in the graph are due to multipath reflections and interferences in signal because of the environment.

If we take another approach, given the measured RSSI, can we confidently determine the beacon's distance from the phone? From the graph, it can be seen that the maximum measured RSSI start to flat out at 2.5m to 3m. Given this flat out value maximum, the range of possible distances can be as close as 2m to as far as 10m, which is meaningless. As a result, we believe that 2m to 2.5m is the maximum distance that can be ranged using iBeacons.

Standard Deviation in RSSI against Distance of Beacon #1



Standard Deviation in RSSI against Distance of Beacon #2



The standard deviations in RSSI have also been calculated. It can be seen that the standard deviation is highest when the beacon is very close to the phone, then reaches its minimum between 0.5m to 2.5m, and rises again. With the small fluctuation in RSSI at the interval, the stability of the measured distance can also be guaranteed.

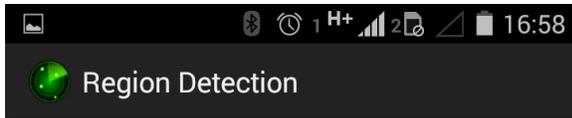
We believe that the initial rise is due to the logarithm nature of RSSI and serious multipath reflections at the deployment point. A slight change in travelled distance can cause a big change in RSSI when the distance is short. Later, the deviation reaches its minimum because of minimal reflections at the distance. The choice of TxPower calibration at 1 metre is proved reasonable. After that, multipath reflections and interferences from the surroundings put the deviation on the rising trend again.

Cell-based Region Detection

After the previous tests above, we can conclude that Bluetooth beacons are never suitable for use in fine location determination - the electric current is often unstable, causing the output power fluctuate, and signals are often blocked, multiplied and absorbed by objects around them. We have slightly changed our approach to exhibition location determination - from exact coordinates to cell-based regions. In other words, we have sliced the exhibition hall various rectangular regions. Beacons are then placed on the wall of each side of the exhibition room and the method of beacon detection is same as above.

Instead of calculating the coordinates of the user, an estimated distance from the beacon is used to make guesses of regions in which the user is located. The guess is made independently for each region, involving 1 to 2 beacons. Sometimes, the power levels received by the mobile device by two beacons might be very similar. It is possible that there are multiple guessed regions. In that case, the conflicts will be solved algorithmically - by taking past history of guesses into account.

In the test scenario, four beacons were deployed at a 3m x 3m area. Each beacon is installed in the middle of the four sides, at coordinates (0, 1.5), (3, 1.5), (1.5, 0), (1.5, 3) respectively. The 3m x 3m area was divided into 9 square regions of 1m x 1m. In the four regions where the beacons are located, whether the user is inside the region is determined using one beacon. In the other regions, it is determined by two beacons.



Start logging

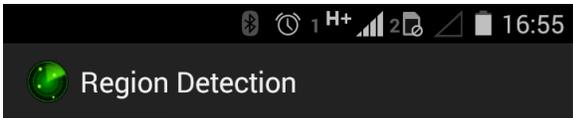
Region detected:

region 1	region 2	region 3
region 4	region 5	region 6
region 7	region 8	region 9

1,3,2,1
0,0,0,0,0,0,0,2,8



Demonstration of region detection



Start logging

Region detected:

region 1	region 2	region 3
region 4	region 5	region 6
region 7	region 8	region 9

2,3,2,0
0,0,0,2,0,0,8,0,0,0



Demonstration of region detection

By using this method of location determination, we have found that the degree of fluctuation of location is significantly lowered when compared to the previous two experiments. When the user is at the middle of a region, the detected region is stable. This method is safe enough to be deployed in use in production sites. Although the detection of region might hop back and forth at region boundaries, it can be solved using more sophisticated algorithms. For example, we can apply tougher restrictions in switching of guessed regions.

Although the position resolution using the region detection method is low, we believe that it can be enhanced by integrating with the traditional one-object-one-beacon system. In an exhibition, a fixed grid of beacons can be deployed for detection of region. For a finer location, extra beacons can be installed alongside each exhibit. Through this two-level deployment scheme, the advantages of both deployment methods can be capitalized.

Design

The whole beacon system involves two parts, the Web CMS, API and an Android application. The CMS is used for adding, editing, removing and previewing beacon details; while the API is for getting beacon data from database quickly; and the Android application is the interface for exhibition visitors to view the extra information of the exhibits.

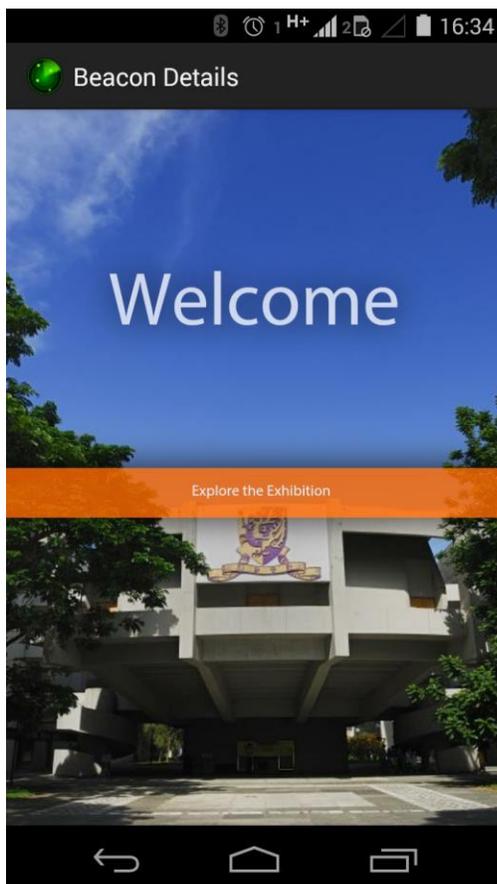
Android

Our application is designed to be as easy to use as possible. No or only little user intervention is required to operate the mobile application.

Interface to be shown to visitors

Welcome screen

A welcome screen is displayed when the user first enters the exhibition. The screen provides contains a full screen background image that can be changed according to different exhibitions. Beneath the welcome message, a button that links to the exhibition main page is shown. Users can tap on the link to learn more about the exhibition. Here in this phase, map data and exhibits are automatically downloaded in background.



Welcome splash screen

Scanning for beacons

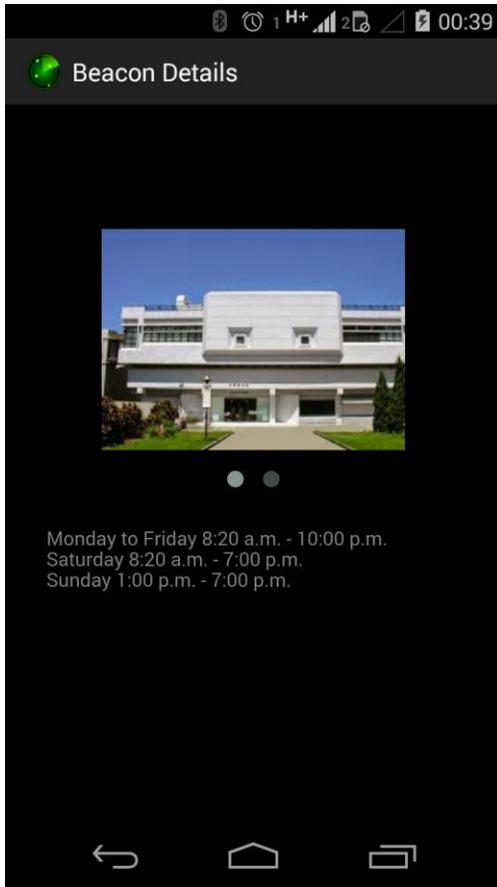
We understand that not all visitors of the exhibition are experts in Bluetooth technology, so, we have to make everything as simple as possible by reducing user interactions with the controls on the phone. Therefore, no button is required to be tapped when scanning for beacons - the procedure of scanning for beacons is automatically done in background.

Exhibits / exhibition information page

After a series of calculations of signal strengths emitted by Bluetooth beacons, we use our own API to obtain data from the database that stores beacon, exhibit or location information. A page of the location will then be chosen to be displayed on screen. In this example, we have New Asia Ch'ien Mu Library. When the user taps on the photo, the whole page goes black, leaving the image carousel visible, making image preview easier.



An example of results page



Background in black for easier image preview

Interface to be shown for internal testing

Our application also has a few pages for calibration and testing of the Bluetooth

beacons. For example, testing of beacon regions, displaying the number and details of visible beacons and specific beacon details, including its coordinates and response from server using the information of that beacon.

The image below shows the details of which region that the user is in and allows users to track log of the data transmitted by a beacon. For instance, RSSI (power level) detected, time, UUID, major and minor.



Region detected:

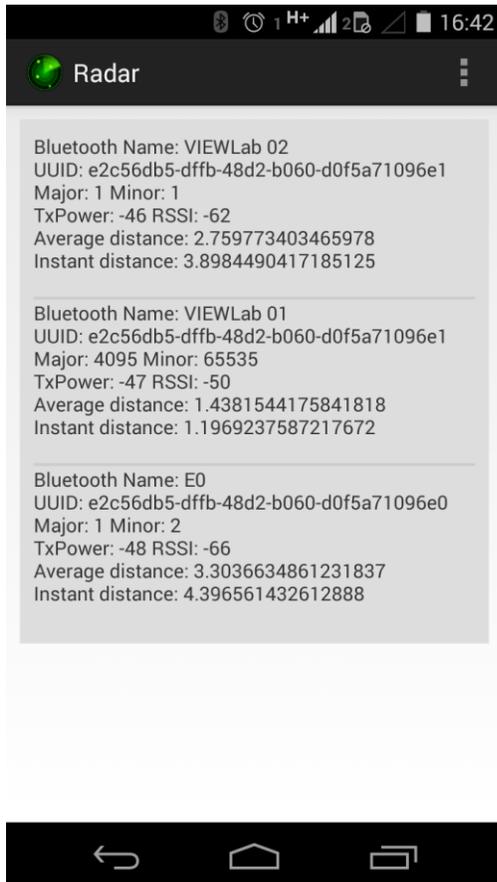
region 1	region 2	region 3
region 4	region 5	region 6
region 7	region 8	region 9

2,3,1,1
0,0,0,9,0,6,0,0,0,5



[An example of region determination](#)

The image below shows a list of visible beacons, along with their details (IDs), for example, UUID, major, minor, RSSI (power level), TxPower (standard power level measured 1 metre apart), name, estimated distance and average distance.



[A list of scanned beacons](#)

Web

Database

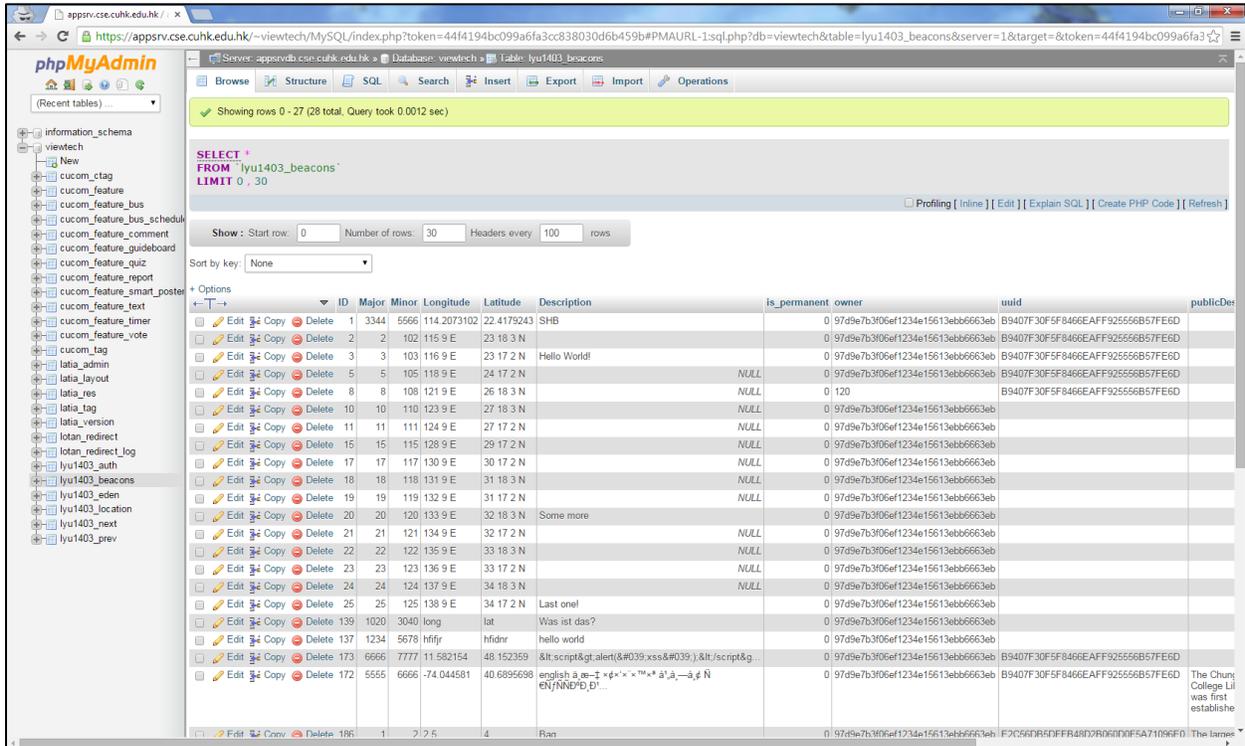
This application is for location estimation in exhibitions and those who hold the exhibition might not be professionals in the field of Bluetooth technology. We have to find a good solution to make setting up and updating exhibits information easy enough to use. Furthermore, in order keep the information in the application up-to-date, we believe maintaining the content inside the mobile application to be dynamic instead of static would reduce the cost of each update and ensure visitors will not receive outdated information if they refuse to update the application.

A MySQL database has been deployed for this purpose. We have created a few tables inside the database to store information of each beacon. For example, the location in term of indoor coordinates relative to the room, IDs, exhibit data like titles and descriptions to be shown to the public.

The screen capture below shows a screen capture of our database storing beacon information, including the ID, UUID, major, minor and its description.

CMS

However, updating the database through command line or phpMyAdmin might still be too difficult for people who work in museums or supermarkets as they are not IT professionals. Therefore, we have created an easy-to-use online CMS to easily add, edit, delete and preview the beacon settings.



Our database storing beacon details

Dashboard page

Dashboard Add Edit Remove Preview

Beacon info

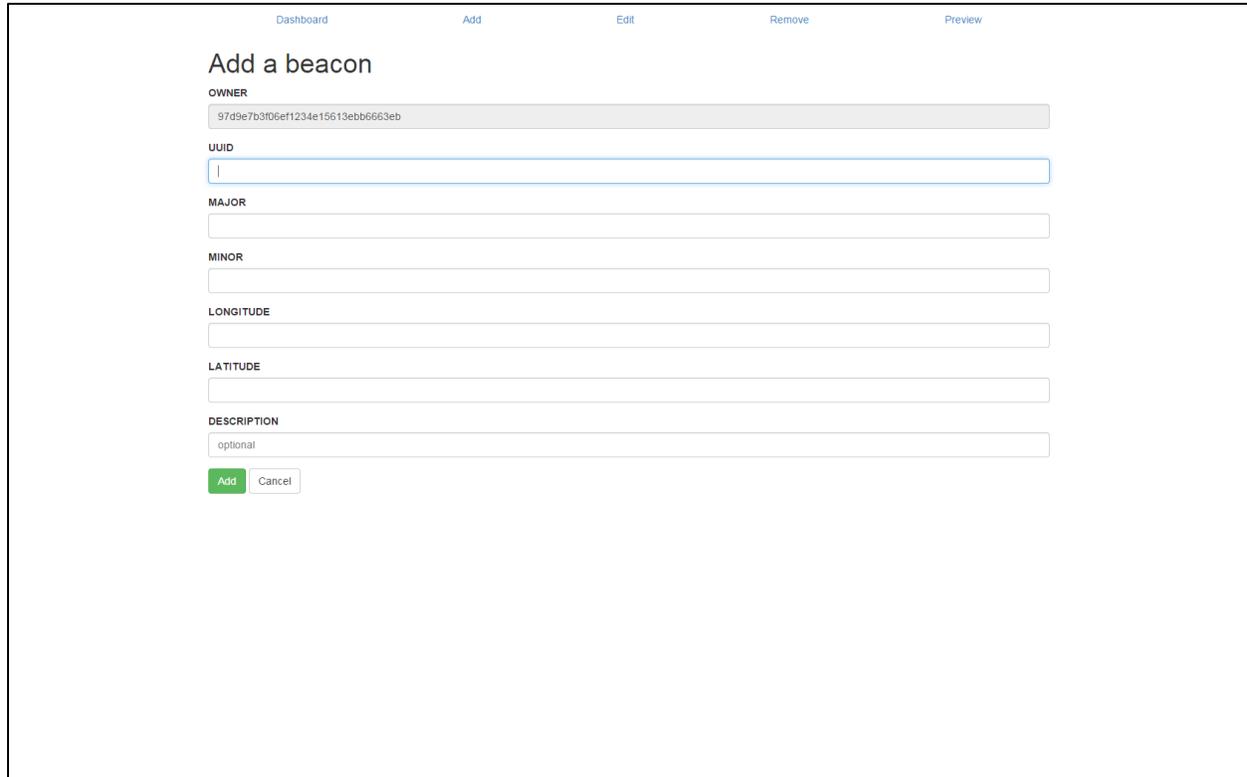
All beacons

ID	UUID	MAJOR	MINOR	LONGITUDE	LATITUDE	DESCRIPTION
1	B9407F30F5F8466EAF925556B57FE6D	3344	5566	114.2073102	22.4179243	SHB
2	B9407F30F5F8466EAF925556B57FE6D	2	102	115 9 E	23 18 3 N	
3	B9407F30F5F8466EAF925556B57FE6D	3	103	116 9 E	23 17 2 N	Hello World!
5	B9407F30F5F8466EAF925556B57FE6D	5	105	118 9 E	24 17 2 N	
8	B9407F30F5F8466EAF925556B57FE6D	8	108	121 9 E	26 18 3 N	
10		10	110	123 9 E	27 18 3 N	
11		11	111	124 9 E	27 17 2 N	
15		15	115	128 9 E	29 17 2 N	
17		17	117	130 9 E	30 17 2 N	
18		18	118	131 9 E	31 18 3 N	
19		19	119	132 9 E	31 17 2 N	
20		20	120	133 9 E	32 18 3 N	Some more
21		21	121	134 9 E	32 17 2 N	
22		22	122	135 9 E	33 18 3 N	
23		23	123	136 9 E	33 17 2 N	
24		24	124	137 9 E	34 18 3 N	
25		25	125	138 9 E	34 17 2 N	Last one!
137		1234	5678	hlfjgr	hlfidnr	hello world
139		1020	3040	long	lat	Was lst das?
172	B9407F30F5F8466EAF925556B57FE6D	5555	6666	-74.044581	40.6895698	english 中文 עברית מנדו русский 日本語
173	B9407F30F5F8466EAF925556B57FE6D	6666	7777	11.582154	48.152359	<script>alert('xss');</script>
186	E2C56DB5DFFB48D2B060D0F5A71096E0	1	2	2.5	4	Bag
187	E2C56DB5DFFB48D2B060D0F5A71096E2	1	1	1	2.5	MATSUSHO retract

Dashboard page showing a list of beacons

Adding a beacon

User can add the UUID, major, minor, coordinates in terms of latitude and longitude and descriptions. A success message will be displayed if the insertion is successful.

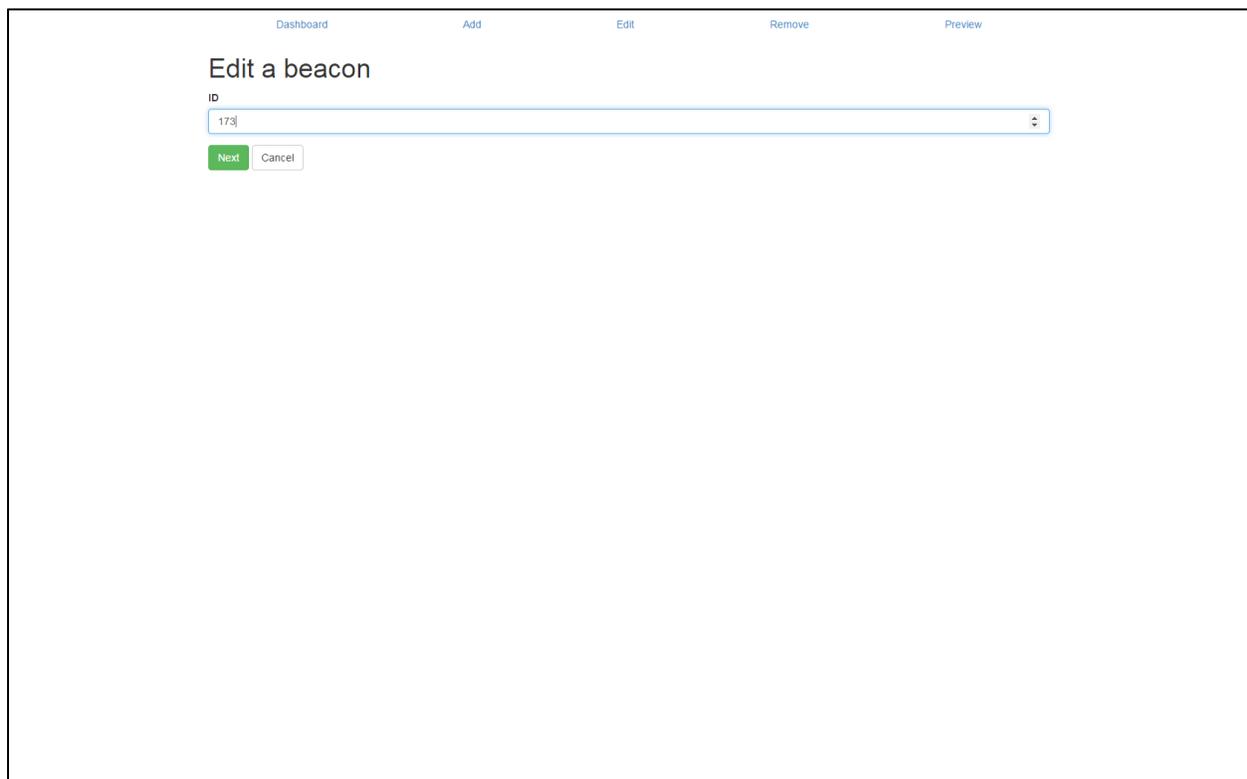


The screenshot shows a web interface for adding a beacon. At the top, there is a navigation bar with links for 'Dashboard', 'Add', 'Edit', 'Remove', and 'Preview'. The main heading is 'Add a beacon'. Below this, there are several input fields: 'OWNER' (pre-filled with '97d9e7b3f06ef1234e15613ebb6663eb'), 'UUID', 'MAJOR', 'MINOR', 'LONGITUDE', 'LATITUDE', and 'DESCRIPTION' (pre-filled with 'optional'). At the bottom, there are two buttons: 'Add' (green) and 'Cancel' (white).

[Page for adding a beacon](#)

Editing a beacon

User can modify a beacon information by entering the beacon ID. After that, a page similar to the adding a beacon is displayed. This page shows the current value of each field of a beacon. If no changes have to be made, just click the cancel button to exit from this page or just click the edit button but without changing the values of each field. A success message will be displayed if the editing is successful.



The screenshot shows a web interface for editing a beacon. At the top, there is a navigation bar with five links: "Dashboard", "Add", "Edit", "Remove", and "Preview". Below the navigation bar, the main heading is "Edit a beacon". Underneath the heading, there is a label "ID" followed by a text input field containing the value "173". Below the input field, there are two buttons: a green "Next" button and a white "Cancel" button.

[Edit a beacon using its ID](#)

Edit a beacon

OWNER

97d9e7b3f06ef1234e15613ebb6663eb

ID

173

UUID

B9407F30F5F846EAF92556B57FE6D

MAJOR

6666

MINOR

7777

LATITUDE

48.152359

LONGITUDE

11.582154

DESCRIPTION

<script>alert('xss');</script>

PUBLIC DESCRIPTION

optional

PUBLIC TITLE

optional

PUBLIC SUBTITLE

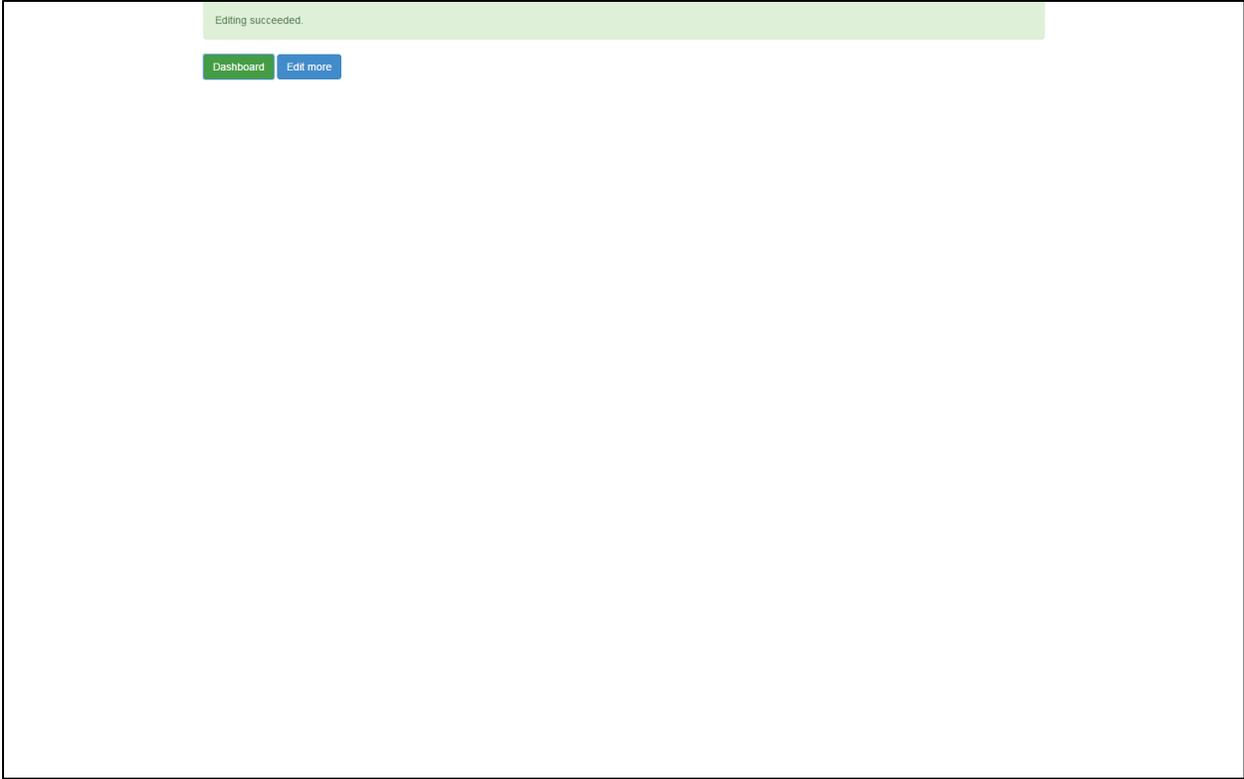
optional

EDIT

DELETE

CANCEL

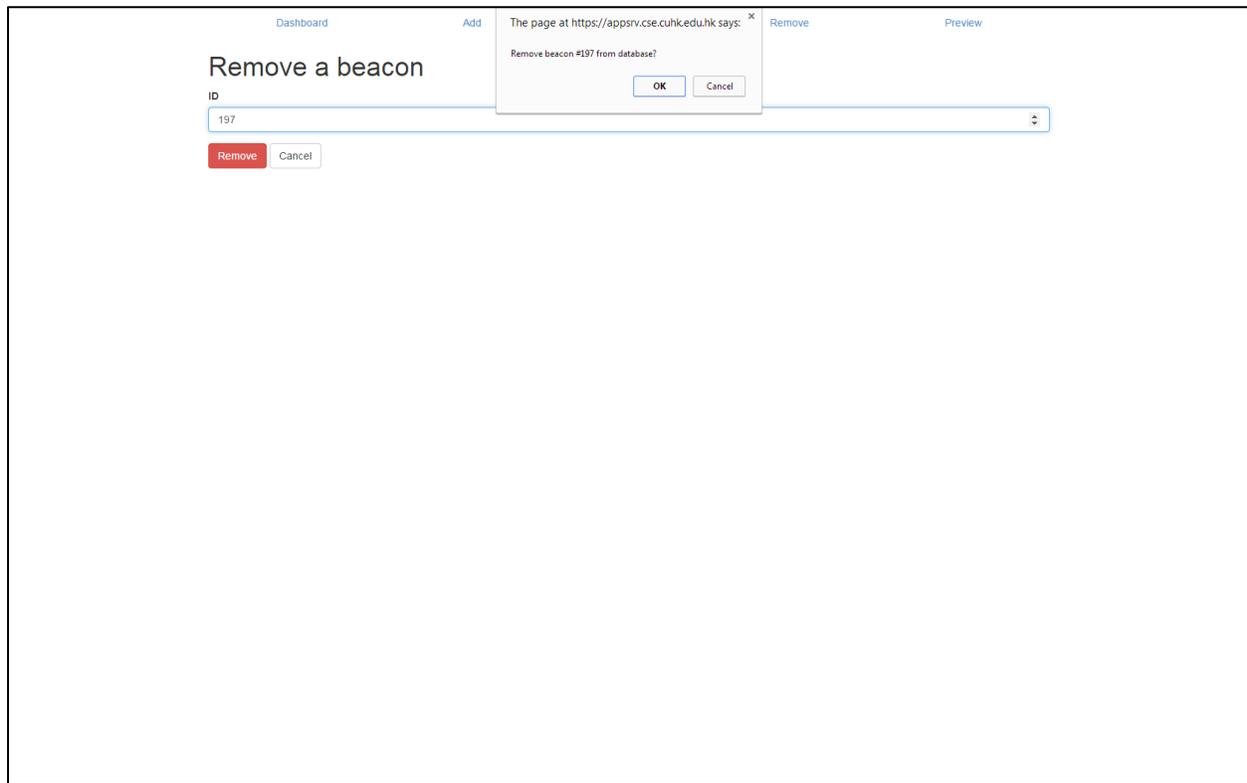
[Edit beacon details](#)



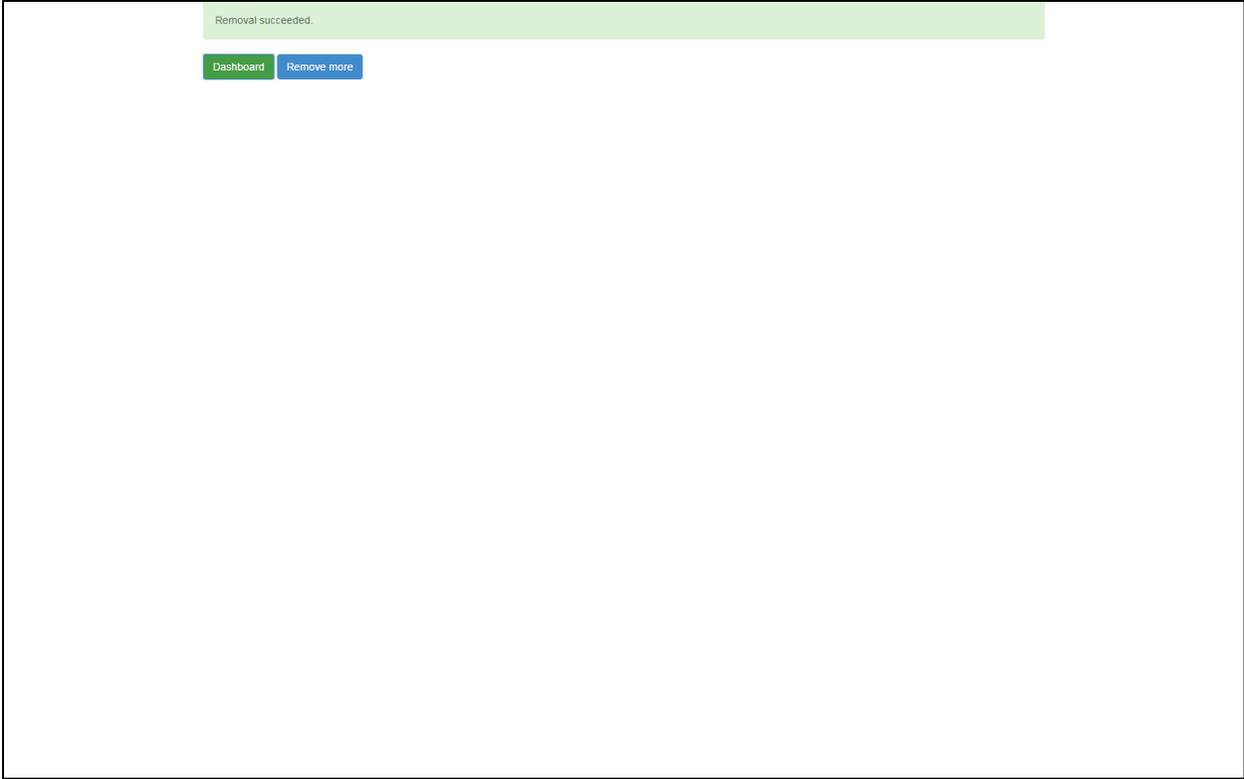
Success message shown after editing a beacon

Removing a beacon

User can remove a beacon information by entering the beacon ID. After that, popup message for removal confirmation is displayed. To cancel the deletion, simple click the cancel button to return to the page. Otherwise, click OK to continue to the deletion. A success message will be displayed if the removal is successful.



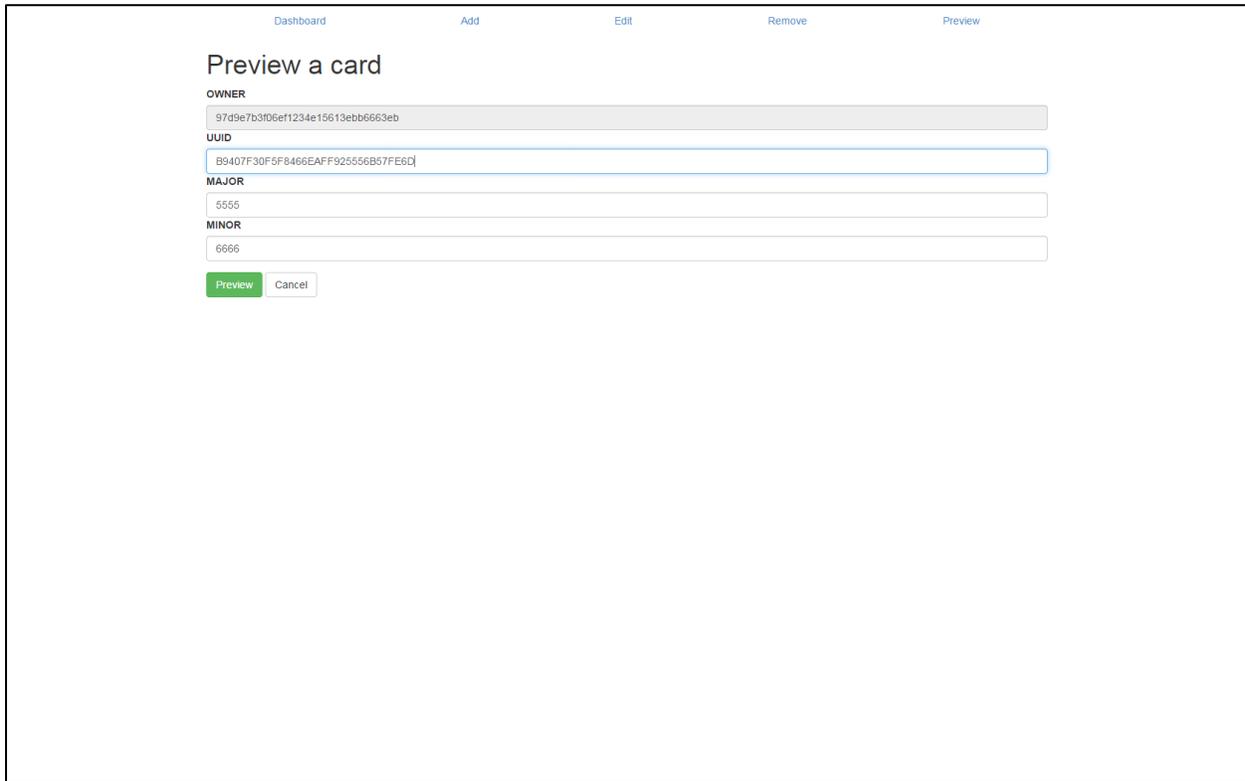
[Remove a beacon using its ID](#)



Success message shown after removing a beacon

Previewing a beacon

To see the beacon details on the Web without using the phone, click the preview button on the top bar. the preview page will then show up. By entering the beacon details, including UUID, major and minor, a preview page of the beacon will appear. This page should be identical when viewed on the phone's application.



The screenshot shows a web application interface with a top navigation bar containing the following links: Dashboard, Add, Edit, Remove, and Preview. The main content area is titled "Preview a card" and contains a form with the following fields:

- OWNER**: A text input field containing the value "97d9e7b3f06ef1234e15613ebb6663eb".
- UUID**: A text input field containing the value "B9407F30F5F8466EAF925556B57FE6D".
- MAJOR**: A text input field containing the value "5555".
- MINOR**: A text input field containing the value "6666".

At the bottom of the form, there are two buttons: a green "Preview" button and a white "Cancel" button.

[Preview a beacon details using its details](#)

Beacon Manager x

← → ↻ <https://appsrv.cse.cuhk.edu.hk/~phngz> ☆ ☰

The Chung Chi College Elisabeth Luce Moore Library



Monday - Friday 8:20 a.m. - 10:00 p.m.
Saturday 8:20 a.m. - 7:00 p.m.
Sunday 1:00 p.m. - 7:00 p.m.

Description

The Chung Chi College Library was first established in 1951. The present library building was built with the generous support of the Henry Luce Foundation through the United Presbyterian Church in the United States of America and was officially dedicated on October 29, 1971 by Mr. Henry Luce's sister, Mrs. Elisabeth Luce Moore, after whom the Library was named.

Radar © 2014

[Preview a beacon in browser, as in the mobile application](#)

REST API

To get response, like calculations of locations and beacon information, from the server easier, we have written our own REST API from scratch.

The screenshot shows a web browser window displaying the 'RADAR REST API DOCUMENTATION' page. The browser's address bar shows the URL: <https://appsrv.cse.cuhk.edu.hk/~phng2/fyp/api/docs/>. The page has a dark blue header with the title 'RADAR REST API DOCUMENTATION' in white. On the left, there is a sidebar with three main sections: 'Important notes' (containing links for 'GET notes' and 'GET API key'), 'Methods' (listing various API endpoints like 'GET getDescription', 'GET getLocationById', etc.), and 'About' (with links for 'GET ourDetails' and 'POST comments'). The main content area is white and contains several sections: 'GET notes', 'HTTP request method' (noting that all API calls use GET), 'Redirection to HTTPS' (explaining automatic redirection to encrypted channels), 'Return type' (stating that all API calls return JSON objects), 'Return format' (referencing JSend's page for details), 'Format' (showing a JSON object structure: `{"status": "status_of_call", "data": {returned_data}}`), and 'Example' (showing a specific JSON response: `{"status": "OK", "data": {"latitude": "22.4179249", "longitude": "114.2073102"}}`). Below these sections, there is a 'GET API key' section with a 'Request an API key' button and a 'New!' badge indicating a demo key: `97d5e7b3f06ef1234e15613ebb6663eb`. At the bottom of the main content area, there is a 'Methods' section with a link for 'GET getDescription'.

[Our API documentation page](#)

The API uses the GET method for requests to ease the deployment and usage. No matter the query result is successful or not, for instance, failures due to invalid parameters, the API can still return a message to users to notify them the results.

The screenshot shows a web browser window displaying the Radar REST API documentation. The page is titled "Radar REST API Documents" and the URL is "https://appsrv.cse.cuhk.edu.hk/~phng2/fyp/api/docs/". The main content area is divided into two sections, each for a different API endpoint.

GET getLocationById

Description
Returns latitude and longitude of a beacon.

Parameters

key	Your API key. It looks like this: 97d9e7b3f06ef1234e15613ebb6663eb
request	Use getLocationById to use this API call.
id	The ID of your beacon (generated by us, not UUID). It looks like this: 1

Example request

```
GET https://appsrv.cse.cuhk.edu.hk/~phng2/fyp/api/?key=97d9e7b3f06ef1234e15613ebb6663eb&request=getLocationById&id=1
```

Example result

Successful result

```
{"status": "OK", "data": {"latitude": "22.4179243", "longitude": "114.2073102"}}
```

Unsuccessful result

```
{"status": "warning", "data": {"error_message": "No quota left"}}
```

GET getLocationByUuidMajorMinor

Description
Returns latitude and longitude of a beacon.

Parameters

key	Your API key. It looks like this: 97d9e7b3f06ef1234e15613ebb6663eb
request	Use getLocationByUuidMajorMinor to use this API call.
uuid	The UUID of your beacon, without hyphens. It looks like this: 89407f30f5f8466a7f92555657f7e6d
major	The major of your beacon. It looks like this:

Example of an API function call

Successful and failed results using the method getLocationByUuidMajorMinor are displayed below:

Successful results

```
{"status": "OK", "data": {"latitude": "22.4179243", "longitude": "114.2073102"}}
```

Failed results

```
{"status": "warning", "data": {"error_message": "No quota left"}}
```

JSON

What's so good about JSON?

We have dug into the details of different formats of delivering data through the Internet.

We have chosen common formats, JSON and XML for comparisons for the API. Below shows some of the similarities and differences of the two formats. Details will be explained below.

Features	JSON	XML
Cross-platform	Yes	Yes
Human-readable and writable	Yes; yes, key-value pair delimited by a colon	Yes; no, value surrounded by key tags
Unicode	Yes	Yes
Size	More compact	Bigger

Cross-platform

Both JSON and XML are now supported in major operating systems, including smartphones. APIs for encoding and decoding JSON and XML are provided in many languages, for example `NSJSONSerialization` in Objective C, `getJSONArray` and `getJSONObject` in Java, `JSON.parse` and `eval` in JavaScript and many more languages. XML can also be easily ported to many systems. With the great portability of JSON, we do not have to worry about different implementations of the API in different operating systems and languages.

Human-readable

Both JSON and XML are designed to be easily read by humans. However, values in XML are surrounded by their key tags, which make writing data in XML format becomes a lot of hard work. Imagine you have missed a slash after the closing tag in example 1 and swapped two closing tags in example 2 when you are coding in XML:

Example 1:

```
<beacon>
  <property>
    <id>2<id>
      <!-- a slash should appear in the 2nd id tag
      above -->
    <public_name>Entrance</public_name>
    <private_name>west_ent_7</private_name>
  </property>
</beacon>
```

Example 2:

```
<beacon>
  <property>
    <id>2</id>
    <public_name>Entrance<private_name>
      <!-- public_name closing should appear after "Entrance" -->
    </public_name>west_ent_7</private_name>
```

```
        <!-- private_name opening tag should appear before "west_ent_7" -  
->  
        </property>  
</beacon>
```

These two cases will not happen in JSON as it does not use closing tags. Debugging in XML uses us way more time than JSON. We want to let our developers to use our make their applications as quick as possible and spend less time on debugging. Therefore, we have chosen JSON to return our response from server.

Speaks Unicode

Both JSON and XML support Unicode. Displaying text in different languages using both formats will not be a problem using our API.

More compact

As JSON do not require closing tags to enclose values, it is very often to have smaller size in JSON than XML objects to represent the identical data. Smaller size implies the transfer of data over the Internet requires less time, and hence, reduces the overhead of processing response on the mobile device locally.

Documentation

We have set up a site for documenting the functions that are provided by the API. The page is available at the API documentation site [25]:

The site does not only contain a list of functions, but also example uses of the API, successful and failed results of function calls parameters and descriptions of the calls.

The two images below show the interface of our API documentation site.

Implementation

Android

Below is part of our codes of the Android application that we develop.

DistanceActivity.java - displaying details, for example UUID, major, minor, coordinate pair and response from server using our REST API, of a beacon.

```
package hk.edu.cuhk.radar;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;

import org.altbeacon.beacon.Beacon;
import org.altbeacon.beacon.BeaconConsumer;
import org.altbeacon.beacon.BeaconManager;
import org.altbeacon.beacon.BeaconParser;
import org.altbeacon.beacon.RangeNotifier;
import org.altbeacon.beacon.Region;
import org.altbeacon.beacon.distance.DistanceCalculator;

import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.StatusLine;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
```

```

import org.json.JSONObject;

import android.app.Activity;
import android.app.Fragment;
import android.os.Bundle;
import android.os.RemoteException;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;

public class DistanceActivity extends Activity implements
BeaconConsumer {

    private static String uuid = "", major = "", minor = "";
    private static int count = 0;
    private static final int INTERVALS = 5;

    BeaconManager distanceBeaconManager =
BeaconManager.getInstanceForApplication(this);

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_distance);

```

```

        // set beacon manager to manage iBeacons; bind beacon
manager
        distanceBeaconManager.getBeaconParsers().add(new
BeaconParser() .setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-
23,p:24-24"));

        distanceBeaconManager.setBackgroundBetweenScanPeriod(0l);
        distanceBeaconManager.setBackgroundScanPeriod(1000l);

        distanceBeaconManager.setForegroundBetweenScanPeriod(0l);
        distanceBeaconManager.setForegroundScanPeriod(1000l);
        distanceBeaconManager.bind(this);
    }

    @Override
    public void onBeaconServiceConnect() {
        distanceBeaconManager.setRangeNotifier(new
RangeNotifier() {
            @Override
            public void
didRangeBeaconsInRegion(Collection<Beacon> beacons, Region
region) {
                Object[] beaconArray = beacons.toArray();

                // find beacon with shortest distance
                int idx = -1, min = Integer.MAX_VALUE;
                for (int i = 0; i < beaconArray.length;
i++) {
                    int d =
((Beacon)beaconArray[i]).getRssi();
                    if (d < min) {
                        min = d;
                        idx = i;
                    }
                }
            }
        });
    }
}

```

```

        }
    }

    // update static fields and load web page
    if nothing active
        if (count == 0 && idx != -1) {
            uuid =
((Beacon) beaconArray[idx]).getId1().toString().replace("-",
"").toUpperCase();

            major =
((Beacon) beaconArray[idx]).getId2().toString();
            minor =
((Beacon) beaconArray[idx]).getId3().toString();

            final TextView textViewUuid =
(TextView) findViewById(R.id.distanceEditTextUuid);
            final TextView textViewMajor =
(TextView) findViewById(R.id.distanceEditTextMajor);
            final TextView textViewMinor =
(TextView) findViewById(R.id.distanceEditTextMinor);
            final TextView textViewX =
(TextView) findViewById(R.id.distanceTextViewLatitudeValue);
            final TextView textViewY =
(TextView) findViewById(R.id.distanceTextViewLongitudeValue);
            final TextView textViewResponse =
(TextView) findViewById(R.id.distanceTextViewResponseValue);

            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    textViewUuid.setText(uuid);

                    textViewMajor.setText(major);

```

```

        textViewMinor.setText(minor);
    }
});

HttpClient webHttpClient = new
DefaultHttpClient();

try {
    HttpResponse response =
webHttpClient.execute(new
HttpGet("https://appsrv.cse.cuhk.edu.hk/~phng2/fyp/api/?key=97d
9e7b3f06ef1234e15613ebb6663eb&request=getLocationByUuidMajorMin
or&uuid=" + uuid + "&major=" + major + "&minor=" + minor));
    StatusLine webStatusLine =
response.getStatusLine();
    if (webStatusLine.getStatusCode()
== HttpStatus.SC_OK) {
        ByteArrayOutputStream webOut =
new ByteArrayOutputStream();

        response.getEntity().writeTo(webOut);
        webOut.close();
        final String webResponseString
= webOut.toString();

        String data = new
JSONObject(webResponseString).getString("data");
        final String lat = new
JSONObject(data).getString("latitude");
        final String lon = new
JSONObject(data).getString("longitude");

        //webStatus.setText(new

```

```

JSONObject (webResponseString).getString("status"));
        runOnUiThread(new Runnable() {
            @Override
            public void run() {

                textViewResponse.setText(webResponseString);

                textViewY.setText(lat);

                textViewX.setText(lon);

            }
        });
    } else {
        // Closes the connection.

        response.getEntity().getContent().close();
        throw new
        IOException(webStatusLine.getReasonPhrase());
    }
} catch (ClientProtocolException e) {
    // TODO Auto-generated catch
    block
        e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch
    block
        e.printStackTrace();
} catch (JSONException e) {
    // TODO Auto-generated catch
    block
        e.printStackTrace();
    }
}
}

```

```

        // increment counter if static fields
equal; decrement otherwise
        if (count < INTERVALS && idx != -1 &&
uuid.equals(((Beacon)beaconArray[idx]).getId1().toString().repl
ace("-", "").toUpperCase())
                &&
major.equals(((Beacon)beaconArray[idx]).getId2().toString()) &&
minor.equals(((Beacon)beaconArray[idx]).getId3().toString())) {
            count++;
        }
        else if (count > 0) {
            count--;
        }
    }
});

    try {
        Region radarRegion = new Region("radarRegion", null,
null, null);

        distanceBeaconManager.startRangingBeaconsInRegion(radarReg
ion);
    } catch (RemoteException e) { }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    distanceBeaconManager.unbind(this);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

```

```

        // Handle action bar item clicks here. The action bar
will
        // automatically handle clicks on the Home/Up button,
so long
        // as you specify a parent activity in
AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {

        public PlaceholderFragment() {
        }

        @Override
        public View onCreateView(LayoutInflater inflater,
ViewGroup container,
            Bundle savedInstanceState) {
            View rootView =
inflater.inflate(R.layout.fragment_distance,
                container, false);
            return rootView;
        }
    }
}

```

MainActivity.java - displaying a list of visible beacons in range

```
package hk.edu.cuhk.radar;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;

import org.altbeacon.beacon.Beacon;
import org.altbeacon.beacon.BeaconConsumer;
import org.altbeacon.beacon.BeaconManager;
import org.altbeacon.beacon.BeaconParser;
import org.altbeacon.beacon.RangeNotifier;
import org.altbeacon.beacon.Region;
import org.altbeacon.beacon.distance.DistanceCalculator;

import android.app.Activity;
import android.app.Fragment;
import android.content.Intent;
import android.os.Bundle;
import android.os.RemoteException;
import android.os.StrictMode;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends Activity implements
BeaconConsumer {
```

```

        BeaconManager distanceBeaconManager =
BeaconManager.getInstanceForApplication(this);

        private static ArrayList<String> listItems = new
ArrayList<String>(Arrays.asList("No beacon detected..."));

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);

            // discouraged to do this
            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);

            if (savedInstanceState == null) {
                getFragmentManager().beginTransaction()
                    .add(R.id.container, new
PlaceholderFragment()).commit();
            }
            distanceBeaconManager.getBeaconParsers().add(new
BeaconParser().setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-
23,p:24-24"));

            distanceBeaconManager.setBackgroundBetweenScanPeriod(01);
            distanceBeaconManager.setBackgroundScanPeriod(10001);

            distanceBeaconManager.setForegroundBetweenScanPeriod(01);
            distanceBeaconManager.setForegroundScanPeriod(10001);
            distanceBeaconManager.bind(this);
            ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this, R.layout.beacon_list_item,

```

```

listItems);
        ListView listView = (ListView)
findViewById(R.id.beaconList);
        listView.setAdapter(adapter);
    }

    @Override
    public void onBeaconServiceConnect() {
        distanceBeaconManager.setRangeNotifier(new
RangeNotifier() {
            @Override
            public void
didRangeBeaconsInRegion(Collection<Beacon> beacons, Region
region) {
                Object dc = Beacon.getDistanceCalculator();

                // update item list
                listItems.clear();
                if (beacons.size() == 0) {
                    listItems.add("#1 No beacon
detected...\n");
                    listItems.add("#2 No beacon
detected...\n");
                    listItems.add("#3 No beacon
detected...\n");
                    listItems.add("#4 No beacon
detected...\n");
                }
                else {
                    Object[] beaconArray = beacons.toArray();
                    for (int i = 0; i <
beaconArray.length; i++) {
                        String tmp = new String("");

```

```

        tmp = tmp + "Bluetooth Name: " +
((Beacon)beaconArray[i]).getBluetoothName() + "\n";
        tmp = tmp + "UUID: " +
((Beacon)beaconArray[i]).getId1() + "\n";
        tmp = tmp + "Major: " +
((Beacon)beaconArray[i]).getId2() + " Minor: " +
((Beacon)beaconArray[i]).getId3() + "\n";
        tmp = tmp + "TxPower: " +
((Beacon)beaconArray[i]).getTxPower() + " RSSI: " +
((Beacon)beaconArray[i]).getRssi() + "\n";
        tmp = tmp + "Average distance: "
+ ((Beacon)beaconArray[i]).getDistance() + "\n";
        tmp = tmp + "Instant distance: "
+ ((DistanceCalculator)
dc).calculateDistance(((Beacon)beaconArray[i]).getTxPower(),
((Beacon)beaconArray[i]).getRssi()) + "\n";
        listItems.add(tmp);
    }
}

// update View
final ArrayAdapter adapter = (ArrayAdapter)
((ListView) findViewById(R.id.beaconList)).getAdapter();
runOnUiThread(new Runnable() {
    @Override
    public void run() {
        adapter.notifyDataSetChanged();
    }
});
}
});

try {

```

```

        Region radarRegion = new Region("radarRegion", null,
null, null);

        distanceBeaconManager.startRangingBeaconsInRegion(radarReg
ion);
    } catch (RemoteException e) {
    }
}

@Override
protected void onResume() {
    super.onResume();
    onBeaconServiceConnect();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    distanceBeaconManager.unbind(this);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    // Inflate the menu; this adds items to the action
bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar
will

```

```
        // automatically handle clicks on the Home/Up button,
so long
        // as you specify a parent activity in
AndroidManifest.xml.
        switch(item.getItemId()) {
        case R.id.actionAbout:
            Intent intent_about=new Intent(this,
AboutActivity.class);
            this.startActivity(intent_about);
            break;

        case R.id.actionMap:
            Intent intent_map=new Intent(this,
MapActivity.class);
            this.startActivity(intent_map);
            break;

        case R.id.actionVideo:
            Intent intent_video=new Intent(this,
VideoActivity.class);
            this.startActivity(intent_video);
            break;

        case R.id.actionCheckIn:
            Intent intent_checkin=new Intent(this,
CheckinActivity.class);
            this.startActivity(intent_checkin);
            break;

        case R.id.actionWeb:
            Intent intent_web=new Intent(this,
WebActivity.class);
            this.startActivity(intent_web);
```

```
        break;

        case R.id.actionProduct:
            Intent intent_product=new Intent(this,
ProductActivity.class);
            this.startActivity(intent_product);
            break;

        case R.id.actionTable:
            Intent intent_table=new Intent(this,
TableActivity.class);
            this.startActivity(intent_table);
            break;

        case R.id.actionPass:
            Intent intent_pass=new Intent(this,
PassActivity.class);
            this.startActivity(intent_pass);
            break;

        case R.id.actionDistance:
            Intent intent_distance=new Intent(this,
DistanceActivity.class);
            this.startActivity(intent_distance);
            break;

        case R.id.actionAccel:
            Intent intent_accel=new Intent(this,
AccelActivity.class);
            this.startActivity(intent_accel);
            break;

        case R.id.actionMakeNotification:
```

```

        Intent intent_make_notification=new Intent(this,
MakeNotificationActivity.class);
        this.startActivity(intent_make_notification);
        break;

    default:
        return super.onOptionsItemSelected(item);
    }
    return true;
}

/**
 * A placeholder fragment containing a simple view.
 */
public static class PlaceholderFragment extends Fragment {

    public PlaceholderFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater,
ViewGroup container,
        Bundle savedInstanceState) {
        View rootView =
inflater.inflate(R.layout.fragment_main, container,
            false);
        return rootView;
    }
}
}

```

MapActivity.java - displaying the region that you are in and provides a logging tool for detecting beacon signal strength

```
package hk.edu.cuhk.radar;

import java.util.Collection;
import org.altbeacon.beacon.Beacon;
import org.altbeacon.beacon.BeaconConsumer;
import org.altbeacon.beacon.BeaconManager;
import org.altbeacon.beacon.BeaconParser;
import org.altbeacon.beacon.RangeNotifier;
import org.altbeacon.beacon.Region;
import org.altbeacon.beacon.distance.DistanceCalculator;

import android.app.Activity;
import android.app.Fragment;
import android.os.Bundle;
import android.os.RemoteException;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class MapActivity extends Activity implements
BeaconConsumer {

    private static String uuid = "", major = "", minor = "";
    private static int count = 0;
    private static final int INTERVALS = 5;
```

```

        BeaconManager distanceBeaconManager =
BeaconManager.getInstanceForApplication(this);

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_map);

    // load the Web View
    WebView webWebView =
(WebView) findViewById(R.id.detailWebView);
    webWebView.setWebViewClient(new WebViewClient());
    webWebView.getSettings().setJavaScriptEnabled(true);

    webWebView.loadUrl("https://appsrv.cse.cuhk.edu.hk/~phng2/
fyp/splash.html");

    // set beacon manager to manage iBeacons; bind beacon
manager
    distanceBeaconManager.getBeaconParsers().add(new
BeaconParser().setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-
23,p:24-24"));

    distanceBeaconManager.setBackgroundBetweenScanPeriod(0l);
    distanceBeaconManager.setBackgroundScanPeriod(1000l);

    distanceBeaconManager.setForegroundBetweenScanPeriod(0l);
    distanceBeaconManager.setForegroundScanPeriod(1000l);
    distanceBeaconManager.bind(this);
}

@Override

```

```

public void onBeaconServiceConnect() {

    distanceBeaconManager.setRangeNotifier(new
RangeNotifier() {
        @Override
        public void
didRangeBeaconsInRegion(Collection<Beacon> beacons, Region
region) {

            Object dc = Beacon.getDistanceCalculator();
            Object[] beaconArray = beacons.toArray();

            // find beacon with shortest distance
            int idx = -1;
            double min = Double.MAX_VALUE;
            for (int i = 0; i < beaconArray.length;
i++) {

                double d = ((DistanceCalculator)
dc).calculateDistance(((Beacon)beaconArray[i]).getTxPower(),
((Beacon)beaconArray[i]).getRssi());
                // double d =
(Beacon)beaconArray[i]).getDistance();
                if (d < min) {
                    min = d;
                    idx = i;
                }
            }

            // update static fields and load web page
if nothing active

            if (count == 0 && min < 2) {
                uuid =
((Beacon)beaconArray[idx]).getId1().toString().replace("-",
"").toUpperCase();

```

```

        major =
((Beacon)beaconArray[idx]).getId2().toString();
        minor =
((Beacon)beaconArray[idx]).getId3().toString();

        final WebView webWebView =
(WebView) findViewById(R.id.detailWebView);
        webWebView.post(new Runnable() {
            @Override
            public void run() {

                webWebView.loadUrl("https://appsrv.cse.cuhk.edu.hk/~phng2/
fyp/display.php?owner=97d9e7b3f06ef1234e15613ebb6663eb&uuid=" +
uuid + "&major=" + major + "&minor=" + minor);

            }

        });

        // increment counter if static fields
equal; decrement otherwise
        if (count < INTERVALS && min < 3 &&
uuid.equals(((Beacon)beaconArray[idx]).getId1().toString().repl
ace("-", "").toUpperCase())

            &&
major.equals(((Beacon)beaconArray[idx]).getId2().toString()) &&
minor.equals(((Beacon)beaconArray[idx]).getId3().toString())) {
            count++;
        }
        else if (count > 0) {
            count--;
            if (count == 0) {
                uuid = "";
                major = "";
            }
        }
    }
}

```

```

        minor = "";
        final WebView webWebView =
(WebView) findViewById(R.id.detailWebView);
        webWebView.post(new Runnable() {
            @Override
            public void run() {

                webWebView.loadUrl("https://appsrv.cse.cuhk.edu.hk/~phng2/
fyp/splash.html");

            }
        });
    }
}

try {
    Region radarRegion = new Region("radarRegion", null,
null, null);

    distanceBeaconManager.startRangingBeaconsInRegion(radarReg
ion);
} catch (RemoteException e) { }

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar
will
    // automatically handle clicks on the Home/Up button,
so long
    // as you specify a parent activity in
AndroidManifest.xml.

```

```

        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {

        public PlaceholderFragment() {
        }

        @Override
        public View onCreateView(LayoutInflater inflater,
            ViewGroup container,
                Bundle savedInstanceState) {
            View rootView =
inflater.inflate(R.layout.fragment_web, container,
                false);
            return rootView;
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        distanceBeaconManager.unbind(this);
    }
}

```

WebActivity.java - shows the result page of the scanned page, i.e. the name and description of the beacon that are public

```
package hk.edu.cuhk.radar;

import java.util.Collection;
import org.altbeacon.beacon.Beacon;
import org.altbeacon.beacon.BeaconConsumer;
import org.altbeacon.beacon.BeaconManager;
import org.altbeacon.beacon.BeaconParser;
import org.altbeacon.beacon.RangeNotifier;
import org.altbeacon.beacon.Region;
import org.altbeacon.beacon.distance.DistanceCalculator;

import android.app.Activity;
import android.app.Fragment;
import android.os.Bundle;
import android.os.RemoteException;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Toast;

public class WebActivity extends Activity implements
BeaconConsumer {

    BeaconManager distanceBeaconManager =
BeaconManager.getInstanceForApplication(this);
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_web);

    // load the Web View
    WebView webWebView =
(WebView) findViewById(R.id.webWebView);
    webWebView.setWebViewClient(new WebViewClient());

    webWebView.getSettings().setJavaScriptEnabled(true);
    webWebView.getSettings().setDatabaseEnabled(true);

    webWebView.getSettings().setDomStorageEnabled(true);

    webWebView.loadUrl("https://appsrv.cse.cuhk.edu.hk/~twyeu
ng2/fyp/location.php");

    // set beacon manager to manage iBeacons; bind
beacon manager
    distanceBeaconManager.getBeaconParsers().add(new
BeaconParser()).setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-
23,p:24-24"));

    distanceBeaconManager.setBackgroundBetweenScanPeriod(0l);

    distanceBeaconManager.setBackgroundScanPeriod(1000l);

    distanceBeaconManager.setForegroundBetweenScanPeriod(0l);

    distanceBeaconManager.setForegroundScanPeriod(1000l);
    distanceBeaconManager.bind(this);
}

```

```

@Override
public void onBeaconServiceConnect() {

    distanceBeaconManager.setRangeNotifier(new
RangeNotifier() {
        @Override
        public void
didRangeBeaconsInRegion(Collection<Beacon> beacons, Region
region) {
            Object dc = Beacon.getDistanceCalculator();
            Object[] beaconArray =
beacons.toArray();

            String tmp = "";
            for (int i = 0; i <
beaconArray.length; i++) {
                tmp = tmp + "uuid[]" +
((Beacon)beaconArray[i]).getId1().toString().replace("-",
"" ).toUpperCase();

                tmp = tmp + "&major[]" +
((Beacon)beaconArray[i]).getId2();

                tmp = tmp + "&minor[]" +
((Beacon)beaconArray[i]).getId3();

                tmp = tmp + "&distance[]" +
((Beacon)beaconArray[i]).getDistance();

                tmp = tmp + "&raw[]" +
((DistanceCalculator)
dc).calculateDistance(((Beacon)beaconArray[i]).getTxPower(),
((Beacon)beaconArray[i]).getRssi());

                tmp = tmp + "&rssi[]" +
((Beacon)beaconArray[i]).getRssi() + "&";
            }
        }
    }
}

```

```

        final String request = tmp;
        final WebView webWebView =
(WebView) findViewById(R.id.webWebView);
        webWebView.post(new Runnable() {
            @Override
            public void run() {

                webWebView.loadUrl("https://appsrv.cse.cuhk.edu.hk/~twyeu
ng2/fyp/location.php?" + request);
            }
        });
        Log.w("LocationURL",
"https://appsrv.cse.cuhk.edu.hk/~twyeung2/fyp/location.php?" +
request);
    }
});

    try {
        Region radarRegion = new Region("radarRegion",
null, null, null);

        distanceBeaconManager.startRangingBeaconsInRegion(radarRe
gion);
    } catch (RemoteException e) { }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action
bar will
    // automatically handle clicks on the Home/Up
button, so long
    // as you specify a parent activity in

```

AndroidManifest.xml.

```
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment
    {

        public PlaceholderFragment() {
        }

        @Override
        public View onCreateView(LayoutInflater inflater,
        ViewGroup container,
            Bundle savedInstanceState) {
            View rootView =
inflater.inflate(R.layout.fragment_web, container,
                false);
            return rootView;
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        distanceBeaconManager.unbind(this);
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="hk.edu.cuhk.radar"
    android:versionCode="0"
    android:versionName="0.2.4 crescent" >

    <uses-permission android:name="android.permission.INTERNET"
/>
    <uses-permission
android:name="android.permission.BLUETOOTH" />
    <uses-permission
android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.VIBRATE"
/>
    <uses-permission
android:name="android.permission.BLUETOOTH_PRIVILEGED" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-feature
android:name="android.hardware.bluetooth_le"
android:required="true"/>

    <uses-sdk
        android:minSdkVersion="18"
        android:targetSdkVersion="19" />

    <application
        android:name="hk.edu.cuhk.radar.RadarApp"
        android:allowBackup="true"
```

```

    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <activity
        android:launchMode="singleInstance"
        android:name="hk.edu.cuhk.radar.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    <activity
        android:name="hk.edu.cuhk.radar.AboutActivity"
        android:label="About" >
    </activity>
    <activity
        android:name="hk.edu.cuhk.radar.MapActivity"
        android:label="@string/title_activity_map" >
    </activity>
    <activity
        android:name="hk.edu.cuhk.radar.VideoActivity"
android:configChanges="orientation|keyboardHidden|screenSize"
        android:label="@string/title_activity_video"
        android:theme="@style/FullscreenTheme" >
    </activity>
    <activity
        android:name="hk.edu.cuhk.radar.CheckinActivity"

```

```
        android:label="@string/title_activity_checkin" >
</activity>
<activity
    android:name="hk.edu.cuhk.radar.SettingsActivity"
    android:label="@string/title_activity_settings" >
</activity>
<activity
    android:launchMode="singleInstance"
    android:name="hk.edu.cuhk.radar.WebActivity"
    android:label="@string/title_activity_web" >
</activity>
<activity
    android:name="hk.edu.cuhk.radar.ProductActivity"
    android:label="@string/title_activity_product" >
</activity>
<activity
    android:name="hk.edu.cuhk.radar.TableActivity"
    android:label="@string/title_activity_table" >
</activity>
<activity
    android:name="hk.edu.cuhk.radar.PassActivity"
    android:label="@string/title_activity_pass" >
</activity>
<activity
    android:name="hk.edu.cuhk.radar.DistanceActivity"
    android:label="@string/title_activity_distance" >
</activity>
<activity
    android:name="hk.edu.cuhk.radar.AccelActivity"
    android:label="@string/title_activity_accel" >
</activity>
<activity
```

```
android:name="hk.edu.cuhk.radar.MakeNotificationActivity"
android:label="@string/title_activity_make_notification" >
    </activity>
    <activity

android:name="hk.edu.cuhk.radar.ShowNotificationActivity"
android:label="@string/title_activity_show_notification" >
    </activity>

    </application>

</manifest>
```

Web

We use PHP to implement the CMS and the API.

.htaccess - redirects user to HTTPS protocol if they are using the unencrypted HTTP protocol

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

add.php - adds a new beacon

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="IE=edge" http-equiv="X-UA-Compatible">
  <meta content="width=device-width, initial-scale=1"
name="viewport">

  <title>Beacon Manager</title><!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <script src="js/jquery-2.1.1.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page
via file:// -->
  <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
```

```

/script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
    <![endif]-->
</head>

<body>
    <div class="container">
        <?php
            include 'include.php';
        ?>

        <h1>Add a beacon</h1>
        <form role="form" action="insert.php" method="get">
            <div class="form-group">
                <label for="inputOwner">OWNER</label>
                <input class="form-control" id="inputOwner"
name="inputOwner" value="97d9e7b3f06ef1234e15613ebb6663eb"
required readonly>
            </div>
            <div class="form-group">
                <label for="inputUuid">UUID</label>
                <input class="form-control" id="inputUuid"
name="inputUuid" required autofocus>
            </div>
            <div class="form-group">
                <label for="inputMajor">MAJOR</label>
                <input type="number" min="0" max="9999"
class="form-control" id="inputMajor" name="inputMajor"
required>
            </div>
            <div class="form-group">

```

```

        <label for="inputMinor">MINOR</label>
        <input type="number" min="0" max="9999"
class="form-control" id="inputMinor" name="inputMinor"
required>
    </div>
    <div class="form-group">
        <label for="inputLongitude">LONGITUDE</label>
        <input type="text" class="form-control"
id="inputLongitude" name="inputLongitude" required>
    </div>
    <div class="form-group">
        <label for="inputLatitude">LATITUDE</label>
        <input type="text" class="form-control"
id="inputLatitude" name="inputLatitude" required>
    </div>
    <div class="form-group">
        <label
for="inputDescription">DESCRIPTION</label>
        <input type="text" class="form-control"
id="inputDescription" name="inputDescription"
placeholder="optional">
    </div>
    <button type="submit" class="btn btn-
success">Add</button>
    <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-default">Cancel</button></a>
    </form>

</div>
</body>
</html>

```

display.php - shows details of a location, for example, pictures, title of the location and description

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta content="IE=edge" http-equiv="X-UA-Compatible" />
    <meta name="viewport" content="width=device-width,
initial-scale=1, maximum-scale=1, user-scalable=0" />

    <title>Beacon Manager</title><!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/owl.carousel.css" rel="stylesheet">
    <link href="css/owl.theme.css" rel="stylesheet">
    <!-- jQuery (necessary for Bootstrap's JavaScript
plugins) -->

    <script src="js/jquery-2.1.1.min.js"></script>
    <!-- Include all compiled plugins (below), or include
individual files as needed -->

    <script src="js/bootstrap.min.js"></script>
    <script src="js/owl.carousel.min.js"></script>
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the
page via file:// -->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
```

```
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>

    <![endif]-->

<style>
    head, body {
        background: #EEEEEE;
        -moz-transition: ease 0.25s;
        -webkit-transition: ease 0.25s;
        -o-transition: ease 0.25s;
        transition: ease 0.25s;
    }

    h1 {
        font-size: 28px;
    }

    h2 {
        font-size: 24px;
    }

    h3 {
        font-size: 14px;
        color: #888888;
    }

    #copyright {
        text-align: center;
        font-size: 12px;
        margin-top: 15px;
        margin-bottom: 5px;
    }
}
```

```

        .hideAll {
            background: #000000 !important;
            color: #000000 !important;
        }

        #noResult {
            display: block;
            position: fixed;
            height: 100%;
            width: 100%;
            z-index: 9999;
            background: #EEEEEE;
            font-size: 20px;
            text-align: center;
            padding-top: 20%;
        }

</style>

</head>

<body>

    <?php
        ini_set("display_errors",1);
        ini_set("display_startup_errors",1);
        error_reporting(-1);
        $con=new
mysqli("appsrvdb.cse.cuhk.edu.hk","viewtech","G5XYcFhy","viewte
ch");

        if (mysqli_connect_errno()) {
            echo "Failed to connect to MySQL: " .

```

```

mysqli_connect_error();
}

$owner=$_GET["owner"];
$uuid=$_GET["uuid"];
$major=$_GET["major"];
$minor=$_GET["minor"];

$owner=htmlspecialchars($owner, ENT_QUOTES);
$uuid=htmlspecialchars($uuid, ENT_QUOTES);
$major=htmlspecialchars($major, ENT_QUOTES);
$minor=htmlspecialchars($minor, ENT_QUOTES);

$owner=mysqli_real_escape_string($con, $owner);
$uuid=mysqli_real_escape_string($con, $uuid);
$major=mysqli_real_escape_string($con, $major);
$minor=mysqli_real_escape_string($con, $minor);

$result=mysqli_query($con, "SELECT * FROM
lyu1403_beacons WHERE owner='" . $owner . "' AND uuid='" .
$uuid . "' AND Major='" . $major . "' AND Minor='" . $minor.
"'");

if (mysqli_num_rows($result)==0) {
    echo '
        <div id="noResult">
            Something wrong occured.<br>
            Try again later.
        </div>
    ';
}

```

```

        $row=mysqli_fetch_array($result);

        echo '
            <div class="container-fluid">
                <div class="col-lg-12">
                    <h1>' . $row["publicTitle"] . '</h1>
                </div>
            </div>

            <!-- image carousel out of col or container for
spanning full width -->

            <center>
                <div class="owl-carousel">';

                $coverPicArray = explode(", ",
$row["picture"]);

                for ($i = 0; $i <
count($coverPicArray); $i++) {
                    echo '';
                }

                echo '
            </div>
        </center>

        <div class="container-fluid">
            <div class="col-lg-12">
                <h3>
                    ' . $row["publicSubtitle"] . '
                </h3>

```

```
<h2>
```

```
    Description
```

```
</h2>
```

```
<p>
```

```
    ' . $row["publicDescription"] . '
```

```
</p>
```

```
<div id="copyright">Radar © 2014</div>
```

```
</div>
```

```
</div>
```

```
<!-- all custom scripts here -->
```

```
<script>
```

```
    $(".owl-carousel").owlCarousel(
```

```
    {
```

```
        stagePadding: 10,
```

```
        margin: 10,
```

```
        loop: true,
```

```
        autoWidth: true,
```

```
        autoHeight: true
```

```
    }
```

```
);
```

```

        $(".owl-carousel").click (function() {
            $("body").toggleClass("hideAll");
        });

</script>

';

echo '
    <div class="hidden">
        <center>
            
        </center>
        <div class="panel panel-info">
            <table class="table table-striped table-
hover">
                <tr>
                    <td align="right">
                        ID
                    </td>
                    <td>
                        ' . $row["ID"] . '
                    </td>
                </tr>
                <tr>
                    <td align="right">
                        Major
                    </td>
                    <td>
                        ' . $row["Major"] . '
                    </td>
                </tr>
            </table>
        </div>
    </div>

```

```

        <tr>
            <td align="right">
                Minor
            </td>
            <td>
                ' . $row["Minor"] . '
            </td>
        </tr>
        <tr>
            <td align="right">
                Longitude
            </td>
            <td>
                ' . $row["Longitude"] . '
            </td>
        </tr>
        <tr>
            <td align="right">
                Latitude
            </td>
            <td>
                ' . $row["Latitude"] . '
            </td>
        </tr>
        <tr>
            <td align="right">
                Location
            </td>
            <td>
                <iframe width="250"
height="250" frameborder="0" style="border:0"
src="https://www.google.com/maps/embed/v1/place?key=AIzaSyBeVdy
imJ5kn0MM9Yp66bxfauWF7ZPVdsM&zoom=18&q=' . $row['Latitude'] .

```

```

', ' . $row['Longitude'] . '">
</iframe>
</td>
</tr>
<tr>
<td align="right">
Notes
</td>
<td>
' . $row["Description"] . '
</td>
</tr>
</table>
</div>
</div>
';

?>

<!--
<a href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"
target="_parent"><button type="button" class="btn btn-success"
autofocus>Dashboard</button></a>
<a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/preview.php"
target="_parent"><button type="button" class="btn btn-
primary">Preview more</button></a>
-->

</body>
</html>

```

edit.php - edits a beacon by entering the beacon ID

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="IE=edge" http-equiv="X-UA-Compatible">
  <meta content="width=device-width, initial-scale=1"
name="viewport">

  <title>Beacon Manager</title><!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins)
-->

  <script src="js/jquery-2.1.1.min.js"></script>
<!-- Include all compiled plugins (below), or include
individual files as needed -->

  <script src="js/bootstrap.min.js"></script>
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page
via file:// -->
  <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
  <![endif]-->
```

```

<style>
  #topBar {
    display: block;
    position: relative;
    width: 100%;
    background: #000000;
    color: #FFFFFF;
    text-align: center;
    padding-top: 5px;
    padding-bottom: 5px;
    font-size: 20px;
  }
</style>

</head>

<body>
  <!--
  <div id="topBar">Something in the system changed. This
page will not work and be removed soon.</div>
  -->
  <div class="container">
    <?php
      include 'include.php';
    ?>

    <h1>Edit a beacon</h1>
    <form role="form" action="modify.php" method="get">
      <div class="form-group">
        <label for="inputID">ID</label>
        <input type="number" min="1" class="form-
control" id="inputID" name="inputID" required autofocus>
      </div>

```

```

        <button type="submit" class="btn btn-
success">Next</button>
        <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-default">Cancel</button></a>
    </form>
</div>
</body>
</html>

```

finish_modify.php - final page for confirmation before making changes to a beacon

```

<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta content="IE=edge" http-equiv="X-UA-Compatible">
        <meta content="width=device-width, initial-scale=1"
name="viewport">

        <title>Beacon Manager</title><!-- Bootstrap -->
        <link href="css/bootstrap.min.css" rel="stylesheet">
        <!-- jQuery (necessary for Bootstrap's JavaScript
plugins) -->

        <script src="js/jquery-2.1.1.min.js"></script>
        <!-- Include all compiled plugins (below), or include
individual files as needed -->

        <script src="js/bootstrap.min.js"></script>
        <!-- HTML5 Shim and Respond.js IE8 support of HTML5

```

```

elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the
page via file:// -->
    <!--[if lt IE 9]>
        <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/s>script>
        <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
    <![endif]-->
</head>

<body>
    <div class="container">
        <?php
            ini_set('display_errors',1);
            ini_set('display_startup_errors',1);
            error_reporting(-1);
            $con=new
mysqli("appsrvdb.cse.cuhk.edu.hk","viewtech","G5XYcFhy","viewte
ch");
            if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " .
mysqli_connect_error();
            }

            if (isset($_GET['inputOwner'])) {
                $inputOwner=$_GET['inputOwner'];
            }

            if (isset($_GET['inputID'])) {
                $inputID=$_GET['inputID'];

```

```

    }

    if (isset($_GET['inputUuid'])) {
        $inputUuid=$_GET['inputUuid'];
    }

    if (isset($_GET['inputMajor'])) {
        $inputMajor=$_GET['inputMajor'];
    }

    if (isset($_GET['inputMinor'])) {
        $inputMinor=$_GET['inputMinor'];
    }

    if (isset($_GET['inputLatitude'])) {
        $inputLatitude=$_GET['inputLatitude'];
    }

    if (isset($_GET['inputLongitude'])) {
        $inputLongitude=$_GET['inputLongitude'];
    }

    if (isset($_GET['inputDescription'])) {
        $inputDescription=$_GET['inputDescription'];
    }

    if (isset($_GET['inputPublicDescription'])) {
        $inputPublicDescription=$_GET['inputPublicDescription'];
    }

    if (isset($_GET['inputPublicTitle'])) {
        $inputPublicTitle=$_GET['inputPublicTitle'];
    }

```

```

    }

    if (isset($_GET['inputPublicSubtitle'])) {

$inputPublicSubtitle=$_GET['inputPublicSubtitle'];
    }

    $inputOwner=htmlspecialchars($inputOwner,
ENT_QUOTES);
    $inputID=htmlspecialchars($inputID, ENT_QUOTES);
    $inputUuid=htmlspecialchars($inputUuid,
ENT_QUOTES);
    $inputMajor=htmlspecialchars($inputMajor,
ENT_QUOTES);
    $inputMinor=htmlspecialchars($inputMinor,
ENT_QUOTES);
    $inputLatitude=htmlspecialchars($inputLatitude,
ENT_QUOTES);
    $inputLongitude=htmlspecialchars($inputLongitude,
ENT_QUOTES);

$inputDescription=htmlspecialchars($inputDescription,
ENT_QUOTES);

$inputPublicDescription=htmlspecialchars($inputPublicDescriptio
n, ENT_QUOTES);

$inputPublicTitle=htmlspecialchars($inputPublicTitle,
ENT_QUOTES);

$inputPublicSubtitle=htmlspecialchars($inputPublicSubtitle,
ENT_QUOTES);

```

```



```

```

        echo '<div class="alert alert-success"
role="alert">Editing succeeded.</div>';
    }
    else {
        die(mysqli_error($con));
    }

?>

    <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-success"
autofocus>Dashboard</button></a>
    <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/edit.php"><button
type="button" class="btn btn-primary">Edit more</button></a>

    </div>
</body>

</html>

```

include.php - the top navigation to be included in each page

```

<?php
    echo '
        <ul class="nav nav-pills nav-justified">
            <li><a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/">Dashboard</a><
/li>
            <li><a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/add.php">Add</a>

```

```

</li>
    <li><a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/edit.php">Edit</
a></li>
    <li><a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/remove.php">Remo
ve</a></li>
    <li><a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/preview.php">Pre
view</a></li>
</ul>
';
?>

```

index.php - the dashboard page displaying all beacon, along with their information, including their UUID, major, minor and description

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta content="IE=edge" http-equiv="X-UA-Compatible">
    <meta content="width=device-width, initial-scale=1"
name="viewport">

    <title>Beacon Manager</title><!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins)
-->

    <script src="js/jquery-2.1.1.min.js"></script>

```

```

<!-- Include all compiled plugins (below), or include
individual files as needed -->

<script src="js/bootstrap.min.js"></script>
<!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page
via file:// -->
<!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
<![endif]-->

<style>
    * {
        word-break: break-all;
    }
</style>

</head>

<body>
    <div class="container">
        <?php
            include 'include.php';
        ?>
        <!--
        <ul class="list-group">
            <li class="list-group-item list-group-item-danger">For

```

```

your security,</li>
    <li class="list-group-item">NO IE6</li>
    <li class="list-group-item">NO GBK</li>
    <li class="list-group-item list-group-item-
success">Thank you.</li>
</ul>
-->
<?php
// Connect to db
ini_set('display_errors',1);
ini_set('display_startup_errors',1);
error_reporting(-1);
$con=new
mysqli("appsrddb.cse.cuhk.edu.hk","viewtech","G5XYcFhy","viewte
ch");
if (mysqli_connect_errno()) {
echo "Failed to connect to MySQL: " .
mysqli_connect_error();
}

$result=mysqli_query($con, "SELECT * FROM
lyu1403_beacons ORDER BY ID ASC");

echo
'
<h1>Beacon info</h1>
<div class="panel panel-info">
    <div class="panel-heading">
        All beacons
    </div>
    <table class="table table-striped table-hover">
        <tr>
            <th>ID</th>

```

```

        <th>UUID</th>
        <th>MAJOR</th>
        <th>MINOR</th>
        <th>LONGITUDE</th>
        <th>LATITUDE</th>
        <th>DESCRIPTION</th>
    </tr>'
;

    while($row = mysqli_fetch_array($result)) {
        echo '
            <tr>
                <td>' . $row['ID'] . '</td>
                <td>' . $row['uuid'] . '</td>
                <td>' . $row['Major'] . '</td>
                <td>' . $row['Minor'] . '</td>
                <td>' . $row['Longitude'] . '</td>
                <td>' . $row['Latitude'] . '</td>
                <td>' . $row['Description'] . '</td>
            </tr>
            ';
        }

        echo '
            </table>
        </div>'
;
    ?>
</div>
</body>
</html>

```

insert.php - a page to insert a new beacon details that are sent from add.php

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta content="IE=edge" http-equiv="X-UA-Compatible">
    <meta content="width=device-width, initial-scale=1"
name="viewport">

    <title>Beacon Manager</title><!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- jQuery (necessary for Bootstrap's JavaScript
plugins) -->

    <script src="js/jquery-2.1.1.min.js"></script>
    <!-- Include all compiled plugins (below), or include
individual files as needed -->

    <script src="js/bootstrap.min.js"></script>
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the
page via file:// -->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
      <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
    <![endif]-->
  </head>
```

```

<body>
  <div class="container">
    <?php
      ini_set('display_errors',1);
      ini_set('display_startup_errors',1);
      error_reporting(-1);
      $con=new
mysqli("appsrvdb.cse.cuhk.edu.hk","viewtech","G5XYcFhy","viewte
ch");

      if (mysqli_connect_errno()) {
          echo "Failed to connect to MySQL: " .
mysqli_connect_error();
      }

      $inputOwner=$_GET['inputOwner'];
      $inputUuid=$_GET['inputUuid'];
      $inputMajor=$_GET['inputMajor'];
      $inputMinor=$_GET['inputMinor'];
      $inputLongitude=$_GET['inputLongitude'];
      $inputLatitude=$_GET['inputLatitude'];
      $inputDescription=$_GET['inputDescription'];

      $inputOwner=htmlspecialchars($inputOwner,
ENT_QUOTES);
      $inputUuid=htmlspecialchars($inputUuid,
ENT_QUOTES);
      $inputMajor=htmlspecialchars($inputMajor,
ENT_QUOTES);
      $inputMinor=htmlspecialchars($inputMinor,
ENT_QUOTES);
      $inputLongitude=htmlspecialchars($inputLongitude,
ENT_QUOTES);

```

```

        $inputLatitude=htmlspecialchars($inputLatitude,
ENT_QUOTES);

$inputDescription=htmlspecialchars($inputDescription,
ENT_QUOTES);

        $inputOwner=mysqli_real_escape_string($con,
$inputOwner);
        $inputUuid=mysqli_real_escape_string($con,
$inputUuid);
        $inputMajor=mysqli_real_escape_string($con,
$inputMajor);
        $inputMinor=mysqli_real_escape_string($con,
$inputMinor);
        $inputLongitude=mysqli_real_escape_string($con,
$inputLongitude);
        $inputLatitude=mysqli_real_escape_string($con,
$inputLatitude);
        $inputDescription=mysqli_real_escape_string($con,
$inputDescription);

        $query="INSERT INTO lyu1403_beacons (owner, Uuid,
Major, Minor, Longitude, Latitude, Description) VALUES
('$inputOwner', '$inputUuid', '$inputMajor', '$inputMinor',
'$inputLongitude', '$inputLatitude', '$inputDescription)";

        if (mysqli_query($con, $query)) {
            echo '<div class="alert alert-success"
role="alert">Insertion succeeded.</div>';
        }
        else {
            die(mysqli_error($con));
        }
    ?>

```

```

        <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-success"
autofocus>Dashboard</button></a>
        <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/add.php"><button
type="button" class="btn btn-primary">Add more</button></a>
        </div>
    </body>
</html>

```

modify.php - a page to modify beacon details after entering the beacon ID in edit.php

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta content="IE=edge" http-equiv="X-UA-Compatible">
    <meta content="width=device-width, initial-scale=1"
name="viewport">

    <title>Beacon Manager</title><!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins)
-->

    <script src="js/jquery-2.1.1.min.js"></script>
<!-- Include all compiled plugins (below), or include
individual files as needed -->

```

```

<script src="js/bootstrap.min.js"></script>
<!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page
via file:// -->
<!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/s>script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
<![endif]-->
</head>

<body>
    <div class="container">
        <?php
            include 'include.php';

            ini_set('display_errors',1);
            ini_set('display_startup_errors',1);
            error_reporting(-1);
            $con=new
mysqli("appsrvdb.cse.cuhk.edu.hk","viewtech","G5XYcFhy","viewte
ch");

            if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " .
mysqli_connect_error();
            }

            echo '<h1>Edit a beacon</h1>';

```

```

$inputId=$_GET["inputID"];

$inputId=htmlspecialchars($inputId, ENT_QUOTES);

$inputId=mysqli_real_escape_string($con, $inputId);

$result=mysqli_query($con, "SELECT * FROM
lyu1403_beacons WHERE id='" . $inputId . "'");

$row=mysqli_fetch_array($result);

echo '
<form role="form" action="finish_modify.php"
method="get">
    <div class="form-group">
        <label for="inputOwner">OWNER</label>
        <input class="form-control" id="inputOwner"
name="inputOwner" value=' . $row['owner'] . ' required
readonly>
    </div>
    <div class="form-group">
        <label for="inputID">ID</label>
        <input type="number" min="0" max="99999"
class="form-control" id="inputID" name="inputID" value=' .
$row['ID'] . ' required readonly>
    </div>
    <div class="form-group">
        <label for="inputUuid">UUID</label>
        <input class="form-control" id="inputUuid"
name="inputUuid" value=' . $row['uuid'] . ' required autofocus>
    </div>
    <div class="form-group">
        <label for="inputMajor">MAJOR</label>

```

```

        <input type="number" min="0" max="99999"
class="form-control" id="inputMajor" name="inputMajor"
value=' . $row['Major'] . ' required>
    </div>
    <div class="form-group">
        <label for="inputMinor">MINOR</label>
        <input type="number" min="0" max="99999"
class="form-control" id="inputMinor" name="inputMinor"
value=' . $row['Minor'] . ' required>
    </div>
    <div class="form-group">
        <label for="inputLatitude">LATITUDE</label>
        <input class="form-control" id="inputLatitude"
name="inputLatitude" value=' . $row['Latitude'] . ' required>
    </div>
    <div class="form-group">
        <label for="inputLongitude">LONGITUDE</label>
        <input class="form-control" id="inputLongitude"
name="inputLongitude" value=' . $row['Longitude'] . ' required>
    </div>
    <div class="form-group">
        <label
for="inputDescription">DESCRIPTION</label>
        <input type="text" class="form-control"
id="inputDescription" name="inputDescription" value=""; echo
$row['Description']; echo "placeholder="optional">
    </div>
    <div class="form-group">
        <label for="inputPublicDescription">PUBLIC
DESCRIPTION</label>
        <input type="text" class="form-control"
id="inputPublicDescription" name="inputPublicDescription"
value=""; echo $row['publicDescription']; echo

```

```

'"placeholder="optional">
    </div>
    <div class="form-group">
        <label for="inputPublicTitle">PUBLIC
TITLE</label>
        <input type="text" class="form-control"
id="inputPublicTitle" name="inputPublicTitle" value="'; echo
$row['publicTitle']; echo '"placeholder="optional">
    </div>
    <div class="form-group">
        <label for="inputPublicSubtitle">PUBLIC
SUBTITLE</label>
        <input type="text" class="form-control"
id="inputPublicSubtitle" name="inputPublicSubtitle" value="';
echo $row['publicSubtitle']; echo '"placeholder="optional">
    </div>
    <button type="submit" class="btn btn-
success">EDIT</button>
    <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/pop.php?inputID=
'; echo $row['ID']; echo'"><button type="button" class="btn
btn-danger">DELETE</button></a>
    <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-default">CANCEL</button></a>
    </form>
';
?>
    </div>
</body>
</html>

```

pop.php - a page to remove a beacon from database after entering the beacon ID from
remove.php

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta content="IE=edge" http-equiv="X-UA-Compatible">
    <meta content="width=device-width, initial-scale=1"
name="viewport">

    <title>Beacon Manager</title><!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- jQuery (necessary for Bootstrap's JavaScript
plugins) -->

    <script src="js/jquery-2.1.1.min.js"></script>
    <!-- Include all compiled plugins (below), or include
individual files as needed -->

    <script src="js/bootstrap.min.js"></script>
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the
page via file:// -->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
      <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
```

```

        <![endif]-->
</head>

<body>
    <div class="container">
        <?php
            ini_set('display_errors',1);
            ini_set('display_startup_errors',1);
            error_reporting(-1);
            $con=new
mysqli("appsrvdb.cse.cuhk.edu.hk","viewtech","G5XYcFhy","viewte
ch");
            if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " .
mysqli_connect_error();
            }

            $inputID=$_GET['inputID'];

            $inputID=htmlspecialchars($inputID, ENT_QUOTES);

            $inputID=mysqli_real_escape_string($con, $inputID);

            $query="DELETE FROM lyu1403_beacons WHERE
ID=$inputID";
            if (mysqli_query($con, $query)) {
                echo '<div class="alert alert-success"
role="alert">Removal succeeded.</div>';
            }
            else {
                die(mysqli_error($con));
            }
        ?>
    </div>

```

```

        <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-success"
autofocus>Dashboard</button></a>
        <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/remove.php"><but
ton type="button" class="btn btn-primary">Remove
more</button></a>

    </div>
</body>
</html>

```

preview.php - a page accepting beacon identification information, including UUID, major and minor, for previewing a beacon details

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta content="IE=edge" http-equiv="X-UA-Compatible">
    <meta content="width=device-width, initial-scale=1"
name="viewport">

    <title>Beacon Manager</title><!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins)
-->

    <script src="js/jquery-2.1.1.min.js"></script>

```

```

<!-- Include all compiled plugins (below), or include
individual files as needed -->

    <script src="js/bootstrap.min.js"></script>
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page
via file:// -->
    <!--[if lt IE 9]>
        <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
        <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
    <![endif]-->
</head>

<body>
    <div class="container">
        <?php
            include 'include.php';
        ?>

        <h1>Preview a card</h1>
        <form role="form" action="display.php" method="get">
            <div class="form-group">
                <label for="owner">OWNER</label>
                <input class="form-control" id="owner"
name="owner" readonly>
                <label for="uuid">UUID</label>
                <input class="form-control" id="uuid"
name="uuid" required autofocus>

```

```

        <label for="major">MAJOR</label>
        <input type="number" min="0" class="form-
control" id="major" name="major" required>
        <label for="minor">MINOR</label>
        <input type="number" min="0" class="form-
control" id="minor" name="minor" required>
    </div>
    <button type="submit" class="btn btn-
success">Preview</button>
    <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-default">Cancel</button></a>
</form>
</div>

<script>
    $("#owner").val
("97d9e7b3f06ef1234e15613ebb6663eb");
    $("#uuid").val
("B9407F30F5F8466EAFF925556B57FE6D");
    $("#major").val ("5555");
    $("#minor").val ("6666");
</script>
</body>
</html>

```

remove.php - a page to remove a beacon from database by first accepting a beacon ID

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="IE=edge" http-equiv="X-UA-Compatible">
  <meta content="width=device-width, initial-scale=1"
name="viewport">

  <title>Beacon Manager</title><!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins)
-->

  <script src="js/jquery-2.1.1.min.js"></script>
<!-- Include all compiled plugins (below), or include
individual files as needed -->

  <script src="js/bootstrap.min.js"></script>
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page
via file:// -->
  <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"><
/script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scr
ipt>
  <![endif]-->
</head>
```

```

<body>
  <div class="container">
    <?php
      include 'include.php';
    ?>

    <h1>Remove a beacon</h1>
    <form role="form" action="pop.php" onsubmit="return
confirm('Remove beacon
#+document.getElementById('inputID').value+' from
database?');" method="get">
      <div class="form-group">
        <label for="inputID">ID</label>
        <input type="number" min="1" class="form-
control" id="inputID" name="inputID" required autofocus>
      </div>
      <button type="submit" class="btn btn-
danger">Remove</button>
      <a
href="http://appsrv.cse.cuhk.edu.hk/~phng2/fyp/"><button
type="button" class="btn btn-default">Cancel</button></a>
    </form>
  </div>
</body>
</html>

```

Limitations

No background scans

As mentioned above, Bluetooth scans drain battery. We have come up with two possible solutions to the battery drain, however, the solutions are palliative. The first one is to reduce the scan frequency in the background. Yet, reducing the scan frequency implies a huge reduction in the accuracy in location determination - we can imagine the situation would be like locating through WiFi. The second solution is to scan on demand. The application will only scan when the user manually requests the application to scan for beacons nearby. However, this application, as an exhibition navigation system, targets the general public, not Bluetooth developers. Therefore, we have to make the application as easy to use as possible. We have to minimize the number of commands and encapsulate all technical details to keep everything user-friendly. Thus, we will not add any 'manual scan' buttons to the user interface.

As a result, these two solutions have been turned down by our team. Beacons will be scanned only when the application is in foreground.

Beacon Distances, Noises and Interferences

The quality of scanning largely depends on the signals emitted by the beacons. After our investigation on the waves emitted by the beacons, there are huge differences in the accuracy of location estimation of different methods of beacons deployment.

Floor and ceiling, neither good

Not to be deployed on the floor

From the experiments above, we have found that concrete, wooden and turf both absorb signals emitted by Bluetooth beacons. The received RSSI (signal power level) by mobile phones drastically decreases and the estimate distance from the beacon would be further than expected.

Not to be deployed on ceilings

One location that we have placed beacons onto is the lamp shield. However, lamp shields are made of metal and are in cylindrical shapes. Metals reflect beacon signals so well that multiple signals are apparently produced from one source, causing interferences of signals. This might result in amplification of signals and received power will be inaccurate.

Neither too close, nor too far

Too close

As beacons are often deployed on walls or ceilings, placing the phone near the beacon (and hence near walls or ceilings) causes unwanted errors. The errors are induced by the reflections by walls and ceilings and two identical waves may be produced by the reflection, causing undesirable scan results.

Too far

According to one of the specification pages of the AltBeacon library, the accuracy of beacons that are far away is far lower than nearby beacons - a beacon that is 20 meters away may result in an estimation of distance between 10 to 40 metres [26]. The error happens when there are large objects blocking the waves emitted by the beacon or wave amplitude amplified by smooth surfaces, like walls.

Background noises

Though BLE implements frequency hopping to prevent interferences with other bands used by other devices, there is a possibility of having a very short time that two devices are using the same frequency. This might happen when two Bluetooth devices are turned on at exactly the same time and they emit waves at the same frequency or the devices are turned on at different times and emit waves at two distinct frequencies. Both situations cause two devices fighting to use the same band at the same instance and cause interferences.

More is less

Deploying more beacons does not always implies we get more accurate results - we still have to deploy them in suitable distances. Placing two beacons too close causes interferences and reduces accuracy.

Not on the same wavelength

When the frequency of the scan is different from the emission frequency, there is a possibility that at some time the phone cannot find any beacon. If we still take this failed scan result in approximating the distance, this odd value largely affects the moving distance estimated by the phone, and hence, affects the scan accuracy. The good news is, the emission frequency by the beacon can be adjusted manually to match the frequency of the scanning device.

Different scanning devices

Let us take Android as an example here. We see a great diversity of specifications in various Android devices - they have very different screen resolutions, memories, processing power and Bluetooth antennas. The huge difference in hardwares of Android devices makes the signal strength detected versus the actual distance to the beacon a huge discrepancy between devices. Not to mention the material of the casing of the phone affects the detected signal strength, the diverse of types of devices make calibration of the phones a huge work to the deployment project. Currently, the only solution for more accurate results is to test the beacons with different phones in different distances with different beacons and determine a formula for finding the distance for that phone.

Publicly available IDs

We can think of beacons are simply Bluetooth devices or WiFi access points - their names are visible to everyone, no authentication is required to read these information, nor these information is encrypted. Bad guys can scan for the UUIDs, majors and minor of the beacons and replicate these values to his own beacons to trick your customers believing they are in the right location. The consequence might lead to an increase of server load if you pull data from the database every time you scan for a beacon, or, more seriously, creates loss to the company if you distribute gifts coupons through on the detection of beacons. There is currently no solution to this kind of attack and we suggest that beacons should not be used in processing sensitive information like personal details.

Preview of Future Works

Providing an Android application and a web interface is not the end of our work - we aim to extend our project further. Features shown below are what we have already planned and started:

Multiple user CMS

We have already revamped the interface of the Web CMS that supports multiple users. Each user has their own account to manage their beacons easily. Instead of typing IDs or long commands, users click on beacons to delete or modify them. Users can also filter their beacons according to different locations that they are deployed. Furthermore, they can also prepare information required for the next exhibition beforehand without modifying current data. Also, we have implemented a backup function so that exhibition managers can restore from them in case of loss or corruption of data. With these easy to use features, we are sure that everyone can manage their exhibition information system without a hassle.

Extend API

We have also taken care for those who are geeky! A REST API has already been implemented to return useful information of a beacon, such as its latitude and longitude and a user's quota on querying beacons information, through a JSON object. Some of the function calls are already available, such as getLocationById and getLocationByUuidMajorMinor. We will extend our API further by providing more functions such as implementing setters and enhancing getters. Please refer to <https://appsrv.cse.cuhk.edu.hk/~phng2/fyp/api/docs/> for documentation and details.

More accurate positioning determination

Currently, the beacon deployment method is one beacon per exhibit. It is useful when this deployment method is used in museums or trade fairs. Though we are able to determine locations through the power levels of at least 3 beacons detected, the fluctuation of the figures is high. Hence, the location determined may not be always very accurate. However, we want to generalize the use of the system by deploying beacons on suitable locations of each room instead on each object or a fixed position and enhance the methods of location estimation. Through these methods of deployment, we expect a more accurate result of location determination can be obtained from either our web service or Android App. We will also look into some estimation of location by trilateration with height of beacons deployed taken into concern and hope to reduce the errors generated by background noise or interferences.

Acknowledgement

Our group would like to show our deepest gratitude to the project supervisor, Professor Michael Lyu for providing valuable suggestions and encouragement on our research project to make sure everything goes well. We would also like to thank Professor Irwin King for guidance in developing the open source indoor location system, which provides inspirations for our direction in the next term. Last but not least, we would also like to thank Mr Edward Yau and Mr Tsz Lung Un of the ViewLab for their opinions and help on the project. This project is not made possible without their advice.

Bibliography

- [1] U.S. Government, "GPS Accuracy," 18 September 2014. [Online]. Available: <http://www.gps.gov/systems/gps/performance/accuracy/>. [Accessed November 2014].
- [2] Washington Post, "What is a cell tower's range?," 27 June 2014. [Online]. Available: http://www.washingtonpost.com/local/what-is-a-cell-towers-range/2014/06/27/a41152ce-fe3b-11e3-b1f4-8e77c632c07b_graphic.html. [Accessed November 2014].
- [3] Hello Bar, "Infographic: RFID vs. NFC," [Online]. Available: <http://blog.atlasrfidstore.com/near-field-communication-infographic>. [Accessed November 2014].
- [4] Kontakt.io, "Kontakt.io iBeacon compatible solution," [Online]. Available: <http://kontakt.io/>. [Accessed November 2014].
- [5] IndoorAtlas, "IndoorAtlas," [Online]. Available: <https://www.indooratlas.com/>. [Accessed November 2014].
- [6] Anssi Kempainen of University of Oulu; J. Haverinen, "Global indoor self-localization based on the ambient magnetic field," October 2009.
- [7] Radius Networks, "Radius Networks | Proximity Services for Businesses and Developers," [Online]. Available: <http://www.radiusnetworks.com/>. [Accessed November 2014].
- [8] Radius Networks, "Why does my Android phone get such erratic distance measurements? - Radius Networks," 19 August 2014. [Online]. Available: <https://radiusnetworks.zendesk.com/hc/en-us/articles/203055040-Why-does-my-Android-phone-get-such-erratic-distance-measurements->. [Accessed November 2014].
- [9] City Gallery, "City Gallery," [Online]. Available: <http://www.citygallery.gov.hk/en/index.html>. [Accessed November 2014].
- [10] Planning Department of the Government of the Hong Kong Special Administrative Region, "CityGallery 展城館 on the App Store on iTunes," [Online]. Available: <https://itunes.apple.com/us/app/citygallery-zhan-cheng->

guan/id638601756?ls=1&mt=8. [Accessed November 2014].

- [11] Planning Department, "City Gallery - Android Apps on Google Play," [Online]. Available:
<https://play.google.com/store/apps/details?id=com.tda.citygallery&rdid=com.tda.citygallery>. [Accessed November 2014].
- [12] Mashable, "Macy's Flagship New York Store Gets Indoor GPS Navigation," [Online]. Available: <http://mashable.com/2012/11/08/macys-indoor-gps/>. [Accessed November 2014].
- [13] Meridian, "Meridian | Build a better mobile app," [Online]. Available: <http://meridianapps.com/>. [Accessed November 2014].
- [14] Macys Inc, "Macy's on the App Store on iTunes," [Online]. Available: <https://itunes.apple.com/us/app/macys/id341036067?mt=8>. [Accessed November 2014].
- [15] Macy's Inc, "Macy's - Android Apps on Google Play," [Online]. Available: <https://play.google.com/store/apps/details?id=com.macys.android&hl=en>. [Accessed November 2014].
- [16] LINE Corporation, "LINE Maps for Indoor, the App Specializing in Indoor Navigation for Shopping Malls and Department Stores, is Released on Android Devices," 12 November 2014. [Online]. Available: <http://linecorp.com/en/pr/news/en/2014/869>. [Accessed November 2014].
- [17] LINE Corporation, "LINE Maps for Indoor - Android Apps on Google Play," 10 November 2014. [Online]. Available: <https://play.google.com/store/apps/details?id=jp.linecorp.LINEMAPS>. [Accessed November 2014].
- [18] Adafruit, "GATT | Introduction to Bluetooth Low Energy | Adafruit Learning System," 26 March 2014. [Online]. Available: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>. [Accessed November 2014].
- [19] "Profiles Overview | Bluetooth Development Portal," 2014. [Online]. Available: <https://developer.bluetooth.org/TechnologyOverview/Pages/Profiles.aspx>. [Accessed November 2014].

- [20] Radio-Electronics.com, "Bluetooth Radio Interface, Modulation & Channels :: Radio-Electronics.com," [Online]. Available: <http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php>. [Accessed November 2014].
- [21] Bluetooth, "The Low Energy Technology Behind Bluetooth Smart," 2014. [Online]. Available: <http://www.bluetooth.com/Pages/low-energy-tech-info.aspx>. [Accessed November 2014].
- [22] Taobao.com, "ibeacon_淘寶搜索," 2014. [Online]. Available: http://s.taobao.com/search?q=ibeacon&js=1&stats_click=search_radio_all%253A1&initiative_id=staobaoz_20141116. [Accessed November 2014].
- [23] Android, "Bluetooth Low Energy | Android Developers," [Online]. Available: <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>. [Accessed November 2014].
- [24] Bluetooth, "Characteristics | Bluetooth Development Portal," 2014. [Online]. Available: <https://developer.bluetooth.org/gatt/characteristics/Pages/CharacteristicsHome.aspx>. [Accessed November 2014].
- [25] P. H. Ng and T. W. Yeung, "Radar REST API Documentation," October 2014. [Online]. Available: <https://appsrv.cse.cuhk.edu.hk/~phng2/fyp/api/docs/>. [Accessed November 2014].
- [26] Radius Networks, "Android Beacon Library," 2014. [Online]. Available: <http://altbeacon.github.io/android-beacon-library/distance-calculations.html>. [Accessed November 2014].

