

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Automation of Internet of Things Connection

Academic Session: May 2015

I, KHOO KOAY CHIN YAU, declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

9, Lorong Melur 4,
Jalan Raja Uda 12300
Butterworth, Pulau Pinang

Dr Ooi Boon Yaik
Supervisor's name

Date: 1 September 2015

Date 1 September 2015

AUTOMATION OF INTERNET OF THINGS CONNECTION

BY

KHOO KOAY CHIN YAU

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled “**AUTOMATION OF INTERNET OF THINGS CONNECTION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : KHOO KOAY CHIN YAU

Date : 1 September 2015

ACKNOWLEDGEMENT

I would like to express my sincere thanks and appreciation to my supervisor, Dr Alex Ooi Boon Yaik who has given me the opportunity to engage in this project and constantly providing valuable advice to me. Thank you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

A device constantly broadcasting Bluetooth Low Energy packets is developed in order to achieve the automation and streamlining of connection. The seamless connection is also made available by having the client device listening for the BLE packets. The server device was created with Raspberry Pi and the client device used is an Android mobile phone. The only wireless technology needed for connection is Bluetooth.

All that is needed to do is to walk into the proximity of the Bluetooth with the client device, and the connection will be automatically initiated, prompting the client for acceptance. Upon acceptance, the client will be able to communicate with the server device and perform the functions provided. The system is also able to perform simple reporting functions such as recording number of users connected, timestamp of connection and more.

Both the server and the client application was developed in Java for its widespread compatibility with a multitude of devices.

TABLE OF CONTENTS

TITLE	i
DECLARATION OF ORIGINALITY	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATION	x
CHAPTER 1 INTRODUCTION	1
1.1 Motivation and Problem Statement	1
1.2 Project Scope	1
1.3 Project Objectives	2
1.4 Impact, Significance and Contribution	2
1.5 Background Information	3
1.6 Report Organisation	3
CHAPTER 2 LITERATURE REVIEW	4
2.1 Apple iBeacon	4
2.2 Near Field Communication	5
2.3 Wi-Fi Direct	6
2.4 BYOD Public System	7
2.5 Critical Remarks of Previous Works	8

CHAPTER 3 SYSTEM DESIGN	9
3.1 System Approach	9
3.1 Methodology	10
3.2 Use Case Diagram	11
3.3 Use Case Specification	12
3.4 Activity Diagram	17
3.5 Sequence Diagram	19
3.6 Technology Involved	20
3.7 Implementation Issues and Challenges	20
3.8 Timeline	21
CHAPTER 4 IMPLEMENTATION AND TESTING	22
4.1 Tools	22
4.2 Requirements	23
4.3 Specification	24
4.3.1 Analysis	24
4.3.2 Design	24
4.3.3 Verification Plan	24
4.4 Implementation and Testing	25
4.4.1 Client's Implementation	25
4.4.2 Server's Implementation	27
4.4.3 Testing	30

CHAPTER 5 CONCLUSION	34
5.1 Project Review	34
5.2 Discussion	34
5.3 Conclusion	35
5.4 Future Work	35
REFERENCES	36

LIST OF TABLES

Table Number	Title	Page
Table 2.5.1	Comparison of previous works	8
Table 3.4.1	Use Case Connect	12
Table 3.4.2	Use Case Navigation	13
Table 3.4.3	Use Case Retrieve Information	14
Table 3.4.4	Use Case System Restart	15
Table 3.4.5	Use Case System Shutdown	16
Table 4.1.1	Laptop Specification	22
Table 4.1.2	Mobile Device Specification	22
Table 4.1.3	Server Specification	22
Table 4.3.1	Verification Plan	24
Table 4.4.1	Use-Case Testing 1	30
Table 4.4.2	Use-Case Testing 2	31
Table 4.4.3	Use-Case Testing 3	31
Table 4.4.4	Use-Case Testing 4	32
Table 4.4.5	Use-Case Testing 5	32
Table 4.4.6	Testing Result	33

LIST OF FIGURES

Figure Number	Title	Page
Figure 3.3.1	Use Case Diagram	11
Figure 3.5.1	Client Activity Diagram	17
Figure 3.5.2	Server Activity Diagram	18
Figure 3.6.1	Sequence Diagram	19
Figure 3.9.1	Gantt Chart	21

LIST OF ABBREVIATION

<i>IoT</i>	Internet of Things
<i>IP</i>	Internet Protocol
<i>QR code</i>	Quick Response code
<i>URL</i>	Uniform Resource Locater
<i>PAN</i>	Personal Area Network
<i>Bluetooth LE</i>	Bluetooth Low Energy/ Bluetooth 4.0
<i>OS</i>	Operating System
<i>DHCP</i>	Dynamic Host Configuration Protocol
<i>NAT</i>	Network Address Translation

Chapter 1: Introduction

1.1 Motivation and Problem Statement

The Internet of Things (IoT) is a phenomenon where everyday objects are paired with a unique identifier that enables them to communicate with each other without the need of human interaction. It has often been depicted on recent media as the next mega trend. To put it simply, much like how the World Wide Web connects the world together where every computer is connected to each other, the Internet of Things does so in an even finer manner – everything from home, car to wearable is connected to the internet and each other. This greatly simplifies our life as the sensors are able to communicate with each other and automate tedious manual tasks, bringing us countless possibilities.

That being said, the IoT still needs to establish connection with each other in order to communicate and work properly. IoT's connectivity is the main issue due to the slow adoption of IPV6 because there isn't nearly enough identifiers available for all the devices with the existing IPV4. Moreover, there are plenty of areas where a stable internet connection isn't widely available which further diminishes the usability of IoT.

Works need to be done in order for them to connect properly to each other and communicate as intended.

1.2 Project Scope

The project aims to develop an automation of connection by utilising existing technology if possible in order to maximise compatibility between devices. This is done by creating a connectivity platform using the Bluetooth wireless technology. After the completion of project, an Android device is able to connect to a server automatically, requiring only a permission prompt from the client. The project also explores possible real world application and provides an example in doing so. Both the application for the mobile device and server is developed in Java. It works under Linux because the intended platform of server deployment is a Raspberry Pi.

1.3 Project Objective

The connectivity platform, after completion, must be able to provide the following functions:

- a) Broadcast Bluetooth communication to nearby devices.
- b) Detect Bluetooth communication when in proximity.
- c) Accept or reject the Bluetooth communication.
- d) Communicate between the server and client.

By the end of the project, a real world application will be developed with the connectivity platform to demonstrate its capabilities.

1.4 Proposed Approach

The proposed approach used in the project is iBeacon transmission and IP over Bluetooth via Bluetooth Personal Area Network.

iBeacon transmission is used to broadcast advertisement which the nearby mobile devices will detect. The mobile devices can then pair with the server device and join the Bluetooth PAN in order to obtain IP address. This is known as Bluetooth Network Access Point. With an attached IP address through Bluetooth, the client mobile devices can then communicate with the server for further interactions.

1.5 Impact, Significance and Contribution

The connectivity platform developed for the project will benefit users in terms of convenience in connecting between devices. By eliminating the hassle of manual connection, paired with automatic listening of client, the users don't have to be aware of the location of server, they can move about with their life and receive notification for connection when they are in proximity. This opens up various possibilities for location-based applications that targets short-range communication.

1.6 Background Information

Connectivity between Internet of Things is usually Wi-Fi, due to its wide adoption nature. However, since Bluetooth is just as widely known to the public as Wi-Fi, it is a suitable alternative to Wi-Fi as a connection between Internet of Things.

In order for devices to communicate to each other, a connection has to first be established between the devices. Data can then be transferred back and forth between the devices to be processed as information that will ultimately benefit the users. For example, data gathered from sensors found on wearable such as a smart watch can be transferred to a smartphone for processing. The results can be seen in the smartphone that notify users of their heart rate, daily activity and so on. The connectivity in this case, is the Bluetooth connection between the smart watch and the smartphone.

The technology used in the project, Bluetooth Low Energy, is the newest iteration of Bluetooth that is also known as Bluetooth Smart and Bluetooth 4.0. Unlike previous Bluetooth versions, BLE has an advertising mode that is able to broadcast their status to nearby devices that has their Bluetooth enabled. BLE is also capable of operating with reduced power consumption, which is very suitable for small devices like IoT.

1.7 Report organisation

This report starts with the introduction of the project, including the motivation and problem statement, project scope, project objective, proposed approach, impact, significance and contribution and background information. These are followed by literature reviews, system design and system implementation and testing. This report will be concluded with the conclusion of the project.

Chapter 2: Literature Review

2.1 Apple iBeacon

iBeacon is an Apple technology that introduces new proximity sensing capabilities to the mobile phones. It allows both iOS and Android application to alert its users when it came into contact with nearby beacons. This works by having the mobile phones listening for signals from Bluetooth Low Energy advertising mode. BLE advertising mode works by broadcasting its data packet at a regular interval to nearby devices, triggering events on the listening devices such as notifications or pop up information.

Apple iBeacon is a Bluetooth broadcasting standard that follows a specific format that must contain an iBeacon prefix, UUID, Major, Minor, and Tx Power.

- iBeacon prefix is a fixed 9 bytes data that indicates that this transmission is actually originated from an iBeacon device.
- UUID is a 16 bytes string that is used as an identification in order to differentiate itself from the others. UUID is meant to categorise a large group of beacons.
- Major is a smaller subset of differentiation from the UUID that is only 2 bytes long. Major is used to distinguish a small group of beacons within the UUID.
- Minor is an individual identifier that pin points the location of the beacon. Among a group of related beacons, each and every beacon has a unique Minor 2-bytes long string, but contains the same UUID and Major.
- Tx Power is the strength of the Bluetooth transmission. It is used to determine the distance to the Apple iBeacon.

2.2 Near Field Communication

Near Field Communication (NFC) is a close-range communication technology that utilizes electromagnetic radio fields to transfer data between two devices.

Contactless in nature, NFC can exchange information and communicates with each other wirelessly, without touching each other. Designed for use in close proximity, NFC's approximate range of operation is 10cm.

NFC typically communicates between a NFC enabled mobile phone and a NFC tag encoded with desired information, but some communicates between two mobile phones as well. Coming upon close proximity with the tag, the mobile phone will launch the programmed responses such as directs to an URL, pushes a message or opens an application.

NFC devices have two states, passive and active. In passive or half-duplex form, NFC devices can transmit information but do not read any information themselves. On the other hand, NFC devices can both read and transmit information in active or full-duplex form, not to mention exchange information between compatible devices and alter information residing in a NFC tag.

Often applauded for its simplicity, NFC requires only minimal configuration and setup for it to work properly. NFC tags needed only to be encoded with information once, after that it can operate without batteries or any other accessories. NFC tags obtain power by sipping it from an active NFC device such as a mobile phone when the mobile phone approaches the tags.

2.3 Wi-Fi Direct

Wi-Fi Direct is a variant of the Wi-Fi wireless technology that allows devices to connect directly without the need of an access point. This enables devices to connect to each other directly to share information quickly and easily even if there isn't any internet connection available. This is especially useful when travelling at a place without a stable Wi-Fi connection.

Wi-Fi Direct is capable of having a one-to-one connection or a group of simultaneous connection. It also supports service discovery in order to identify devices with desired functionalities.

Developed with convenience in mind, Wi-Fi Direct doesn't require any set-up procedures before using. Connection happens automatically without any password required as well, the user simply has to confirm any incoming connection.

The disadvantages of Wi-Fi Direct is that any open connections will be disconnected when Wi-Fi Direct is turned on. Besides, some manufacturers have their own standard that differs slightly to the others, which may cause incompatible connection.

2.4 BYOD Public System

This paper describes a system where a mobile device is the only controller to the public displays. It is a Bring Your Own Device system to eliminate the need of controllers in order to minimise costs and brings convenience to the users as well as the admin.

The proposed system utilises the BYOD concept by installing a server to the public interactive displays that allows socket-based connection with the mobile devices. Wi-Fi access points are setup in the public displays for connection between the server and mobile devices. After setups are done, the mobile devices only needed to connect to the access point provided by the public display to proceed with the interactions and navigations of the public displays. However, the mobile devices do need to install a separate application in order to connect and communicate with the public displays.

Some of the functionalities provided include uploading a presentation slide into the public display from the mobile device, navigate the public display's presentations, and input text or comments into the public display.

2.5 Critical Remarks of Previous Works

Table 2.5.1 Comparison of previous works

	NFC	Wi-Fi Direct	Apple iBeacon
Range	Short (10cm)	Long (100m)	Medium (50m)
Accuracy	High	Low	Medium (50m)
Engagement	Active	Passive	Passive
Power	No	Yes	Yes
Availability	Low	High	High
Apps	No	No	Yes
Radio	Electromagnetic Radio Field	Radio transmission	Radio transmission

NFC is by far the most accurate technology to be use in IoT connection. However, countless opportunities could be lost due to its limited operation range, rendering the technology useless for application requiring more than 10cm of operating range, which is true in most cases.

Wi-Fi Direct on the other hand, proves to be a great potential connectivity of IoT with its impressive range and high availability among the existing devices. Unfortunately, accuracy plays a major part in the communication between devices as well. Due to its high range, signals are often conflicted among each other causing unstable connection between devices. This is not a problem where the number of devices are low, but wide deployment of IoT is the key aspect in terms of business value, meaning that low accuracy in crowded situations are unacceptable.

Apple iBeacon is a standard of Bluetooth Low Energy advertising mode. Equipped with considerable range, accuracy and availability, it seems that iBeacon is the better choice among the existing solutions. Still, it lacks a mechanism to send large amount of data over a high bandwidth connection because Bluetooth connections are low in bandwidth. This is critical since quick and smooth content transmission is fairly important, especially IoT devices are often deployed on places like shops and stores where users might get impatient and leave before contents are received.

Chapter 3: System Design

3.1 System Approach

Inspired by the work done in BYOD Public System by our fellow UTAR alumni, the connectivity platform proposed in the project will be used to create a similar interactive display to demonstrate the capabilities of the platform.

The system created in the previous work requires a manual Wi-Fi connection to be established between the mobile device and the public display before interaction can occur. Therefore by eliminating that manual connection, the system can be further improved by simplifying the entry of usage for the users.

Other than that, the previous system do not have the capability of notifying the user of its own existence, even though users might be interested in interacting with the display, they might missed it due to the lack of notification. On the other hand, the connectivity platform proposed in the project broadcasts iBeacons constantly to nearby devices. Any devices in proximity with a compatible application will receive a notification and be prompted a connection automatically. This will greatly broadens the usability of the system.

As the users navigate the contents on the public display, information can be pulled down into their own mobile phones for further consumption simply with a touch. The information can be transmit in the form of QR code or URL.

The system can also log all the connections made to the server for analysis purposes. Every mobile devices connected to the server will be recorded by saving their MAC addresses and the timestamp of connection. These data can be converted into graphs and diagrams to determine the users' connection frequency.

3.2 Methodology

The project adopts prototyping as the System Development Model of choice. Prototyping is a great match to the work done in the project as the requirements are constantly updated throughout its lifetime.

Initially, a basic requirement gathering was done to grasp the preliminary understanding of the system. Motivation and problem statement together with several other requirements was identified in this stage, fuelling the development of prototype later on.

The first prototype created uses a different approach compared to the latest iteration, where the main connection used was Wi-Fi instead of Bluetooth. After several iteration of review, analysis, design and implementation, it is decided that Bluetooth pairing is a better option of connection compared to Wi-Fi due to its power-saving capability and ease-of-use, where it doesn't require a password for connection.

Several diagrams were designed and created throughout the development that provide a better view on the working of the system.

3.3 Use Case Diagram

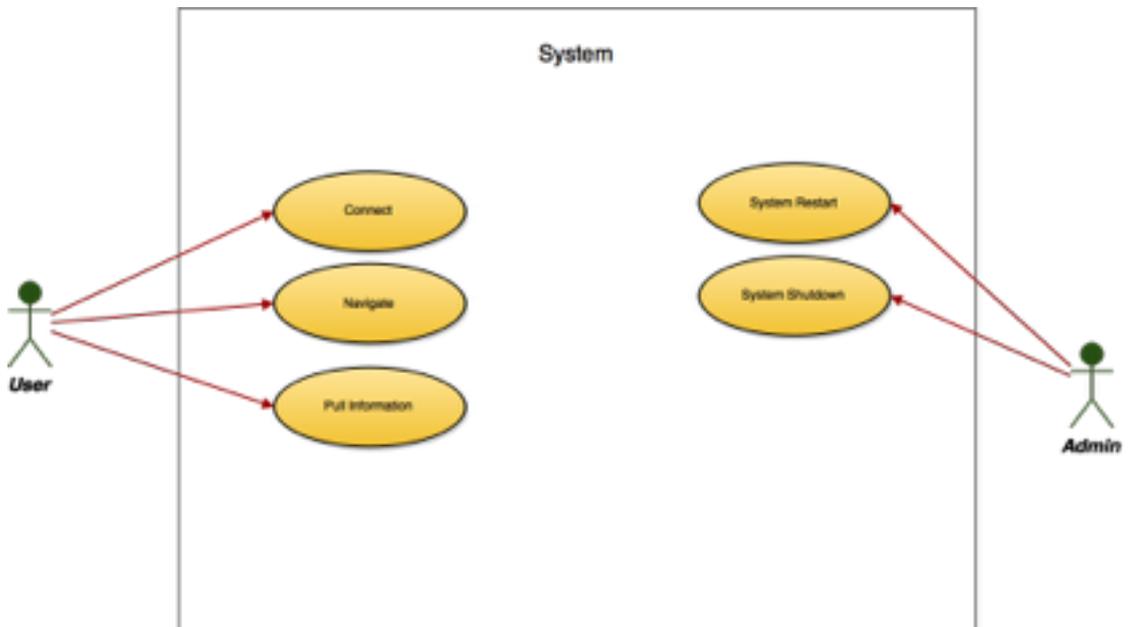


Figure 3.3.1 Use Case Diagram

3.4 Use Case Specification

Table 3.4.1 Use Case Connect

Use Case Name	Connect	Use Case ID	1
Importance Level	High	Use Case Type	Detail, Essential
Actor	User		
Stakeholders and Interests	User - Connect to the server with mobile phones		
Brief Description	Describes how a user connects to the server with mobile phones		
Trigger	User wants to connect to the server		
Relationships	Association	User	
	Include	-	
	Extend	-	
	Generalization	-	
Normal Flow of Events	1	Application detects the Bluetooth beacon.	
	2	User agrees to the Bluetooth PAN connection.	
Sub Flows	Not applicable		
Alternate/ Exceptional Flows	Not applicable		

Table 3.4.2 Use Case Navigation

Use Case Name	Navigation	Use Case ID	2
Importance Level	High	Use Case Type	Detail, Essential
Actor	User		
Stakeholders and Interests	User - Navigate the advertisement with mobile phones		
Brief Description	Describes the navigation of the advertisement with mobile phones		
Trigger	User wants to navigate the advertisement with mobile phones		
Relationships	Association	User	
	Include	-	
	Extend	-	
	Generalization	-	
Normal Flow of Events	1	User touches the on-screen navigators.	
	2	The advertisements changes based on the user's inputs.	
Sub Flows	Not applicable		
Alternate/ Exceptional Flows	Not applicable		

Table 3.4.3 Use Case Retrieve Information

Use Case Name	Retrieve Information	Use Case ID	3
Importance Level	High	Use Case Type	Detail, Essential
Actor	User		
Stakeholders and Interests	User - Retrieve interested information from the advertisement into the mobile phones		
Brief Description	Describes the retrieval of information from the advertisements		
Trigger	User wants to retrieve interested information from the advertisements		
Relationships	Association	User	
	Include	-	
	Extend	-	
	Generalization	-	
Normal Flow of Events	1	User touches the on-screen button.	
	2	Information is downloaded into the mobile device.	
Sub Flows	Not applicable		
Alternate/ Exceptional Flows	Not applicable		

Table 3.4.4 Use Case System Restart

Use Case Name	System Restart	Use Case ID	4
Importance Level	High	Use Case Type	Detail, Essential
Actor	Admin		
Stakeholders and Interests	Admin - Restart the server		
Brief Description	Describes how the admin restart the server		
Trigger	Admin wants to restart the server		
Relationships	Association	Admin	
	Include	-	
	Extend	-	
	Generalization	-	
Normal Flow of Events	1	Admin restarts the server by turning off the power and turning on the power.	
Sub Flows	Not applicable		
Alternate/ Exceptional Flows	Not applicable		

Table 3.4.5 Use Case System Shutdown

Use Case Name	System Shutdown	Use Case ID	5
Importance Level	High	Use Case Type	Detail, Essential
Actor	Admin		
Stakeholders and Interests	Admin - Shutdown the server		
Brief Description	Describes how the admin shutdown the server		
Trigger	Admin wants to shutdown the server		
Relationships	Association	Admin	
	Include	-	
	Extend	-	
	Generalization	-	
Normal Flow of Events	1	Admin shutdown the server by turning off the power.	
Sub Flows	Not applicable		
Alternate/ Exceptional Flows	Not applicable		

3.5 Activity Diagram

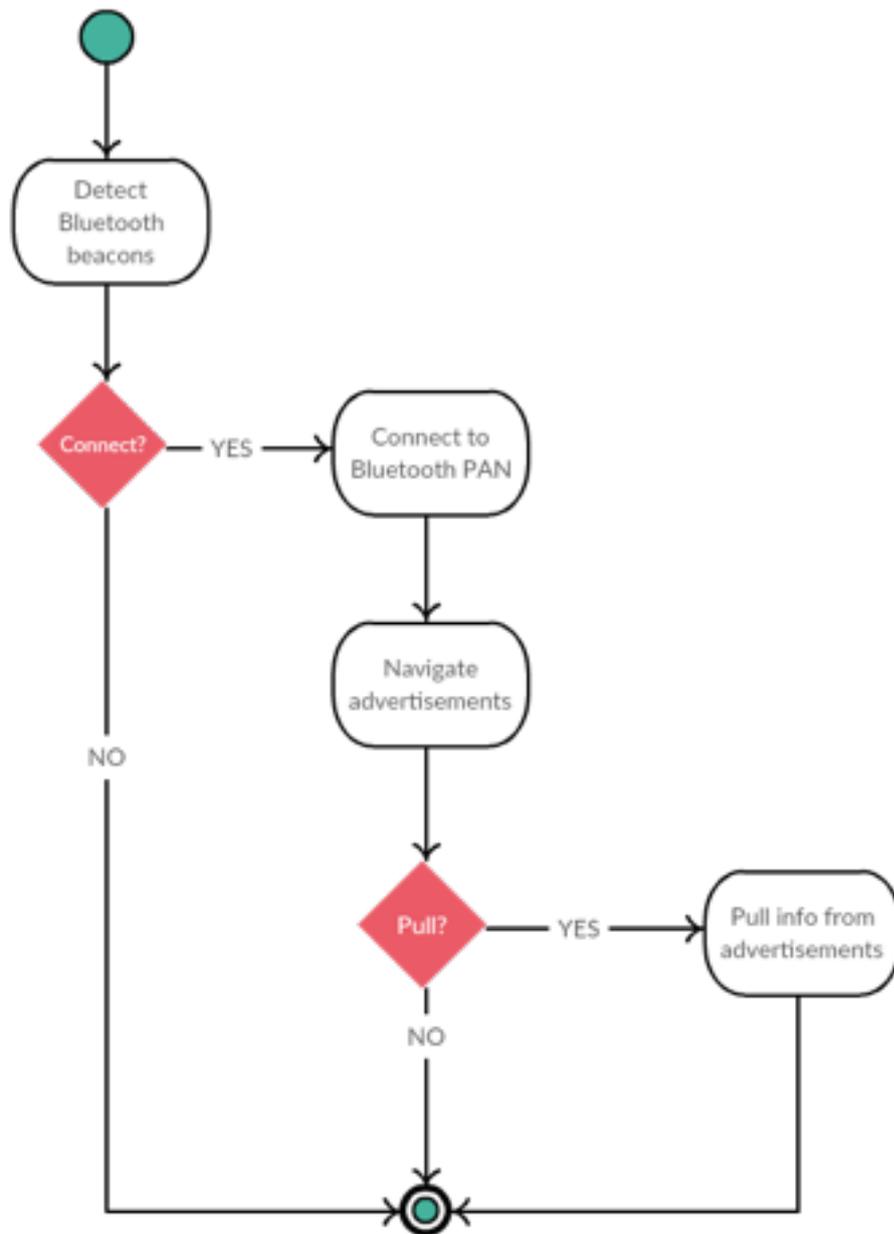


Figure 3.5.1 Client Activity Diagram

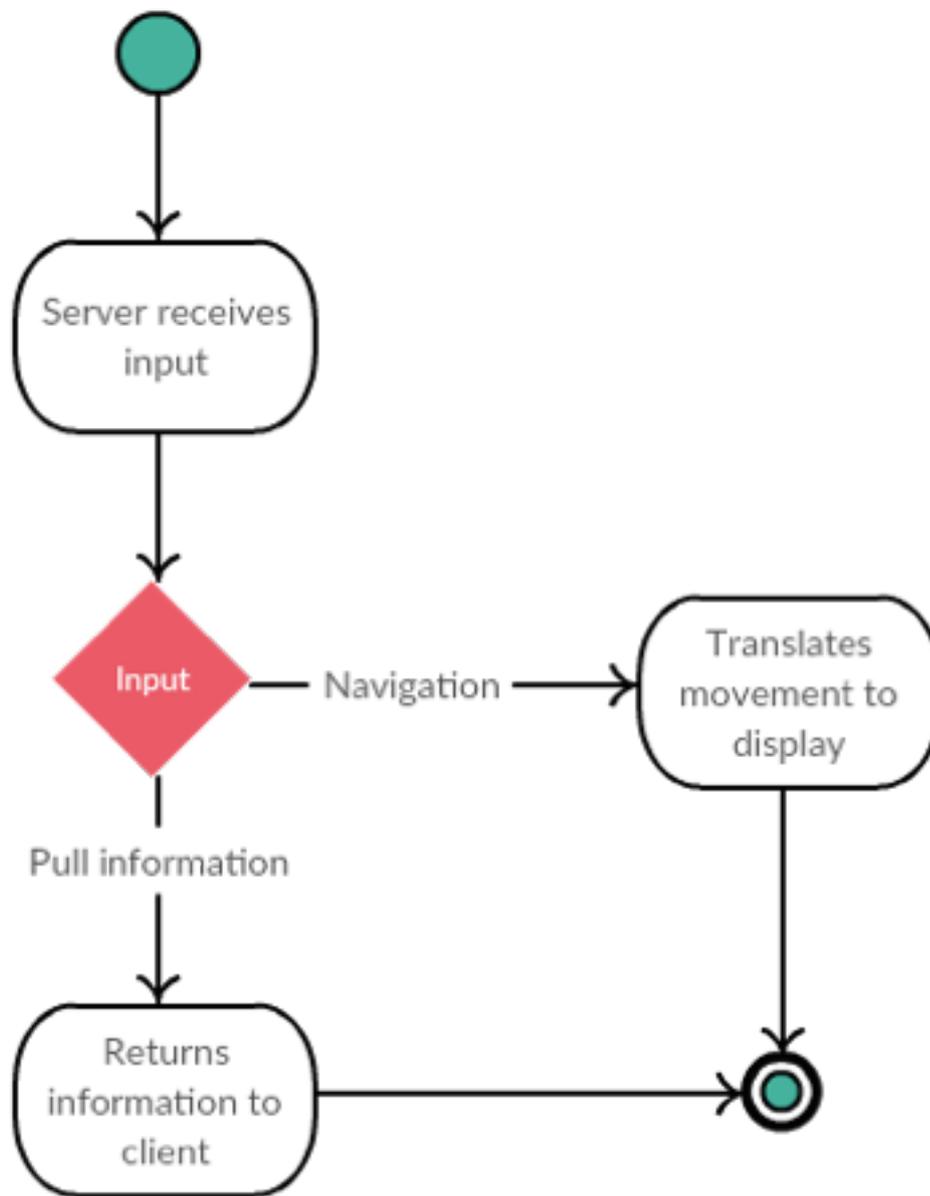


Figure 3.5.2 Server Activity Diagram

3.6 Sequence Diagram

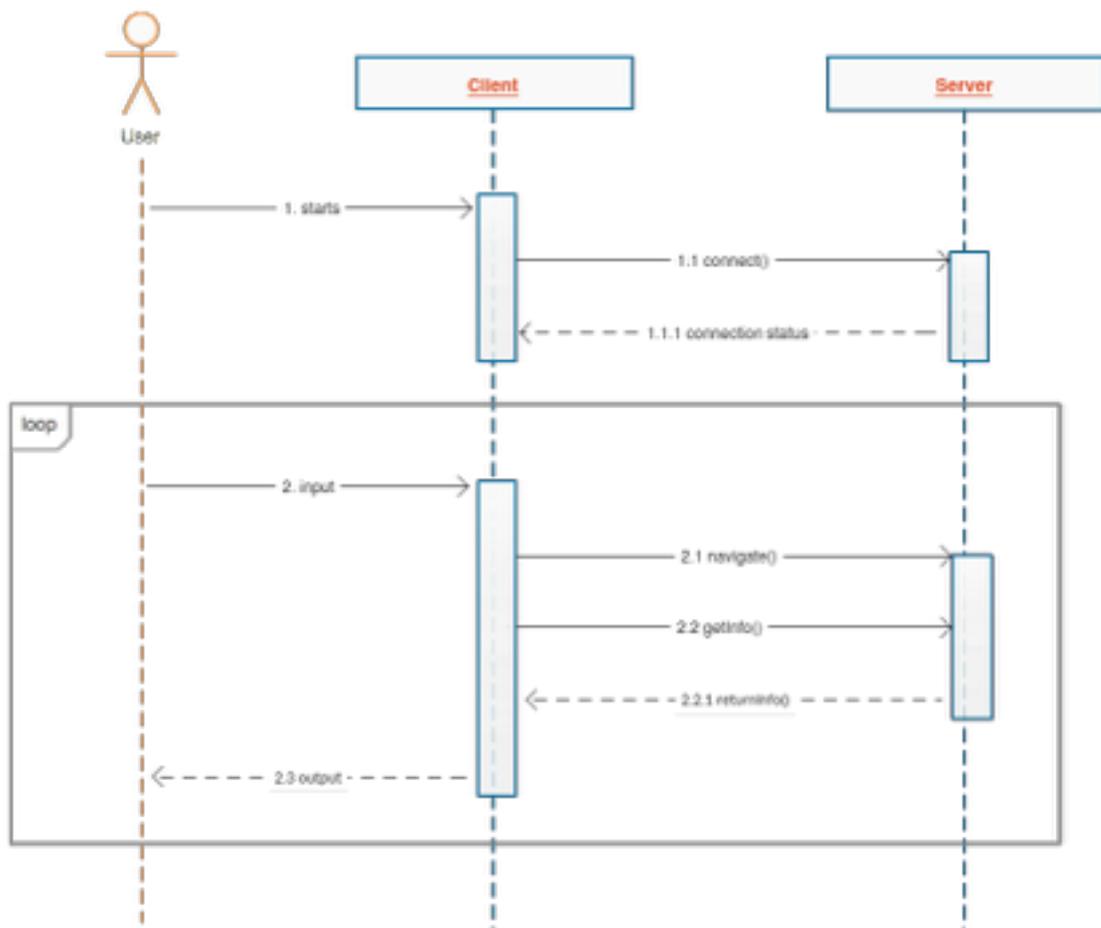


Figure 3.6.1 Sequence Diagram

3.7 Technology Involved

The technology utilised in the project includes Bluetooth, Bash command language, Java programming language, Raspbian Linux operating system and Android mobile operating system.

Several components of Bluetooth were used in the project, namely the Bluetooth Low Energy variant that allows an advertisement mode, known as beacons, and Bluetooth Personal Area Network that allows IP over Bluetooth to access servers.

The Java programming language is a the language of choice in developing both the Linux's server application and the Android's client application.

Development softwares such as Android Studio and Eclipse were used to develop the project.

3.8 Implementation Issues and Challenges

The system developed in the project works fine in tested environment. It is able to detect the Bluetooth beacons and connect to Bluetooth Personal Area Network successfully. Besides that, it is also able to navigate the advertisements after the mobile phones had established connection to the server. Data can be retrieved from the server with no issues.

However, the system do have a few limitation and challenges. The Android OS has to be version 4.4 above to work properly, this is due to Bluetooth LE not been supported in earlier versions of Android OS.

Moreover, the system only allows a single connection between client and server at a time. This can be interpreted both as a limitation or a feature. The display was intended to be navigated by a single user only, therefore this limitation brings no harm to the system.

Lastly, the communication between the client and server is delayed and not instantaneous. This could be the problem of bandwidth and should be fixed if a better connection and hardware is used.

3.9 Timeline

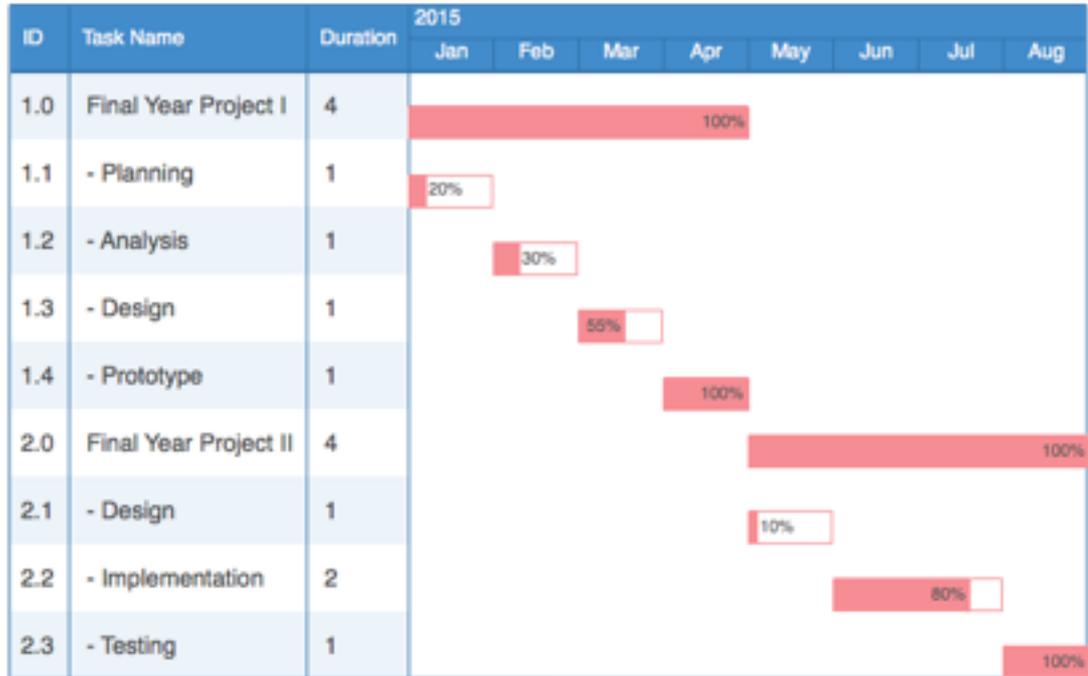


Figure 3.9.1 Gantt Chart

Chapter 4 System Implementation and Testing

4.1 Tools

The hardware used in the project are an Apple MacBook Pro laptop, a Xiaomi Mi4i mobile device, a Raspberry Pi 2 Model B server and a monitor with HDMI port.

Table 4.1.1 Laptop Specification

Processor	2.7 GHz Intel Core i5
RAM	8GB
Operating System	OSX Yosemite
Graphic Card	Intel Iris Graphics 6100

Table 4.1.2 Mobile Device Specification

Processor	Qualcomm Snapdragon 615
RAM	2GB
Operating System	Android OS 5.0 Lollipop
Bluetooth	Bluetooth 4.1

Table 4.1.3 Server Specification

Processor	Quadcore Broadcom BCM2836
RAM	1GB
Operating System	Raspbian Linux OS
Bluetooth	Bluetooth 4.0 CSR dongle

4.2 Requirements

Several requirements of the system are needed, they are as follows:

- i. The mobile client device must have Bluetooth 4.0 or above.
- ii. The mobile client device must have Android OS 4.4 or above.
- iii. The mobile client device must have the Bluetooth internet access function.
- iv. The server device must have a dongle with Bluetooth 4.0 or above.
- v. Any monitor with HDMI port.

If these essential requirements are fulfilled, the mobile client device will be able to communicate with the server by listening to the Bluetooth beacons and connect to the Bluetooth PAN in order to obtain an IP address. The server will be hooked into the monitor and display advertisements for viewing. The advertisements can be navigated by the mobile client after the connections are established. If the users are interested in any of the advertisements, they can retrieve more information regarding that advertisement. The system admins can also retrieve the logs of users' connection to be used as reporting functions.

4.3 Specification

4.3.1 Analysis

The verification plan is carried out. The first verification plan is to test whether the Bluetooth beacon can be detected by the mobile devices. The second and the third is to find out if the mobile devices can be connected to the Bluetooth PAN and navigate the advertisements respectively. Lastly, the functionality to retrieve information from the advertisements is verified.

4.3.2 Design

The mobile application is installed into an Android mobile device. As for the server, it is installed into a Raspberry Pi running the Raspbian Linux OS. The mobile device is responsible to test the detection of Bluetooth beacon and the connection with the server.

4.3.3 Verification Plan

Table 4.3.1 Verification Plan

Verification Plan	Expected Result
1. Detection of Bluetooth beacon	The mobile device will detect the Bluetooth beacon
2. Connection to the server	The mobile device will connect to the Bluetooth PAN of the server
3 Navigation of advertisement	The mobile device will send input to the server for advertisement navigation
4. Retrieval of information	The mobile device will retrieve information from the server advertisement

4.4 Implementation and Testing

The implementations are divided into two parts, the client's and the server's.

4.4.1 Client's Implementation

The very first thing the mobile application needs to do is to detect the Bluetooth beacons. This can be done by utilising the AltBeacons Open Sourced API written by the generous folks at Radius Networks. Since iBeacons is a Bluetooth standard proposed by Apple, there isn't any support of it in Android OS. Therefore a third party open sourced API was used in the project.

```
if (region.matchesBeacon(beacon)) {  
    address = beacon.getBluetoothAddress();  
}
```

If a beacon with matching identifiers are found, its MAC address is recorded to be used later.

```
try {  
    Class<?> classBluetoothPan = Class.forName(className);  
    Constructor<?> ctor =  
        classBluetoothPan.getDeclaredConstructor(Context.class,  
            BluetoothProfile.ServiceListener.class);  
    ctor.setAccessible(true);  
    Object instance = ctor.newInstance(getApplicationContext(), ne  
        BluetoothPanServiceListener(getApplicationContext()));  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

With the MAC address of the server obtained, the mobile device can initiate a connection to the Bluetooth PAN in order to be assigned an IP address.

```

try {
    clientSocket = new Socket(ipAddress, 8381);
    oos = new ObjectOutputStream(clientSocket.getOutputStream());
} catch (Exception e) {
    e.printStackTrace();
}

```

After the connection between the client and server is established, a socket is needed for them to communicate with each other.

```

left.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        try {
            oos.writeObject("LEFT");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

right.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        try {
            oos.writeObject("RIGHT");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

```

In order to navigate the advertisements, the client have to sent these commands to the server. The server will receive the instructions, be it a Left or a Right, and translate them into movements to change the advertisements displayed on the monitor.

4.4.2 Server's Implementation

The server used is a Linux Raspbian Raspberry Pi, however any Debian-Based Linux OS will mostly do fine as well. To implement the servers, it has to be configured properly.

```
sudo apt-get install bluez bluez-utils python-dbus python-gobject
sudo apt-get install isc-dhcp-server
sudo apt-get install bridge-utils
```

Linux uses Bluez Bluetooth stack, therefore it is essential to install it and its relevant utilities. A DHCP server is also required to distribute IP addresses to the mobile devices too. Since bridging between interfaces is needed as well, the bridge-utils is installed.

```
#!/bin/bash

flag="01 1A"
device="hci0"
ogf="0x08"
ocf="0x0008"
manufacturer="FF"
company="4C 00 02 15"

uuid="11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11"
major="00 00"
minor="00 00"
rssi="C8 00"
octet_no="1E"
octet_length="02"
data_length="1A"

sudo hcitool -i $device cmd $ogf $ocf $octet_no $octet_length $flag
$data_length $manufacturer $company $uuid $major $minor $rssi

sudo hciconfig $device leadv
```

First of all, the server needs to start transmitting the Bluetooth beacons. The code shown above does that via Bash commands. The beacons will have a 16 bytes UUID, which was set as “11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11” for simple debugging purposes. With an UUID set, any client device that searches for that particular UUID will be able to detect the server device.

```
#!/bin/bash

sudo brctl addbr pan0 wlan0

sudo ifconfig pan0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up

sudo /etc/init.d/isc-dhcp-server start

sudo iptables -A POSTROUTING -t nat -o wlan0 -j MASQUERADE
```

Next, the Bluetooth Personal Area Network needs to be setup in order to assign the mobile devices with IP addresses through DHCP. Starts by adding a bridge named pan0 that links to the interface wlan0. Next setup a static IP to pan0 so that the mobile devices can communicate to the server easily. Lastly, configure the bridge to forward all packets from private network to public networks through IP masquerading.

```
try {
    server = new ServerSocket(8381, 0);
    while (true) {
        try {
            connection = server.accept();
            output = new
                ObjectOutputStream(connection.getOutputStream());
            input = new
                ObjectInputStream(connection.getInputStream());
            navigate();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            output.close();
            input.close();
            connection.close();
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

Setup the socket server for the communication between the client device and server. This enables the client to navigate the advertisements after the sockets are connected.

```
if (message.equals("LEFT")) {
    try {
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_LEFT);
        r.keyRelease(KeyEvent.VK_LEFT);
    } catch (Exception e) {
        e.printStackTrace();
    }
} else (message.equals("RIGHT")) {
    try {
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_RIGHT);
        r.keyRelease(KeyEvent.VK_RIGHT);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

These are the translation of client's instructions into actual movements. If the server receives these inputs, it will navigate on the advertisements based on them.

4.4.3 Testing

The project applied one of the Black-Box Testing techniques, Use-Case Testing to test and verify the system. Use-Case Testing is done by testing each use-cases to determine if the functionality work as requirement stated.

Table 4.4.1 Use-Case Testing 1

Use Case ID	1	Version	1.0
Feature	F001		
Purpose	To connect to the server		
Actor	User		
Trigger	Detection of Bluetooth beacons		
Precondition	Application is installed in the mobile device		
Scenario Name	Step	Action	
Main Flow	1	The mobile device detected the presence of the Bluetooth beacons.	
	2	The Bluetooth beacons is of matching identifiers.	
	3	A prompt of connection agreement.	
	4	User agreed to connect the Bluetooth PAN.	
	5	The mobile device is connected to the server.	
Alternate Flow - Disagree Connection	4.1	User disagreed to connect to the Bluetooth PAN.	
Rules	None		
Author	Khoo		

Table 4.4.2 Use-Case Testing 2

Use Case ID	2	Version	1.0
Feature	F002		
Purpose	To navigate the advertisements		
Actor	User		
Trigger	User touches the navigation inputs		
Precondition	Mobile device is connected to the server		
Scenario Name	Step	Action	
Main Flow	1	User touches the navigation inputs.	
	2	The advertisement changes.	
Rules	None		
Author	Khoo		

Table 4.4.3 Use-Case Testing 3

Use Case ID	3	Version	1.0
Feature	F003		
Purpose	To retrieve information from the advertisement		
Actor	User		
Trigger	User touches the on-screen button		
Precondition	Mobile device is connected to the server		
Scenario Name	Step	Action	
Main Flow	1	User touches the on-screen button.	
	2	Information is downloaded into the mobile device.	
Rules	None		
Author	Khoo		

Table 4.4.4 Use-Case Testing 4

Use Case ID	4	Version	1.0
Feature	F004		
Purpose	To restart the server		
Actor	Admin		
Trigger	Admin turns off the power		
Precondition	The server is turned on		
Scenario Name	Step	Action	
Main Flow	1	Admin turns off the power.	
	2	Admin turns on the power again.	
Rules	None		
Author	Khoo		

Table 4.4.5 Use-Case Testing 5

Use Case ID	5	Version	1.0
Feature	F005		
Purpose	To shutdown the server		
Actor	Admin		
Trigger	Admin turns off the power		
Precondition	The server is turned on		
Scenario Name	Step	Action	
Main Flow	1	Admin turns off the power.	
Rules	None		
Author	Khoo		

Table 4.4.6 Testing Result

Functionality	Expected Result	Actual Result	Passed	
			Y	N
Transmission of Bluetooth beacons	The server transmits the Bluetooth beacon	The server transmits the Bluetooth beacon	x	
Detection of the Bluetooth beacons	The mobile device detects the Bluetooth beacon	The mobile device detects the Bluetooth beacon	x	
Connection between mobile device and server	The mobile device connects to the server	The mobile device connects to the server	x	
Navigation of advertisement	Advertisements are navigated	Advertisements are navigated	x	
Retrieval of information from advertisements	Information is retrieved from the advertisements	Information is retrieved from the advertisements	x	

Chapter 5: Conclusion

5.1 Project Review

Initially the project was develop with both Bluetooth and Wi-Fi as the means of connecting the server and mobile devices. However, the Wi-Fi connection isn't as easy as it seems, since it needed a passphrase for secure connection which would destroy the purpose of seamless connection for the project. Therefore the project took a turn of direction and changed to connecting via Bluetooth Personal Area Network instead. This provides the mobile devices with IP addresses for communication purposes as well as eliminating the need of inputting passwords for connection.

The objective of the project was successfully met by utilising several technologies such as the Bluetooth Low Energy advertisement mode, Bluetooth Personal Area Network, Bash command lines and Java programming language, together with some of the open sourced API that provides essential functions for the system.

5.2 Discussion

The final deliverable of the project is a connectivity platform that automates connection between compatible devices. The project also developed a proof of concept application that demonstrates the capabilities of the connectivity platform. The application allows a mobile device to detect and connect to a advertisement-displaying server with only a single input, the permission of the user. The user can then proceed to navigate and interact with the advertisement display.

5.3 Conclusion

To conclude, the project introduces an automation of connection between devices such as the Internet of Things. It does so by having the server transmitting Bluetooth Low Energy beacons together with the client listening for it. After detection, the client will connect to the Bluetooth Personal Area Network and obtain IP address via DHCP and access the server through NAT and IP masquerading.

The example of application provided in the project uses the connectivity platform to simplify and automate the connection between a mobile device and the public advertisement display.

5.4 Future Work

Future improvement on the project can be done to strengthen the connection between the devices and increases the functionality of the servers as well as the mobile client.

References:

Galeev, Mikhail. 'Bluetooth 4.0: An Introduction To Bluetooth Low Energy'. *EETimes*. N.p., 2015. Web. 3 Mar. 2015.

Getting Started With Ibeacon. 1st ed. Apple, 2014. Web. 3 Mar. 2015.

Nearfieldcommunication.org,. 'Near Field Communication: What Is Near Field Communication?'. N.p., 2015. Web. 3 Mar. 2015.

Wi-fi.org,. 'Wi-Fi Direct | Wi-Fi Alliance'. N.p., 2015. Web. 3 Mar. 2015.