UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT PAPER

Author's full name : **SAYED MUHAMMAD ARIF FIKRI BIN SAYED ABDULLAH**

Date of Birth : **05TH MARCH 1995**

Title : **IOT-BASED OFFICE MONITORING AND CONTROL SYSTEM**

Academic Session : **2017/2018**

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organisation where research was done)*
☑	**OPEN ACCESS**	I agree that my thesis be published and accessed online (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

1. The thesis is the property of Universiti Teknologi Malaysia

2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.

Certified by:

SIGNATURE	**SIGNATURE OF SUPERVISOR**
950305115351	**DR MOHD ADIB BIN SARIJARI**
(NEW IC NO/PASSPORT)	**NAME OF SUPERVISOR**
Date: 12TH JUNE 2018	Date: 12TH JUNE 2018

NOTES: * If the thesis is CONFIDENTAL or RESTRICTED, please attach the letter from the organisation concerned stating the reason/s and duration for the confidentiality or restriction.

"I hereby declare that I have read this final year project report and in my opinion this report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Engineering (Electrical- Electronic)"

Signature : …….....

Name of Supervisor : DR MOHD ADIB BIN SARIJARI

Date : 12 JUNE 2018

IOT-BASED OFFICE MONITORING AND CONTROL SYSTEM

SAYED MUHAMMAD ARIF FIKRI BIN SAYED ABDULLAH

A report submitted in partial fulfilment of the
requirements for the award of the degree of
Bachelor of Engineering (Electrical – Electronic)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2018

I declare that this report entitled "*IoT-based Office Monitoring and Control System*" is the result of my own research except as cited in the references. The report has been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……

Name : SAYED MUHAMMAD ARIF FIKRI BIN SAYED ABDULLAH

Date : 12 JUNE 2018

"*My dearest parents and my siblings*"

This is for all of you.

ACKNOWLEDGEMENT

In the name of Allah, the most Gracious and the most Merciful, as giving me the strength to finish my thesis. The utmost gratitude to Him for giving me the courage to complete the thesis. First, I wanted to acknowledge Universiti Teknologi Malaysia for a great facility which has providing a good researching experience in completing my thesis.

Next, I would like to express my gratitude to my supervisor of Final Year Project who is Dr Mohd Adib bin Sarijari for his invaluable advice, guidance as well as support in the process to evolve the system in this project. Then, further appreciation especially to all the UTM-Telekom lab group member, Mr. Abdul Hadi Fikri, Mr. Hamdan and Mr. Muhd Aizuddin which has giving the guidance, help and advice in the process to finish the project. Thanks to them because with their support I managed to understand the new technology that was quite awkward for me at the first place and I never expected can finish the project.

Special thanks to my beloved family especially my parent as they never give up in giving the support in many ways to me, not only for this final project but since the beginning I start my study in the Universiti Teknologi Malaysia. Their support and advice are very valuable for me as I can further my study in Bachelor of Engineering (Electrical-Electronics).

I also would like to thank to all my friend and for those who contribute directly or indirectly towards preparing and completing my final year project. Finally, not forgetting to all the person that giving me support and advice since I started my study journey until now.

ABSTRACT

Internet of things (IoT) is a new technology that has received great attention in the recent years. The technology allowed interconnection between smart devices through the Internet network. This work proposed one of IoT applications which is Office Monitoring and Control System. The system will monitor the access to the office and control the electrical devices in the office. The system has the ability to provide secure access method by using Near Field Communication (NFC) technology in which NFC has higher security level system as compared to RFID. Raspberry pi 3 module is used as the gateway and the main controller of the system, while Raspberry Pi Zero W is used to implement the control system. The cloud will be setup to enable the user to access the database and the server from a web-based software, through the gateway device. Next, the Graphical User Interface (GUI) is implemented to provide a user-friendly interaction environment. The problem on how to detect the connection between the client and the server has been found by using the IoT method called *Heartbeat* which can notify the user about the connectivity status between server and client. The work produces the system that can monitor the person enter and exit the building and can control the lighting lamp in the building through IoT platform.

ABSTRAK

Internet-of-Things (IoT) merupakan teknologi baru yang mendapat perhatian yang sangat baik pada tahun-tahun sejak kebelakangan ini. Teknologi ini membenarkan sambungan antara peranti pintar melalui rangkaian internet. Kerja ini mencadangkan satu aplikasi IoT iaitu sistem pengawasan dan kawalan pejabat yang berasaskan IoT. Sistem ini akan memantau akses ke pejabat dan mengawal peranti elektrik di pejabat. Sistem ini mempunyai keupayaan untuk menghasilkan sistem keselamatan yang selamat dengan menggunakan teknologi Near Field Communication (NFC) di mana NFC merupakan system yang mempunyai tahap keselamatan yang lebih tinggi. Raspberry Pi 3 modul digunakan sebagai gateway dan sistem utama, manakala Raspberry Pi Zero W digunakan untuk melaksanakan sistem kawalan. Cloud akan diatur untuk membolehkan pengguna mengakses pangkalan data dan Server dari perisian berasaskan web dan gateway. Seterusnya, GUI dihasilkan untuk menyediakan kemudahan interaksi mesra pengguna. Masalah tentang bagaimana untuk mengesan sambungan antara Client dan Server telah dijumpai dengan menggunakan kaedah IoT yang dikenali sebagai *Heartbeat* yang boleh memberitahu pengguna mengenai status sambungan antara Client dan Server. Projek kerja ini menghasilkan sistem yang boleh memantau orang masuk dan keluar bangunan dan dapat mengawal lampu di bangunan melalui kaedah platform IoT.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	-	Internet of Things
NFC	-	Near Field Communication
GUI	-	Graphical User Interface
HTTP	-	Hypertext Transfer Protocol
MQTT	-	Message Queuing Telemetry Transport
SQL	-	Structure Query Language
PHP	-	Hypertext Pre-processor
HTML	-	Hypertext Mark-up Language
PCB	-	Printing Circuit Board

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Internet of things or in short IoT is a new technology that received a great attention in the recent years. Today, the technology is used to solve many problems in the daily life. The technology is very helpful in improving the conventional system in many fields. It has a significant potential which can increase the productivity of the system which allow the user to remote, manage and control the system at anytime and anywhere as well as the system developer. The IoT platform already improve many conventional systems including has been a part of Industry 4.0. The IoT framework that will be used in this project is IoTWave platform that consists of three main components which are the Cloud, the Things and the Network. All of the three components will work together to produce a complete IoT system.

1.2 Problem Statement

Today, the term *smart office* was famously used in many electronics products produced for office automation. Among the common products are: door security system. This is because without proper or strong security system, the administrator or the owner of the office will not have the power to control and prevent the problem from happening.

The new technology today has made the things become smart which they can communicate with each other and produce the effective product such as smart home or smart pay system. The human's ability is very limited as we cannot monitor the safety of the office or our house, 24 hours per day. We cannot do that because it can give bad effect to our health.

In a private building such as an office, an unauthorized access has led to many dangerous cases such as robbery. In order to prevent and avoid the problem from happening, IoT- based security system is one of the effective solutions which allow the user to monitor a person entering and exiting the office in a real-time, anytime and anywhere. In addition, the system also can prevent from any unauthorized access into the building as the system only allow a list of specific or authorized person to enter the office or house.

A control system is proposed to provide the control to the office electrical components such as lamp, fan and air conditioner. This idea allows the user controls their electrical components ON or OFF at any time and from anywhere. The control system allows users to monitor and control the electrical components by IoT-based that connect the component to the Internet.

One on the main problems in IoT-based system is to detect error or malfunction on the controller of the system itself. When an error situation happened to the controller of the system, it is very important to alert the owner immediately. In this project, we will come out with a solution that can notify the user promptly when the error occurs in the system.

For these problem, the solutions of the problems need to be conceived. It is essential to provide a long-lasting solution that can maintain or overcome the stated problems. Generally, IoT-based technology has the potential to connect everything and that the reason why it has been chosen as an effective solution to overcome the problem statement provided.

1.3 Objectives

IoT based security system for home or office which can allow the user to monitor, control and access the system in real time design based on several objectives:

- Develop IoT-based system to manage the access and monitor the authorize person enter and exit the building.
- Develop IoT-based system to monitor and control office automation system.
- Provide solution or method to detect either the client or server is connected or disconnected to the server.

1.4 Scope of Study

The designation of IoT-based Office Security, Control and Access system is based on IoT technology. The system consists of three main parts of IoT which are: the Things, the Cloud and the Network. The system includes a Near Field Communication (NFC) reader, a processing unit board, an alarm, a magnetic lock, a lamp controller, a door gap sensor, a Raspberry Pi Zero W, and a Raspberry Pi 3.

The Things consist of the hardware components of the system which are the Near Field Communication (NFC) reader, magnetic lock, door gap sensor, fluorescent lighting, Raspberry Pi Zero W, and Raspberry Pi 3 as the gateway of the system. All the hardware and the software of the system will be integrated, forming a complete IoT Things system.

In the Cloud and Network, it will consist of a server, database, enterprise services and the interconnection between it. The server of the system will save the data of the system in the database. The network will manage the connectivity of the things and the network including the data transfer between the things and the web-based software.

The research part is to provide a method to detect the connection within client and the server. A program need to be design and implement to make sure the client and the server is always connected to each other. Next, the notification to the user is fundamental to make sure that the user alert when something happen to the client or

the main controller of the system. The summary of the scope of this work is shown in Fig. 1.1.

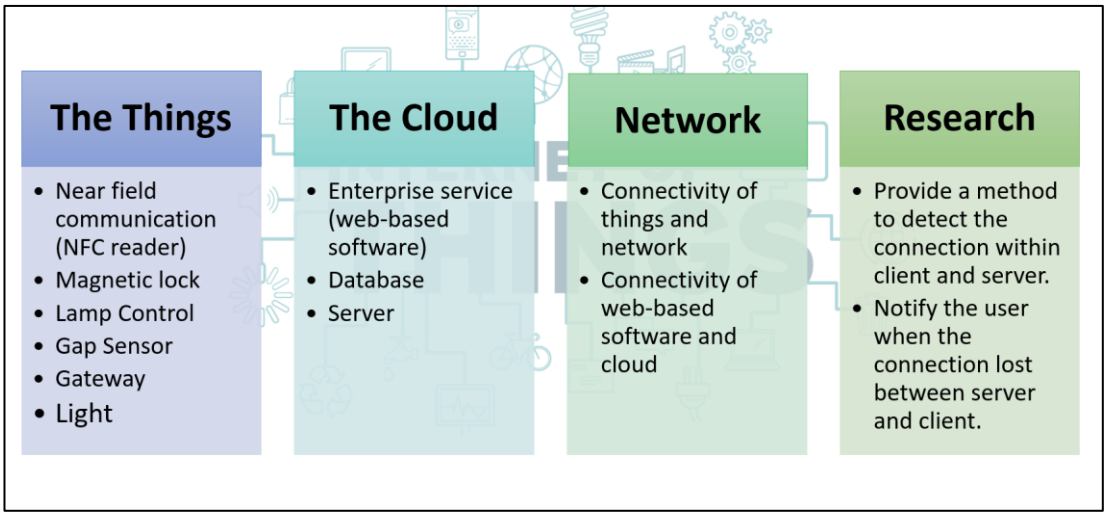

Figure 1.1: Scope of Works

1.5 Thesis Outlines

This thesis consists of six chapters. Chapter 1 introduce of the project will be developed. In Chapter 2 the explanation about literature review of the project which focus to the technology and related work to produce the proposed project. Then, Chapter 3 will discuss about the methodology on how to develop the IoT-based proposed project. After that, Chapter 4 content is about result achieved of project which consist the full result of the full system. Next in Chapter 5, the project management will be explain based on the Gant chart of the project. Lastly, Chapter 5 will explain about the overall conclusion of the project.

1.6 Summary

As a conclusion, this chapter is the introduction of the project program which include the objective of the project, the problem and the overview about the new technology which is Internet of things.

In the next chapter, the research about the technology, components and related work will be discussed as a process in producing the project product.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the previous chapter, we already discussed about the introduction of Internet of Things based office monitoring and control system. Next, in order to produce the IoT based office monitoring and control system for an office, the research about the related technology that will be used in the project must be done. In this chapter, the research about related technology and work will be discussed including the research about Internet of things in term of its definition.

2.2 Internet of Things (IoT)

Internet of things is one of the popular technology that many researcher futures their research in the technology. There is no specific definition as many researchers define IoT based on their own research. The several IoT definitions are stated as below:

- J. Gubbi et al [1], define IoT produce or create an intelligent system for our environment by allowing the transformation of the traditional mobile computing such as smartphones into connectivity existing object.
- Based on A.V Jerald [2], IoT is a part of future internet with a self-configured capability by following the certain communication protocols and standard.

- L. Atzori et al [3], stated that the presence of diversity of devices around us is the main idea of IoT.
- IoT term itself can mean something else to different people [4].

As conclusion, different researcher gives the different definition about the internet of things technology. IoT itself also can become the different definition for different people. IoT is the new technology that can benefit the people which it can help the humans to solve the daily problem in their life for example the smart home system has enable the human to control their home electrical appliances remotely. IoT is the technology that can helps in giving a better life for human in daily life.

2.3 Office Monitoring and Control System

The proposed system consists monitoring and control part which can allow the user to monitor the office and also can remotely control the electrical appliances such as lamp and air conditioning, remotely by using the developed IoT based office monitoring and control system. In the monitoring part, the user can monitor persons entering and exiting the office, through a web-based software. A person who wants to enter the office need to scan their identification card. Only the authorize person can enter the office. Near Field Communication (NFC) technology is used for the authentication method, to recognize the users. for the control system, the system has the ability to control the lamp in the office by using Raspberry Pi Zero W as the main controller of the lamp control system. Basically, the user can control all the function of the system including controlling part, through a web-based software developed for the system. All the Things will be connected to the Raspberry Pi 3 as the main controller of the whole system as well as the gateway of the system that connect the local part to the server.

2.4 Fault detection in Controller

In the robust system, the inventor need to make sure that the system will be functioning well and always aware of the many possibilities that may occur to a complete system. The main thing of the system which is the controller must be programmed and have the solution to overcome any error occur when the system is running.

One of the possibilities that will commonly occur to the system is a fault to the system. A faulty that can occur, for example an error in the system. In the proposed system, we want to detect any the error that may occur to the system. For example, the lost connection of the internet or the absent of the electrical power may make the error to the system or the system will completely shut down.

One of the solutions that can already be executed when the error occurs in the system is that the controller will automatically shut down the system and restart the controller by itself. Next, a method need to be used in order to detect the connectivity between the main controller and the server of the system.

The solution on how to overcome the connection status between the main server and the client can be done by making the client send the data to the server periodically in specific time interval. If the server receives the data, we know that the client is connected to the server. Next, if the server does not receive any data from the clients, it means that some error occurs at the client either it offline or has a technical problem.

2.5 Near Field Communication

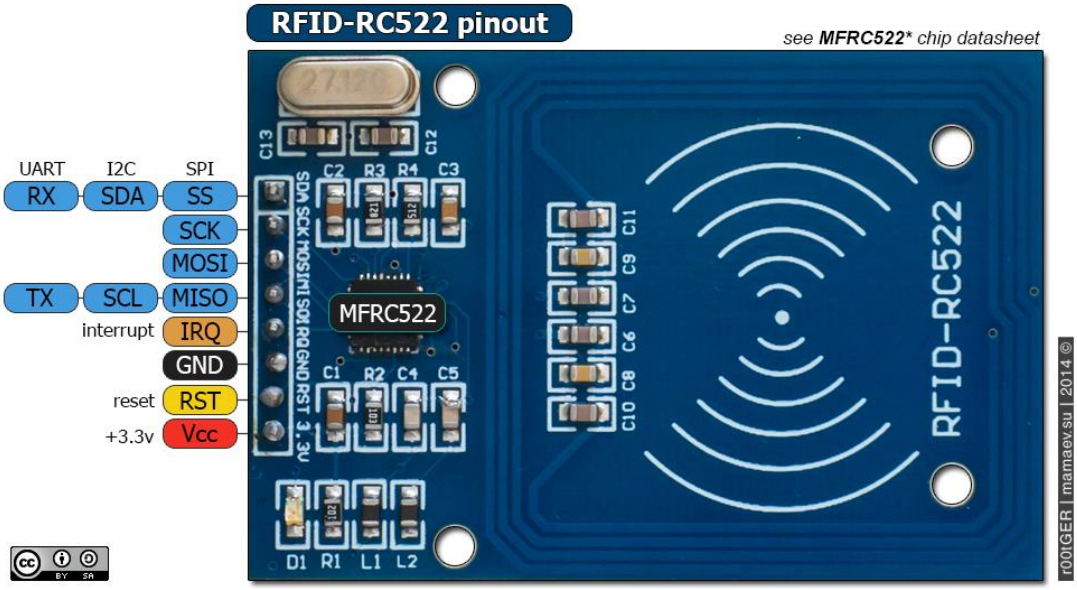

Figure 2.1: NFC for Raspberry Pi

The Near Filed Communication (NFC) technology is the popular technology used in security system today. In the proposed system, the security technology for monitoring part is important so that the user or admin can monitor the office. So, near field communication technology will be used as identification method for the project. Actually, NFC is the improvement of the previous technology which is Radio-Frequency Identification (RFID). NFC operate at the same high frequency passive RFID system which is 13.56 MHz but it can only communicate at low range distance. The reason why the NFC become a popular choice as it has high security level compare to the RFID.

NFC has a peer-to-peer communication feature which enable an NFC device able to be both as a reader and as a tag. As for example, smartphones today also come along with NFC technology today as they can share information by tapping the two smartphones or NFC devices together, which this feature turns sharing data such as contact info into a simple task. Next, NFC also can read the passive NFC tags such as ATM card and out identification card as it comes along with NFC technology. That

why the NFC has become the most reliable and be the choice for the consumer. As state before, almost all new brand smartphone has already build-in NFC technology. Then, the popular application using NFC technology are smart poster, identification card, debit card, credit card and smartphone pairing to transferring the data.

2.6 Raspberry Pi Zero W

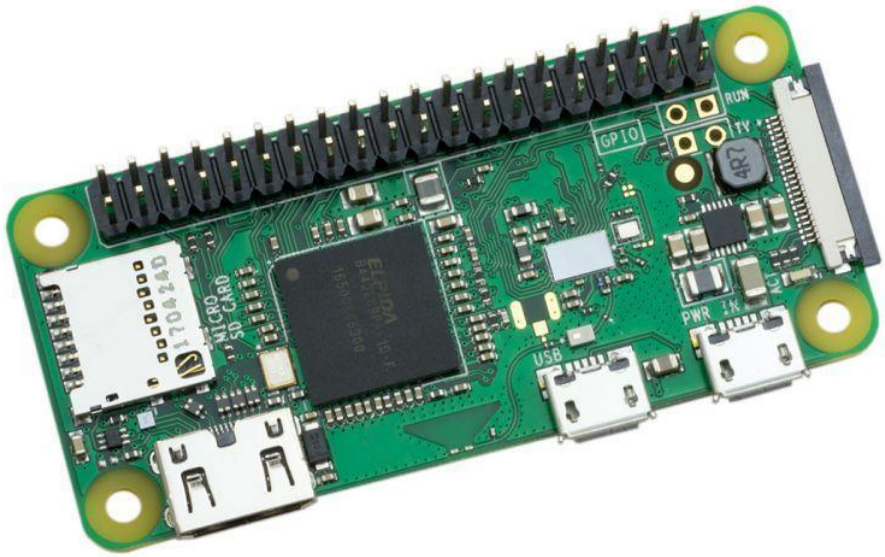

Figure 2.2: Raspberry Pi Zero W

Raspberry Pi Zero W is the extends of the Pi Zero family. The Pi Zero has all the functionality of the original Pi Zero but come with the added connectivity [5]. The Raspberry Pi Zero W is a powerful board which come in small size and the price is also affordable relevant to the functionality of the board. The Raspberry PI Zero board is powerful board that can help the developer to develop small IoT project. The specification of the board is stated in Table 2.1.

Table 2.1: Raspberry Pi Zero W Specification

No.	Specification	Description
1.	CPU	1 GHz, single-core CPU
2.	RAM	512 MB
5.	Networking	802.11 /b/g/n wireless LAN
6.	Bluetooth	Bluetooth 4.1 classic, Bluetooth low energy
7.	Storage	MicroSD card
8.	GPIO	40-pin header
9.	Port	Mini HDMI, USB On-The-Go ports, micro USB power, Camera Serial Interface(CSI) connector, composite video and reset header.

2.7 Raspberry Pi 3

Figure 2.3: Raspberry Pi 3

The raspberry pi 3 is the latest version of the raspberry pi. In the project, raspberry pi-3 will be the chosen one as it has many features that can allow to produce the final product for IoT based product. The raspberry pi 3 is a very high-performance board which has the features such as Ethernet port, Wi-Fi, Bluetooth, HDMI port and the USB port [9]. All the features complete the characteristic of a complete computer. Raspberry pi 3 board is suitable for IoT based project as it can become the gateway for the system as it has high processing capability. The full specification of the raspberry pi is show in Table 2.2.

Table 2.2: Raspberry Pi 3 Specification

No.	Specification	Description
1.	System on Chip (SoC)	Broadcom BCM2837
2.	CPU	4x ARM Cortex-A53,1.2 GHz
3.	GPU	Broadcom VideoCore IV
4.	RAM	1GB LPDDR2
5.	Networking	10/100 Ethernet, 2.4 GHz 802.11n wireless
6.	Bluetooth	Bluetooth 4.1 classic, Bluetooth low energy
7.	Storage	MicroSD card
8.	GPIO	40-pin header
9.	Port	HDMI,3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface(CSI), Display Serial Interface.

2.8 Database

The database is one of the most important things in every system today. The database store the information of the system so that the user can retrieve the information from it at any point of time when the system run. Basically, a relational database represents information in form of tables using rows and columns. The database can be classified into two categories that is structured and unstructured databased. The structured databased is Structure Query Language (SQL) while the unstructured is Not Only Structure Query Language (NoSQL). Both of the database has their own advantages and disadvantages according to certain application.

Table 2.3: Comparison SQL and NoSQL

Entity	SQL	NoSQL
Type	Relational	Non-Relational
Data	Structured Data stored in Tables	Un-structured stored in JSON files but the graph database does support relationship
Schema	Static	Dynamic
Scalability	Vertical	Horizontal
Language	Structured Query Language	Un-structured Query Language
Joins	Helpful to design complex queries	No joins do not have the powerful interface to prepare complex query
Flexible	Rigid schema bound to relationship	Non-rigid schema and flexible
Transaction	ACID	CAP theorem
Auto elasticity	Requires downtime in most cases	Automatic, no outage required

2.9 Related Works

IoT based office monitoring and the control system are proposed to produce the system that can monitor the security system which the main door and allow the user to control the electrical appliances remotely in the office. The admin can monitor the access data in the real-time. Next, it also provides the admin the data history of the person that access the office. So, at the same time the system also provides a good security environment to the office as the admin can monitor the people access the door of the office.

The control technology in the system is proposed to allow the admin control and monitor the status of the electrical appliances such as the lamp and air conditioning. The main idea is to control any electrical appliances through the web-based software. The web-based software is one of the recent popular technology that provide Graphical User Interface (GUI) which allow the users to control any system easily. Recently, many developers use that technology to provide user friendly environment.

In the related work, we will discuss about the related project work has been done by several developers. The Table 2.4 show the list of the related work of the IoT based system.

Table 2.4: Related work review

TITLE AND AUTHOR	DESCRIPTION	COMMENT
IoT BASED HOME INTERCOM SYSTEM [6] NUR ZAFIRAH BINTI AZMAN (2016)	The work done in this report is contributing the intercom system based on IoT platform. The system provides the user interface for video or audio authentication, capture image and wireless doorbell.	The system does not provide any web-based software so that the user only can access the system through android software. The system is limited to android user only. Next, the system will not notify the user about the error occur to the system when the error incident occurs to the system.
AN ACCESS SYSTEM FOR BUILDINGS BASED ON SMARD CARDS [7]	The system using a smart card which is NFC or RFID to access point that connected to the server via the internet.	The system can run in many buildings by access point using the internet connection. The weak point of the system is that the system will collapse

M.POPA, H. CIOCARLIE (2011)		and need to restart when the internet connection lost. The system not have any notification function that will notify the user when the system is collapse.
IoT-BASED SECURITY AND ACCESS SYSTEM FOR UTM RESIDENTIAL COLLEGE BUILDING [8] HERIANTO BIN UDIN (2017)	The project in the report is about implementing IoT based security access to building to prevent from not authorize person access to the building. The system includes three main things only which are magnetic lock, NFC reader and raspberry pi. This system is suitable for implement in the office and building.	The lost connection between the server and the client can make the system unable to update the person enter and exit the building. The system does not notify the user when the connection between server and client lost.
SMART DOOR SYSTEM FOR HOME SECURITY USING RASPBERRY PI 3 [9] NASER ABBAS HUSSEIN (2017)	The proposed system provided the security system that suitable for home and office. The system includes raspberry pi3, raspberry pi camera and keypad. The authorized person can enter the door by giving the correct password to the keypad and the door will open. The system has the ability to notify the user when the about the guest by taking the picture of the guest and sent it to the admin through the email. So,	The system only notifies the user through the email. There is no specific web-based software used to make the system easier for the admin to handle it. Next, the system only notifies the admin about the guest by taking that picture. The system will not notify the user about the error occur when the error incident happens to the system.

	the admin will decide to give the permission to enter or not.	
DESIGN AND EVALUATION OF WIRELESS HOME AUTOMATION SYSTEMS [10] MANIKANDAN J. (2016)	The project produces many design of home automation which are using different type of technology include Ethernet-based, IR-based, Bluetooth-based, RF-based, GSM-based and voice-based home-automation systems. All the system allows the user to control the electrical appliances at home such as lamp and fan without need to switch off on. the switch buttons. But, they can control them through the proposed design such as mobile phone apps using Bluetooth technology.	All the design produces the automation just only works inside the house only. They did not provide IoT technology which can allow the user automation system using the internet. Besides, the proposed system such as Bluetooth just work in short range. So, the user need to be near to the controller to control the system using mobile software.

2.10 Comparison with Commercial Product

The proposed product will be compared with existing commercial product that already in market today. All the commercial product that compared are represented in the Table 2.5.

Table 2.5: Comparison with commercial product

FEATURES	DOT COM	MAGNET	IoT PRODUCT BY HERIANTO	IoT PROPOSED PRODUCT
MULTI-TIER USER	×	×	✓	✓
ACCESS CARD	RFID	RFID	NFC i.e. identification card	NFC i.e. identification card
CLOUD-BASED	✓	×	✓	✓
ERROR SOLUTION (Notify user)	×	×	×	✓
ALARM	×	×	×	✓
LAMP CONTROL	×	×	×	✓
WEB-BASED SOFTWARE	✓	×	✓	✓
CUSTOMIZABLE AND UPGRADEABLE	×	×	✓	✓
SUPPPORTED USER	LIMITED	LIMITED	UNLIMITED	UNLIMITED

2.11 Summary

The Internet of Things (IoT) technology can allow things to be controlled via the Internet. IoT actually can help people solve their daily life issues. Authentication technology such as Near Field Communication (NFC) can solve the unauthorized access issue. The new processor module technology such as Raspberry Pi Zero W and Raspberry Pi 3 enable IoT application such as control system which allow smart office system able to be implemented. The system allows the data exchange through the Internet from the cloud database and local database by using Raspberry Pi 3. IoT also allow the system to notify the user in real-time when error occurs on the main controller of the system. In Chapter 3, the methodology of the proposed project will be discussed in detail.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, we will present the method taken on the development of the monitoring and control system for the office based on Internet of Things (IoT) technology, the implementation flow and the requirement of the proposed system. The methodology uses to develop the system will be discussed.

3.2 The System Architecture

The proposed system developed is based on IOTWave IoT framework. IOTWave is a complete IoT framework or architecture that can be used to design or developed any IoT applications [11]. The system architecture of the proposed IoT-based Office Monitoring and Control System can be found in Figure 3.1.

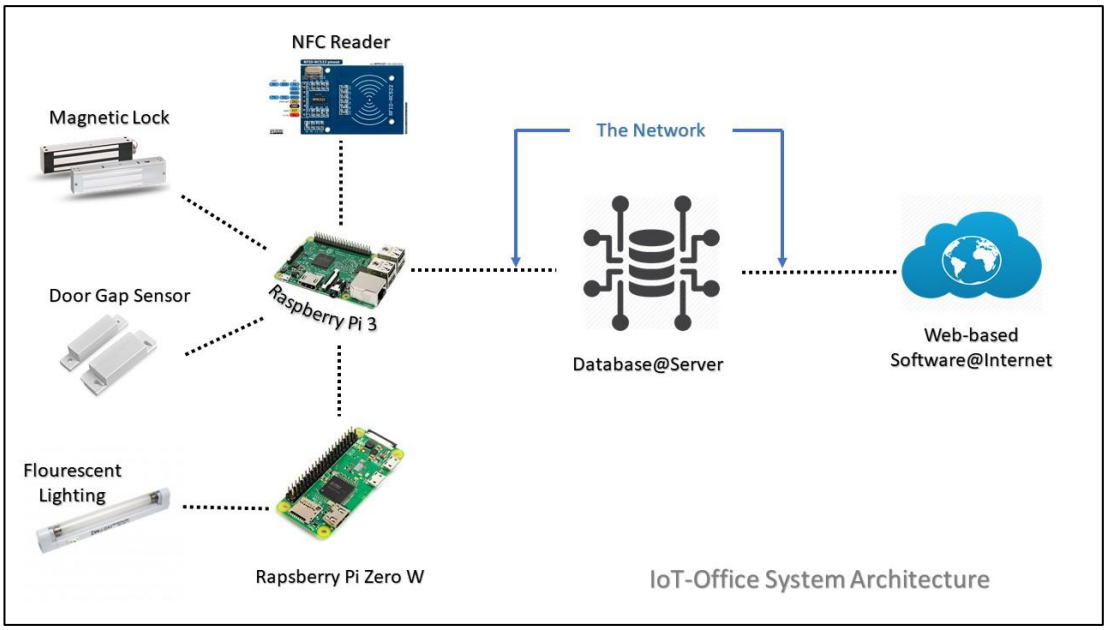

Figure 3.1: System Architecture

The architecture of the system consists of two controllers which are the Raspberry Pi 3 and the Raspberry Pi Zero W. Raspberry Pi 3 is the main controller of the system. The hardware that connected to the Raspberry Pi 3 are the door gap sensor, magnetic lock and the Near Field Communication (NFC) reader. All the Things at the main controller combined to form a complete system that provide security technology to the door at the building.

Next, the Raspberry Pi Zero W is the controller of the lighting control of the system. The fluorescent lamp can be controlled by the Raspberry Pi Zero W controller. The Raspberry Pi Zero W controller is then connected to the main controller which is the Raspberry Pi 3 wirelessly. The Raspberry Pi 3 is the gateway of the system which is the bridge that connect the system to the cloud. The things can send the data over the network via gateway.

The network is important for the interconnection between the device and the web-based software. This is the fundamental part of the system as the web-based software is the platform used for the user to access system. The system framework of the system is shown in Table 3.1.

Table 3.1: The System Framework

The Things	The Cloud	The Network
• NFC reader • Magnetic Lock • Door Gap Sensor • Fluorescent Lighting • Raspberry Pi Zero W • Raspberry Pi 3	• The Apache and Node JS • MySQL Database • Graphical User Interface • Web-based Software	• Message Queuing Telemetry Transport(MQTT) • Local Area Network • Wi-Fi • Interconnection technologies

3.3 The Things

In IoT system, the Things can be anything including a car, building, pen or even an animal which have been set an address for it by a unique ID called based on IP, which enable it to be identify when connecting to the Internet. The presence of IoT technology also allow the Things to communicate with each other based on the application. In this project, the Things are referring to the office facilities such as Door, windows, lamp, alarm, etc., as well as the devices that can monitor the system by the user which is smartphone of the laptop through web-based software.

In the office, there are many components required for the system to work. The components or the things are NFC reader, magnetic lock, door gap sensor, fluorescent lighting, Raspberry Pi 3 and Raspberry Pi Zero W. Raspberry Pi act as the main controller of the system as well as a gateway for the proposed system. The additional features such as camera, intruder detector and IP camera can be adding to make the advance improvement for the system.

3.3.1 Raspberry Pi 3

In the proposed system, Raspberry pi 3 is the most important thing in the system as it will be the main controller which is the main brain of the system. Raspberry pi 3 will control the other Things process, give the instruction and collecting the data. Raspberry pi 3 also act as the gateway for the system and it will pass the data to the cloud and receive the data from the server. More details about the Raspberry Pi 3 has been reviewed in Chapter 2.

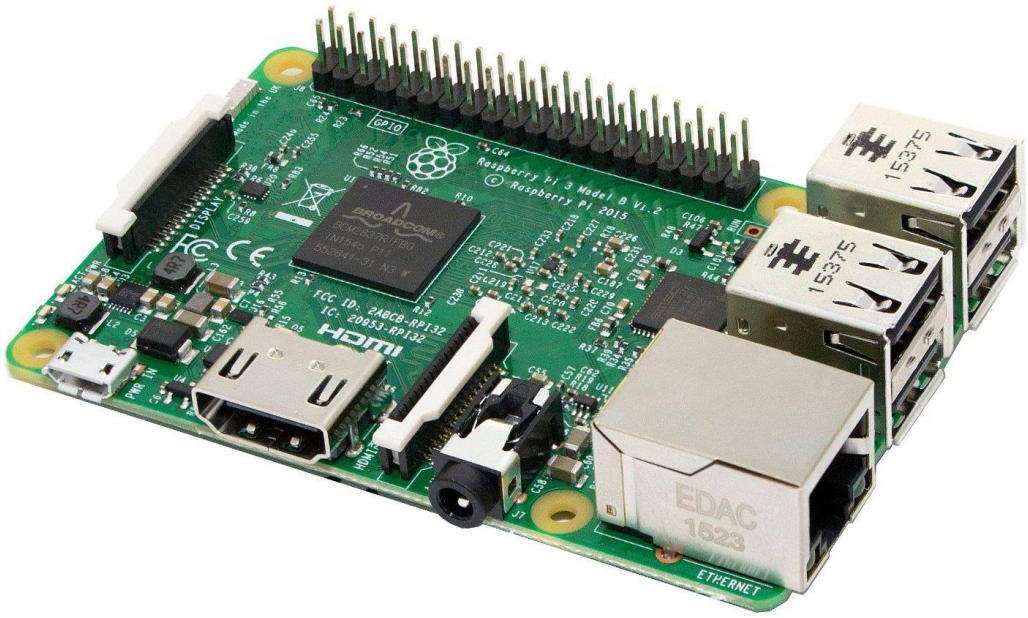

Figure 3.2: Raspberry Pi 3

3.3.2 Raspberry Pi Zero W

The Raspberry Pi Zero W is one of the subsystems of the full system. The Pi Zero W act as the main controller of the lighting control of the system. The hardware that connect to the Pi Zero W is fluorescent lamp. In the proposed system, the lighting can be access or control through the web -based software. The Pi Zero W connect to the main controller which the Raspberry Pi 3 wirelessly through the Network.

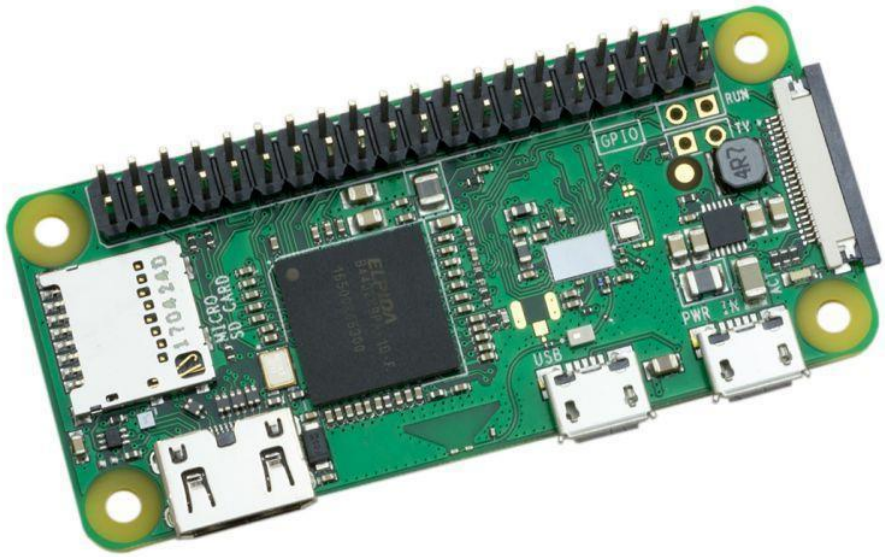

Figure 3.3: Raspberry Pi Zero W

3.3.3 NFC Reader

We used NFC technology that already present in UTM's student matric card or the Malaysian identification card (IC) as shown in Figure 3.4. Hence, the user does not need to buy any extra card or tag as the identification item. NFC reader will become the user authenticator by reading the student card or identification card then sent the card ID to the Raspberry Pi. The Raspberry Pi 3 will check to the database if the user is authorized or not to access or enter the office.

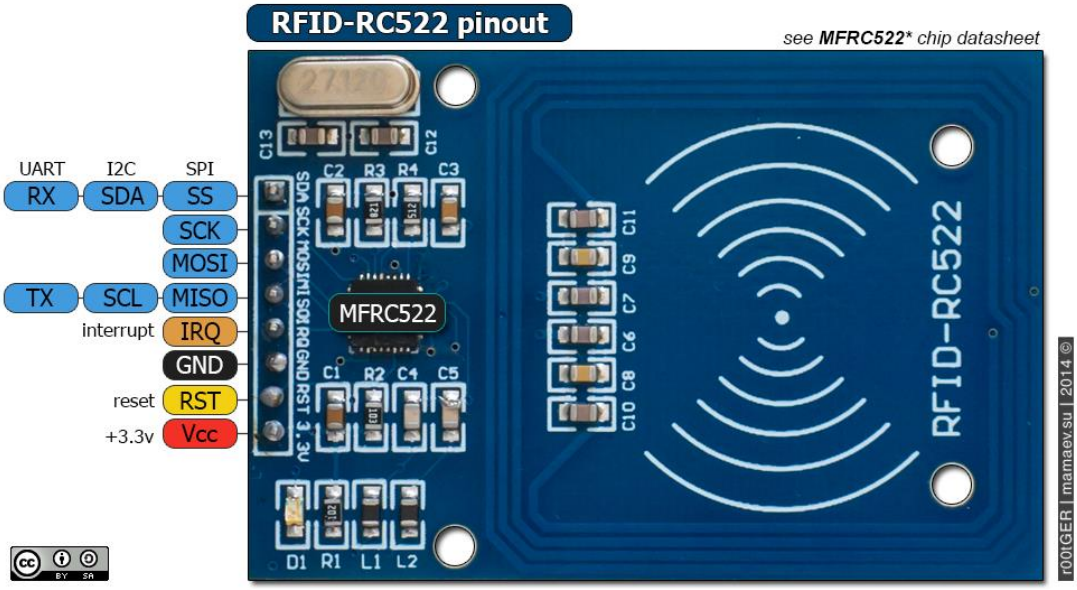

Figure 3.4: NFC reader

.

3.3.4 Magnetic Lock

The magnetic lock is one of the main Things in this system as shown in Figure 3.5. It will be installed at the main door of the building. The magnetic lock is the devices that will lock/unlock the door. In the system, the magnetic lock will lock the door by default and only unlock the door when it receives the instruction from the main controller to open the door.

Figure 3.5: Magnetic Lock

3.4 The Cloud

In the project, the Cloud include the server, database and enterprise services software, where user will refer or access to send and receive the data or information. The server basically will get the information or data and then store them in the database if necessary. The Cloud need to set up before it can function according to the programmer. The enterprise services software is needed to allow the user access the system via web browser. The database used for the system will be discussed in Sub-section 3.4.1.

3.4.1 Database

MySQL is the database used for the system. It is a popular Database system as it can be used in various kinds of environments with a lot of features. It also inexpensive and can handle a larger Database. The database can be created beyond 4 GB and can manage 50 million rows. In addition, the database is easy to learn and use so that it not takes much time to understand and work with MySQL database.

3.5 The Network

The Network is the interconnection between the Cloud and the Things. It is the data transmission medium for the Cloud and the Things in the system. The Raspberry Pi 3 will connect to the Cloud by using Wi-Fi technology. The network between Raspberry Pi Zero W and the Raspberry Pi 3 also connected via Wi-Fi technology.

The Message Queuing Telemetry Transport (MQTT) is used for controlling the fluorescent lamp at the Raspberry Pi Zero W. MQTT is designed extremely lightweight publish or subscribe message transport. The publish and subscribe process will occur at the Raspberry Pi Zero W as the control system used MQTT to control the fluorescent lamp.

3.6 The System Flow

When the system runs, the magnetic lock will lock the door by default. The door only opens when the NFC reader reads the authorized NFC card from inside or outside the office. The access log of the authorized person accessing the door will be updated to the database.

The control light system provides the normal switch at the office which can control by any person in the office either to switch the fluorescent light off or on. The web-based software allows the admin to monitor the light status and turn off the light from the web-based software. When the light is turned off from the web-based software, the person at the office cannot open the light via the switch as the web-based is the master switch of the light.

3.7 Graphic User Interfaces (GUI) Description

Graphic user interface is developed enable the user to access the system easily as shown in Figure 3.6. The command language is not user friendly for the normal person. The command language basically only used by the developer. The GUI help very much in producing user friendly system.

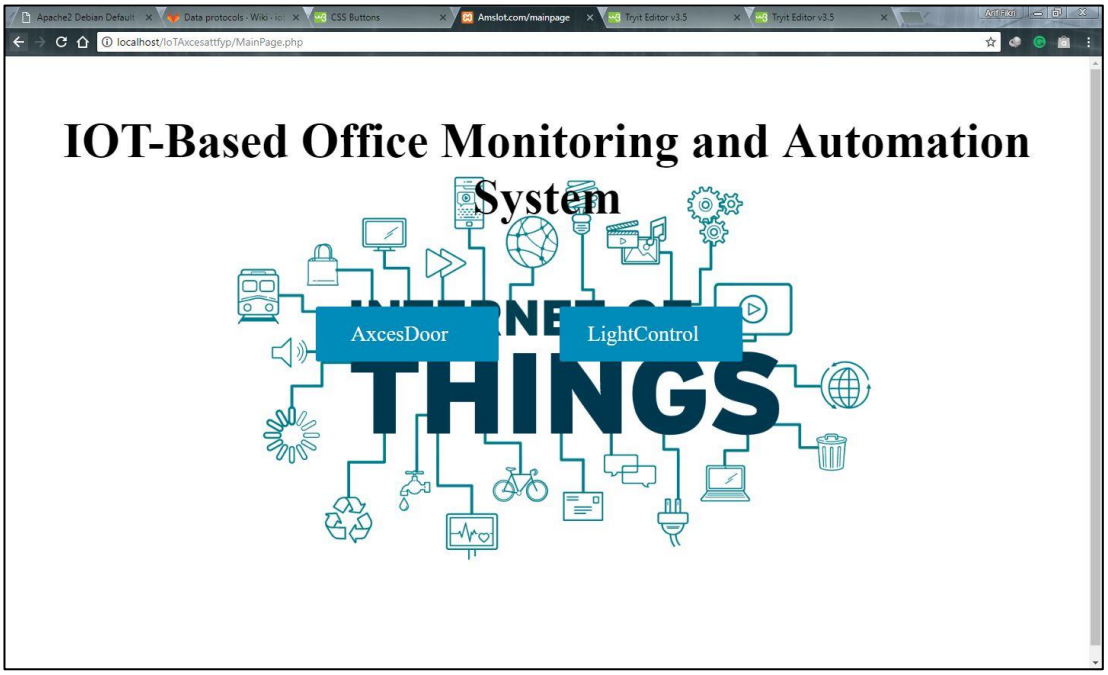

Figure 3.6: GUI for user

The web-based software allows the user to access the system by only using the existing browser. Besides, web-based software can be access through any devices that support web-based software. Today almost all devices support web-based software.

Hence, the user can access the system via their laptop or the smartphones by connecting to the system website (domain).

The web-based system provides the pages for the door log and the control lighting. The user can access the door log of the system to see the door log. The page will display the log activity of the person entering and exiting the office. In the control lighting page, the user can see the status of the fluorescent lighting and control it from the page.

3.8 The Heartbeat

Heartbeat is the name that given to the novel method that we proposed to detect the connection of the client and the server whether it is online or offline. The heartbeat system architecture consists of the client, the server and the telegram messaging software. The heartbeat process is the process of the client sending the data to the server periodically to ensure the connection is online. The connection status between the client and server will be notified to users by sending the message to the user via telegram as shown in Figure 3.7.

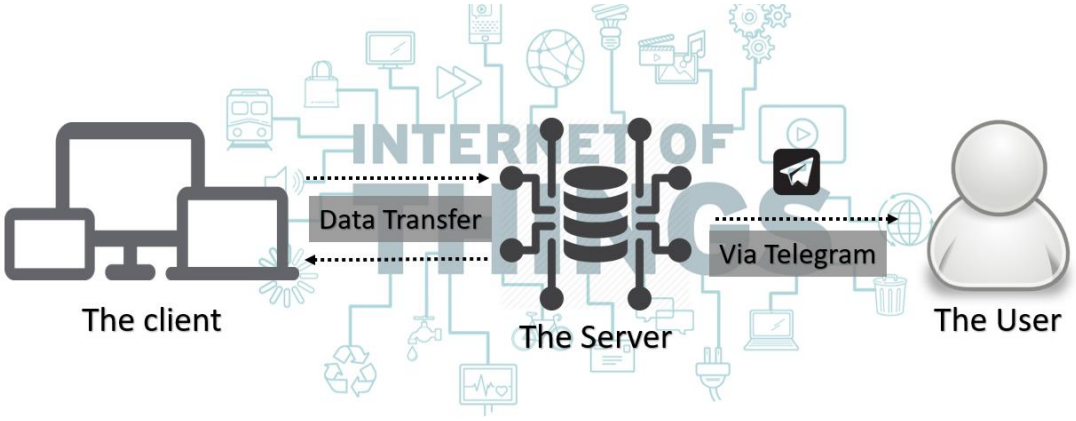

Figure 3.7: The Heartbeat Architecture

3.9 Summary

For the conclusion, the methodology of the proposed project has been discussed. The methodology about the Things, the Cloud and the Network of the project has been explained in this chapter. The heartbeat method also explained in this chapter. The latest technology has been used to produce an efficient Internet of Things (IoT) product. In the next chapter, the result that achieved in the project will be discussed and explained.

CHAPTER 4

RESULT AND ANALYSIS

4.1 Introduction

In Chapter 3, we already discussed about the methodology to develop the IoT-based Office Monitoring and Control system. The methodology includes all the important element of the IoT work which are including the things, the cloud and the network. In this chapter, the result of the work done will be shown and discuss. All the result obtained are based on the work plan or the Gantt chart of the project. In the project, the result includes the description of the hardware and software setup.

4.2 Hardware Configuration

The system need to be implement and install to a model as a prototype for the system. The prototype model has been designed for implementing and install the system using a software which is Solidworks 2014. The Solidworks is the software that used to draft the design of the prototype model before we come out with the final design of the prototype model. The model is needed to install the main controller and the light control of the system.

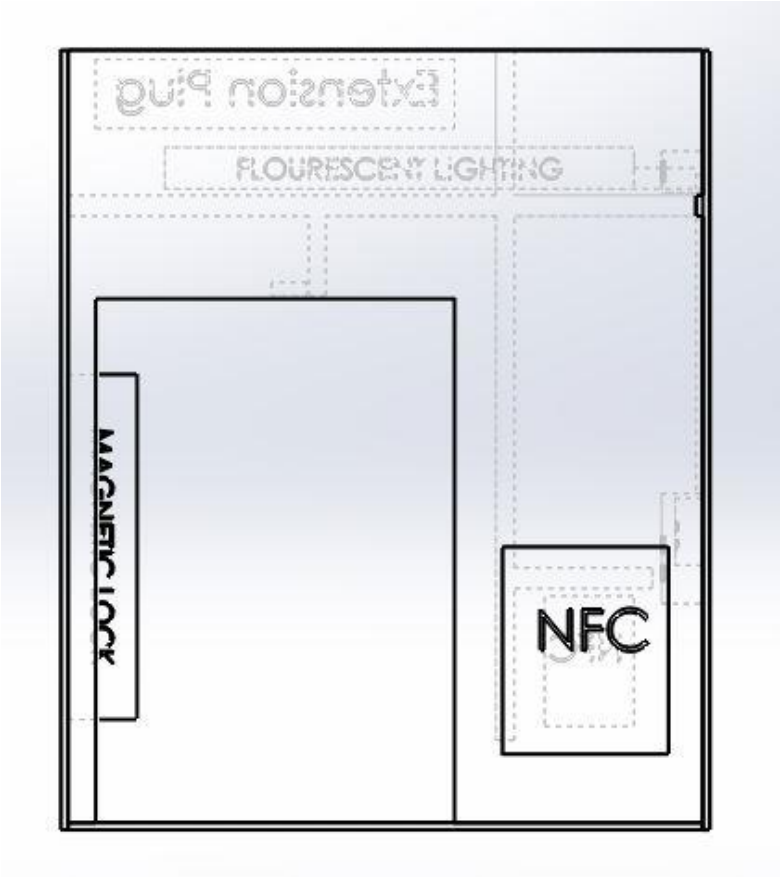

Figure 4.1: Prototype Model Front View

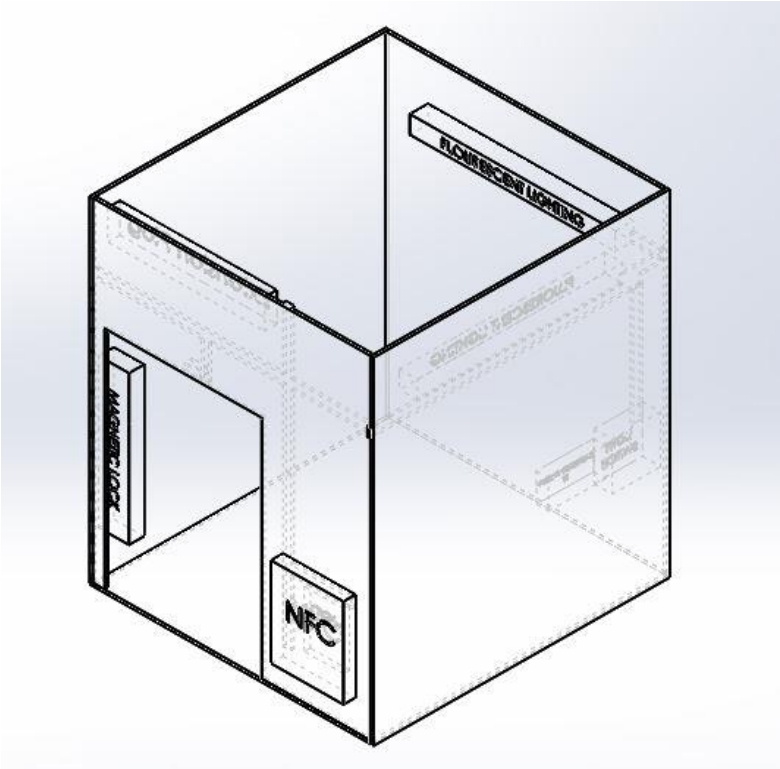

Figure 4.2: Prototype Model 3D View

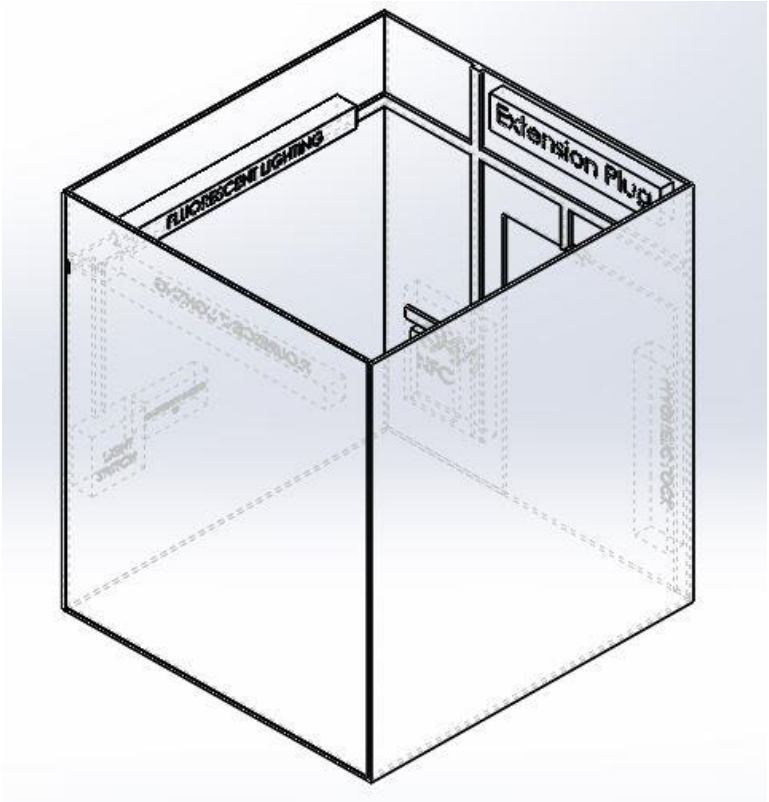

Figure 4.3: Prototype Model 3D View

Figures 4.1, 4.2, and 4.3 show the design that has been used for installing the system. The prototype is designed complete with the hardware that need to install including the main controller part and the light control of the system. Figure 4.1 shows the front view of the prototype model with NFC casing contain the main controller and the NFC reader. Next, the light control of the system can be seen in the Figure 4.2 and Figure 4.3.

4.2.1 Main Controller

The main controller of the system is the Raspberry Pi 3. The Raspberry Pi 3 is installed in the NFC casing. The things which are the magnetic lock, door gap sensor and NFC reader is connected to the Raspberry Pi 3 through a Printed Circuit Board PCB that created especially for the connection of all the things at the main controller. The technical and complete installation of the main controller is shown in the Figure below.

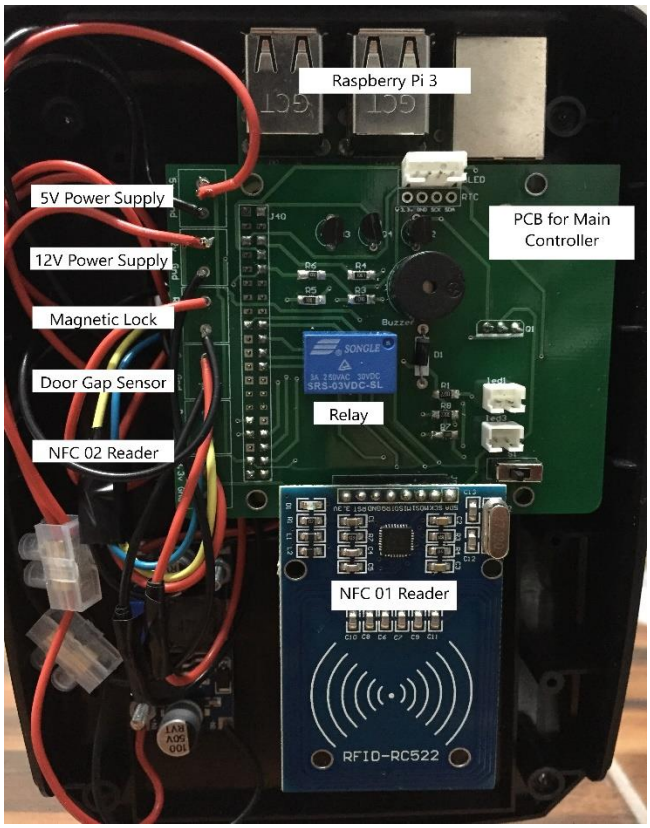

Figure 4.4: Main Controller

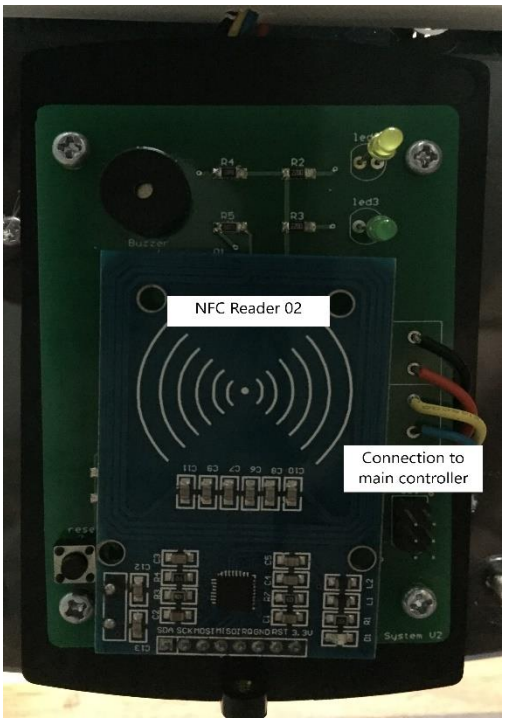

Figure 4.5: NFC Reader 02

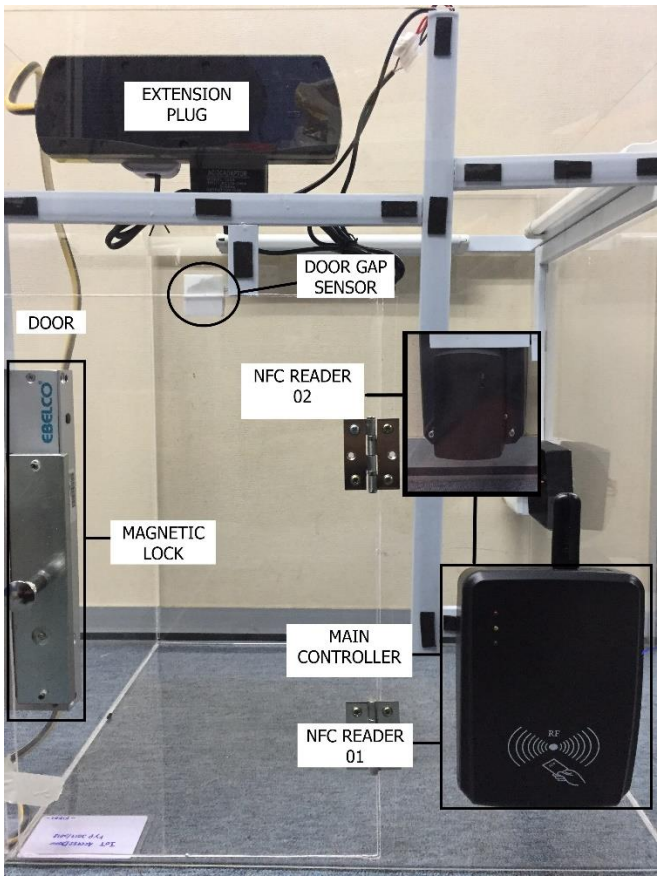

Figure 4.6: Main Controller System

The Figure 4.4 and Figure 4.5 shows the technical part of the system. The Figure 4.4 show the PCB board that used to connect all the things that connect to the main controller. The connection wire has already labeled in the Figure 4.4. The NFC reader 01 is connected to the main controller and only read the NFC card from the outside of the prototype model. It means that the NFC reader 01 is for the person enter the door. The NFC reader 02 is install behind the main controller which is in the prototype model. The connection between main controller and the NFC reader has already labeled. The NFC reader 02 is for the person exit the building.

The Figure 4.6 shows the complete installation of the main controller system. The casing of NFC reader contains the main controller of the system. The raspberry Pi 3 is connected with the magnetic lock, NFC reader 01, NFC reader 02 and the door gap sensor. The system provided two NFC reader which are NFC reader 01 and NFC reader 02. The NFC reader 01 is outside the prototype model illustrate outside the

building and the NFC reader 02 is inside the prototype model illustrate inside the building. So, the person need to scan before entering and exiting the building. The door log about the person enter or exit the building can be access through the database and can be monitor through web-based software.

4.2.2 Light Control

The light control is the separated subsystem that control the fluorescent lighting. The main controller for the light control part is the Raspberry Pi Zero W. The light control system is implemented at the Raspberry Pi Zero W. The things which are the fluorescent lighting is connected to the Raspberry Pi Zero W by connecting wires. There is no specific PCB used for the light control system.

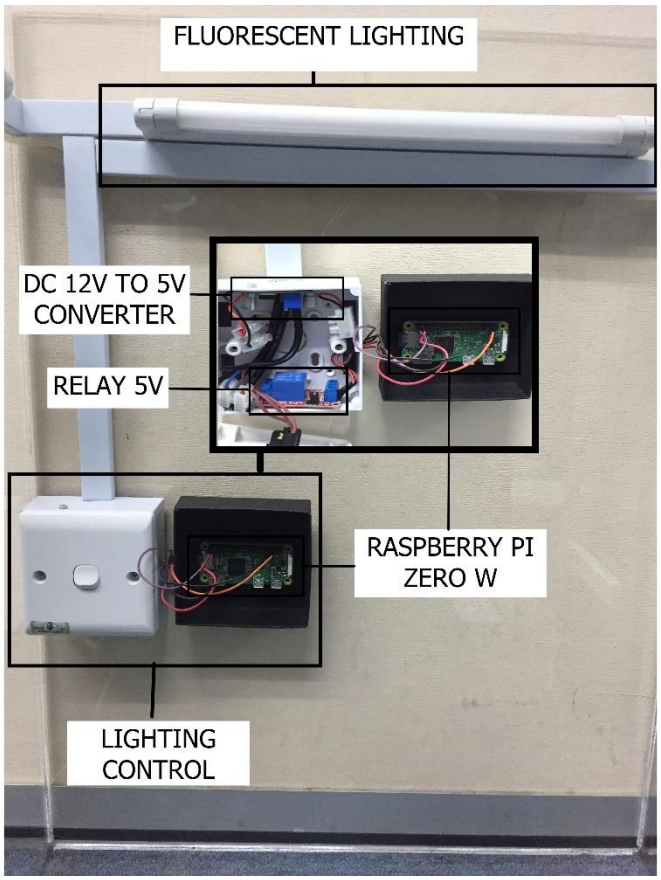

Figure 4.7: The Light Control

In Figure 4.7, we can see the hardware configuration of the lighting control. The Raspberry Zero W is the main controller for the light control program. The components that connect to the raspberry pi are DC 12 V to 5V converter and relay 5V. The DC 12V to 5V converter used to power up the Raspberry Pi Zero W and the Relay 5V is to control the switch of the fluorescent lighting by the program in the raspberry pi. The Raspberry Pi Zero W is connected to the main controller wirelessly through Wi-Fi.

4.3 Programming Configuration.

The programming configuration is very important to implement the software that follow the functionality need for the full system. The programming environment to implement the project using the software Microsoft Visual Studio Code. The programming language used for implementing the system are Node JS, PHP, and HTML. Besides, Visual Studio Code support in controlling the Raspberry Pi remotely using bash terminal and the coding can be directly upload to the Raspberry Pi.

There are four background program that run to complete the full system. The programs are the main controller, light control, the server and the MQTT broker. The main controller program run at the main control of the system which is the Raspberry Pi 3. The Node JS file that contain the main controller program for the door is implemented at the Raspberry Pi 3.

Next, the light control program will be implement and run at the Raspberry Pi Zero W. The server and the MQTT broker program is implemented at the server of the system. The system run with two separate server program which are the HTTP server and MQTT broker. The main controller is connected to the HTTP server and the light control connected with MQTT broker.

The HTTP server program contain the program that communicate the main control with the database. When the main control program runs, the data of person enter and exit the door will be update to the database. Then, MQTT broker is used to control the light. The light control program will connect to the MQTT broker.

The web-based software implemented using Node Js, HTML and PHP language. In the Door Log page, the data for the door log is update from the database of the system. The page need to be refresh manually when there are updated data in database door log. The page does not automatically update the user enter and exit the door in real-time. Only the server updates the door log data in real time, but the web-based page ned to manually refresh to see the updated door log data.

The light control used MQTT as the server for the light control system. The web-based light control page is connected to the MQTT broker. The web-based page and the light control program subscribe the same topic to make sure the light can be control through the web-based software in real-time. When the user clicks the button on or off the light from the web-based software, the program will publish the message and topic. The publish message will subscribe by the light control program at the Raspberry Pi Zero W which already subscribe the same topic. The program run the command either to on or off the light. The figure below shows the interaction between the MQTT broker and MQTT client of the light control system.

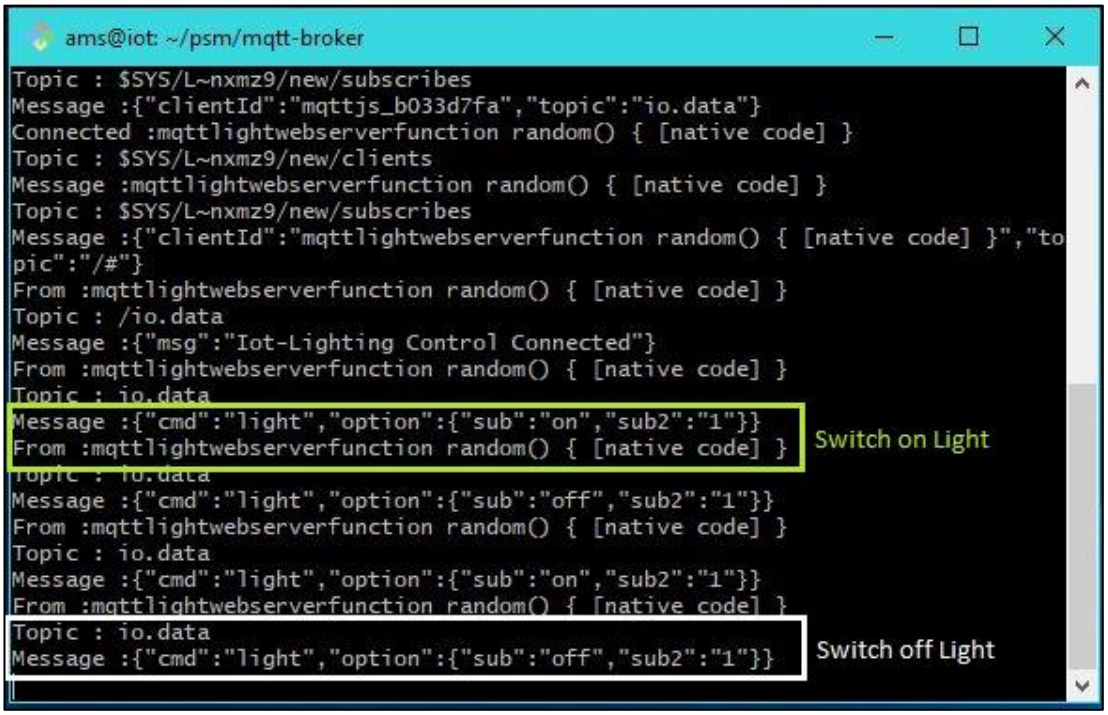

Figure 4.8: MQTT-broker Program

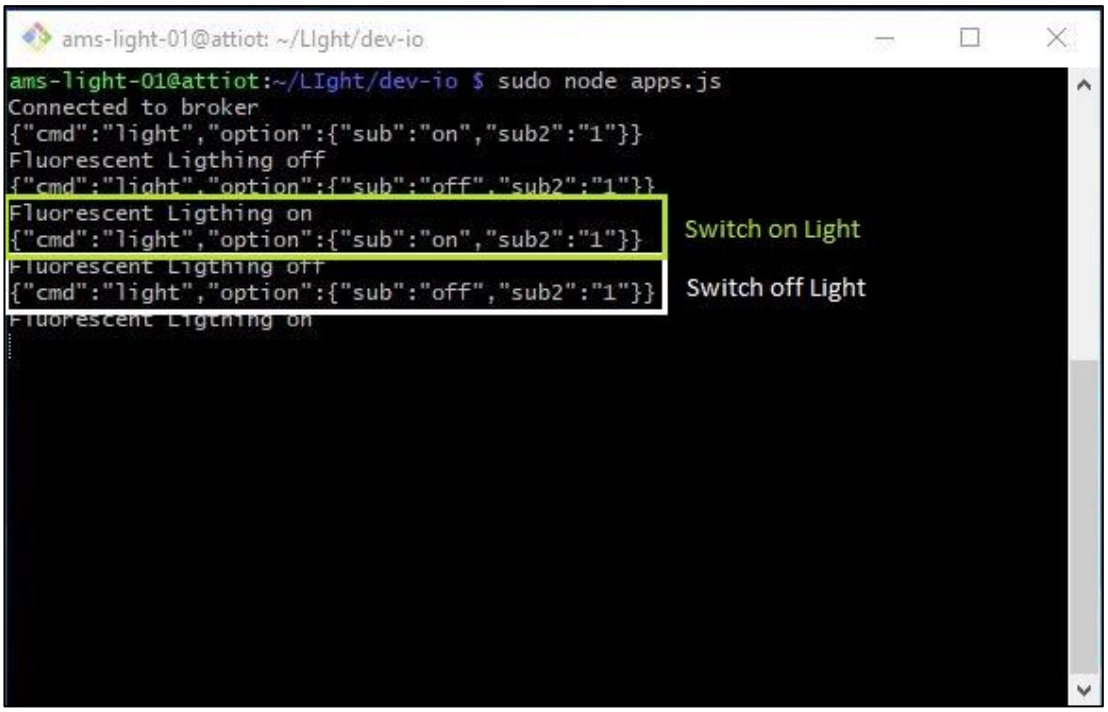

Figure 4.9: MQTT-client Program

4.4 Graphical User Interface GUI

The Graphical User Interface (GUI) is the fundamental element in the IoT-based Office Monitoring and Control system. The system is developed for the user and not for the developer. So, the GUI is the method that can make a user-friendly environment in using the system. The purpose of the GUI is to make the user easily use the system. In this project, web-based software used GUI for the user to easily access the system by click and click the button provided in the web pages. The IoT-based Office Monitoring and Control System provided three pages for the user to easily access the system. The web pages that has been implemented are the main page, the door log and the lighting control. The figures below show the GUI web pages that has been implemented for the system.

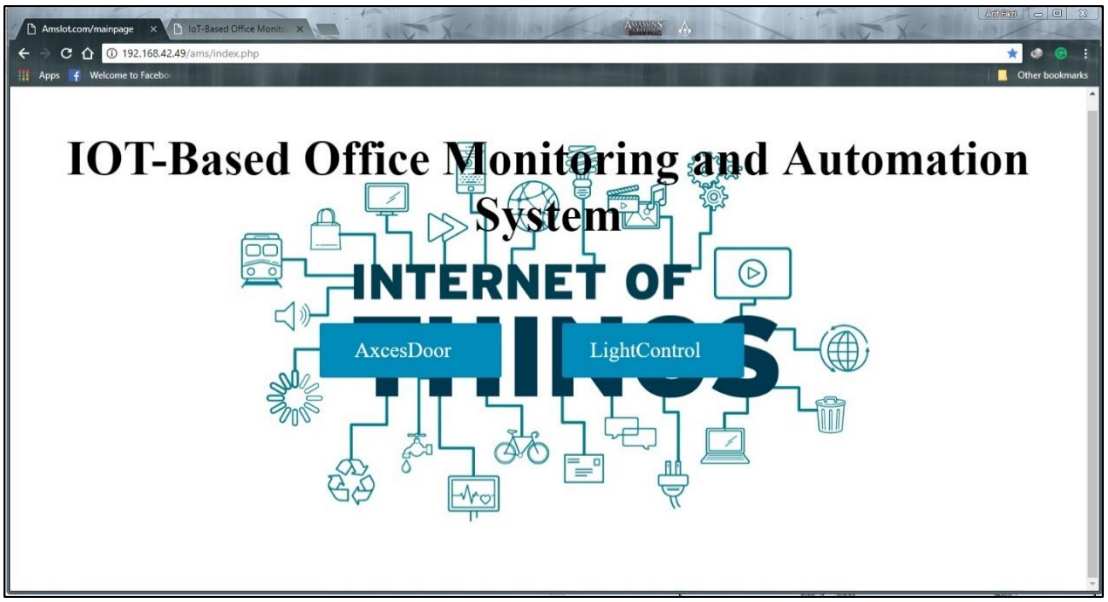

Figure 4.10: Main Page Website

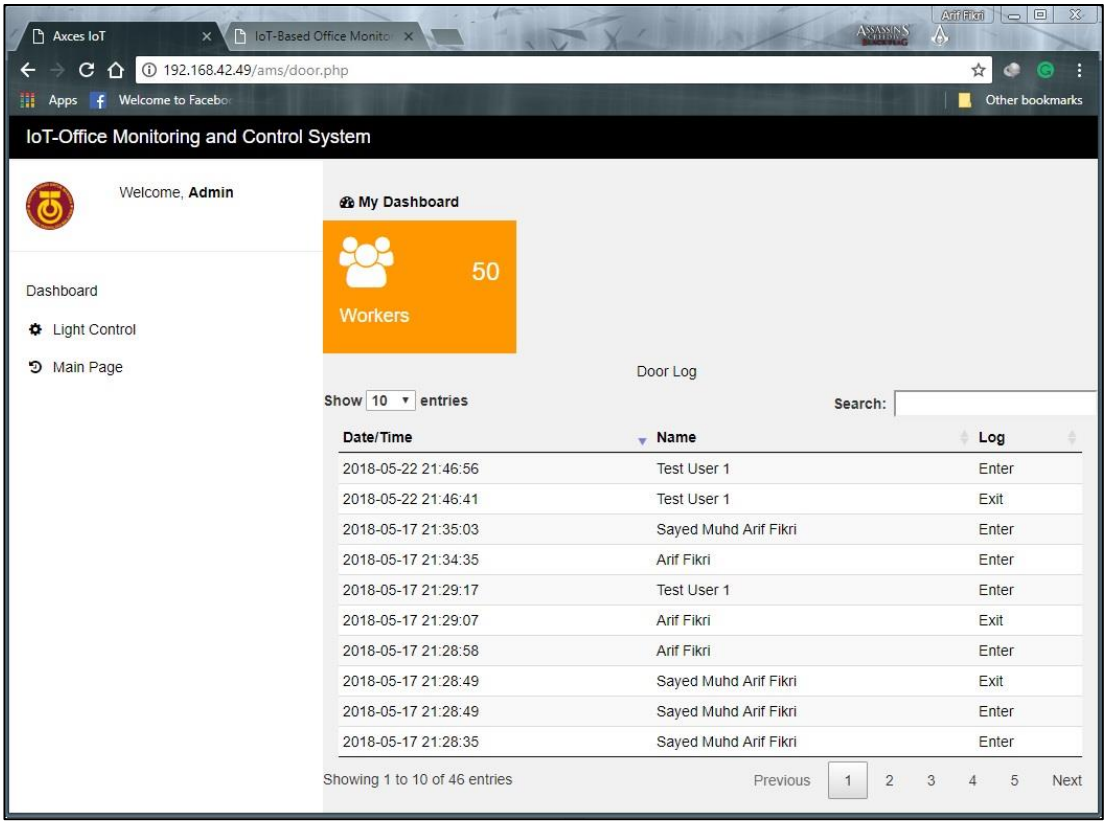

Figure 4.11: The Door Log

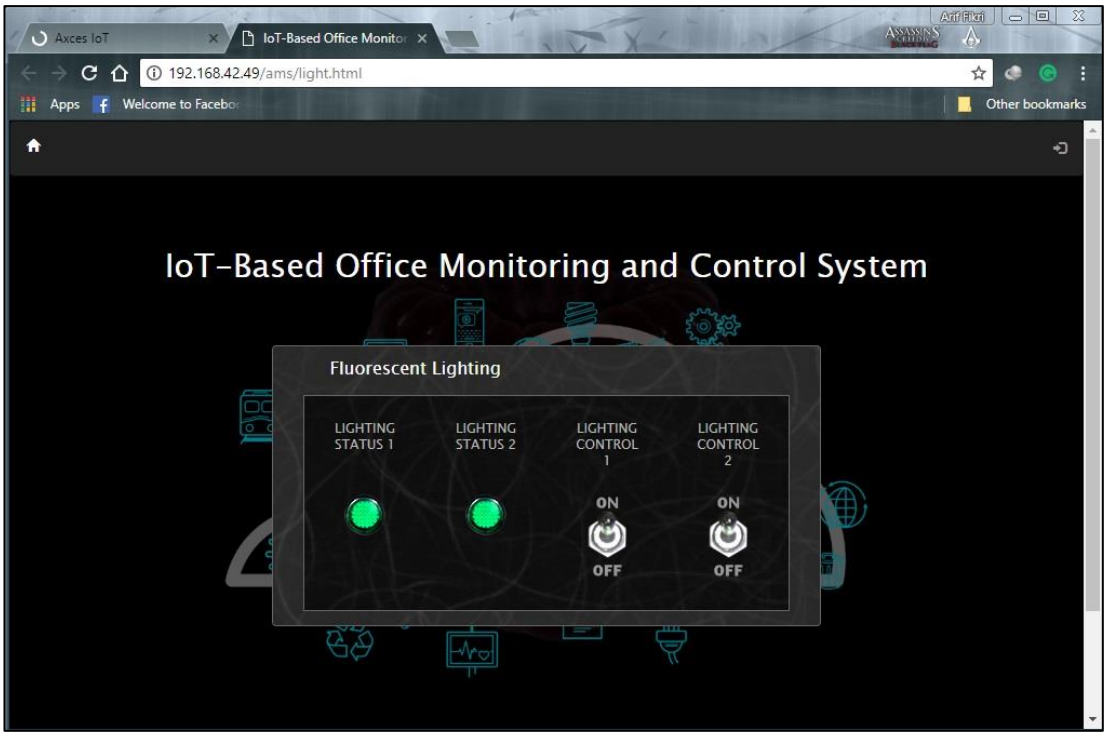

Figure 4.12: Lighting Control Webpage

The GUI page website is developed and implemented by using the Node Js, php and html language. The software used is Microsoft Visual Studio Code to write the coding of the full program. The programming language used are easily to understand and can be learn in a period to finish the project. The Node Js, PHP and Html has been used as the programming language as the language can easily understand and be learn in a short time as the language used the High-level language. So, the process to develop the system will not take long time for the learning process of the programming language.

4.5 The Database

The database used for the system is MySQL database. MySQL provide the easier way to manage the data in phpMyAdmin page. The page helps the developer to manage the database easily without using command language. The database is used to save the data of the user and the log activity of the person enter and exit the buildings.

The database. The information that updated in the door log activity database are the time, name and log activity. The door log data will automatically update to the database when the authorize person enter or exit the buildings. The figures below show the dashboard of the MySQL database.

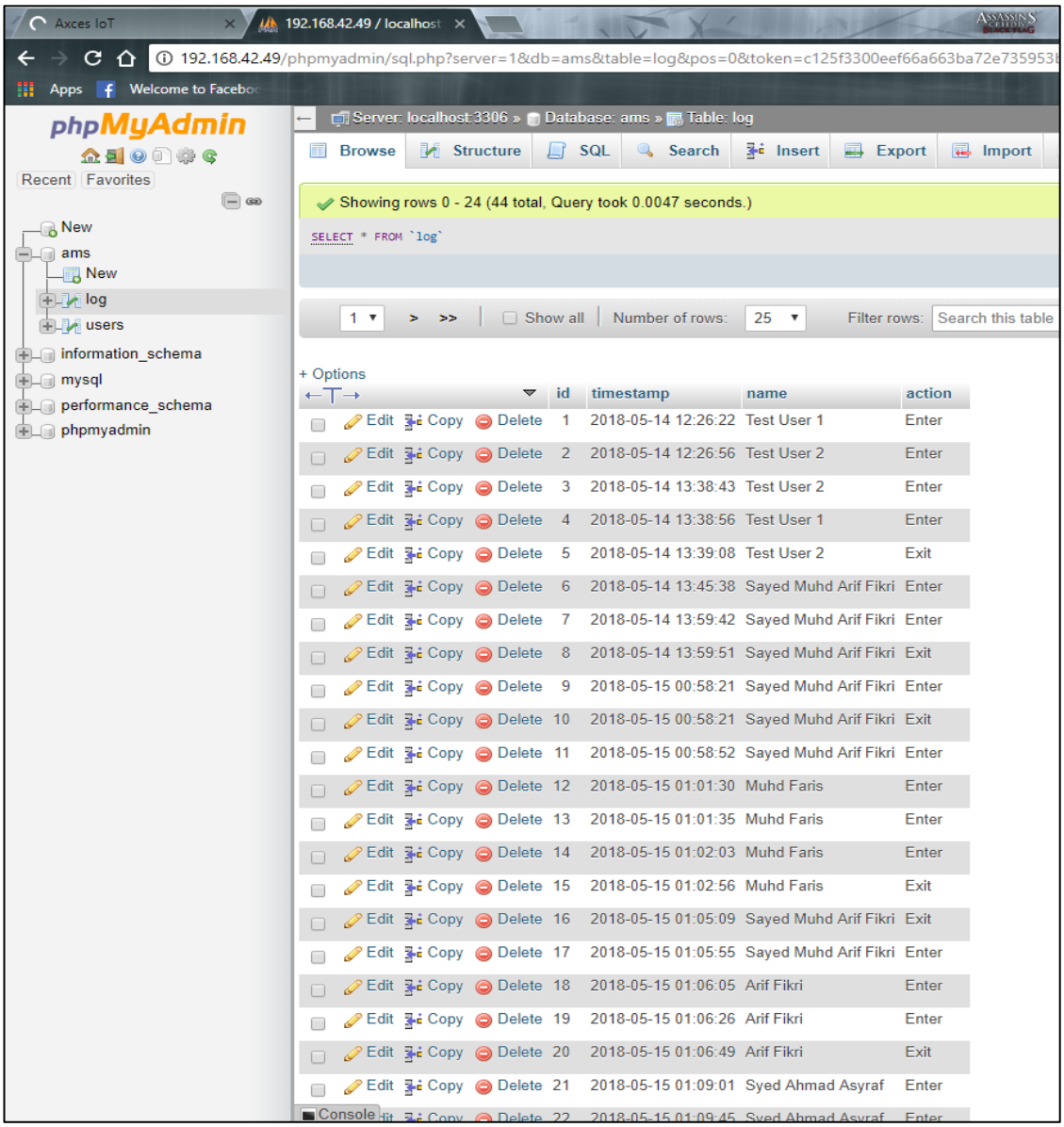

Figure 4.13: MySQL Door Log Database

4.6 The Heartbeat

The heartbeat is the proposed solution used for solving the connectivity problem between the sever and the client. The heartbeat program is running in the same program in the main controller and the HTTP server. The purpose is to make sure that the main controller always connected to the server. When the server detected the client not connected to it, the server will automatically send the message to the user via telegram.

Telegram provide free source for the developer to develop the program that need to use telegram API. The telegram provides the bot that can help in developing a program that need to use telegram for sending the message. The bots are third-party application that run inside the Telegram. The bots that used to create the new bot account for the project called BotFather. In developing the heartbeat system, BotFather is used to create new bot account.

After successfully create the bot account for the system, the BotFather will give the special token that only works for the system. The token only works to send message to the bot account that has been created before. The bot account that has been created for this system named amsiotattfypbot. The programming language will include the token to make sure the program can send the message.

Figure 4.14 show the token that has been include in the server program of the system. This can enable the server to send the message to the user to notify the status of the main system. The system will not send the message to the user in specific time, but the server only will send the message to the server in real-time only. If the client is disconnected to the server, a message will automatically send to the user to notify that the client is disconnected to the server. So that, the user can know that something happened to the client or the main controller. If the client or the main controller reconnect to the server, a message will automatically send in a real-time to notify the user about the client or the main controller is alive and connected to the server.

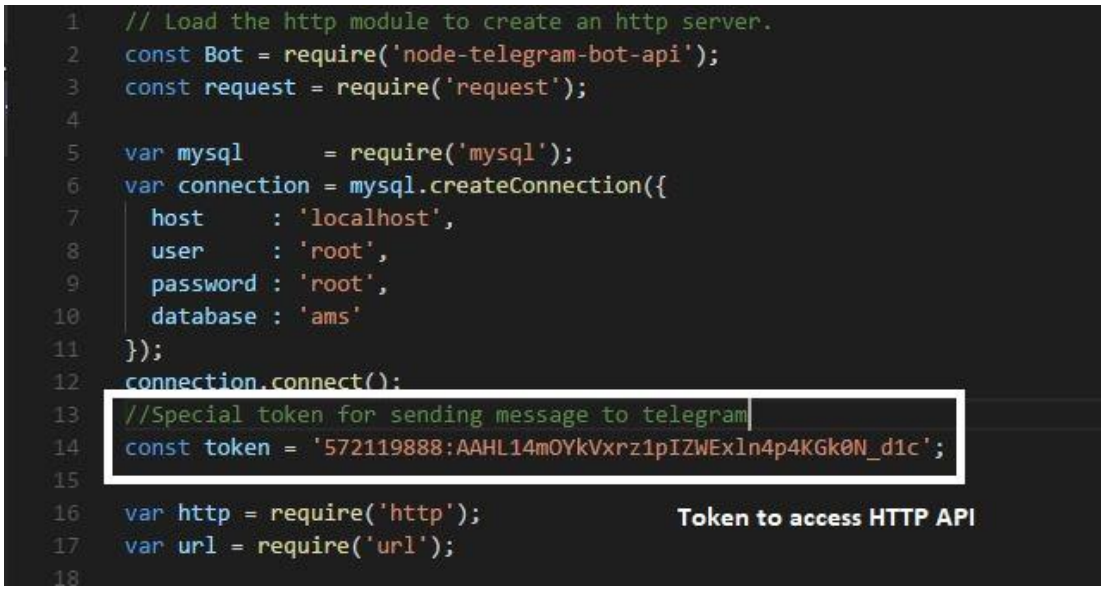

Figure 4.14: Token to Access HTTP API

Figure 4.15 and Figure 4.16 show the heartbeat server and the heartbeat client run in the bash terminal. When the client is connected to the server, a message will automatically send to the user tell that the device is connected to the server. The client will always send the data to the server periodically in five seconds to make sure that the device is always connected to the server. When the client stops working and lost connection to the server, the server will wait for the data from the client for seven seconds and if there is no data receive by the server, the server will automatically send the message to the user to notify that the device is disconnected from the server.

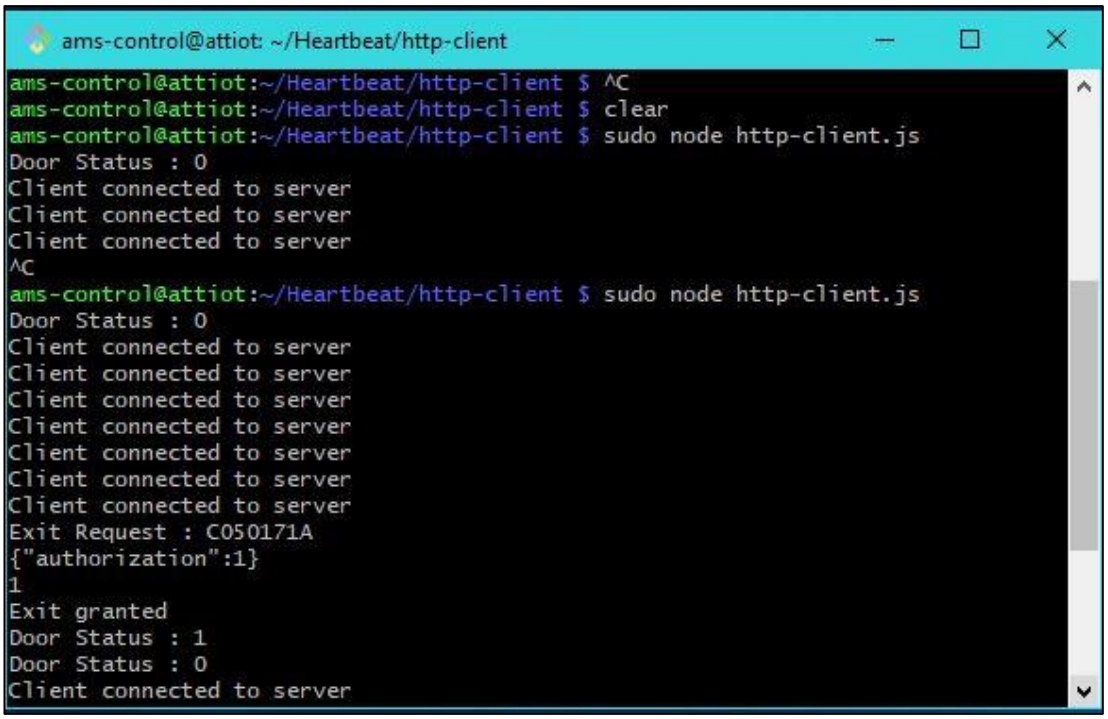

Figure 4.15: Heartbeat Client/ Main Controller

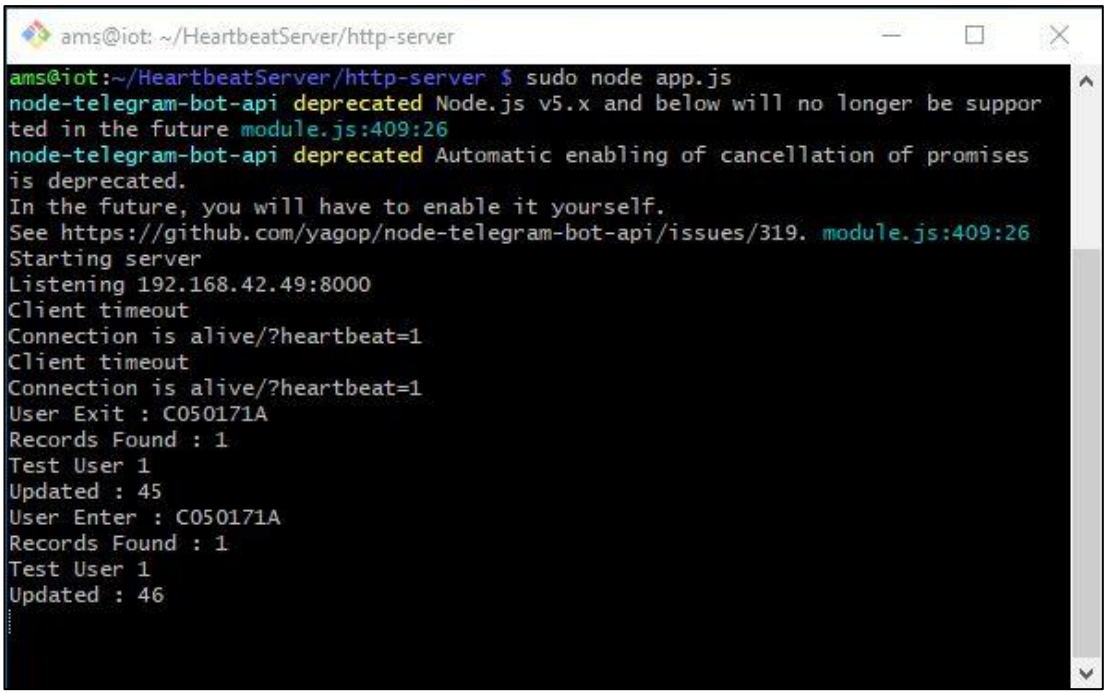

Figure 4.16: Heartbeat Server

Figure 4.17 show the message that already send from the server notify about the status of main controller device. An additional feature has been added to the system is the server will send the message to the user when the authorize person access the door. The user can know who is access the door as the server send the message about the detail of the person access the door in real-time.

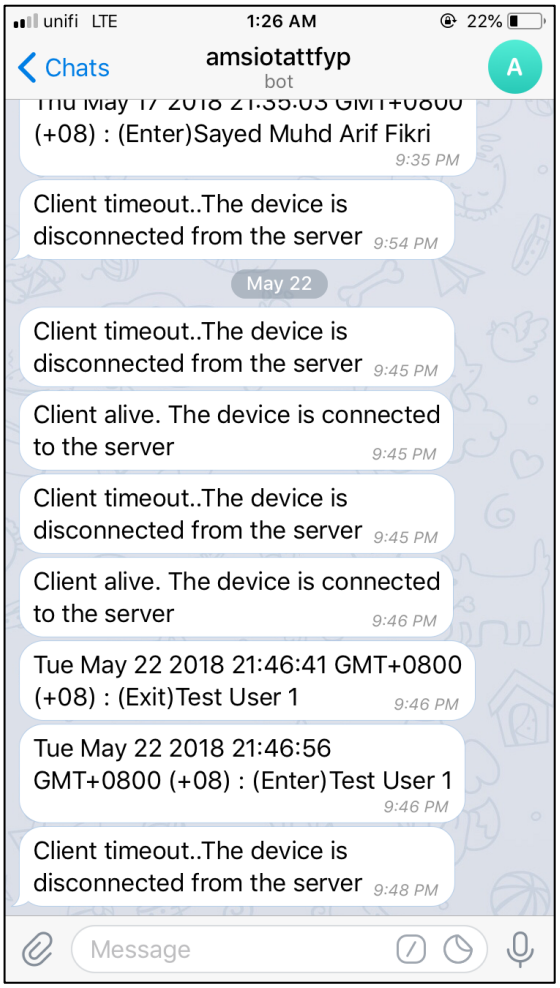

Figure 4.17: The message sends Via Telegram Apps

4.7 Summary

The result achieved in the project fulfilled the objective of IoT-based Office Monitoring and Control System. The system provides the monitoring system for the door log access for the office. Next, the user can access the control of the light in the office through web-based software. The Graphical User Interfaces to create the user-friendly environment has been developed. The solution on how to detect the connection between the client and the server has been implement. Telegram has been used as a medium to sending the message to the user to notify about the status of the main controller. The full system has been implemented and develop using the latest technology. In the next chapter, the project management of FYP 2 will be discussed.

CHAPTER 5

PROJECT MANAGEMENT

5.1 Introduction

All the result achieve in the project has been discussed in the previous chapter which are full system development including the GUI web-based software and the Heartbeat. In this chapter, project management will be discussed about the planning of the project in the both semester in finishing the final product. All the material that used in the project listed in the Bill of the project material.

5.2 Project Planning

Gant chart is important in this project as it represent the project planning for final year project including semester one and semester two. The Gant charts will be the reference for the project plan management so that the project can be complete within the given time. All the project activities in the Gant charts must be complete in time to make sure that the IoT-based proposed project finish in time. Tables below shows the Gant charts of semester one and semester two.

Table 5.1: Gant Chart for Final Year Project 1

No	Task Activity	Year 2017/2018														
		W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15
1	Discuss title of project with supervisor															
2	Literature review															
3	Identify the objective and scope of the project															
4	Study the proposed project															
5	Learn and explore the Raspberry pi platform															
6	Identify the hardware and software requirement.															
7	FYP1 presentation															
9	Documentation															
10	Report Submission															

Table 5.2: Gant Chart for Final Year Project 2

No	Task Activity	Year 2017/2018														
		W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15
1	Design the prototype using Solidworks															
2	Build the prototype model with all the hardware need															
3	Develop the user interface															
4	Develop the light control system															
5	UI development for the system															
6	Develop heartbeat program															
7	Integrated and testing the complete system															
9	Journal Paper Submission															
10	EES 2018															
11	Documentation															
12	Report Submission															

Table 5.1 show the activity that has been done in the previous semester. The previous semester is focusing on the theory part on how to implement the system based on the objective of the project. In Table 5.2, the planning activities are focusing on developing and implementing the final product of the project to meet the objective of

the proposed project. The planning is important to make sure that the final product can be finished in time and not exceed the due date given to finish the project.

5.3 Bill of Materials

Table 5.3 show the total expense of the cost for developing the prototype model and implementing the full system. All the list of the material used to develop the hardware and the system listed are in Table 5.3. The total cost will be total up to see the total expense in producing the final product of the system including the prototype model.

Table 5.3: Bill of Materials

Item	Unit Price (RM)	Units	Total (RM)
Raspberry Pi 3	295.00	1	295.00
Raspberry Pi Zero W	50.00	3	150.00
NFC reader	15.00	2	30.00
MicroSD 16GB	30.00	3	90.00
Magnetic Lock	74.00	1	74.00
Terminal Connector	1.50	1	1.50
Dupond Cable male-female	7.00	1	7.00
12V 2A Adapter	25.00	1	25.00
Relay Board High/Low Triggered 5V	8.00	1	8.00
Socket Extension Plug	19.60	1	19.60
8W Florescent Lighting	12.00	2	24.00
Switch	4.00	1	4.00
PVC Casing	4.80	1	4.80
D-link DWA-123 Wireless N 150 USB Adapter	23.00	1	23.00
Arduino Magnetic Contact Switch Door Sensor	14.70	1	14.70
TOTAL			770.60

5.4 Summary

In conclusion the project management is important to make sure that the project can be done systematically according to the planning. The Gant chat of the both semester helps in managing the project of IoT-based Office monitoring and control system. The project management can be the master mind of the project to make sure the cost is enough to produce the final product and the proposed system can be finished according the planning that has been done in the bot Gant chart for FYP1 and FYP2. In Chapter 6, the conclusion about the overall project in FYP 1 will be discussed.

CHAPTER 6

CONCLUSION

6.1 Introduction

Internet of things is the technology that can make the smart things communicate with each other to perform a complete IoT-based system. The technology enables them to connect with each other by using the internet connection. Generally, the IoT system must contain three main things which are the things, the cloud and the network. If one of the important element is malfunction or lost, the system fails to perform its functionality to become IoT devices.

In conclusion, the IoT-based Office Monitoring and Control system has been successfully developed. The system provides the monitoring system for the door log access of the office. The user or the admin can access the web-based of the door log pages to see the list of data of the door log activity of the office which already save in the database. Next, the system can notify the user about who is accessing the door in real-time by receiving a telegram message from the server containing the detail of the door log.

The light control system in the project provide the user to control the light at the office in real time using MQTT technology which allow the user to control the light from the website in the real time. The problem on the how to detect either the device is connected to the server or not has already solved with Heartbeat method.

The system has been tested and the program is working accordingly. The GUI system can make the user can access and control the system easily. All the hardware

and software part of the system works with each other well. The system is ready to implement in any office. Nevertheless, the developer can still make improvements to the system such as adding more electrical appliances to control. Overall, it can be concluded that the proposed system is functioning well according to the scope of the project.

6.2 Contribution

In FYP2 project, the IoT-base Office Monitoring and Control System has successfully developed and implemented. The system is running accordingly follow the objective of the system. The achievement of the project can be summarized in the following:

- The access security system has been develop using Near Field Communication technology. Only authorize person can access the door and if the person NFC card id is not in the database the person cannot access the door.
- The control system has been develop using MQTT technology to synchronize the command from the web-based and the hardware program. The user can control the light from the web0based software.
- The Graphical User Interface has been developed for the user to easily access the monitoring system and control system through the website. The website can be access in any device so that the user just opens the web browser and go to the system web pages to access the system.
- The solution on how to detect the client and the server is connected or not has been developed and the system is implemented in the main controller of the system. When the device is disconnected from the server, the server will immediately notify the user about the device status by sending a message via Telegram application.

6.3 Future Works

The proposed system is running according to the objective and scope of work of the project. In addition, the upgrade can be done to the system to produce more functionality and robust system. The security technology for the door system can be upgrade by adding the new technology such as camera for face recognize or fingerprint scanner to produce more secure monitoring and security system for the door. Next, more electrical appliances such as fan and air conditional or add more fluorescent light can be installed to improve the control system. So, if more electrical appliance can be added then more devices can be control through the web-based software. Then, in the project, the *Heartbeat* system was produced to notify the user about the status of the system either online or offline. In future, the analysis of the status connection between the client and the server can be done in order to produce more efficient notification system.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] A. V. Jerald, S. A. Rabara, and D. P. Bai, "Algorithmic Approach to Security Architecture for Integrated IoT Smart Services Environment," 2017 World Congress on Computing and Communication Technologies (WCCCT), 2017.

[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] M. Fell, "Roadmap for The Emerging Internet of Things", Carre & Strauss,2014

[5] "RASPBERRY PI ZERO W," *raspberrypi.* [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-zero-w/.

[6] N.Z Azman, "IoT-Based Home Intercom System",FYP Thesis, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2016.

[7] M. Popa and H. Ciocarlie, "An access system for buildings based on smart cards," 2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2011.

[8] H. Udin, "IoT-Based Security and Access System for UTM Residential College Building", FYP Thesis, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2017.

[9] N. A. Hussein and I. A. Mansoori, "Smart Door System for Home Security Using Raspberry pi3," 2017 International Conference on Computer and Applications (ICCA), 2017.

[10] M. J., “Design and evaluation of wireless home automation systems,” 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), 2016.

[11] M. N. A. Hassan, “IoT-Based Vehicle Access Management System”, FYP Thesis, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2017.

[12] “Raspberry Pi 3 Model B,” Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. nosql [Accessed 13 Dec. 2017].

[13] “MQTT,” MQTT RSS. [Online]. Available: http://mqtt.org/. nosql [Accessed 13 Dec. 2017].

[14] R. Minerva, A. Biru, and D. Rotondi,” Towards a definition of the Internet of Things(IoT),”IEEE Internet Things, no. 1,p. 86, 2015.

[15] P. Jayaram, “SQL vs. NoSQL - DZone Database,” dzone.com, 24-Mar-2017. [Online]. Available: https://dzone.com/articles/sql-vs-nosql.

[16] A. D. Deshmukh and U. B. Shinde, “A low cost environment monitoring system using raspberry Pi and arduino with Zigbee,” 2016 International Conference on Inventive Computation Technologies (ICICT), 2016.

[17] P. Ray, “A Survey on Internet of Things Architectures,” EAI Endorsed Transactions on Internet of Things, vol. 2, no. 5, p. 151714, Jan. 2016.

[18] Y.-W.Bai and S.-C.Chen,’Design and Implementation of Home Intercom and Security Control Systems for Buildings,’ in 2007 IEEE International Symposium on Consumer Electronics, 2007,pp. 1-6.

[19] dzone.com. (2017). SQL vs. NoSQL - DZone Database. [online] Available at: http://dzone.com/articles/sql-vs-nosql [Accessed 13 Dec. 2017].

[20] N. A. Zaini, N. Zaini, M. F. A. Latip, and N. Hamzah, “Remote monitoring 40 system based on a Wi-Fi controlled car using Raspberry Pi,” in *2016 IEEE Conference on Systems, Process and Control (ICSPC),* 2016, pp. 224–229.

[21] S. Tozlu, M. Senel, Wei Mao, and A. Keshavarzian, “Wi-Fi enabled sensors for internet of things: A practical approach,” *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 134–143, Jun. 2012.

[22] M. Kang, E. Noh, and K. Kim, “NFC transmitter coil placement to minimise degradation of A4WP wireless power transfer efficiency,” *Electron. Lett.*, vol. 53, no. 9, pp. 616–618, Apr. 2017.

APPENDIX A- MAIN CONTROL

```
var gpio = require('wiringpi-node');
var mfrc522 = require("mfrc522-rpi");
var request = require('request');
var http = require('http');
var nfcExit = require('./lib/nfcexit');

/* Default HTTP server request options
 *  Use "GET" method
 *  Create json type body for nfc,button and time
 *  Set headers to application/json
 */

setInterval(function () {
   request({
      uri: "http://192.168.42.49:8000?heartbeat=1",
      method: "GET",
      timeout: 10000,
      followRedirect: true,
      maxRedirects: 10
   }, function (error, response, body) {
      console.log(body);
   });
}, 5000);

var uidEnter = null;
var holdEnter = null;

// Function to get timestamp
function getDateTime() {
   var date = new Date();

   var hour = date.getHours();
   hour = (hour < 10 ? "0" : "") + hour;

   var min = date.getMinutes();
   min = (min < 10 ? "0" : "") + min;

   var sec = date.getSeconds();
```

```
    sec = (sec < 10 ? "0" : "") + sec;

    var year = date.getFullYear();

    var month = date.getMonth() + 1;
    month = (month < 10 ? "0" : "") + month;

    var day = date.getDate();
    day = (day < 10 ? "0" : "") + day;

    return year + "/" + month + "/" + day + " @ " + hour + ":" + min + ":" + sec;
}

var beep = 0;
var doorAlert = null;
var gapTimeout = null;
var newGapState = 0;
var gapState = 0;
var MagneticLock = 4; //7; //4
var BuzzerSound = 19;  //24; //19
var DoorGapSensor = 6; //22; //6
var Led1 = 16;  //27; //16

gpio.wiringPiSetupGpio();

gpio.pinMode(Led1, gpio.OUTPUT);
gpio.pinMode(MagneticLock, gpio.OUTPUT);
gpio.pinMode(BuzzerSound, gpio.OUTPUT);

gpio.digitalWrite(Led1, gpio.LOW); //LOW ON, HIGH OFF
gpio.digitalWrite(BuzzerSound, gpio.LOW); //LOW OFF, HIGH ON
gpio.digitalWrite(MagneticLock, gpio.LOW); //LOW ACTIVE HIGH INACTIVE



mfrc522.initWiringPi(0);

//NFC Enter
setInterval(function () {
    mfrc522.reset();

    //# Scan for cards
    var response = mfrc522.findCard();
    if (!response.status) {
        gpio.digitalWrite(Led1, gpio.HIGH);
        return;
    }
```

```
   //# Get the UID of the card
   response = mfrc522.getUid();
   if (!response.status) {
      gpio.digitalWrite(Led1, gpio.HIGH);
      return;
   }
   //# If we have the UID, continue
   const uid = response.data;
   var uidHex = uid[0].toString(16) + uid[1].toString(16) + uid[2].toString(16) +
uid[3].toString(16);

   if(uidEnter == null || uidEnter!=uidHex){
      if(holdEnter!=null){
         clearTimeout(holdEnter);
      }
      holdEnter = setTimeout(function(){
         uidEnter = null;
      },2000);
      uidEnter = uidHex;
      enterRequest(uidHex.toUpperCase());
   }else{
      if(holdEnter!=null){
         clearTimeout(holdEnter);
      }
      holdEnter = setTimeout(function(){
         uidEnter = null;
      },2000);
   }
   //# Stop
   mfrc522.stopCrypto();
}, 500);
//NFC Exit
nfcExit.on('uid',function(uid){
   exitRequest(uid);
});

function exitRequest(uid){
   console.log("Exit Request : "+uid);
   request({
      uri: "http://192.168.42.49:8000?reqExit="+uid,
      method: "GET",
      timeout: 10000,
      followRedirect: true,
      maxRedirects: 10
   }, function (error, response, body) {
      console.log(body);
      var res = JSON.parse(body);
```

```
        console.log(res.authorization);
        if(res.authorization){
            console.log("Exit granted");
            gpio.digitalWrite(MagneticLock, gpio.HIGH);
            gpio.digitalWrite(BuzzerSound, gpio.HIGH);
            setTimeout(function(){
                gpio.digitalWrite(BuzzerSound, gpio.LOW);
            },500);
            setTimeout(function () {
                gpio.digitalWrite(MagneticLock, gpio.LOW);
                if(gpio.digitalRead(DoorGapSensor)){
                    doorAlert = setInterval(function(){
                        if(beep){
                            beep = 0;
                        }else{
                            beep = 1;
                        }
                        gpio.digitalWrite(BuzzerSound, beep);
                    },50);
                }
            }, 5000);

        }else{
            console.log("Exit denied");
            gpio.digitalWrite(MagneticLock, gpio.LOW);
            gpio.digitalWrite(BuzzerSound, gpio.HIGH);
            setTimeout(function(){
                gpio.digitalWrite(BuzzerSound, gpio.LOW);
            },1000);
        }
    });
}

function enterRequest(uid){
    console.log("Enter Request : "+uid);
    request({
        uri: "http://192.168.42.49:8000?reqEnter="+uid,
        method: "GET",
        timeout: 10000,
        followRedirect: true,
        maxRedirects: 10
    }, function (error, response, body) {
        var res = JSON.parse(body);
        console.log(res.authorization);
        if(res.authorization){
            console.log("Enter granted");
            gpio.digitalWrite(MagneticLock, gpio.HIGH);
```

```
            gpio.digitalWrite(BuzzerSound, gpio.HIGH);
            setTimeout(function(){
               gpio.digitalWrite(BuzzerSound, gpio.LOW);
            },500);
            setTimeout(function () {
               gpio.digitalWrite(MagneticLock, gpio.LOW);
               if(gpio.digitalRead(DoorGapSensor)){
                  doorAlert = setInterval(function(){
                     if(beep){
                        beep = 0;
                     }else{
                        beep = 1;
                     }
                     gpio.digitalWrite(BuzzerSound, beep);
                  },50);
               }
            }, 5000);
         }else{
            console.log("Enter denied");
            gpio.digitalWrite(MagneticLock, gpio.LOW);
            gpio.digitalWrite(BuzzerSound, gpio.HIGH);
            setTimeout(function(){
               gpio.digitalWrite(BuzzerSound, gpio.LOW);
            },1000);
         }
      });
   }
   console.log("Door Status : "+gpio.digitalRead(DoorGapSensor));
   if(gpio.digitalRead(DoorGapSensor)){
      doorAlert = setInterval(function(){
         if(beep){
            beep = 0;
        }else{
            beep = 1;
         }
         gpio.digitalWrite(BuzzerSound, beep);
      },50);
   }
   function doorStatus(state){
      console.log("Door Status : "+state);
      if(!state){
         if(doorAlert!=null){
            clearInterval(doorAlert);
         }
         gpio.digitalWrite(BuzzerSound, gpio.LOW);
      }
   }
```

```
gpio.wiringPiISR(DoorGapSensor, gpio.INT_EDGE_BOTH, function(delta) {
   newGapState = gpio.digitalRead(DoorGapSensor);
   if(gapTimeout != null){
     clearTimeout(gapTimeout);
   }
   gapTimeout = setTimeout(function(){
      if(newGapState==gapState){
         return;
      }
      gapState = newGapState;
      doorStatus(gapState);
   },250);});
```

APPENDIX B - LIGHT CONTROL

```
// Control Using MQTT
/*
+-----+-----+---------------------+------+---+---Pi 3---+---+------+---------------------+-----+-----
+

Device Physical wpi BCM
Fluorescent Lighting
WPi 27

+-----+-----+---------------------+------+---+---Pi 3---+---+------+---------------------+-----+-----
+
*/
var mqtt = require('mqtt');
var gpio = require('wiringpi-node');

var Lighting1 = 27;

gpio.wiringPiSetup();

gpio.pinMode(Lighting1,gpio.OUTPUT);
gpio.digitalWrite(Lighting1, gpio.LOW);

//-------------------------------------- MQTT -------------------------

// User API KEY
var apikey = '';

// Device Developer ID
var devid = '';

var url = 'mqtt://192.168.42.49' ; // use mqtts for secure connection

var settings = {
  port: 1883,   // use 8883 for secure connection
  clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),

};
```

```
// Data format
var data = {
   //"device_developer_id": devid,
   "data": {}
 };

// Connect to MQTT broker
var client = mqtt.connect(url, settings);

client.on('connect', function () {
   console.log("Connected to broker");
   client.subscribe('io.data');
})
client.on('message', function (topic, message) {

   try{
      console.log(message.toString());
      var data = JSON.parse(message.toString());
      parseCommand(data);
   }catch(e){console.log('Error: '+e);}
});

function parseCommand(cmd)
{
   switch (cmd.cmd) {
      case 'light':
         if(cmd.option.sub =='on')
         {
            if(cmd.option.sub2 == '1')
            {
               gpio.digitalWrite(Lighting1,gpio.HIGH);
               console.log('Fluorescent Ligthing off');
            }
         }
         else if(cmd.option.sub =='off')
         {
            if(cmd.option.sub2 == '1')
            {
               gpio.digitalWrite(Lighting1,gpio.LOW);
               console.log('Fluorescent Ligthing on')
            }

         }
         break; } }
```

APPENDIX C - MAIN SERVER

```
const Bot = require('node-telegram-bot-api');
const request = require('request');
var mysql      = require('mysql');
var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'root',
  password : 'root',
  database : 'ams'
});
connection.connect();
//Special token for sending message to telegram
const token = '572119888:AAHL14mOYkVxrz1pIZWExln4p4KGk0N_d1c';
var http = require('http');
var url = require('url');
console.log("Starting server");
var host = "192.168.42.49";
var port ="8000"
var isAlive = false;
var telegramId = 330441680;
const bot = new Bot(token, {polling: true});
bot.on('message', (msg) => {
  const chatId = msg.chat.id;
  console.log(chatId);
  console.log(msg);
  // send a message to the chat acknowledging receipt of their message
  bot.sendMessage(chatId, 'Received your message');
});

// Emptied the record

// Configure our HTTP server to respond all requests.

//Timer (Expired)
var timerExpired = null;
var timerCount = 7500;

var server = http.createServer(function (request, response) {
```

```
  var q = url.parse(request.url, true).query;
  //console.log(q);

  if(q.heartbeat ==1){
    //Wait for "data" receive event
    if(timerExpired == null){
      timerExpired = setTimeout(connectionDead,timerCount);
    }else{
      clearTimeout(timerExpired);
      timerExpired = setTimeout(connectionDead,timerCount);
    }

    if(!isAlive){
      console.log("Connection is alive" + request.url);
      bot.sendMessage(telegramId, "Client alive. The device is connected to the
server");
    }
    isAlive = true;
    response.writeHead(200, {"Content-type":"text/plain"});
    response.end("Client connected to server");

    return;
  }

  if(typeof q.reqEnter != 'undefined'){
    console.log("User Enter : "+q.reqEnter);
    verifyUser("Enter",q.reqEnter,response);
    return;
  }

  if(typeof q.reqExit != 'undefined'){
    console.log("User Exit : "+q.reqExit);
    verifyUser("Exit",q.reqExit,response);
    return;
  }

  console.log("Unknown request!");
});


// Listen on port 8000
server.listen(port,host, function(){
  console.log("Listening " + host + ":" +port);
  if(timerExpired == null){
     timerExpired = setTimeout(connectionDead,timerCount);
  }else{
    clearTimeout(timerExpired);
```

```
    timerExpired = setTimeout(connectionDead,timerCount);
  }

  //bot.sendMessage(telegramId, "Listening" + host + ":" +port);
});

function connectionDead(){
  //No heart beat
  console.log("Client timeout");
  bot.sendMessage(telegramId, "Client timeout..The device is disconnected from the
server");
  isAlive = false;
  //send message to telegram
}

/*// Function to get timestamp if necessary
function getDateTime() {
    var date = new Date();

    var hour = date.getHours();
    hour = (hour < 10 ? "0" : "") + hour;

    var min  = date.getMinutes();
    min = (min < 10 ? "0" : "") + min;

    var sec  = date.getSeconds();
    sec = (sec < 10 ? "0" : "") + sec;

    var year = date.getFullYear();

    var month = date.getMonth() + 1;
    month = (month < 10 ? "0" : "") + month;

    var day  = date.getDate();
    day = (day < 10 ? "0" : "") + day;

    return year + "/" + month + "/" + day + " @ " + hour + ":" + min + ":" + sec;
}*/

function verifyUser(action,uid,response){
  var auth = {
    authorization:0
  };

  connection.query('SELECT `id`,`name` FROM `users` WHERE `uid`= ?', [uid],function
(error, results, fields) {
    if (error){
```

```
      console.log(error);
      response.writeHead(200, {"Content-type":"application/json"});
      response.end(JSON.stringify(auth));
      return 0;
    }
    if(results.length < 1){
      console.log("Records Found : 0");
      response.writeHead(200, {"Content-type":"application/json"});
      response.end(JSON.stringify(auth));
      return 0;
    }
    console.log("Records Found : "+results.length);
    console.log(results[0].name);
    auth.authorization = 1;
    response.writeHead(200, {"Content-type":"application/json"});
    response.end(JSON.stringify(auth));

    updateLog(action,results[0].name);
  });
}
function updateLog(action,name){
  var date = new Date();
  bot.sendMessage(telegramId, date+" : ("+action+")"+name);
  var sql = "INSERT INTO log (action, name) VALUES (?,?)";
  connection.query(sql, [action,name],function (error, results, fields) {
    if (error){
      console.log(error);
      return;
    }
    console.log("Updated : "+results.insertId);     });}
```

APPENDIX D - LIGHT MQTT-BROKER

```
/* Include mosca library to start mqtt broker
 *  https://www.npmjs.com/package/mosca
 */
var mosca = require('mosca');
// Setup Mosca to listen on port 465 for native MQTT and port 443 for MQTT over
websocket
var settings = {
   port:1883,     //native mqtt
   http:{          //mqtt over websocket
      port:3001,
      bundle:true,
      static:"./"
   }
};
// Create and start MQTT broker with `settings` configuration
var server = new mosca.Server(settings);
// Display a message when the MQTT broker is ready
server.on('ready', function(){
   console.log("MOsca @ mqtt = 1883 & mqtt-ws = 3001");
});
//  Display a message and a client id when the client is connected to the broker
server.on('clientConnected', function(client) {
   console.log("Connected :"+client.id);
});

//  Display a message and a client id when the client is disconnected from the
broker
server.on('clientDisconnected', function(client) {
   console.log("Disconnected :"+client.id);
});
// Display the message published by the client
server.on('published', function(packet, client) {
   if(typeof client != 'undefined')
   console.log("From :"+client.id);
   console.log("Topic : "+packet.topic);
   console.log("Message :"+packet.payload.toString());
});
```

APPENDIX E - GUI MAIN PAGE

```
<head>
<title>AmsIot.com/mainpage</title>
<style>
        body {
          background-image: url("things.png");
          height: 100%;
          /* Center and scale the image nicely */
          background-position: center;
          background-repeat: no-repeat;
          background-size: cover;
        }
       /*Style the header*/
        .header {
       padding: 23px;
       text-align: center;
       font-size: 32px;
       }
       .button {
       display: inline-block;
       border-radius: 3px;
       background-color: #008CBA;
       border: none;
       color:  #FFFFFF;
       text-align: center;
       font-size: 28px;
       padding: 20px;
       width: 200px;
       transition: all 0.5s;
       cursor: pointer;
       margin-right: -320px;
       margin-left : 400px;
       margin-top : 40px;
       }
       .button span {
       cursor: pointer;
        display: inline-block;
        position: relative;
        transition: 0.5s;
```

```
        }
       .button span:after {
        content: '\00bb';
       position: relative;
        opacity: 0;
       top: 0;
       right: 5px;
       transition: 0.5s;
      }
       .button:hover span {
       padding-right: 10px;
       }
       .button:hover span:after {
        opacity: 1;
         right: 0;
       }
        </style>
         <nav class="navbar navbar-inverse">
       <div class="container-fluid">
         <div class="navbar-header">
           <a class="navbar-brand" href="/IoTAxcesattfyp/MainPage.php"></a>
         </div>
       </div>
          </nav>
        <body>
        <div class="header">
          <h1>IOT-Based Office Monitoring and Automation System</h1>
          </div>
           <div>
         <a class="button" href="door.php"><span>AxcesDoor </span></button1>
         <a class="button" href="light.html"><span>LightControl </span></button2>
          </body>
</html>
```

APPENDIX F - GUI DOOR LOG

```
<!DOCTYPE html>
<html>
<title>Axces IoT</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/datatables.min.css">

<style>
html,body,h1,h2,h3,h4,h5 {font-family: "Raleway", sans-serif}
</style>
<body class="w3-light-grey">

<!-- Top container -->
<div class="w3-bar w3-top w3-black w3-large" style="z-index:4">
  <button class="w3-bar-item w3-button w3-hide-large w3-hover-none w3-hover-
text-light-grey" onclick="w3_open();"><i class="fa fa-bars"></i>  Menu</button>
  <span class="w3-bar-item w3-center">IoT-Office Monitoring and Control
System</span>

</div>

<!-- Sidebar/menu -->
<nav class="w3-sidebar w3-collapse w3-white w3-animate-left" style="z-
index:3;width:300px;" id="mySidebar"><br>
  <div class="w3-container w3-row">
    <div class="w3-col s4">
      <img src="Logo Universiti Teknologi Malaysia (UTM).png" class="w3-circle w3-
margin-right" style="width:46px">
    </div>
    <div class="w3-col s8 w3-bar">
      <span>Welcome, <strong>Admin</strong></span><br>

    </div>
```

```
 </div>
 <hr>
 <div class="w3-container">
  <h5>Dashboard</h5>
 </div>
 <div class="w3-bar-block">
  <a href="#" class="w3-bar-item w3-button w3-padding-16 w3-hide-large w3-dark-grey w3-hover-black" onclick="w3_close()" title="close menu"><i class="fa fa-remove fa-fw"></i>  Close Menu</a>
  <a href="light.html" class="w3-bar-item w3-button w3-padding"><i class="fa fa-cog fa-fw"></i>  Light Control</a>
  <a href="index.php" class="w3-bar-item w3-button w3-padding"><i class="fa fa-history fa-fw"></i>  Main Page</a></div>
</nav>

<!-- Overlay effect when opening sidebar on small screens -->
<div class="w3-overlay w3-hide-large w3-animate-opacity" onclick="w3_close()" style="cursor:pointer" title="close side menu" id="myOverlay"></div>

<!-- !PAGE CONTENT! -->
<div class="w3-main" style="margin-left:300px;margin-top:43px;">

 <!-- Header -->
 <header class="w3-container" style="padding-top:22px">
  <h5><b><i class="fa fa-dashboard"></i> My Dashboard</b></h5>
 </header>


 <div class="w3-quarter">
  <div class="w3-container w3-orange w3-text-white w3-padding-16">
   <div class="w3-left">
    <i class="fa fa-users w3-xxxlarge"></i>
   </div>
   <div class="w3-right">
     <h3>50</h3>
   </div>
   <div class="w3-clear">
   </div>
   <h4>Workers</h4>
  </div>
 </div>

 <div class="w3-twothird" style="margin-left:300px;overflow-x:auto;">
  <h5>Door Log</h5>
  <!--table class="w3-table w3-bordered">
      <tr>
       <th>Name</th>
```

```
    <th>Times</th>
    <th>Day</th>
    <th>Dates</th>
    <th>Log</th>
   </tr>
   <tr>
    <td></td>
    <td></td>
    <td></td>
   </tr>
   <tr>
    <td></td>
    <td></td>
    <td><i></i></td>
   </tr>
   <tr>
    <td><i </i></td>
    <td></td>
    <td></i></td>
   </tr>
   <tr>
    <td></i></td>
    <td></td>
    <td><i></i></td>
   </tr>
   <tr>
    <td></i></td>
    <td></td>
    <td><i></i></td>
   </tr>
   <tr>
    <td></i></td>
    <td></td>
    <td><i></i></td>
   </tr>
  </table-->
</div>

<table class="table table-striped compact" cellspacing="0" id="dtLog">
<thead>
 <tr>
  <th>Date/Time</th>
  <th>Name</th>
  <th>Log</th>
 </tr>
 </thead>
 <tbody>
```

```
      </tbody>
   </table>

</div>

<script>
// Get the Sidebar
var mySidebar = document.getElementById("mySidebar");

// Get the DIV with overlay effect
var overlayBg = document.getElementById("myOverlay");

// Toggle between showing and hiding the sidebar, and add overlay effect
function w3_open() {
   if (mySidebar.style.display === 'block') {
      mySidebar.style.display = 'none';
      overlayBg.style.display = "none";
   } else {
      mySidebar.style.display = 'block';
      overlayBg.style.display = "block";
   }
}

// Close the sidebar with the close button
function w3_close() {
   mySidebar.style.display = "none";
   overlayBg.style.display = "none";
}
</script>

</body>
</html>

<script type="text/javascript" src="js/jquery-3.3.1.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
<script type="text/javascript" src="js/datatables.min.js"></script>
<script type="text/javascript" src="js/door.js"></script>
```

APPENDIX G - GUI LIGHT CONTROL

```
<!DOCTYPE html>
<html>

<head>
   <title>IoT-Based Office Monitoring and Control System</title>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
   <meta name="doc-type" content="Web Page" />
   <link rel="stylesheet" href="css/bootstrap.min.css" type="text/css" />
   <link rel="stylesheet" href="css/style.css" type="text/css" />
</head>
<style>
    body {
         background-image: url("things.png");
         height: 100%;

         /* Center and scale the image nicely */
         background-position: center;
         background-repeat: no-repeat;
         background-size: cover;
        }
</style>

<nav class="navbar navbar-inverse">
      <div class="container-fluid">
        <div class="navbar-header">
           <br><a span class="glyphicon glyphicon-home" href="index.php"></a>
         </div>

       <ul class="nav navbar-nav navbar-right">
        <li><a onClick=loginView() href="#"><span class="glyphicon glyphicon-log-
in"></span></a></li>
       </ul>
      </div>
    </nav>
<body>
   <div id="main">
      <br>
      <br>
```

```
<h1 id="title">
   IoT-Based Office Monitoring and Control System
</h1>
<br>
<br>

<div class="panel" style="text-align:center">
   <div class="col-sm-6"style="text-align:center">
      <h2>Fluorescent Lighting </h2>

      <div class="container gpio_pins center">
         <div class="row">
            <div class="col-xs-3">
             <br><p>LIGHTING STATUS 1</p>
               <br>
               <br>
               <img id="led1Indicator" src="images/indicator/1.png"
alt="indicator">

            </div>

            <div class="col-xs-3">
                  <br><p>LIGHTING STATUS 2</p>
                    <br>
                    <br>
                    <img id="led2Indicator" src="images/indicator/1.png"
alt="indicator">

                </div>

            <div class="col-xs-3">
              <br> <p>LIGHTING CONTROL 1</p>
               <br>
               <img id="led1Control" src="images/button/vertical_on_off/1.png"
alt="indicator">
            </div>

            <div class="col-xs-3">
                  <br> <p>LIGHTING CONTROL 2</p>
                   <br>
                   <img id="led2Control"
src="images/button/vertical_on_off/1.png" alt="indicator">
                </div>

         </div>
      </div>
   </div>
```

```
      </div>
    </div>
    <div style="text-align:center">
      <br><br><div class="panel not_too_wide" style="text-align:center">
        <div class="divider"></div>
        <h2>Credits</h2>
        <p>
          DanielBull @ https://github.com/NeonHorizon/berryio
        </p>
      </div>
    </div>
</body>

</html>

<script src="js/jquery-3.3.1.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/control.js"></script>
<script src="js/font.js"></script>
<script src="js/paho-mqtt.js"></script>

<script>
    var mqtt = null;

    // User API KEY
    var apiKey ="";

    // Device Developer ID
    var devid = '';

    $(document).ready(function () {
      mqttConnect();
    });

    function mqttConnect() {
      if (mqtt != null) {
        mqtt.disconnect();
        mqtt = null;
      }

      mqtt = new Paho.MQTT.Client("192.168.42.49", Number(3001), "/mqtt",
'mqttlightwebserver' + Math.random);
      mqtt.onConnectionLost = onMqttConnectionLost;
      mqtt.onMessageArrived = onMqttMessageArrived;
      mqtt.connect({
```

```
        keepAliveInterval: 86400,
        onSuccess: onMqttConnectSuccess,
        onFailure: onMqttConnectFail,
        reconnect: true
      });

      function onMqttConnectSuccess() {
        console.log("MQTT > onMqttConnectSuccess");
        $('#msg').text("MQTT > onMqttConnectSuccess");
        mqtt.subscribe(apiKey + "/#");

        var message = new Paho.MQTT.Message(JSON.stringify({
          "msg": "Iot-Lighting Control Connected"
        }));
        message.destinationName = apiKey + "/io.data";
        mqtt.send(message);
      }

      function onMqttConnectFail(res) {
        console.error("MQTT > onMqttConnectFail :" + res.errorMessage);
        $('#msg').text("MQTT > onMqttConnectFail :" + res.errorMessage);
      }

      function onMqttConnectionLost(responseObject) {
        if (responseObject.errorCode !== 0) {
          console.warn("MQTT > onMqttConnectionLost:" +
responseObject.errorMessage);
          $('#msg').text("MQTT > onMqttConnectionLost:" +
responseObject.errorMessage);
        } else {
          $('#msg').text("MQTT > onMqttConnectionLost");
        }
      }

      function onMqttMessageArrived(message) {
        console.log(message.topic + ": " + message.payloadString);

        var n = message.topic.indexOf('/')+1;
        var topic = message.topic.substr(n);
        $('#topic').text(topic);
        $('#payload').text(message.payloadString);
      }
    }
</script>
```