

A PROTOTYPE OF SMART PARCEL BOX

BY

Lee Jia Heng

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)
COMMUNICATIONS AND NETWORKING**

Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2022

REPORT STATUS DECLARATION FORM

Title: A PROTOTYPE OF SMART PARCEL BOX


Academic Session: JAN 2022

I LEE JIA HENG
(CAPITAL LETTER)


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

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

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

113, JALAN PANTAI
34350 KUALA KURAU
PERAK

TAN LYK YIN

Supervisor's name

Date: 20 April 2022

Date: 20 APRIL 2022

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 20 April 2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Lee Jia Heng (ID No: 18ACB02515) has completed this final year project entitled “**A PROTOTYPE OF SMART PARCEL BOX**” under the supervision of Ms. Tan Lyk Yin (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.


Yours truly,



(LEE JIA HENG)

DECLARATION OF ORIGINALITY

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

Signature :  _____

Name : LEE JIA HENG

Date : 20 April 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ms Tan Lyk Yin who has given me this bright opportunity to engage in the development of the Prototype of Smart Parcel Box. It is my first step to establish a career in Internet of Things (IoT) and smartdevice field. Her guidance and inspiration given to me had help me a lot in completing this project. A million thanks to you.

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

ABSTRACT

A Parcel Box is parcel receiving box installed in front of the house. The downside of a traditional design parcel box is that the parcel has the potential to be damaged or stolen from the unlocked parcel box. Moreover, a recipient will need to check the parcel box constantly to prevent the parcel from being stolen or damaged.

In the proposed solution, a prototype of smart parcel box is developed. It can be unlocked to receive a parcel and locked after receiving the parcel. This is to prevent the parcel from being stolen by keeping it locked in the parcel box. The smart parcel box can be unlocked by scanning the QR code on the parcel to verify whether the parcel is the one that belongs to the recipient. The smart parcel box will update the database with the parcel received time right after it received a parcel and locked it. This is to avoid someone trying to use the same QR code to unlock the parcel box again.

The proposed solution also come with a website that helps user to easily monitor the parcel information and the parcel box. The website will show the user of the incoming parcel and received parcel and able to verify the received parcel. Besides, when someone is trying to use an unmatched QR code to access the parcel box, the system will record the time and the QR code used and display it on the website.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
SAMPLE OF SUBMISSION SHEET FOR FYP	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	xii
LIST OF TABLES	xv
LIST OF ABBREVIATIONS	xvi
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Project Objectives	2
1.3 Project Scope and Direction	3
1.4 Contributions	3
1.5 Report Organization	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Raspberry Pi vs Arduino	5
2.1.1 Raspberry Pi	5
2.1.2 Arduino	5
2.1.3 Comparison between Raspberry Pi and Arduino	6
2.2 Barcode vs QR Code	7
2.2.1 Barcode	7
2.2.2 QR Code	7
2.2.3 Comparison between Barcode and QR Code	8
2.3 Survey on the Problem	9
2.4 Review of the Existing Systems	9
2.4.1 Arduino Based Smart Box	10
2.4.2 Smart Freight Box	12
2.4.3 Comparison between Two Existing Systems	15

CHAPTER 3 SYSTEM METHODOLOGY /APPROACH	16
3.1 System Development Models	16
3.1.1 System Development Model 1: Waterfall Model	16
3.1.2 System Development Model 2: Spiral Model	17
3.3.3 System Development Model 3: Prototyping Model	18
3.3.4 Selected Model	19
3.2 System Requirement	19
3.2.1 Hardware	19
3.2.2 Software	22
3.3 Functional Requirement	23
3.3.1 QR Code Detection	23
3.3.2 QR Code Verification	23
3.3.3 Photo Capture	23
3.3.4 Notification to Receiver	24
3.3.5 Website	24
3.3.5.1 User Registration	24
3.3.5.2 User Login	24
3.3.5.3 Insert New Tracking Number	24
3.3.5.4 Incoming Parcel Tracking	24
3.3.5.5 Received Parcel Confirmation	24
3.3.5.6 Abnormal Activity Checking	25
3.3.5.7 Lock or Unlock Parcel Box Manually	25
3.3.5.8 View History Record	25
3.3.5.9 Account Password Reset	25
3.3.5.10 Account Deletion	25
3.4 Project Timeline	26
3.5 Estimated Cost	28
3.6 Concluding Remark	29
CHAPTER 4 SYSTEM DESIGN	30
4.1 System Architecture	30
4.1.1 Overall System Design	30
4.1.2 Website Architecture Design	31
4.2 Functional Modules in the System	31
4.2.1 QR Code Detection	31

4.2.2 QR Code Verification	32
4.2.3 Photo Capture	32
4.2.4 Notification to Receiver	32
4.2.5 Website	32
4.2.5.1 User Registration	33
4.2.5.2 User Login	33
4.2.5.3 Insert New Tracking Number	33
4.2.5.4 Incoming Parcel Tracking	33
4.2.5.5 Received Parcel Confirmation	33
4.2.5.6 Abnormal Activity Checking	34
4.2.5.7 Lock or Unlock Parcel Box Manually	34
4.2.5.8 View History Record	34
4.2.5.9 Account Password Reset	34
4.2.5.10 Account Deletion	34
4.3 System Flowchart	35
4.3.1 Overall System	35
4.3.2 User Registration	36
4.3.3 User Login	37
4.3.3 Insert New Tracking Number	38
4.3.5 Account Password Reset	39
4.3.6 Account Deletion	40
4.4 Database design	40
4.4.1 Entity Relationship Diagram (ERD)	41
4.4.2 Data Dictionary - Users Database	41
4.4.3 Data Dictionary - Tracking Database	42
4.4.4 Data Dictionary – Abnormal Database	43
4.5 Algorithm design	44
4.5.1 Hashing Algorithm	44
4.6 Graphical User Interface Design (Website)	44
4.6.1 Login Page	44
4.6.2 Account Register Page	45
4.6.3 Home Page	46
4.6.4 Insert New Tracking Number Page	47
4.6.5 Parcel Box Control Page	47
4.6.6 History Record Page	48
4.6.7 History Record Page – Parcel History Record	48
4.6.8 History Record Page – Abnormal Activity History	49

4.6.9 Account Setting Page	50
4.6.10 Account Setting Page – Change Password	50
4.6.11 Account Setting Page – Delete Account	51
4.7 Use Case Diagram and Description	52
4.7.1 Use Case Diagram	52
4.7.2 Use Case Description	52
4.7.2.1 Use Case Description for UC001	52
4.7.2.2 Use Case Description for UC002	53
4.7.2.3 Use Case Description for UC003	53
4.7.2.4 Use Case Description for UC004	53
4.7.2.5 Use Case Description for UC005	54
4.7.2.6 Use Case Description for UC006	54
4.7.2.7 Use Case Description for UC007	54
4.7.2.8 Use Case Description for UC008	55
4.7.2.9 Use Case Description for UC009	55
4.7.2.10 Use Case Description for UC010	55
4.7.2.11 Use Case Description for UC011	56
4.8 Concluding Remark	56

CHAPTER 5 SYSTEM IMPLEMENTATION **57**

5.1 Hardware Setup	57
5.2 Software Setup	60
5.2.1 Raspberry Pi OS	60
5.2.2 OpenCV / pyzbar	61
5.2.3 WAMP Server	61
5.2.4 Telegram Bot	63
5.3 Setting and Configuration	64
5.3.1 Raspberry Pi Camera Module	64
5.3.2 WAMP Server	65
5.3.3 Telegram Bot	68
5.3.4 Main Python Script	69
5.4 System Operation	72
5.4.1 Main Function	72
5.4.2 Parcel Box Control Function	73
5.4.3 Telegram Bot Notification	74
5.5 Concluding Remark	74

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	75
6.1 System Testing and Performance Metrics	75
6.2 Testing Setup and Result	75
6.2.1 Testing on the Raspberry Pi Camera Module	75
6.2.2 Testing on the Servo	78
6.2.3 Testing on the Database	78
6.2.4 Testing on the Telegram Bot	79
6.3 Project Challenges	79
6.4 Objectives Evaluation	80
6.5 Concluding Remark	80
CHAPTER 7 CONCLUSION	81
7.1 Conclusion	81
7.2 Future Work	82
BIBLIOGRAPHY	83
APPENDIX A	A-1
A.1 Weekly Report	A-1
A.2 Poster	A-13
PLAGIARISM CHECK RESULT	
CHECK LISTS	

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1	Raspberry Pi	5
Figure 2.2	Arduino board	6
Figure 2.3	Example of barcode	7
Figure 2.4	Example of QR code	8
Figure 2.5	Statistic on the report	9
Figure 2.6	Block diagram of Arduino based smart parcel box system	10
Figure 2.7	Front side of smart parcel box	12
Figure 2.8	Flow chart of the smart freight box	13
Figure 3.1	Waterfall Model	16
Figure 3.2	Spiral Model	17
Figure 3.3	Prototype Model	18
Figure 3.4	Raspberry Pi 1 Model A+	20
Figure 3.5	Raspberry Pi Camera Module (5Megapixel)	20
Figure 3.6	SG90 Micro Servo	21
Figure 3.7	Jumper Wires	21
Figure 3.8	Wi-Fi Dongle	21
Figure 3.9	Raspberry Pi OS	22
Figure 3.10	WAMP Server	22
Figure 3.11	Python Programming Language	22
Figure 3.12	OpenCV	23
Figure 3.13	Telegram	23
Figure 4.1	System Architecture Diagram	30
Figure 4.2	Website Architecture Diagram	31
Figure 4.3	System Flowchart	35
Figure 4.4	User Registration Flowchart	36
Figure 4.5	User Login Flowchart	37
Figure 4.6	Account Password Reset Flowchart	38
Figure 4.7	Account Password Reset Flowchart	39

Figure 4.8	Account Deletion Flowchart	40
Figure 4.9	ERD	41
Figure 4.10	Result of Hashing Algorithm	44
Figure 4.11	GUI of User Login Page	44
Figure 4.12	GUI of Account Registration Page	45
Figure 4.13	GUI of Home Page	46
Figure 4.14	GUI of Insert New Tracking Number Page	47
Figure 4.15	GUI of Parcel Box Control Page	47
Figure 4.16	GUI of History Record Page	48
Figure 4.17	GUI of History Record Page – Parcel Record History	48
Figure 4.18	GUI of History Record Page – Abnormal Activity History	49
Figure 4.19	GUI of Account Setting Page	50
Figure 4.20	GUI of Account Setting – Change Password	50
Figure 4.21	GUI of Account Setting – Delete Account	51
Figure 4.22	Use Case Diagram	52
Figure 5.1	System Prototype of Smart Parcel Box	57
Figure 5.2	Front View of System Prototype	58
Figure 5.3	Wiring Diagram of System	58
Figure 5.4	Raspberry Pi OS Download Page	60
Figure 5.5	Raspberry Pi Imager	60
Figure 5.6	WAMP Server Download Page	61
Figure 5.7	WAMP Server Running Status	61
Figure 5.8	WAMP Server Homepage	62
Figure 5.9	Telegram BotFather	63
Figure 5.10	Chat History with BotFather	63
Figure 5.11	Raspberry Pi Configuration	64
Figure 5.12	PHPMyAdmin Page	65
Figure 5.13	Create Database Page	65
Figure 5.14	Create Table Page	66
Figure 5.15	User Accounts Page	66
Figure 5.16	Create User Account Page	67
Figure 5.17	Chat with Telegram Bot	68
Figure 5.18	Telegram Chat ID	68
Figure 5.19	Telegram Bot Python Script	69
Figure 5.20	Main Python Script (1)	69
Figure 5.21	Main Python Script (2)	70
Figure 5.22	Main Python Script (3)	70

Figure 5.23	Parcel Box Control Function Script (1)	71
Figure 5.24	Parcel Box Control Function (2)	71
Figure 5.25	Result of the Main Function	72
Figure 5.26	Result of Parcel Box Control Function	73
Figure 5.27	Telegram Bot Result	74
Figure 6.1	Flaws on QR Code (1)	77
Figure 6.2	Flaws on QR Code (2)	77

LIST OF TABLES

Table Number	Title	Page
Table 2.1	Comparison between Raspberry Pi and Arduino	6
Table 2.2	Comparison between Barcode and QR Code	8
Table 2.3	Comparison between Two Existing Systems	15
Table 3.1	Specifications of Raspberry Pi	20
Table 3.2	Project Timeline for FYP1	26
Table 3.3	Project Timeline for FYP2	27
Table 3.4	Estimated Cost for the Development of System	28
Table 4.1	Data Dictionary – Users Database	41
Table 4.2	Data Dictionary - Tracking Database	42
Table 4.3	Data Dictionary - Abnormal Database	43
Table 4.4	Use Case Description of UC001	52
Table 4.5	Use Case Description of UC002	53
Table 4.6	Use Case Description of UC003	53
Table 4.7	Use Case Description of UC004	53
Table 4.8	Use Case Description of UC005	54
Table 4.9	Use Case Description of UC006	54
Table 4.10	Use Case Description of UC007	54
Table 4.11	Use Case Description of UC008	55
Table 4.12	Use Case Description of UC009	55
Table 4.13	Use Case Description of UC010	55
Table 4.14	Use Case Description of UC011	56
Table 5.1	Pin Connection between Raspberry Pi and SG90 Micro Servo	59
Table 6.1	Result of System Testing (1)	76
Table 6.2	Result of System Testing (2)	76
Table 6.3	Result of System Testing (3)	76
Table 6.4	Result of System Testing (4)	76
Table 6.5	Result of System Testing (5)	76
Table 6.6	Result of System Testing (6)	78
Table 6.7	Result of System Testing (7)	78
Table 6.8	Result of System Testing (8)	79
Table 6.9	Result of System Testing (9)	79

LIST OF ABBREVIATIONS

<i>QR</i>	Quick Response
<i>IDE</i>	Integrated Development Environment
<i>IoT</i>	Internet of Thing
<i>URL</i>	Uniform Resource Locator
<i>RFID</i>	Radio Frequency Identification
<i>GSM</i>	Global System Mobile
<i>CNC</i>	Computer Numerical Control
<i>MCU</i>	Micro-Controller Unit
<i>DC</i>	Direct Current
<i>IC</i>	Identity Card
<i>SDLC</i>	System Development Life Cycle
<i>SoC</i>	System on a Chip
<i>MB</i>	Megabyte
<i>MP</i>	Megapixel
<i>HDMI</i>	High-Definition Multimedia Interface
<i>USB</i>	Universal Serial Bus
<i>GPIO</i>	General Purpose Input/Output
<i>OS</i>	Operating System
<i>WAMP</i>	Windows Apache2 MySQL PHP
<i>SQL</i>	Structured Query Language
<i>PHP</i>	Hypertext Preprocessor
<i>FYP</i>	Final Year Project
<i>FOC</i>	Free of Charge
<i>GUI</i>	Graphical User Interface
<i>ERD</i>	Entity Relationship Diagram
<i>MD</i>	Message-Digest
<i>BLOB</i>	Binary Large Object
<i>SD</i>	Secure Digital
<i>API</i>	Application Programming Interface
<i>RAM</i>	Random Access Memory
<i>LED</i>	Light-Emitted Diode

CHAPTER 1

Introduction

Nowadays, with the convenience that brought from the Internet, people tend to choose to shop online instead of shopping in a physical store to avoid the crowd and can compare the price easily [1]. Besides, the outbreak of the Covid-19 pandemic forces people to stay at home and practice social distancing. Because of this pandemic, people have no choice but to do everything online, including shopping as well. The number of online shoppers is increasing rapidly [2], from groceries, and daily supplies to clothing or luxury products, everything that is needed, you can be buy online. Since digital shopping is trending these years, it had directly increased the demand for the delivery service.

Delivery service is a process to deliver the products and goods to the recipient. Based on the Cambridge Dictionary [3], a mailbox is a metal box placed outside of a person's house, where to send and deliver letters. The parcel box, which is similar to the mailbox but the parcel box, is bigger because it is used to collect the parcel. Parcel box or mailbox see as an important role in the delivery service. The deliveryman puts the parcel inside the parcel box and the recipient gets the parcel from the parcel box.

1.1 Problem Statement and Motivation

People encounter some of the problems with the current process of parcel receiving, but there are rarely have the solution to solve the problems. Imagine that if the recipient is not at home when the deliveryman arrives with their parcel, the deliveryman can choose whether to leave the parcel in the parcel box or at the doorstep or come back at another time when the recipient is at home. If the deliveryman decides to leave the parcel in the parcel box or at the doorstep (in the case that the recipient has no parcel box outside the house), there is a chance that the parcel box could be stolen or damaged [4]. The weather is also one of the concerns that could damage the parcel, for example, on a rainy day, an uncollected parcel just happens to be at the doorstep or inside of a non-waterproof parcel box, the parcel will be damaged from the rain. In another case as well, if the

CHAPTER 1

deliveryman decides to come back at another time when the recipient is at home, the delivery delay and cost are increased. Because the deliveryman will have to visit the recipient's house again. Besides, when the recipient urgently needs the parcel without wanting to wait for the next delivery arrangement, they must drive to the warehouse to pick up their parcels, which increased the cost of the delivery services.

Now that the problem is stated from the point of view of the deliveryman and recipient, what solution can we propose to help them solve the problem? There are some companies that provide parcel receiving services, for example, PostCo and Parcel365 [5]. These companies charge people to help them receive parcels when they are not at home. The parcel recipient needs to go to the service centre to pick up their parcel. Although this solution is solving the problem of the absence of a recipient, it also requires the recipient to make an extra trip to the parcel receiving service centre to pick up their parcel. The cost of delivery services is increased because the recipient needs to pay for the parcel receiving service and might be time-consuming if the service centre is extremely far from their house. The proposed smart parcel box will help people to receive the parcel when the owners are away from home, or even when they are outstation. If the mobile phone is connected to the Internet, the recipient can check the smart parcel box has received the parcel.

With the number of online shoppers increasing, the process of delivery service is getting important. Especially the guarantee of receiving the correct parcels safely is one of the main targets that advertised by the delivery service companies. Hence, the proposed smart parcel box can assist in increasing security, reducing the delivery delay, and reducing the delivery cost during the delivery process.

1.2 Project Objectives

The aim of the project is to introduce an alternate way for people to receive their parcels. The prototype of the smart parcel box will function as a recipient that will help to receive the parcel from the deliveryman. The entire process of the parcel receive will be done automatically without any instruction or help from a person. The features of the proposed system include a database that stores all the parcel information, parcel verification and authentication, and a security system for the parcel box itself.

The three main objectives of this project including:

- a. To design a prototype smart parcel box to receive a parcel using Raspberry Pi.
- b. To implement a verification process to unlock the smart parcel box.
- c. To implement a module informing the user that the parcel is received.

The first objective of the project is to design a prototype smart parcel box using Raspberry Pi. The purpose is to resolve the problem that the parcel is out of delivery, but the parcel's recipient is away from the home. The second objective is to have a process of parcel verification to unlock the smart parcel box. This is to resolve the problem of unauthorized access to the smart parcel box and to make sure the smart parcel box received the correct parcel. While the third objective is to implement a notification module to inform the user of the received parcel. The purpose of this objective is to inform and update the recipient about the received parcel so that recipient could check on the smart parcel box once they come home.

This project is using Raspberry Pi as the main controller to control the whole smart parcel box system. It can process the information of QR code scanned by the camera module and verifies the information with the database, the front door will open once the parcel is verified. Then, the Raspberry Pi will notify the recipient that the parcel had received.

1.3 Project Scope and Direction

This project develops a prototype of a smart parcel box. This project will be developed by using Raspberry Pi and configured with a computer. This project will be able to solve the parcel receiving problem faced by the deliveryman and recipient. Besides, since this project is just a prototype product of a smart parcel box, therefore it is only limited to receiving one parcel at one time, and it is also limited to one deliveryman and one recipient. The prototype of the smart parcel box is not weather-proof.

1.4 Contributions

The proposed system of the smart parcel box in this project will be able to solve the problem faced by the current parcel delivery process when the recipient is not at home. The absence of the recipient in the parcel delivery process will cause a waste of time

CHAPTER 1

for both the deliveryman and the recipient himself. The smart parcel box will be playing the role of a recipient in the parcel delivery process when the recipient is not at home, it will have the feature of parcel verification and some security measurement as well.

The smart parcel box itself is smart enough to verify whether the parcel that will be received belongs to the recipient. After that, the smart parcel box will open the door to receive the parcel and notify the recipient that the parcel is received. Otherwise, the smart parcel box will not open the door and try to notify the deliveryman that the parcel does not belong to the recipient.

In addition, the smart parcel box also encourages the practice of zero-contact delivery. Especially in the Covid-19 pandemic, any contact between humans is potentially to transmit the covid-19 virus. Therefore, the smart parcel box is a way to avoid direct contact between the deliveryman and recipient to keep them safe.

1.5 Report Organization

This project consists of seven chapters. Chapter One, the introduction, discuss the problem statement and motivation of this project, followed by the project objective, project scope, and the contributions of the project. While Chapter Two is the research of the literature reviews. Furthermore, Chapter Three is the system methodology and approach. Moreover, the fourth chapter is about the system design. Chapter Five, is the system implementation. While the system evaluation and discussion are discussed in Chapter Six. Finally, Chapter Seven is the conclusion of this project.

CHAPTER 2

Literature Reviews

2.1 Raspberry Pi vs Arduino

2.1.1 Raspberry Pi

Raspberry Pi could be used as the main component in the development of a smart parcel box. Raspberry Pi is like a functional computer but in a tiny and low-cost package. Raspberry Pi is also known as a single-board computer, just like any other computer that is made up of different components, but it is built on a single printed circuit board [6]. Raspberry Pi is like a mini-size, general-purpose computer and comes up with Linux based operating system, so it can do multitasking. Figure 2.1 shows the Raspberry Pi.

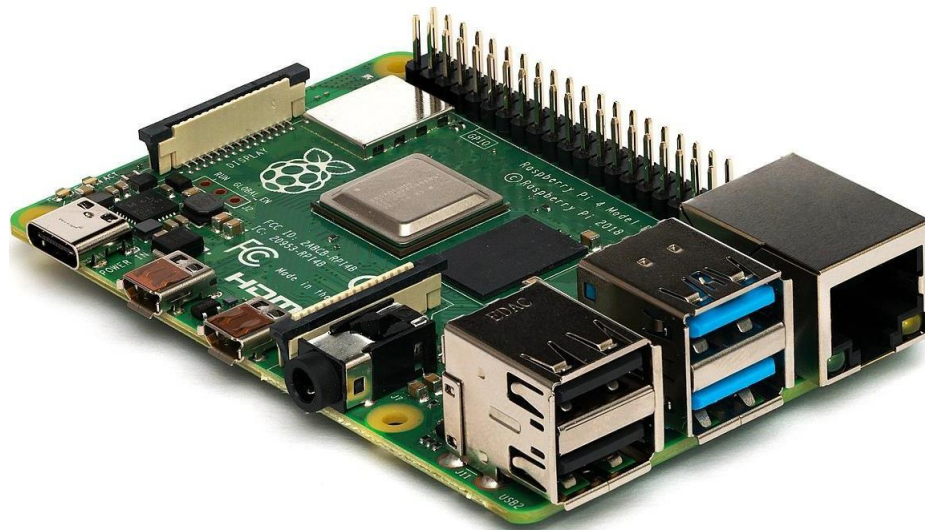


Figure 2.1. Raspberry Pi

2.1.2 Arduino

According to the Arduino website [7], Arduino is an open-source electronics circuit board, it is more like a controller. Arduino can read input and transform it as output. For example, Arduino reads the input from a sensor or a Twitter message, and then Arduino could turn on an LED or do some online activity like posting something on social media. The output depends on what the user configures it to perform. The Arduino boards could be configured by Arduino programming language and Arduino Software,

CHAPTER 2

which is Arduino IDE (Integrated development environment). Arduino is cheap in price, open-source, cross-platform, and extensible software. Figure 2.2 shows the Arduino board.

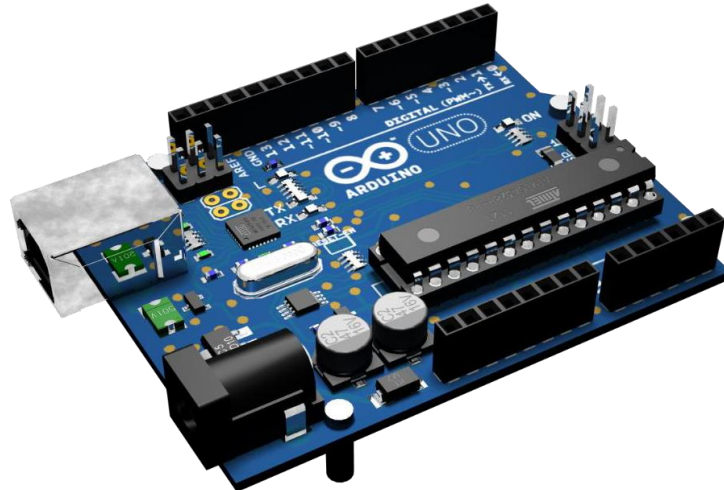


Figure 2.2. Arduino board

2.1.3 Comparison between Raspberry Pi and Arduino

Table 2.1 Comparison between Raspberry Pi and Arduino

Comparison	Raspberry Pi	Arduino
Definition	Minicomputer	Microcontroller
Complexity	Requires complex tasks like installation of libraries and software	Not complex
Cost	Expensive	Lower cost
Connectivity	Easily connected to the Internet	Require integration of external hardware to connect to the Internet
Languages Supported	Python, C, C++, and ruby	Arduino language, C and C++
Operating System	Linux OS	Does not support OS

Although Arduino is sound better in the project of automation, but Raspberry Pi is more suitable for the development of the smart parcel box. First, Raspberry Pi supported languages including Python, but Arduino supports its own language, which is Arduino

programming language and C or C++, this makes Raspberry Pi easy to program compared to Arduino [8]. Besides, Raspberry Pi can connect to a network as it comes with a native Ethernet port and Wi-Fi connectivity. However, Arduino is not network-enabled but some wiring and coding are still able to get Arduino to connect to a network [9], which the process is complicated compared to Raspberry Pi. Moreover, connection to the Internet is important because the parcel box needs to be online to notify the recipient whenever a parcel had received, or the recipient could update with the new parcel information. After that, Raspberry Pi is using Linux operating system while Arduino does not support the use of the operating system. The support of the operating system is to make the configuration on the Raspberry Pi a lot easier.

2.2 Barcode vs QR Code

2.2.1 Barcode

Barcode is a code that is formed by a pattern of lines of varying widths and space between each line, and the barcode can be read by a one-dimensional machine [10]. A barcode typically contains some simple information like numbers, characters, and other symbols. Besides, a barcode can only hold for about 8 to 25 characters, and only store the information in one direction, which is horizontally from left to right. Figure 2.3 shows the example of a barcode.



Figure 2.3 Example of barcode

2.2.2 QR Code

According to a website, MPC [11], a QR code, known as a matrix code, is a two-dimensional (2-D) barcode. Unlike a 1-D barcode, a QR code can store the information in two directions, which is horizontally and vertically. Also, a QR code can store

information like an email address, plain text, and website URLs. Figure 2.4 shows the example of QR code.



Figure 2.4 Example of QR code

2.2.3 Comparison between Barcode and QR Code

Table 2.2 Comparison between Barcode and QR Code

Comparison	Barcode	QR Code
Amount of Information Stored	Lesser	More (It is used to store multimedia data)
Correction Ability	No	Yes

In this project, QR Code has chosen to store the information used for the purpose of parcel verification. The reason to choose a QR code over a barcode is that the information that can store in a QR code is way more than a barcode can hold. Other than that, QR codes also have a correction ability, which means that QR code is still able to be read although it is covered by dirt or scratches [12].

2.3 Survey on the Problem

As the problem mentioned in the problem statement from Chapter 1, the parcels or mails usually got stolen or damaged as its lacks of security provided by the current existing parcel box. According to a survey conducted [13], the survey had shown that even back in the year 2017, there are about 37% of the United States consumers do not trust crowdsourced, a delivery service provider. Among 70% of the respondents had stated

that their deliveries arrive late and about 42% of them suffered from a damaged delivery. Moreover, based on [14], in the past six months, there are 51% of online shoppers experienced at least one package lost on their online order. About 16% of the online shoppers state that the reason they have not received a package is that their package was stolen. Besides, there are 57%, which is half of the online shoppers say that their parcels are usually been left in an unsecured area like the front porch.

On the other hand, based on a report by C+R Research [15], the number of stolen parcel is increasing in 2020 compared to 2019. The report demonstrates that there are 43% of the respondents experienced the problem of a package stolen, which had risen from the year 2019 (36%). There are 64%, which is nearly two-thirds of the respondents mentioned that they had experienced the package stolen more than once. Figure 2.5 shows the statistics of the report.

**Package Theft:
2019 vs. 2020**



Figure 2.5 Statistic on the report

2.4 Review of the Existing Systems

Some researchers have done some research on this problem. They also had developed some prototypes of smart parcel box as well. The current practice that has been used to solve this problem include the use of Arduino, IoT (Internet of Thing), RFID (Radio Frequency Identification) technology, and GSM (Global System Mobile) technology.

2.4.1 Arduino Based Smart Box

According to Nonthaputha [16], they are solving the problem by using Arduino based smart boxes for parcel receiving purposes. In the research paper, they mentioned that Arduino is the best open-source board for using software programming to control the hardware and many applications is Arduino based like smart energy meters and wireless motion detecting systems. In their proposed system, they use two Arduino boards, which are Arduino#1 and Arduino#2. Arduino#1 is first used to handle the magnetic door when the user key in the passcode, Arduino#1 is then checking the type of incoming package, for example, parcels or mail through motion sensors. While Arduino#2 is controlling a mini-CNC (Computer numerical control) for signing the parcels or mails after the Arduino#1 had checked the parcels or mails, after that, Arduino#2 will connect to MCU (Micro-controller unit) node and send the message by Line application to the parcel recipient's mobile phone after signing the parcel. Figure 2.6 shows the block diagram of the system of Arduino based smart parcel box.

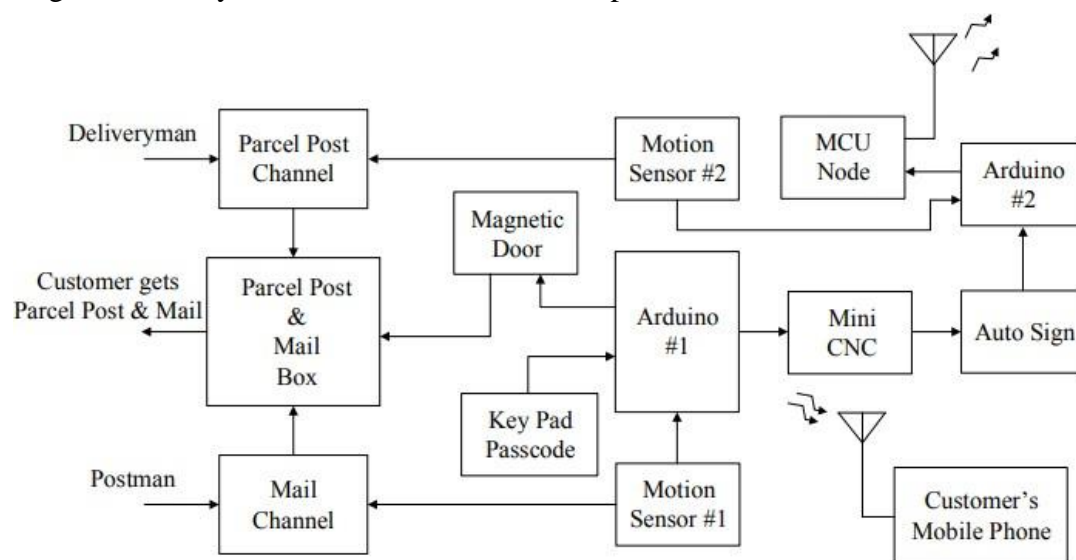


Figure 2.6 Block diagram of Arduino based smart parcel box system

The limitation of this project is that it is using Arduino as the main controller of the project. As discussed earlier about why Raspberry Pi is better than Arduino for the development of smart parcel box. In this case, the authors use two Arduino controllers to develop a smart parcel box, while it only requires one Raspberry Pi to develop a smart parcel box. Unlike Raspberry Pi, which supports operating system while Arduino does not, Raspberry Pi makes the configuration process easier than Arduino. Besides, in the authors' work, they use two Arduino controllers, which makes the configuration of the product more complicated and requires more time. Other than that, the connectivity to

CHAPTER 2

the internet is also one of the important fields to develop the smart parcel box, because the user needs to receive notification from the smart parcel box when the box had received a package. Compared to the Arduino, Raspberry Pi have better performance in the connectivity to the internet since it comes with an Ethernet port. However, if you want to connect to the internet with Arduino, wiring, and coding must be configured well before you connect to the internet.

From the view of the design of the authors' smart parcel box, even though they have the security password for the user to secure everything inside of the parcel box. However, the parcel verification system is absent in this smart box system. Meaning that anyone can put anything into the parcel box without any permission or verification. In another word, if the delivery man put the wrong parcel into the parcel box, they have to wait for the user to unlock the parcel box to take it. Besides, anyone from the outside of the house could put anything into the parcel box, such as rubbish or some dangerous explosive item. Other than that, the user's signature is easy to get as well, if someone just puts a piece of paper into the box and the mini-CNC will automatically sign for anyone. In order to solve this problem, I propose that for any incoming package, no matter it is a parcel or a mail, the smart parcel box needs to verify that the package is belongs to the user, or else the parcel box will just reject to receive that package. Figure 2.7 show the front side of the smart parcel box.



Figure 2.7 Front side of smart parcel box

2.4.2 Smart Freight Box

In another proposed work of developing a smart parcel box [17], the authors had developed an automated package delivery accepting system named “Smart Freight Box”. In their proposed work, they are using Arduino Uno as the main controller of the system, the system including a camera and a DC (direct current) motor that controls the conveyer belt. The user of the smart freight box only can trigger the whole process of accepting the package, the deliveryman will call and verify the package with the user then the door will only be open for the deliveryman. After that, the camera will be turned on once the door is opened, the camera will record everything that happens inside the freight box and stored it in the device. If there is another parcel that needs to be received, the Arduino will turn on the DC motor in order to move the received parcel into the house through back door. Figure 2.8 shows the flow chart of the smart freight box.

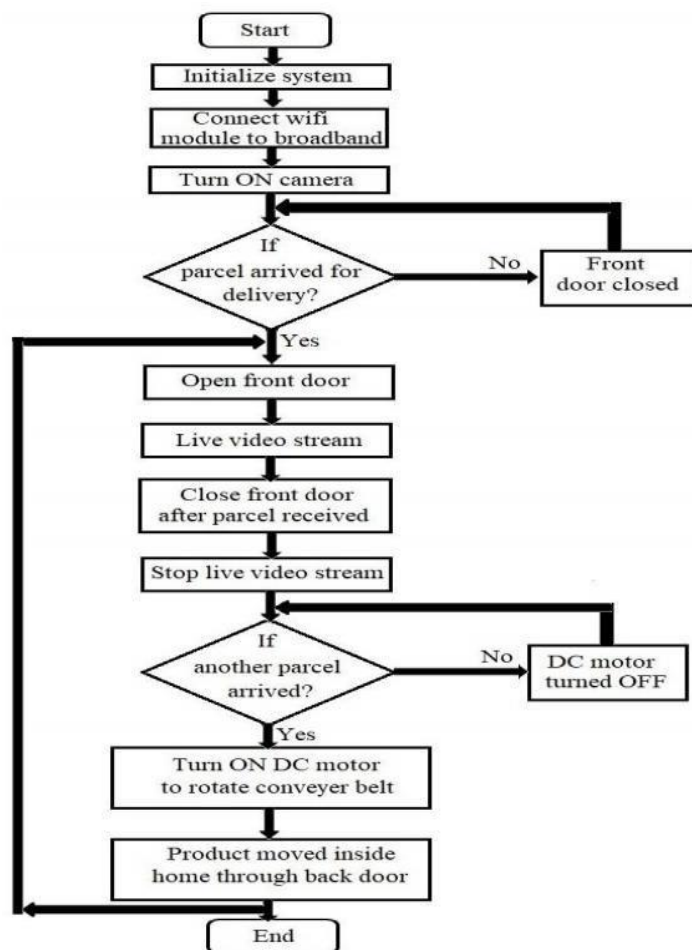


Figure 2.8 Flow chart of the smart freight box

The weakness of the proposed work by the authors is that the security measurement is not enough. Since the back door of the freight box is not locked by the password or any other security system, and the package received can be easily slid into the house through the back door. Meaning that anyone has the access to collect the package if someone broke into the house. In another case, if the user has a pet in the house, the parcel will potentially damage by the pet. The solution to solve this weakness is to implement some security measurement like a keypad to type the password in order to open the back door of freight box, or the freight box needs the user to unlock using any other device or application in the phone. In short, the freight box must be locked and unlocked by the authorised user to secure and protect the parcel inside the box.

Moreover, the user is the only person to trigger the whole accepting parcel process, from opening the front door of the smart freight box until the parcel is moved inside the house and then waiting for another incoming parcel. This whole process can be very time

CHAPTER 2

consuming for the deliveryman. The reason for this circumstance is the deliveryman first needs to call the parcel recipient and verify the parcel with the recipient, and then the recipient will open the smart freight box for the deliveryman. In the case when the recipient's cell phone is not in the service zone, or the deliveryman is unable to contact the recipient, then the smart freight box will never be open, so the deliveryman has no choice but to come back at another time. The same thing happens if the recipient has no internet access to connect with the freight box to open the front door for receiving the parcel when the deliveryman is outside the house. This is not solving the problem of the development of a smart parcel box, which is the deliveryman need to come back again if the parcel recipient is not at home.

By implementing a package verification system on the smart parcel box will be able to solve the problem stated in the previous paragraph. The proposed package verification system in this project will let the smart parcel box verify if the incoming parcel is belonging to the recipient. A database can be created for the recipient to insert the information for the package verification purpose, information like the recipient IC (IdentityCard) number, recipient phone number, or package tracking number provided by the delivery service company. Whenever the recipient shopping online, the recipient will receive a package tracking number from the delivery service company. After that, the recipient inserts and stores the package tracking number in the database for the purpose of package verification. When the delivery man comes with the parcel for the recipient, the deliveryman is asked to scan a barcode or QR code (Quick Response code), which contains the information about the package tracking number, by using the scanner provided on the smart parcel box. Then the smart parcel box will check the information from the barcode or QR code and check it with the database that stores all the package tracking numbers. If the information is matched with the database's information, then the front door of the smart parcel box will open for the deliveryman to receive the package, or else the parcel box will reject the package by keeping the front door closed.

2.4.3 Comparison between Two Existing Systems

Table 2.3 Comparison between Two Existing Systems

Comparison	Arduino Based Smart Box	Smart Freight Box
Embedded Platform Used	Arduino	Arduino Uno
Parcel Verification Process	No	No
Notification Module	Yes	No
Control Remotely	No	Yes
Advanced Security Measurement	No	No

Table 2.3 shows the comparison between two existing systems. Both systems are using Arduino based controller as the embedded platform. Two systems also lack in parcel verification process and advanced security system. Besides, Arduino based smart box having notification module while without control remotely system. For the smart freight box, it can be control remotely but without a notification module.

CHAPTER 3

SYSTEM METHODOLOGY /APPROACH

3.1 Methodology

To make sure the proposed project is meeting all the user requirements and expectations, System Development Life Cycle (SDLC) is the process to determine the user expectation of the proposed project.

3.1.1 System Development Model 1: Waterfall Model

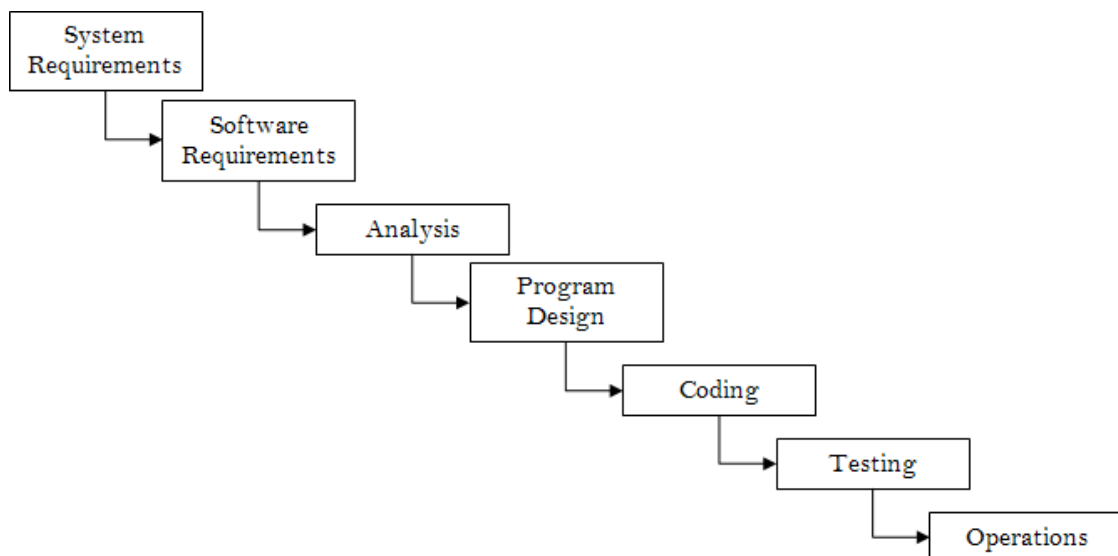


Figure 3.1 Waterfall Model

The earliest SDLC approach used for software development is the waterfall model. The waterfall model demonstrates the software development process as a sequential flow of events. This explains that any step of the development process can start only after the previous stage has finished. The phases in the waterfall model do not overlap with each other [18].

Moreover, waterfall development has the advantages of providing for segmentation and is easy to monitor. A schedule can be made with deadlines for each step of development, and a product can be brought through the different phases of the development process

one after another. In short, the waterfall model is simple and straightforward to comprehend and use, targets that are well understood, and tasks are simple to organise.

However, the downside of the waterfall model is that it does not allow for any reviews or making any adjustments. It is very hard to go back and fix anything that was not well-documented or handled in the design stage once the system has entered the testing stage. Besides, there is a lot of risk and uncertainty in the waterfall model.

3.1.2 System Development Model 2: Spiral Model

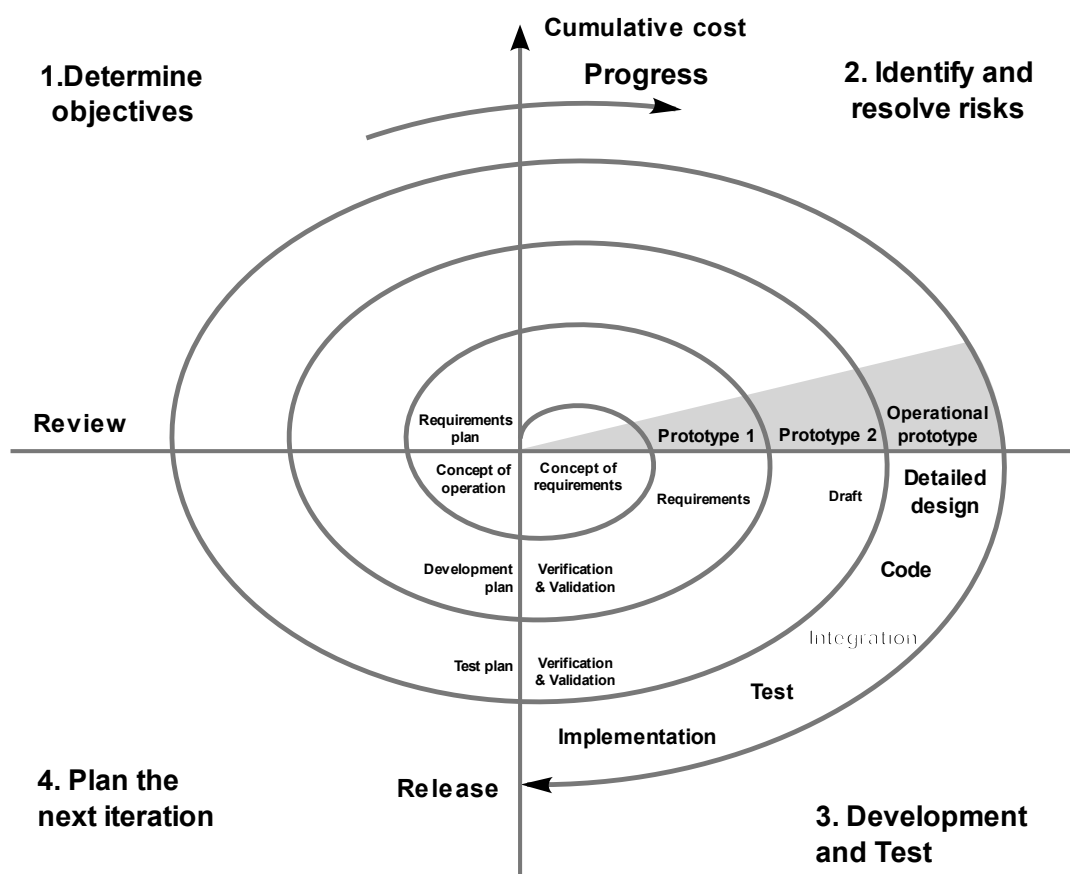


Figure 3.2 Spiral Model

The spiral methodology is an iterative approach. It is the model that combines the prototyping and linear development processes, which is the waterfall model. The spiral model is more focused on risk analysis. Therefore, this approach is best applied to large and complex projects with a high level of risk. Every iteration begins with planning and finishes with the client's feedback on the result [19].

The advantage of applying the spiral model is that it allows the changes of requirements, and because of good risk management, it is considered as cost-effective. So, it is very suitable to apply for a large and complicated project.

Even so, the risk analysis phase is very critical to the project's success. Moreover, it is necessary to hire more skilled professionals, particularly for risk analysis. Thus, the spiral model is not recommended for small projects.

3.1.3 System Development Model 3: Prototyping Model

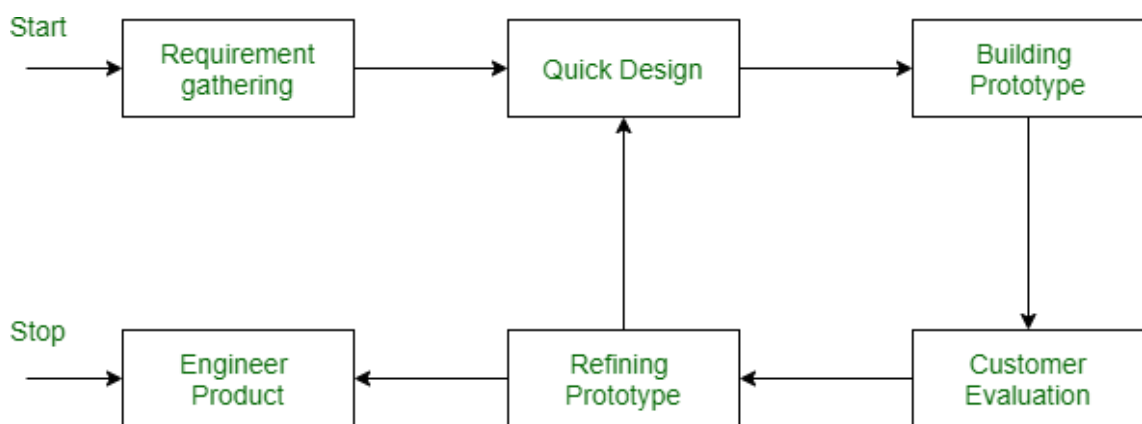


Figure 3.3 Prototype Model

The prototyping model is the type of methodology that focuses on the development of a prototype of a system. The prototype of the system will not be having the full functionality of the system, instead, it will be having some key functions of the system for the client to have a basic understanding of how the system is going to work. After the client had interacted and tested with the system prototype, they will be giving some feedback and opinions to the system developer. After a few tests on the system prototype, a complete function system will be developed that met with most of the user requirements [20]-[21].

The prototyping model is the approach that helps the development team to have a better understanding of the clients' requirements. Because once the clients get the latest prototype, they have tested on the functionality of the system and are able to give feedback and suggest some modifications to the system. Other than that, the prototyping model will help the development team to present some concepts of the system to

CHAPTER 3

investors to get the funding.

However, the prototyping model typically requires to be done at the cost of a development team, thus it is not suitable for a project with a huge budget. In the case of changes and modifications made too often, the rhythm of the development somehow can be disturbed.

3.1.4 Selected Model

After comparing three SDLC models, the model adopted in this project is the prototyping model. The prototyping methodology is suitable for this project because is a small-scale project and this methodology will help to reduce time and costs in the process of development. Besides, any changes or modifications to the prototype can be made easily and shortly because the users can provide their immediate feedback or request on the changes to the system.

First of all, the information regarding on the parcel delivery process and the design of the parcel box is gathered. After that, a simple prototype is designed and ready to build in stage three. The process continues with the user evaluation and giving feedback on the current design of the prototype. In the fifth stage, the prototype will be refining based on the user's feedback and make modify the prototype (if any), which is back to stage two. If the prototype is satisfied the user's requirement, the project will move to the last stage, which is building the product.

3.2 System Requirement

3.2.1 Hardware

The hardware used in this project is Raspberry Pi Model A+, Raspberry Pi camera module, jumper wires, Wi-Fi dongle, and an SG90 Micro Servo. The Raspberry Pi is used to control the whole system, including retrieving information from the database, processing the data read by the camera, and controlling the micro servo to unlock or lock the smart parcel box. The Raspberry Pi camera used in this project is the earlier version of the Raspberry Pi Camera Module, 5 Megapixel, Rev 1.3 camera module. The jumper wires are used to connect the SG90 Micro Servo to the GPIO pins on Raspberry Pi. The

CHAPTER 3

Wi-Fi dongle allows Raspberry Pi to connect to a network.

Table 3.1 Specifications of Raspberry Pi

Description	Specifications
Model	Raspberry Pi 1 Model A+
Processor	Broadcom BCM2835 Single core (ARM11) SoC
Memory	512 MB
USB	1 x USB 2.0 port
Connection	GPIO Header 40-pin
HDMI	Rev 1.3 and 1.4
Storage	microSD card slot for loading operating system and data storage
Power	USB connector for 5V/2A dc



Figure 3.4 Raspberry Pi 1 Model A+

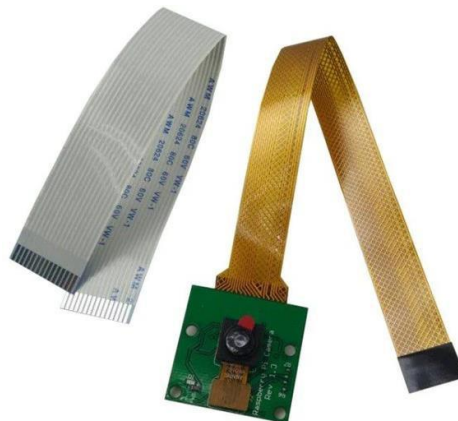


Figure 3.5 Raspberry Pi Camera Module (5 Megapixel)



Figure 3.6 SG90 Micro Servo



Figure 3.7 Jumper Wires



Figure 3.8 Wi-Fi Dongle

3.2.2 Software

In addition, the software used in this project is Raspberry Pi Operation System (previously known as Raspbian OS) is a Debian based free operating system for the Raspberry Pi hardware. The Windows Apache2 MySQL PHP (WAMP) Server will be using in this project. WAMP Server is a Windows web development environment that will allow the user to create a web application with Apache2, MySQL, and PHP. The programming language used in this project is Python 3. Python 3 is a high-level programming language and easy to write as its easy syntax for quick coding, and python 3 comes with many useful libraries that make the complex algorithm easy. OpenCV, which is a programming library will be used to process the real-time image and video. Besides, another programming library that will be used to analyse the QR codes is pyzbar. While Telegram will be using to send a notification to the receiver.



Figure 3.9 Raspberry Pi OS

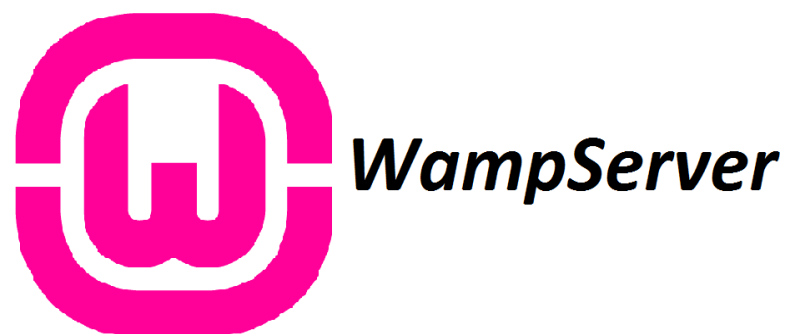


Figure 3.10 WAMP Server



Figure 3.11 Python Programming Language



Figure 3.12 OpenCV



Figure 3.13 Telegram

3.3 Functional Requirement

3.3.1 QR Code Detection

The system is required to have the ability to detect the QR code that shows in front of the camera and able to retrieve the information carried by the QR code.

3.3.2 QR Code Verification

After retrieving the information of the QR code, the system needs to verify the information with the information stored inside the database. The parcel box only will unlock if the result is matched, otherwise the parcel box will remain locked.

3.3.3 Photo Capture

The system is required to capture the photo whenever the QR code is detected. No matter the result is valid or invalid. The photo taken will be uploaded to the database according to the QR code verification result.

3.3.4 Notification to Receiver

Right after the QR code is detected by the system, a notification will be sent to the receiver via telegram bot. The system will notify the receiver whether the QR code verification is valid or invalid.

3.3.5 Website

There will be a website for users to keep track of all the parcels. Some features like tracking the incoming parcel, confirming the received parcel, checking on the abnormal activity, unlock or lock the parcel box manually, viewing history records, and account password reset and account deletion. Besides, the user needs to register an account on the website to login to the website.

3.3.5.1 User Registration

Users will need a unique username, password, first name, last name, and email to register an account to login to the website. Any invalid input like a duplicate username with other users will cause the registration to fail.

3.3.5.2 User Login

Users need to input their username and password to login to the website. Any invalid input will cause the login to fail.

3.3.5.3 Insert New Tracking Number

Users are allowed to insert new tracking numbers into the database. It is the responsibility of users to insert the correct tracking number into the database.

3.3.5.4 Incoming Parcel Tracking

On the main page of the website, there will be a list of incoming parcels with a button next to it. When the users click on the button, the users will be redirected to a parcel tracking website to see the status of their parcels.

3.3.5.5 Received Parcel Confirmation

On the main page of the website, there will be a list of received parcels listed in one

CHAPTER 3

section, each record will show the parcel tracking number, parcel received time, courier name, photo of QR code captured, and a button below it. The function of the button is to let the user confirm and verify the received parcels.

3.3.5.6 Checking on Abnormal Activity

When the system is recording an abnormal activity, the recorded event will display in one section of the website main page. The information like the QR code used and recorded event time will be shown to the user. There will be a button for the user to click on when they have checked on the abnormal activity.

3.3.5.7 Lock or Unlock Parcel Box Manually

The website will be having a page for users to lock or unlock the parcel box manually in case of some unexpected situations. For instance, when the QR code is incomplete or broken, which the system is unable to verify the parcel. Therefore, the deliveryman needs to contact with the receiver to unlock it manually.

3.3.5.8 View History Record

When the users confirmed the received parcels and checked on the abnormal activity, the record will move to the history record and disappear from the main page. Therefore, the website will have the feature of let users view all the past received parcel and abnormal activity.

3.3.5.9 Account Password Reset

Some of the users have the habit of changing their account password frequently. So, the website will also let the user to reset their password by inputting their username, account old password, and new password.

3.3.5.10 Account Deletion

The account deletion is let user to delete their account from the website. For the situation like when a user is moving to another country, and they feel to have the need to delete their account.

3.4 Project Milestone

The following tables demonstrate the project timeline for the development of the prototype of the smart parcel box.

Table 3.2 Project Timeline for FYPI

Project Task	Project Week												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Data Collection	■												
Identifying Problem Statement		■											
Identifying Objectives and Project Scopes		■											
Literature Review Research			■	■									
Determine System Requirements					■								
Determine System Functionalities						■	■						
Outline System Architecture								■					
Outline System Flow Chart								■					
Gathering Hardware									■				
Hardware Setup										■			
Software Setup										■			
System Testing											■	■	
Presentation													■
Documentation	■	■	■	■	■	■	■	■	■	■	■	■	

Table 3.3 Project Timeline for FYP2

Project Task	Project Week												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Knowledge Research	■												
Configuring Telegram Bot		■											
Determine Functionality of Website			■										
Outline Website System Flow			■	■									
Wireframe Website Layout					■								
Website Development						■	■						
System Prototyping							■	■	■				
System Testing								■	■	■			
Finalize System Prototype											■		
Finalize Report												■	
Presentation													■
Documentation	■	■	■	■	■	■	■	■	■	■	■	■	

3.5 Estimated Cost

Table 3.4 Estimated Cost for the Development of System

Items	Price	Quantity	Total Price
Raspberry Pi A+	RM 90.94	1	RM 90.94
5MP Camera Board for Raspberry Pi	RM 40.00	1	RM 40.00
5V 2.5A Adapter micro-B USB cable	RM 20.00	1	RM 20.00
32GB Micro SD Card with NOOBS for RPI	RM 55.00	1	RM 55.00
Dupont Jumper Wires Bundle	RM 0.99	1	RM 0.99
WIFI5 AC600 Dongle	RM 22.90	1	RM 22.90
Raspberry Pi OS	FOC	-	FOC
OpenCV	FOC	-	FOC
WAMP Server	FOC	-	FOC
Telegram	FOC	-	FOC
Grand Total			RM 229.83

The total estimated cost of the development of prototype of smart parcel box is about RM 229.83. All cost is spent on hardware, all the software used is FOC (Free of charge).

3.6 Concluding Remark

In short, this chapter discussed about three different system development models and choose one suitable model to apply in this project, which is the prototyping model. System requirements and functional requirements are also explained in detail. The estimated cost of the project is also calculated and shown in this chapter. Project milestones of FYP1 and FYP2 are displayed in the table form.

CHAPTER 4

System Design

4.1 System Architecture

4.1.1 Overall System Design

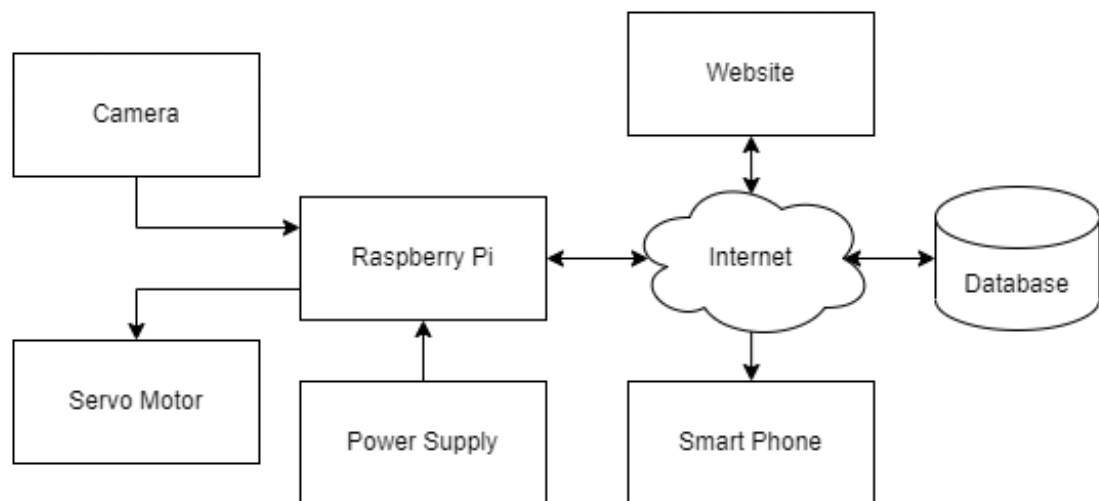


Figure 4.1 System Architecture Diagram

Figure 4.1 illustrates the system architecture design of the prototype of smart parcel box system. The Raspberry Pi is connecting to the power adapter with the output of 5V/2.5A. The Raspberry Pi is also connected to the camera module, SG90 micro servo. The camera module is using to scan and read the information in the QR Code, while the SG90 micro servo is controlled by the Raspberry Pi to unlock or lock the parcel box. Moreover, the Raspberry Pi also connects to the Internet as well as the database and smartphone. The database is used to store the parcel information that will be updated by the user using the website and the system once the parcel is received. The Raspberry Pi will retrieve the parcel information from the database through the Internet and compares it to the data read by the camera. After that, the notification via telegram will be sent to the receiver's smartphone once the parcel is received.

4.1.2 Website Architecture Design

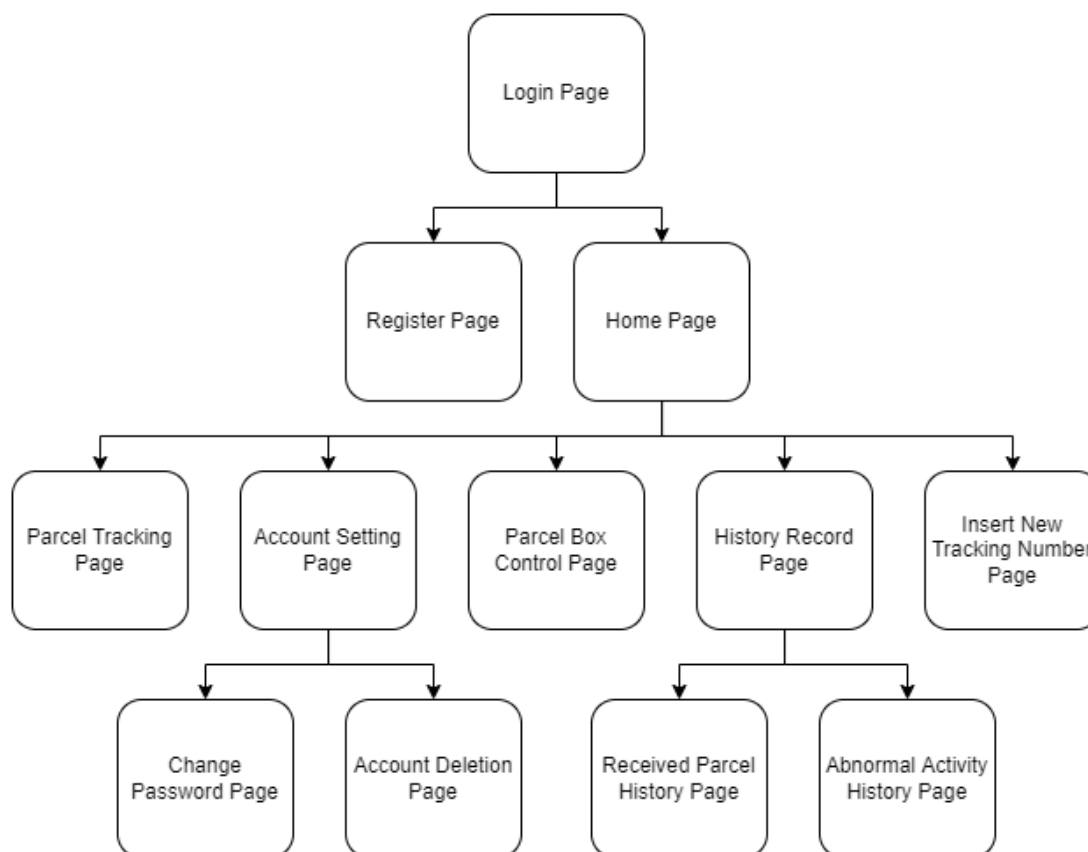


Figure 4.2 Website Architecture Diagram

Figure 4.2 demonstrates the website architecture diagram for this project. The user will first be landing at the login page. If the user wants to register an account, he or she will be taken to the user registration page. Otherwise, the user will be direct to the home page with the correct input of username and password. On the home page, the user will find a list of incoming parcels, received parcels and abnormal activity. After that, the user could direct to many pages of the website, for instance, parcel tracking page, account setting page, parcel box control page, history record page, and insert new tracking number page. At the account setting page, user can choose to go to change password page or account deletion page. As well in history record page, user can choose to go to received parcel history page or abnormal activity history page.

4.2 Functional Modules in the System

4.2.1 QR Code Detection

In this module, the 5MP Camera Board for Raspberry Pi will be using to detect the QR

CHAPTER 4

code in the frame. With the help of the OpenCV library, which allows the system to process real-time image and video capture from the 5MP Camera Board for Raspberry Pi.

4.2.2 QR Code Verification

In this module, after the 5MP Camera Board for Raspberry Pi send the QR code data to the system. After the system receives the QR code data, it will then analyse the QR code to retrieve the information carried by the QR code with the help of pyzbar library. Then, the system will retrieve the information (parcel tracking number) from MySQL database and perform verification based on the information (parcel tracking number) received. If the data is verified, the parcel box door will unlock by controlling the SG90 Micro Servo and a notification will be sent to receiver via telegram. Otherwise, this activity will consider as abnormal activity, the door will remain locked and a notification will be sent to receiver as well.

4.2.3 Photo Capture

After the 5MP Camera Board for Raspberry Pi had detected a QR code, the system will capture a photo. Whether the information from the QR code is valid or invalid, a photo will be captured and uploaded to the MySQL database as proof for receiver to review.

4.2.4 Notification to Receiver

After an event has happened, regardless it is the parcel received or abnormal activity, the system will notify the receiver about the event via telegram. A telegram bot account is created for the purpose of notification.

4.2.5 Website

In this module, a website is designed for the purpose of having a platform to let receiver manages and monitors their parcel box with a better GUI (Graphical User Interface) layout. The website will be sharing the database with the Raspberry Pi to have a better information synchronisation.

4.2.5.1 User Registration

In this module, users require a unique username, password, first name, last name, and email to register an account. Any invalid input like duplicate username with other users will cause the registration to fail. The system will check the user input with MySQL database to avoid duplicate username and email. If the user input is checked with duplicate data in the database, the registration process is unsuccessful, the user needs to start over the process. The password will be encrypted before being stored into the database.

4.2.5.2 User Login

At the login page, user is asked to input username and password. The system will check the user input to compare it with the data stored in the database. If the user input is unmatched with data in the database, user is unable to login to the website. The purpose is to verify the user account.

4.2.5.3 Insert New Tracking Number

User is allowed to manually insert new tracking number to the database. User needs to key in the tracking number and choose a courier from the pre-defined list to insert into database. System will check if the tracking number is already existed in the database before continuing the process. If there is the same tracking number in the database, the process will be aborted.

4.2.5.4 Incoming Parcel Tracking

The system will retrieve the information from the database, filter and select the parcels that are not yet received. Then the information of the parcels will be displayed in one section of the website. The system will provide a button to redirect the user to a parcel tracking website.

4.2.5.5 Received Parcel Confirmation

The system will retrieve the information from the database, filter and select the parcels that are received but not yet confirmed by the receiver. Then the information of the parcels will be displayed in one section of the website. The system will provide a button to let user confirm and verify the received parcel and mark it as verified.

4.2.5.6 Checking on Abnormal Activity

Every abnormal activity will be recorded in a database, the website will retrieve the information from the database and filter out with the record that is unchecked by the user. A button will provide by the system to let the user to acknowledge and check on the abnormal activity, after that the system will mark it as checked abnormal activity.

4.2.5.7 Lock or Unlock Parcel Box Manually

In this module, the website will take the user to a webpage that hosted on the Raspberry Pi and control the SG90 Micro Servo to lock or unlock the parcel box.

4.2.5.8 View History Record

When the user verified the received parcel, the system will mark it as verified received parcel; while the user acknowledges and check an abnormal activity, the system will mark it as checked abnormal activity. The website will retrieve the information and filter the information that marked as verified received parcel or checked abnormal activity and showing on the history record page.

4.2.5.9 Account Password Reset

To reset the account password, user is required to input the username, old password, and new password to complete the process. The website will first check on the username and old password, if the username and old password are matched, then the system will proceed with the password reset process. Otherwise, the website will abort the password reset process when the unmatched username and old password are detected.

4.2.5.10 Account Deletion

The account deletion process requires user to key in the username and password to continue the process. The system will check the user input with the information in the database, only the valid user input will proceed with the account deletion process. When the process is completed, the user account information will be removed from the database. If the user wants to use the website again, he/she would need to register a new account.

4.3 System Flowchart

4.3.1 Overall System

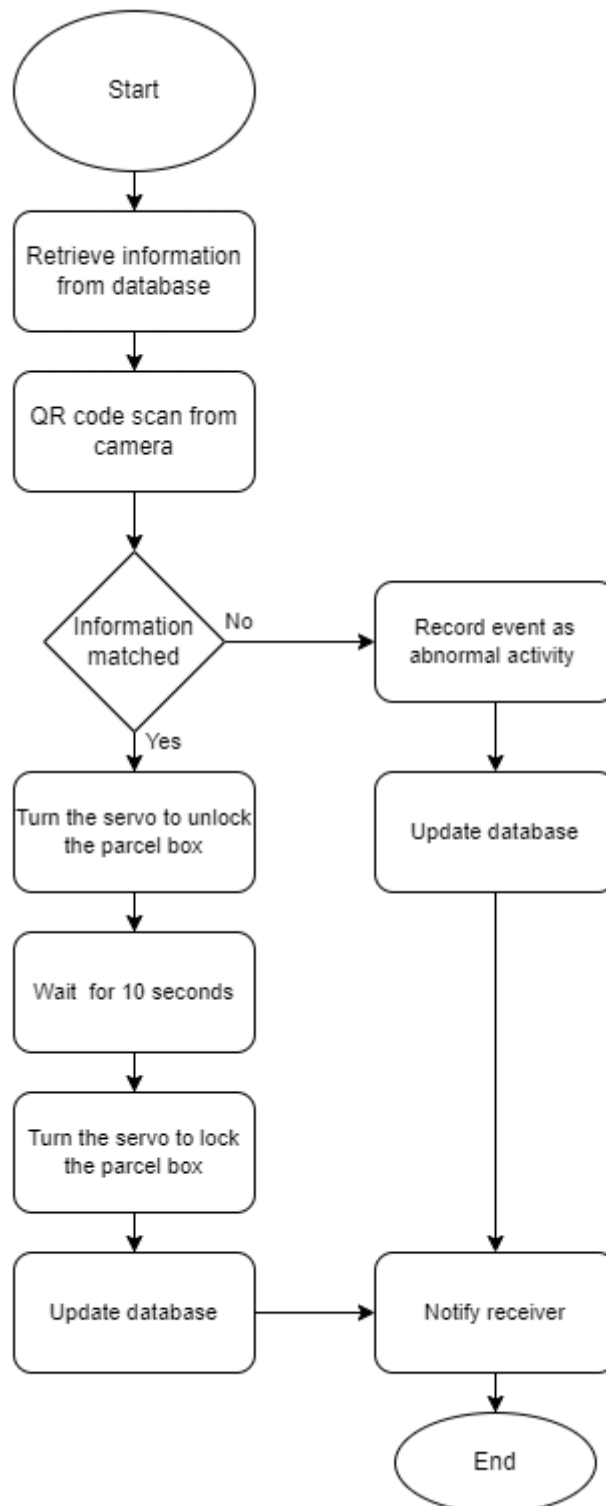


Figure 4.3 System Flowchart

Once the system started, the first thing to do is connect with the database and retrieve the information (parcel tracking number) and store it in the system. After that, the

system starts to wait for the camera to scan QR code and compare the information received with the information retrieved from the database. If the information is matched, the servo motor will be turned to unlock the parcel box. After 10 seconds, the system will turn back the servo motor to lock the parcel box and update the database by marking the parcel as received based on the tracking number. If the information is unmatched, the system will record it as abnormal activity and update the database with the event time and the information detected from QR code. Notification of abnormal activity will be sent to receiver as well.

4.3.2 User Registration

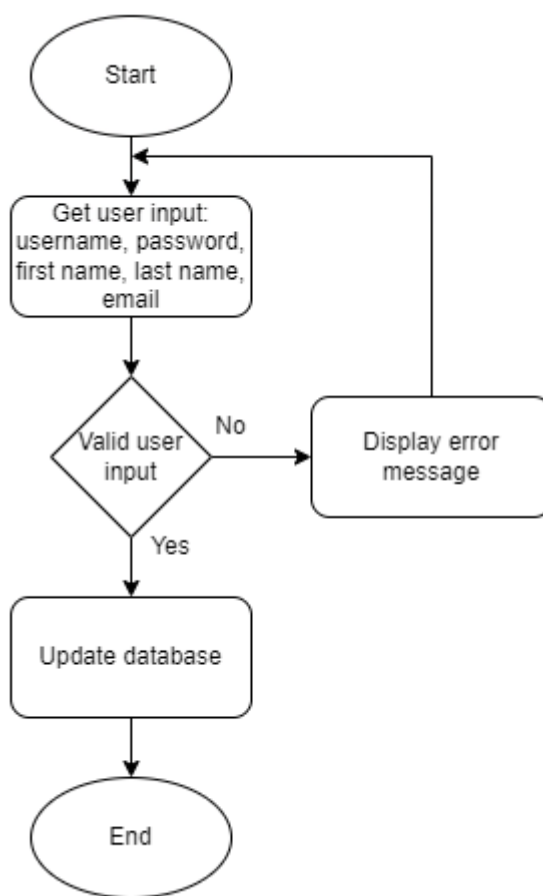


Figure 4.4 User Registration Flowchart

The system starts with getting user input, which are unique username, password, first name, last name, and email. Following by validating user input, the invalid input condition including duplicate username, duplicate email, and any blank value. If the user input is valid, system will update the database by inserting a new record into user account database.

4.3.3 User Login

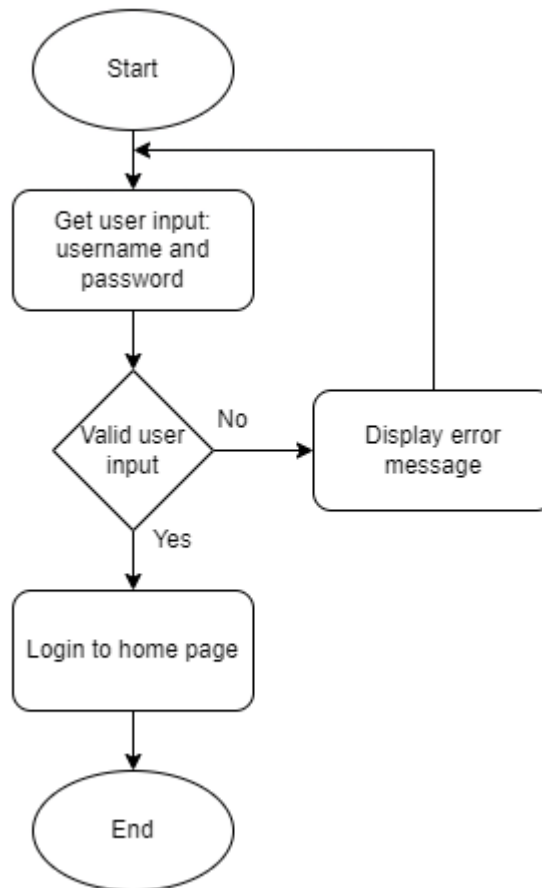


Figure 4.5 User Login Flowchart

For user login, the system will ask user input for username and password, then validate the user input. Invalid input condition including unmatched username or password, and any blank value. User will be logged in to home page if the user input is valid.

4.3.4 Insert New Tracking Number

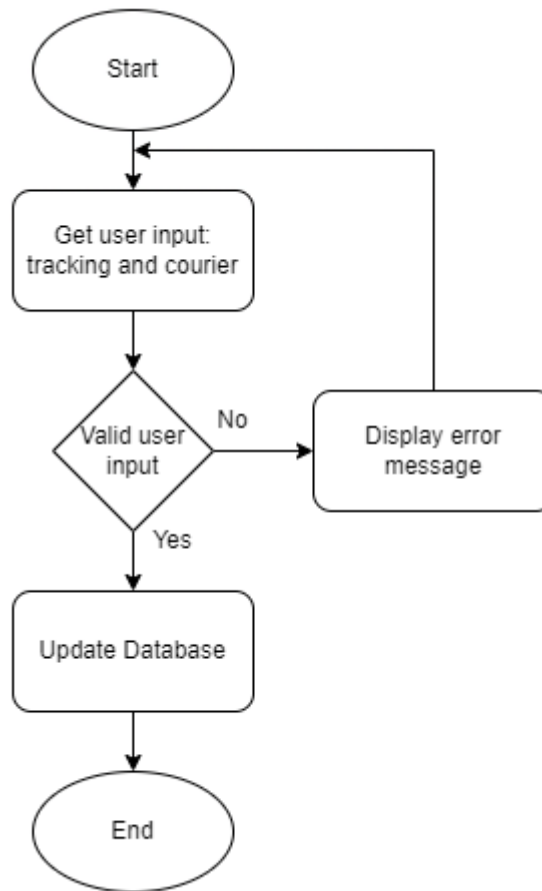


Figure 4.6 Account Password Reset Flowchart

User is required to key in tracking number and select the courier from the pre-defined list. System will validate the user input before continuing the process. Invalid user input including duplicate tracking number with the database and blank value. System will update the database if the user input is valid. The user is responsible to insert a correct new tracking number.

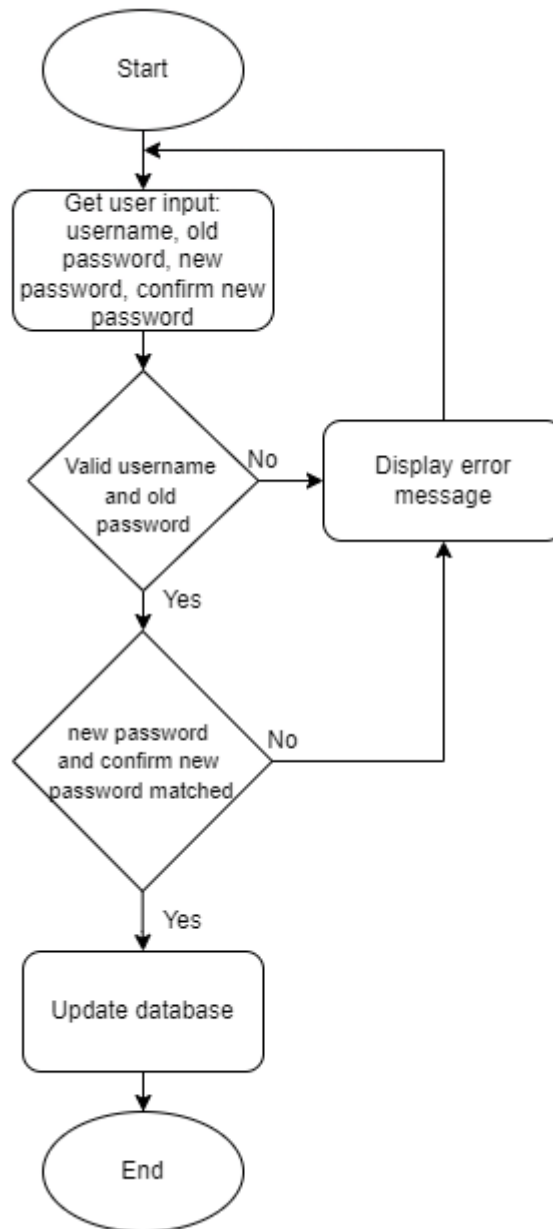
4.3.5 Account Password Reset

Figure 4.7 Account Password Reset Flowchart

To reset account password, user required to input username, old password, new password and confirm new password, then system will validate user input before continuing the process. Invalid input conditions including unmatched username or old password, unmatched new password and confirm new password, and any blank value. If the user input is validated, system will update the database.

4.3.6 Account Deletion

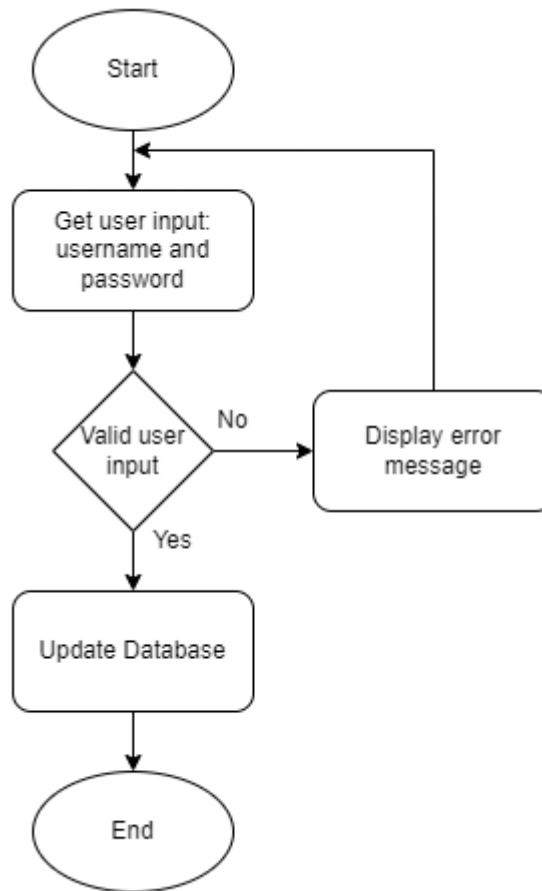


Figure 4.8 Account Deletion Flowchart

User is required to input username and password to delete account. System will validate user input before continuing the process. Invalid user input condition including unmatched username or password, and any blank value. The system will update database by removing the user information from the user account database if the user input is validated.

4.4 Database Design

There is one database with three tables needed for this project, which are users, tracking, and abnormal. The user database is to store user account information for the purpose of website login. While the tracking database is to store the parcel information. The last database is abnormal database, which is to store the abnormal activity detected by the system.

4.4.1 Entity Relationship Diagram (ERD)

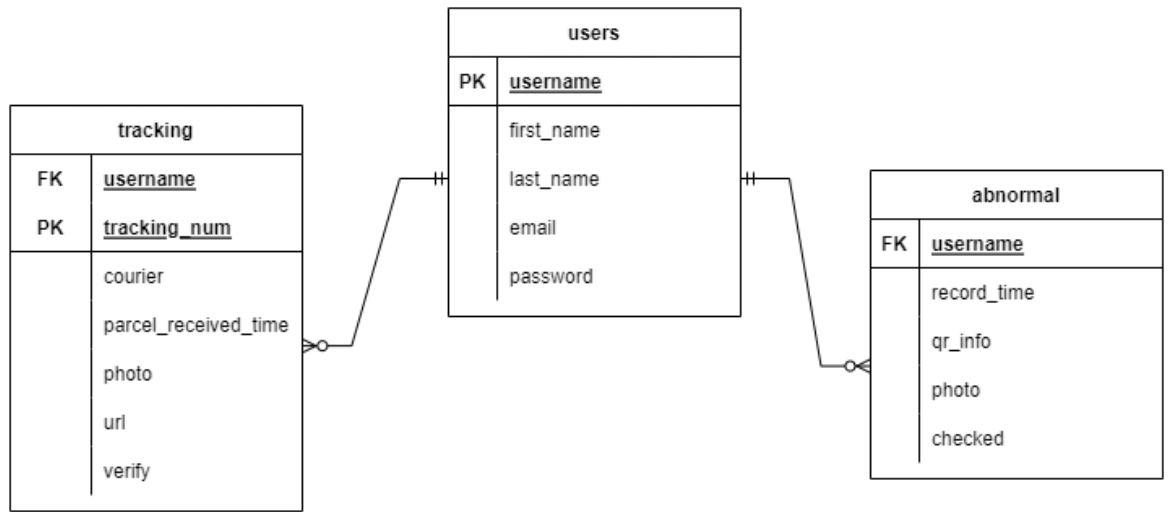


Figure 4.9 ERD

Figure 4.9 shows the ERD of the database. The ERD explained that every user will have zero to many tracking records and abnormal records. While every record of racking and abnormal is belongs to only one user. Username is unique as it is a primary key in users database. In tracking database, username is a foreign key from users database and has tracking_num as the primary key. While in abnormal database, the username is foreign key from users database.

4.4.2 Data Dictionary - Users Database

Table 4.1 illustrate the data dictionary of the users database. All fields are using the database is using varchar with the size of 255. This database is using to store information like username, first name, last name, email, and password. Username must be unique for every record, which means that there is no duplicate record for username and email. The password stored is encrypted using MD5 encryption.

Table 4.1 Data Dictionary – Users Database

Field Name	Data type	Field Size	Description	Example
username	varchar	255	To store user's username	jiaheng56
first_name	varchar	255	To store user's first name	JiaHeng
last_name	varchar	255	To store user's last name	Lee

email	varchar	255	To store user's email	jiaheng56@hotmail.com
password	varchar	255	To store user's password	f175dd5bf9491b1b1c18a28c3d43b89e

4.4.3 Data Dictionary - Tracking Database

Table 4.2 shows the data dictionary of the tracking database that stored the information about the parcel. Username is the same username user register with, and the purpose is to identify which parcel is belongs to which user. The tracking_num is storing tracking number of the parcel that the system using to verify the QR code scanned by the camera. The courier is the name of the service provider for that parcel. Parcel_received_time is to record the timestamp of the parcel received time, null value means that the parcel is not yet received. Photo is to store the photo captured, by the camera when the parcel is received, null value means the photo is not yet taken. Photo stored in a BLOB (Binary Large Object) data form. Next, the url is storing the URL (Uniform Resource Locator) to the parcel tracking website, which is <https://tracking.my>. Lastly, the verify is represent the confirmation on the received parcel, “0” represents the parcel is unconfirmed and unverified by the receiver; while “1” means the received parcel was confirmed and verified by the receiver.

Table 4.2 Data Dictionary - Tracking Database

Field Name	Data type	Field Size	Description	Example
username	varchar	255	To store user's username	jiaheng56
tracking_num	varchar	255	To store parcel's tracking number	SPXMY01110110118C
courier	varchar	255	To store courier's name	shopee
parcel_received_time	timestamp	-	To store parcel received time	2022-03-07 21:40:45
photo	longblob	-	To store photo	[BLOB – 98.9 KiB]

url	varchar	255	To store parcel tracking's url	https://www.tracking.my/shopee/SPXMY01110110118C
verify	bit	1	To store received parcel confirmation status	0

4.4.4 Data Dictionary - Abnormal Database

Table 4.3 demonstrates the data dictionary of the abnormal database, which is used to store the abnormal activity information. Username is used to identify the abnormal activity is recorded under which user's smart parcel box. Record_time is the timestamp that the abnormal activity recorded. While qr_info is the information of the QR code detected. Photo is to store the photo captured when the abnormal activity happened and store in BLOB data form. The checked field is to identify whether the receiver has checked on the abnormal activity. A null value means that the receiver has not yet acknowledged and checked on the abnormal activity; while "1" means the receiver had acknowledged and checked on the abnormal activity.

Table 4.3 Data Dictionary - Abnormal Database

Field Name	Data type	Field Size	Description	Example
username	varchar	255	To store user's username	jiaheng56
record_time	timestamp	-	To store user's first name	2022-03-11 15:32:47
qr_info	varchar	255	To store user's last name	12336699acc
photo	longblob	-	To store user's email	[BLOB – 85.9 KiB]
checked	tinyint	4	To store user's password	1

4.5 Algorithm Design

4.5.1 Hashing Algorithm

In the users database, the user input a plaintext of password, but the hashing algorithm is using to hash the password before storing into the database. Figure 4.10 shows the result of the hashing algorithm.

username	first_name	last_name	email	password
jiaheng56	JiaHeng	Lee	jiaheng_56@hotmail.com	f175dd5bf9491b1b1c18a28c3d43b89e

Figure 4.10 Result of Hashing Algorithm

4.6 Graphical User Interface Design (Website)

4.6.1 Login Page

Figure 4.11 GUI of User Login Page

Figure 4.11 shows the GUI design of login page. User is required to input username and password and click on the login button. There is a “Sign up” link that will take user to account registration page.

4.6.2 Account Registration Page

The image shows a registration form with a teal header containing the word "Register". Below the header are six input fields, each with a label above it: "Username", "First Name", "Last Name", "Email", "Password", and "Confirm password". At the bottom of the form is a teal "Register" button and a link that says "Already a member? [Sign in](#)".

Figure 4.12 GUI of Account Registration Page

Figure 4.12 illustrates the GUI of account registration page. The page will need 6 user inputs from user to register an account. The “Register” button is for user to submit the user input. The “Sign in” link is taking user to login page.

4.6.3 Home page

My Smart Parcel Box

Home New Parcel Box My Account

You are now logged in

Incoming Parcel

Tracking Number	Courier	
SPXMY01110110118C	shopee	<input type="button" value="Track"/>
H12345H	shopee	<input type="button" value="Track"/>
123ABCderf	skynet	<input type="button" value="Track"/>
620001021227493	jt	<input type="button" value="Track"/>

Received Parcel

Parcel Received On:
2022-03-09 15:29:27

Tracking Number:
SPXMY01846813120C

Courier: shopee

Abnormal Activity

Tracking Number:
SPXMY01846813120C

Recorded on : 2022-03-11 15:32:47

Figure 4.13 GUI of Home Page

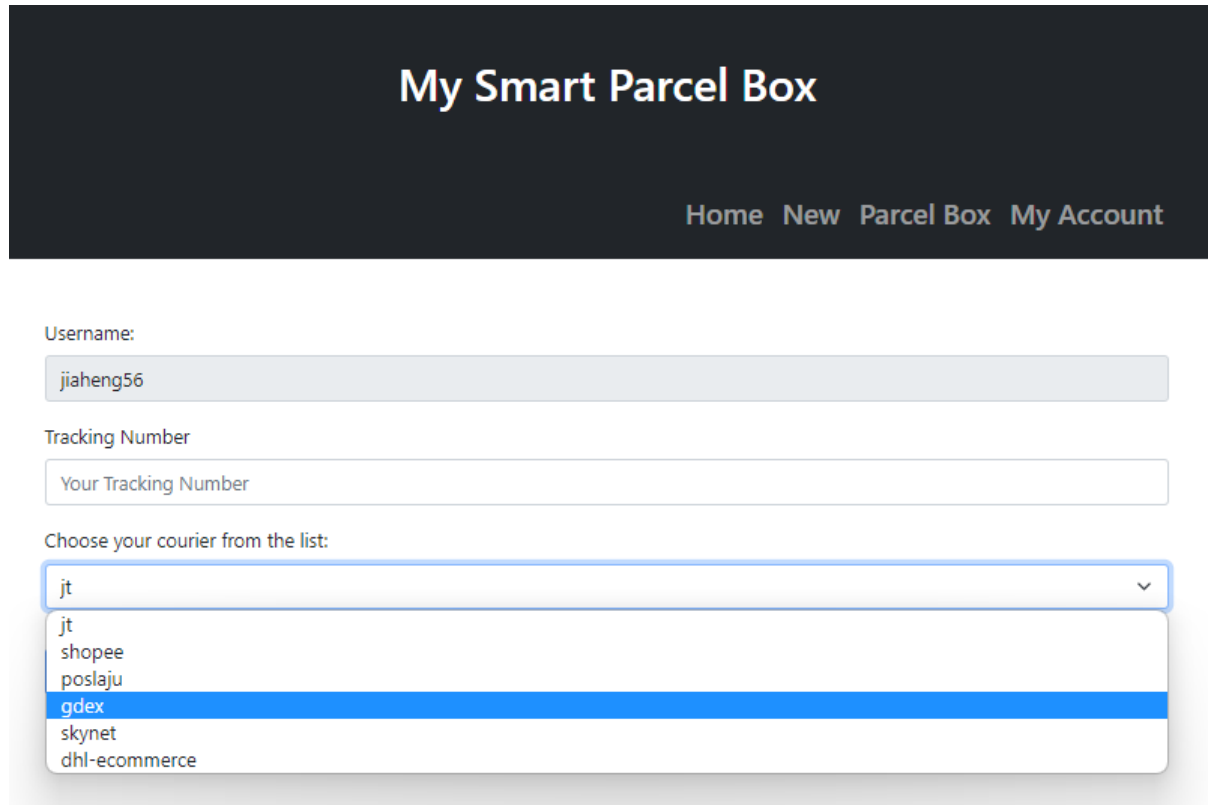
The GUI of home page is presented in figure 4.13. The navigation bar will take user to the respective page. There are three main sections on home page, which are the incoming parcel section, the received parcel section, and the abnormal activity section.

Incoming Parcel Section - a list of incoming parcels is shown to the user and there is a “Track” button that will take user to parcel tracking website.

Received Parcel Section – a list of received parcels and its information is showing to user and a “Verify” button for user to acknowledge and confirm the received parcel.

Abnormal Activity – a list of abnormal activity and related information is showing to user and a “Check” button for user to mark as checked on the abnormal activity.

4.6.4 Insert New Tracking Number Page



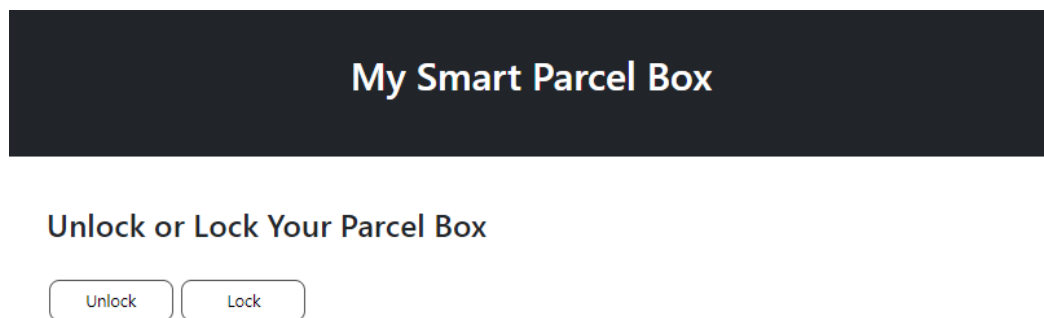
The screenshot shows the 'My Smart Parcel Box' interface. At the top, there is a dark header with the title 'My Smart Parcel Box' in white. Below the header, there are navigation links: 'Home', 'New Parcel Box', and 'My Account'. The main content area contains a form with the following elements:

- Username:** A text input field containing 'jjaheng56'.
- Tracking Number:** A text input field with the placeholder text 'Your Tracking Number'.
- Choose your courier from the list:** A dropdown menu with a blue border. The current selection is 'jt'. The dropdown list is open, showing the following options: 'jt', 'shopee', 'poslaju', 'gdex', 'skynet', and 'dhl-ecommerce'. The 'gdex' option is highlighted with a blue background.

Figure 4.14 GUI of Insert New Tracking Number Page

Figure 4.14 demonstrates the GUI of insert new tracking number page. The username is pre-filled and unable to modify, the purpose is to avoid typing error from the user. It requires user to key in the tracking number and chooses a courier from the pre-defined list. A submit button is placed under the courier to let user submit the form.

4.6.5 Parcel Box Control Page



The screenshot shows the 'My Smart Parcel Box' interface. At the top, there is a dark header with the title 'My Smart Parcel Box' in white. Below the header, there is a section titled 'Unlock or Lock Your Parcel Box'. Underneath this title, there are two buttons: 'Unlock' and 'Lock'.

Figure 4.15 GUI of Parcel Box Control Page

Figure 4.15 shows the GUI of Parcel Control Page. The “Unlock” and “Lock” button is for user to control the smart parcel box manually.

4.6.6 History Record Page

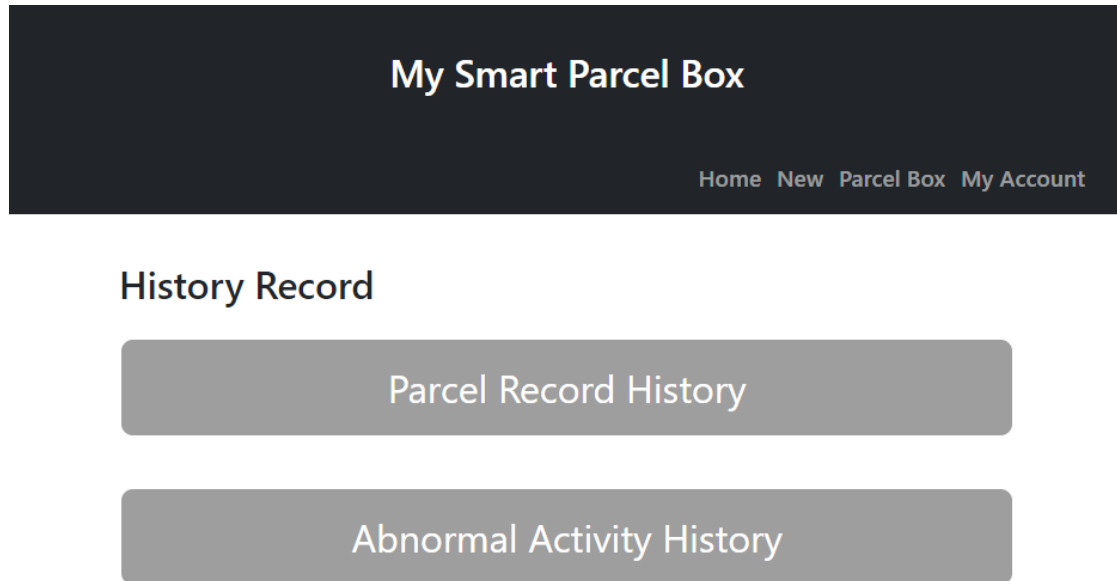


Figure 4.16 GUI of History Record Page

Figure 4.16 shows the GUI of history record page. There are two main sections to let user choose which type of history record he/she would like to see, which are parcel record history and abnormal activity history.

4.6.7 History Record Page – Parcel Record History

Tracking Number	Courier	Parcel Received Time	View Route History	View Photo
5920116398372871	dhl-ecommerce	2022-03-07 21:43:56	View	Open
PL020960647565	poslaju	2022-03-07 21:40:45	View	Open
ygfyuwefwe12	jt	2022-02-22 22:22:22	View	Open
12336699acc	jt	2022-03-09 16:53:44	View	Open

Figure 4.17 GUI of History Record Page – Parcel Record History

Figure 4.17 represents the GUI of the history record page for parcel record history. On

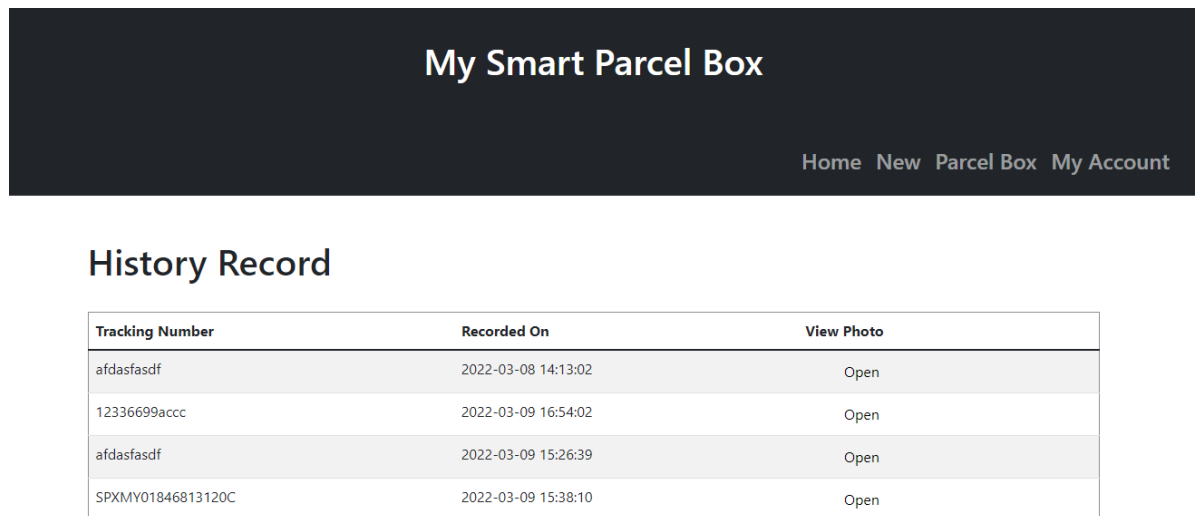
CHAPTER 4

this page, the information of the received parcel will be shown to user, including tracking number courier, parcel received time, route history, and photo. and there are two buttons for each record, which are “View” and “Open”.

View – to let user view the route taken for the parcel, which will redirect user to parcel tracking website.

Open – to let user see the photo taken for the parcel in a new tab.

4.6.8 History Record Page – Abnormal Activity History



The screenshot shows a web application interface for 'My Smart Parcel Box'. The header is dark with the title 'My Smart Parcel Box' in white. Below the header, there are navigation links: 'Home', 'New Parcel Box', and 'My Account'. The main content area is titled 'History Record' and contains a table with the following data:

Tracking Number	Recorded On	View Photo
afdasfasdf	2022-03-08 14:13:02	Open
12336699acc	2022-03-09 16:54:02	Open
afdasfasdf	2022-03-09 15:26:39	Open
SPXMY01846813120C	2022-03-09 15:38:10	Open

Figure 4.18 GUI of History Record Page – Abnormal Activity History

Figure 4.18 shows the GUI of history record page for abnormal activity history. The information of each abnormal activity will be shown to user, including tracking number, recorded time, and photo. The button “Open” is to let user see the photo taken for the abnormal activity in new tab.

4.6.9 Account Setting Page

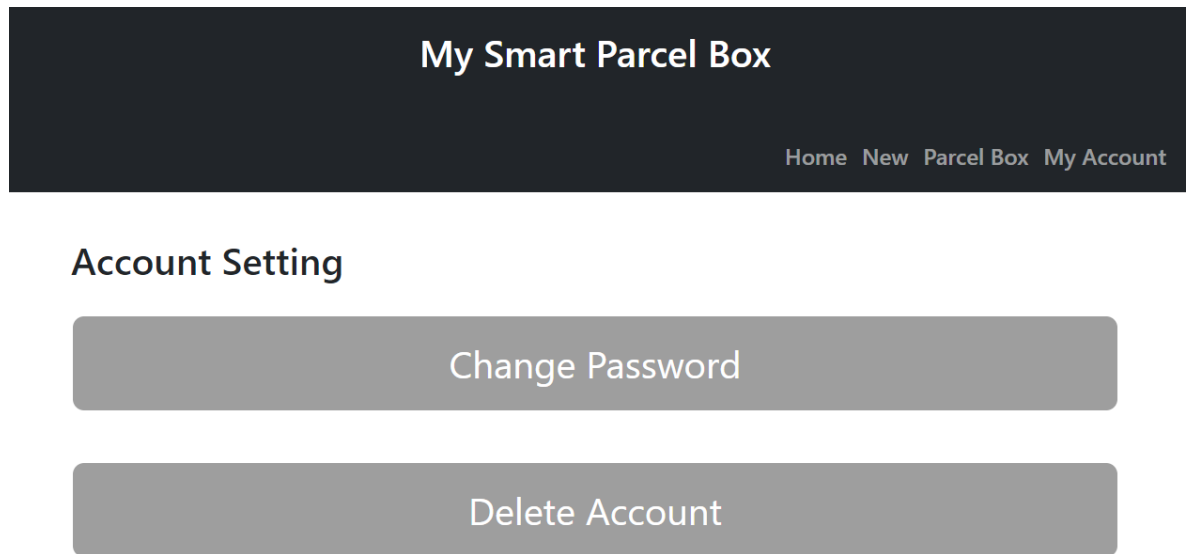


Figure 4.19 GUI of Account Setting Page

Figure 4.19 shows the GUI of account setting page. There are two main sections to let user choose which type of account setting he/she would want, which are change password and delete account.

4.6.10 Account Setting Page – Change Password

Figure 4.20 GUI of Account Setting – Change Password

Figure 4.20 shows the GUI of account setting page for change password. On this page, four user input is required, including username, old password, new password, and confirm new password. The submit button is for user to submit the form and change the password.

4.6.11 Account Setting Page – Delete Account

My Smart Parcel Box

Home New Parcel Box My Account

Account Delete

Username:

Password:

Are you sure you want to delete your account?

Figure 4.21 GUI of Account Setting – Delete Account

Figure 4.21 shows the GUI of account setting page for account deletion. On this page, two user input is required, which are username and confirm password. The “Yes, I’m Sure” button is in red, the purpose is to warn the user the account deletion is not able to undo.

4.7 Use Case Diagram and Description

4.7.1 Use Case Diagram

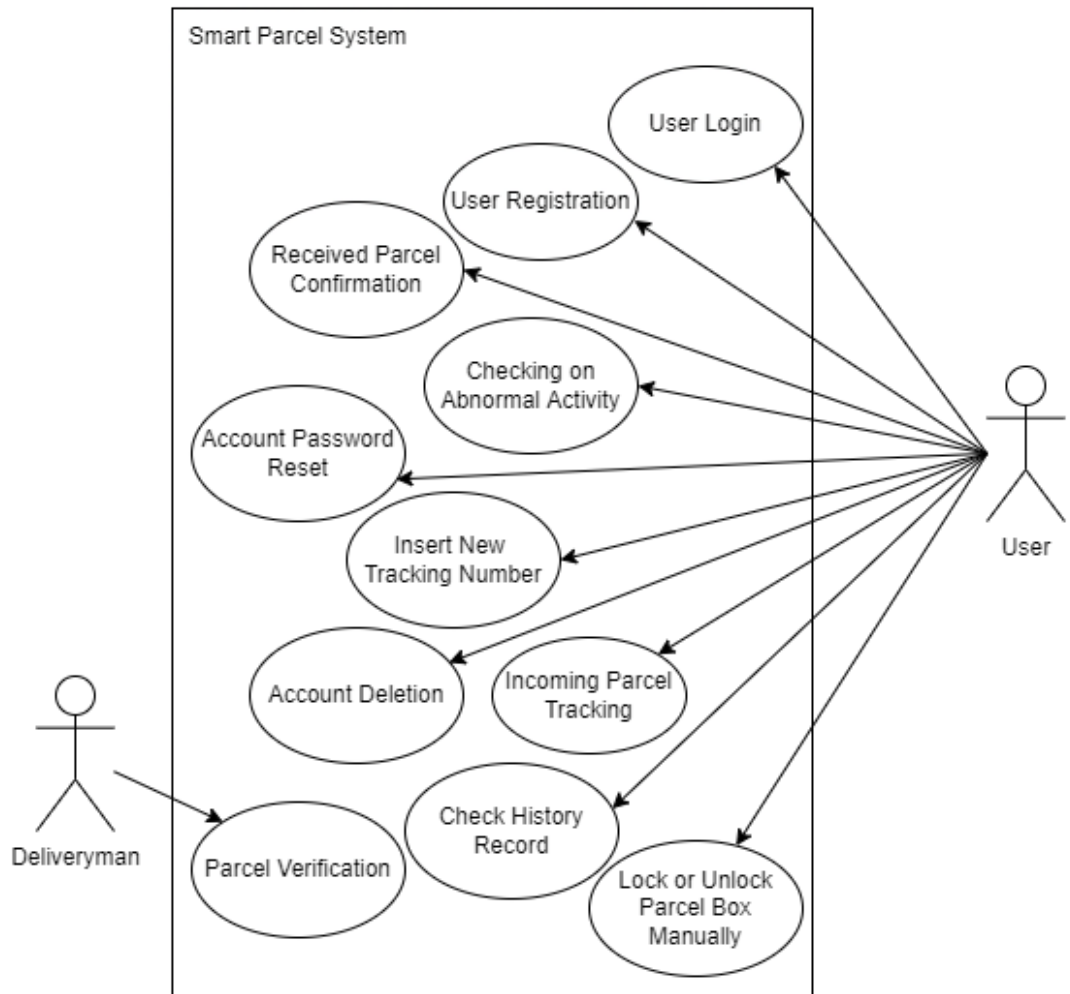


Figure 4.22 Use Case Diagram

4.7.2 Use Case Description

4.7.2.1 Use Case Description of UC001

Table 4.4 Use Case Description of UC001

Use Case Name: Parcel Verification	ID: UC001
Actor: Deliveryman	
Description: Deliveryman unlock smart parcel box by scanning QR code.	
Trigger: 1. Deliveryman place the QR code in front of the camera.	
Precondition: The QR code must be matched the information in the database.	
Normal Flow of Events: 1. Smart parcel box should unlock if the QR code is valid.	
Alternate/ Exceptional Flows: 1. Smart parcel box should remain locked if the QR code is invalid. 2. System is not able to analyse the QR code if the QR code is incomplete or broken.	

4.7.2.2 Use Case Description of UC002

Table 4.5 Use Case Description of UC002

Use Case Name: User Registration	ID: UC002
Actor: User	
Description: User register an account on the website.	
Trigger: 1. User submits username, first name, last name, email, password, and confirm password.	
Precondition: 1. Username and email must not duplicate with information in the database. 2. Password and confirm password must be matched.	
Normal Flow of Events: 1. User successfully registers an account and login into website automatically.	
Alternate/ Exceptional Flows: 1. Process failed if the input username or email is duplicate with information in database. 2. Process failed if the password and confirm password are not matched. 3. Process failed if any blank value is submitted.	

4.7.2.3 Use Case Description of UC003

Table 4.6 Use Case Description of UC003

Use Case Name: User Login	ID: UC003
Actor: User	
Description: User login to website with a valid username and password.	
Trigger: 1. User key in username and password on login page.	
Precondition: Username and password must match with information in the database.	
Normal Flow of Events: 1. User successfully login to the website.	
Alternate/ Exceptional Flows: 1. User unable to login if the username and password are unmatched with information in database.	

4.7.2.4 Use Case Description of UC004

Table 4.7 Use Case Description of UC004

Use Case Name: Received Parcel Confirmation	ID: UC004
Actor: User	
Description: User allowed to acknowledge and confirm the received parcel.	
Trigger: 1. User click on the “verify” button for the received parcel.	
Precondition: User has an unverified received parcel.	
Normal Flow of Events: 1. The parcel is marked as verified and will display in history record.	
Alternate/ Exceptional Flows: -	

4.7.2.5 Use Case Description of UC005

Table 4.8 Use Case Description of UC005

Use Case Name: Checking on Abnormal Activity	ID: UC005
Actor: User	
Description: User allowed check on abnormal activity record.	
Trigger: 1. User click on the “check” button for abnormal activity.	
Precondition: User has unchecked abnormal activity.	
Normal Flow of Events: 1. The abnormal activity is marked as checked and will display in history record.	
Alternate/ Exceptional Flows: -	

4.7.2.6 Use Case Description of UC006

Table 4.9 Use Case Description of UC006

Use Case Name: Insert New Tracking Number	ID: UC006
Actor: User	
Description: User allowed to insert new tracking number into database.	
Trigger: 1. User inserts tracking number and selects a courier.	
Precondition: The tracking number must not be duplicate with the record in the database.	
Normal Flow of Events: 1. The tracking number successfully inserted into database.	
Alternate/ Exceptional Flows: 1. Process failed if user inserts tracking number that already existed in the database. 2. Process failed if user submit blank value.	

4.7.2.7 Use Case Description of UC007

Table 4.10 Use Case Description of UC007

Use Case Name: Incoming Parcel Tracking	ID: UC007
Actor: User	
Description: User is allowed to track the incoming parcel.	
Trigger: 1. User clicks on the “track” button beside the incoming parcel.	
Precondition: User has incoming parcel.	
Normal Flow of Events: 1. User will be redirected to the parcel tracking website based on the tracking number and courier.	
Alternate/ Exceptional Flows: 1. The parcel is unable to track if user inserted a wrong parcel tracking number or wrong courier into the database.	

4.7.2.8 Use Case Description of UC008

Table 4.11 Use Case Description of UC008

Use Case Name: Lock or Unlock Parcel Box Manually	ID: UC008
Actor: User	
Description: User allowed to lock or unlock the parcel box manually.	
Trigger: 1. User click on the “lock” or “unlock” button on the website.	
Precondition: The smart parcel box is online and hosting the parcel control website.	
Normal Flow of Events: 1. Smart parcel box will be controlled by the user to lock or unlock.	
Alternate/ Exceptional Flows: 1. User unable to go to parcel box control page if the Raspberry Pi do not host the website.	

4.7.2.9 Use Case Description of UC009

Table 4.12 Use Case Description of UC009

Use Case Name: Check History Record	ID: UC009
Actor: User	
Description: User allowed to see history record of the received parcel and abnormal activity.	
Trigger: 1. User goes to history record page.	
Precondition: User has confirmed received parcel or checked abnormal activity.	
Normal Flow of Events: 1. User should see a list of confirmed received parcel or checked abnormal activity.	
Alternate/ Exceptional Flows: 1. User unable to see any record if the confirmed received parcel or checked abnormal activity data are empty.	

4.7.2.10 Use Case Description of UC010

Table 4.13 Use Case Description of UC010

Use Case Name: Account Password Reset	ID: UC010
Actor: User	
Description: User allowed to reset account password.	
Trigger: 1. User key in username, old password, new password, and confirm new password in account password reset page.	
Precondition: Username and old password must be matched with information in database, new password and confirm new password must be matched.	
Normal Flow of Events: 1. User successfully changed account password.	
Alternate/ Exceptional Flows: 1. Process failed if username or old password unmatched with the information in database. 2. Process failed if new password and confirm new password are unmatched.	

4.7.2.11 Use Case Description of UC011

Table 4.14 Use Case Description of UC011

Use Case Name: Account Deletion	ID: UC011
Actor: User	
Description: User allowed delete account permanently.	
Trigger: 1. User key in username and password in account deletion page.	
Precondition: Username and password must be matched with the information in database.	
Normal Flow of Events: <ol style="list-style-type: none"> 1. Account will be permanently deleted if the username and password matched with the information in the database. 	
Alternate/ Exceptional Flows: <ol style="list-style-type: none"> 1. Process failed if username or password unmatched with the information in the database. 	

4.8 Concluding Remark

Chapter Four had shown the system architecture design of the overall system and website architecture design. Moreover, the functional module in the system had explained in detail in this chapter. ERD and data dictionary for every database are also presented in this chapter. A database algorithm, which is hashing algorithm is explained as well. Besides, this chapter also presented about the GUI design of the website and explained on each use cases description. Finally, this chapter concludes with the use case diagram and its description.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Hardware Setup

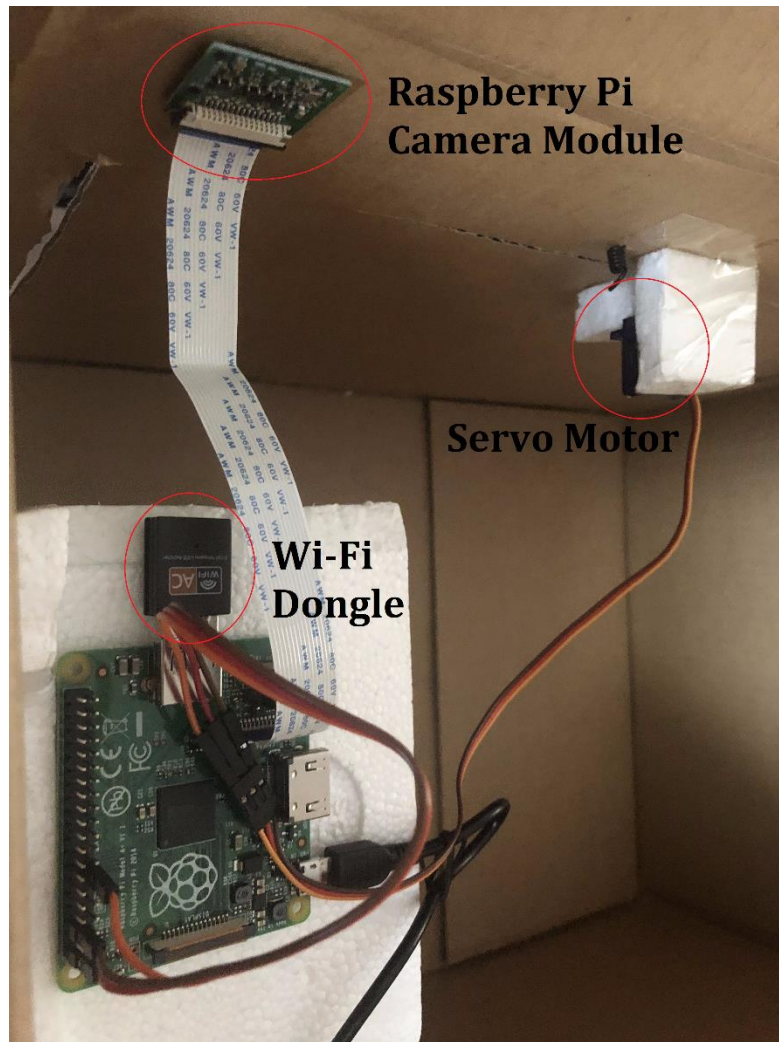


Figure 5.1 System Prototype of Smart Parcel Box

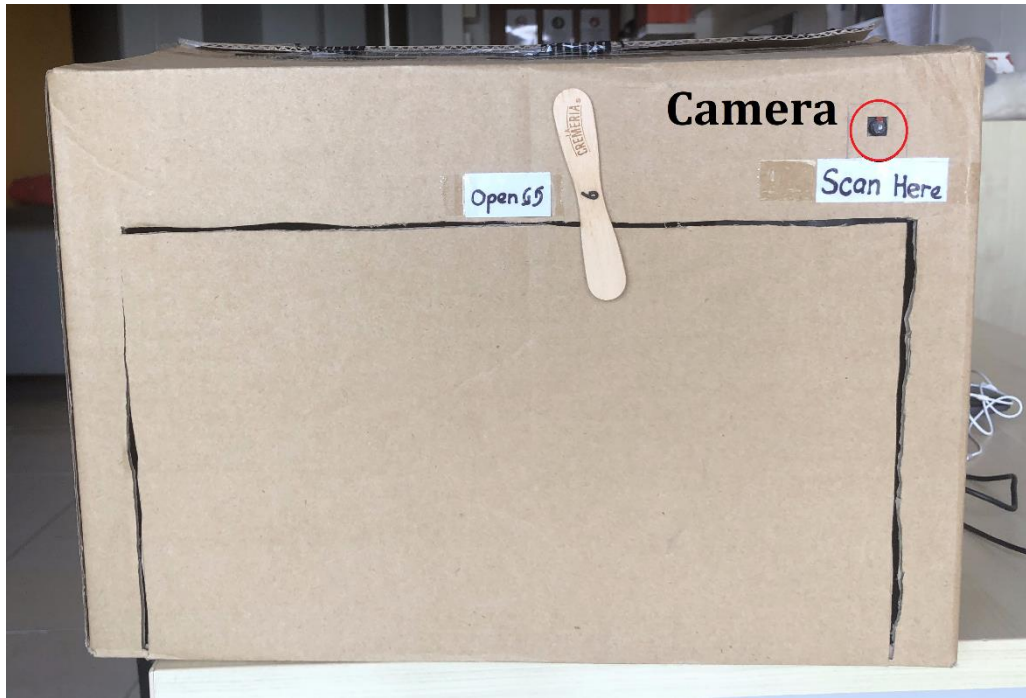


Figure 5.2 Front View of System Prototype

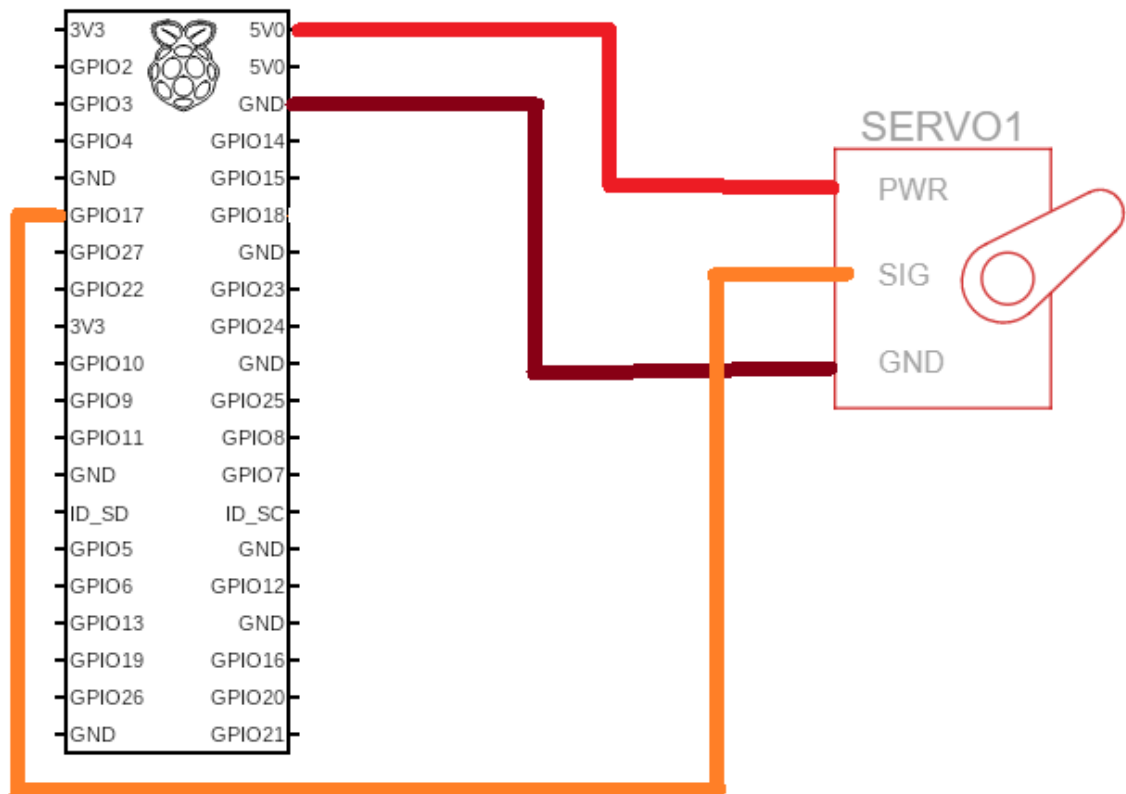


Figure 5.3 Wiring Diagram of System

Table 5.1 Pin Connection between Raspberry Pi and SG90 Micro Servo

Component	Pin	Pin	Component
Raspberry Pi A+ GPIO Pinout	02 (5V)	PWR (Red)	SG90 Micro Servo
	06 (Ground)	GND (Brown)	
	11 (GPIO17)	SIG (Orange)	

Set up the hardware as shown in figure 5.1. Connect the camera module to the Raspberry Pi and place the camera lens face out the parcel box, like the diagram shown in figure 5.2. As for the wiring between Raspberry Pi and SG90 Micro Servo, the connection was presented in figure 5.3 and table 5.1. The red colour wire of SG90 Micro Servo connects to a 5V pin (GPIO pin 02) and the brown colour wire of SG90 Micro Servo must connect to the ground pin (GPIO pin 06), while the orange colour wire of SG90 Micro Servo needs to connect to a GPIO output pin, which is GPIO pin 11 (also known as GPIO17). The WIFI5 AC600 Dongle is needed for the Raspberry Pi to connect to the internet. Once the hardware has been setup correctly, turn on the power supply and the system is good to go.

5.2 Software Setup

5.2.1 Raspberry Pi OS

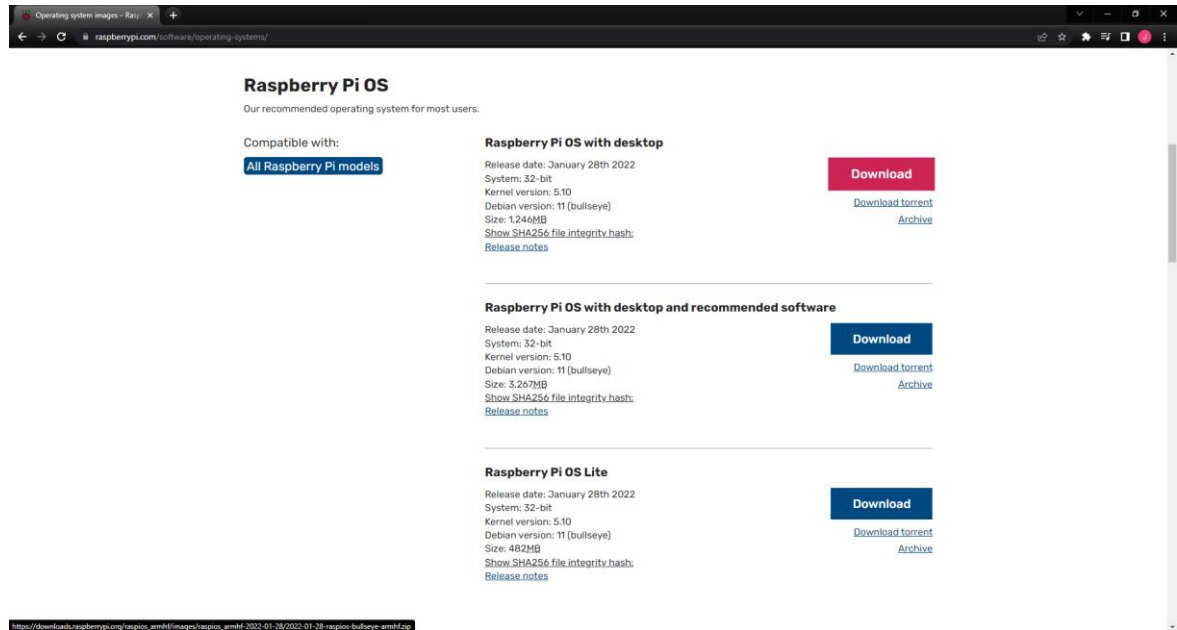


Figure 5.4 Raspberry Pi OS Download Page



Figure 5.5 Raspberry Pi Imager

Download the first OS (Raspberry Pi OS with desktop) on the download page. After that, use the Raspberry Pi Imager to flash the file into the SD card (as shown in figure 5.5).

CHAPTER 5

5.2.2 OpenCV / pyzbar

To install OpenCV and pyzbar library in the Raspberry Pi, type in the following commands in the terminal and wait for it to install.

- `sudo pip3 install opencv-python`
- `sudo pip3 install pyzbar`

5.2.3 WAMP Server



Figure 5.6 WAMP Server Download Page

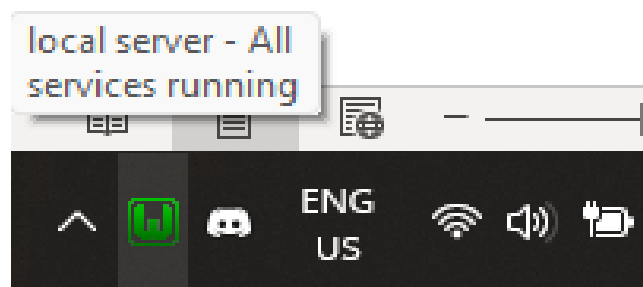


Figure 5.7 WAMP Server Running Status



Figure 5.8 WAMP Server Homepage

Download from the WAMP Server official website and install it then start the WAMP Server. After a while, the WAMP Server icon should turn green and show that all service is running, as shown in figure 5.7. Access 127.0.0.1 on the web browser, the page will look like figure 5.8 if the WAMP Server is properly installed and running. To access WAMP Server from the Raspberry Pi, a library must be installed in the Raspberry Pi using the following command:

- `sudo pip3 install mysql-connector`

5.2.4 Telegram Bot



Figure 5.9 Telegram BotFather

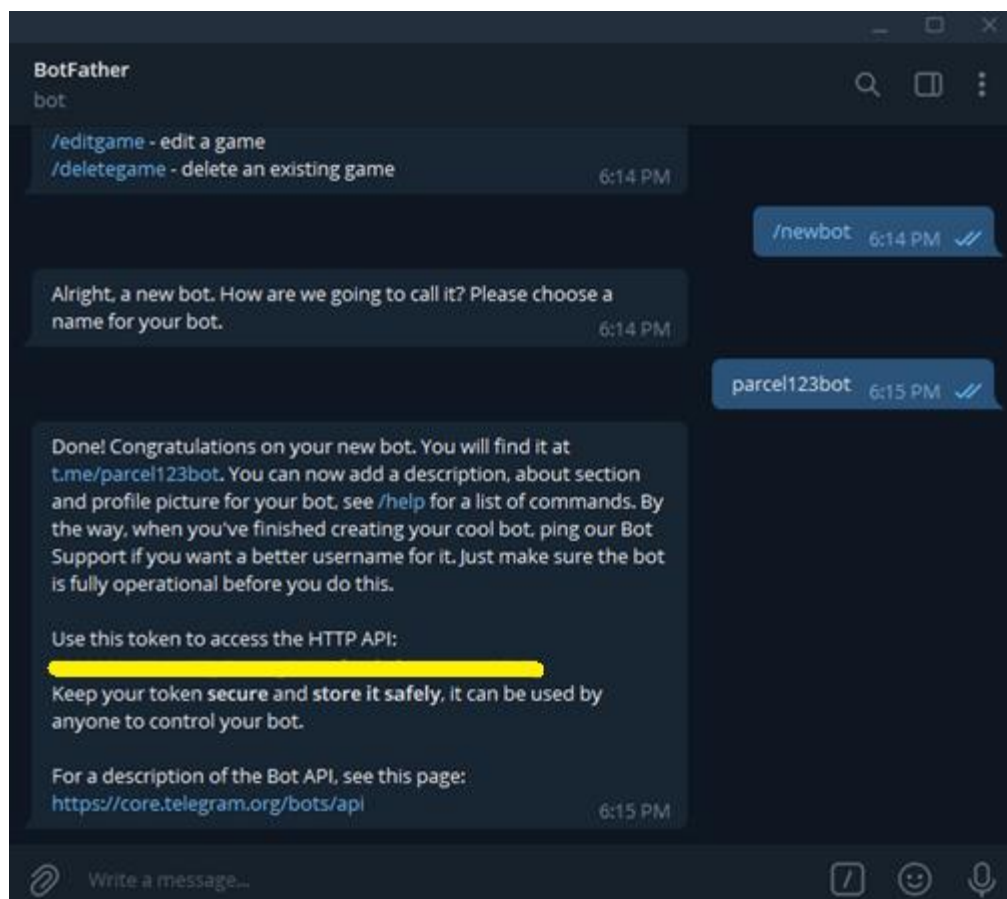


Figure 5.10 Chat History with BotFather

CHAPTER 5

On the telegram application, search for BotFather and start the conversation with “/start”, followed by “/newbot” and choose a unique bot name. Remember and save the API (Application Programming Interface) token.

5.3 Setting and Configuration

5.3.1 Raspberry Pi Camera Module

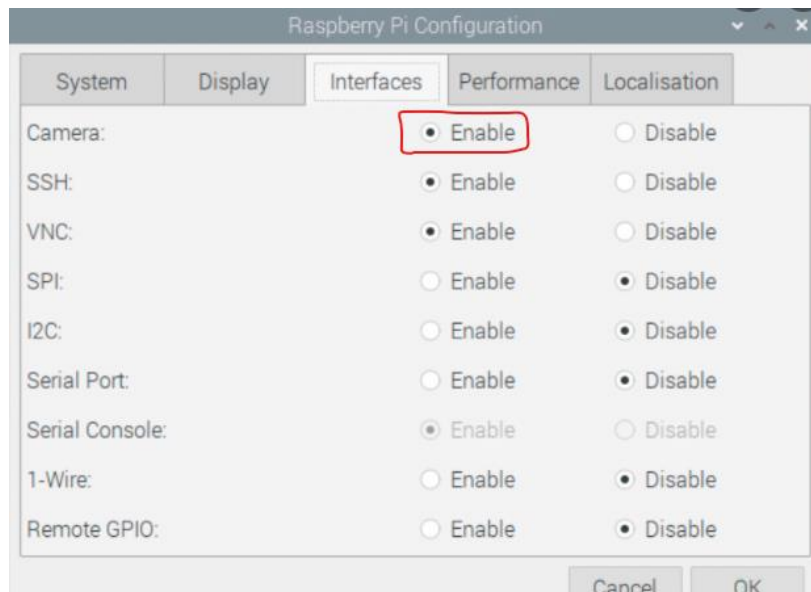


Figure 5.11 Raspberry Pi Configuration

To enable the camera interfaces in Raspberry Pi, follow the path:

Raspberry Pi Menu => Preferences => Raspberry Pi Configuration

After that, a window will pop up as shown in figure 5.11, enable the interface of camera (circled in red).

5.3.2 WAMP Server



Figure 5.12 PHPMYAdmin Page

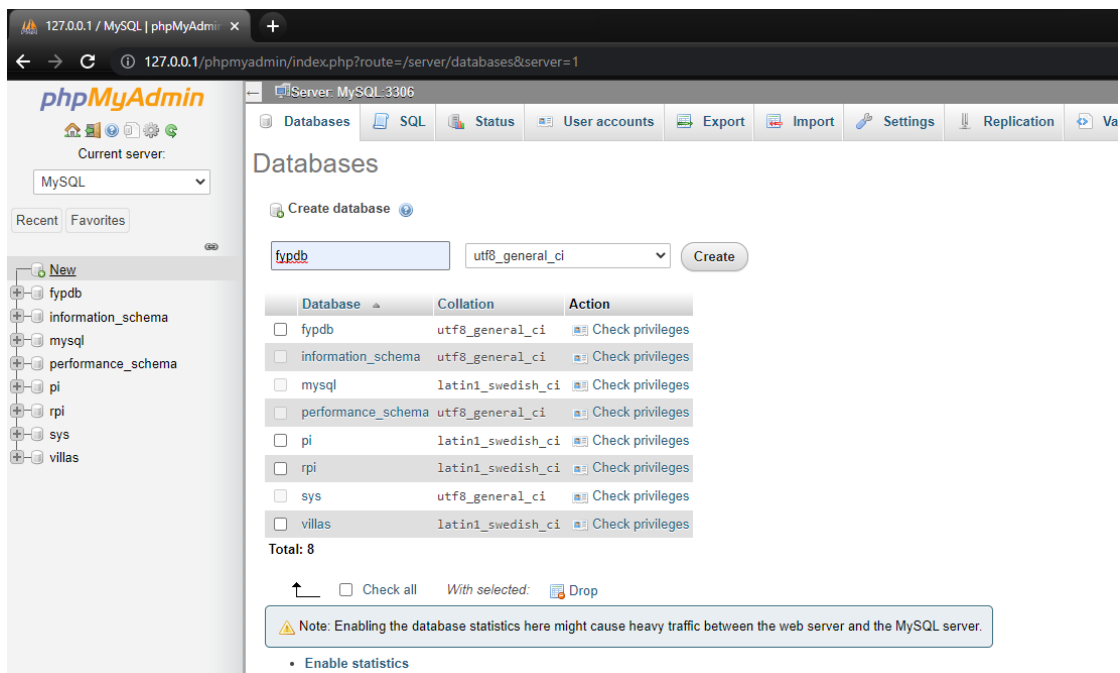


Figure 5.13 Create Database Page

In the web browser, go to 127.0.0.1/phpmyadmin/ click on go as the password by default is blank. Then, like the figure 5.13 shown, click on “New” button on the left side of the page and create a database by giving it a name and click “Create”.

CHAPTER 5

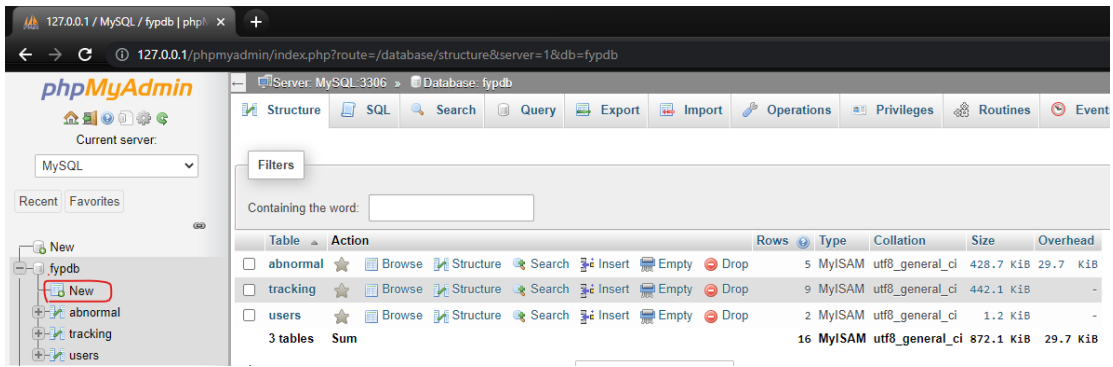


Figure 5.14 Create Table Page

As shown in figure 5.14, go into the database created in previous step and create three tables to store the data. The table details may go to Chapter Four - 4.4 Database Design. Follow the data dictionary to create 3 tables in the database.

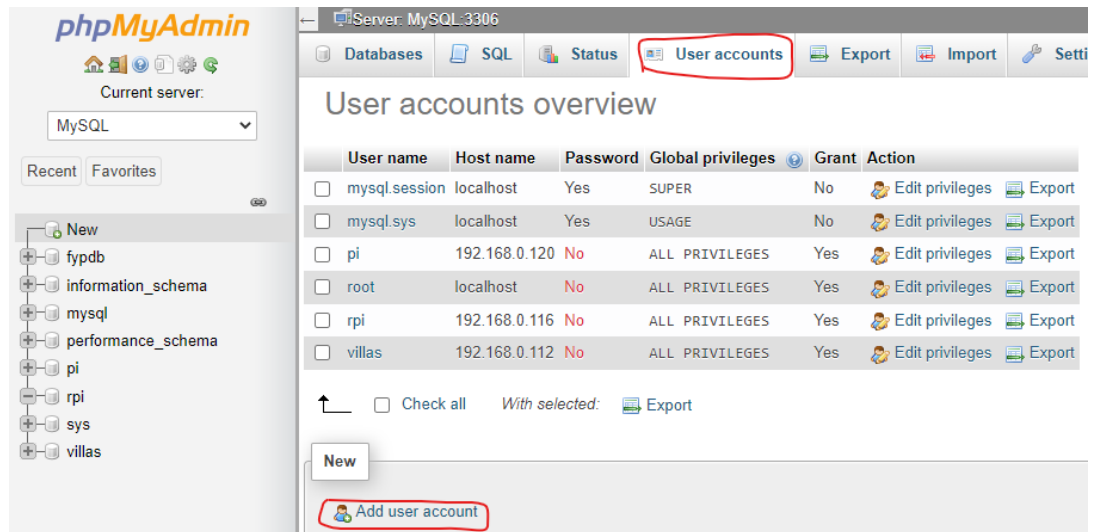


Figure 5.15 User Accounts Page

The screenshot shows the 'Add user account' page in phpMyAdmin. The 'Login Information' section includes fields for 'User name' (jiaheng), 'Host name' (192.168.0.116), 'Password', and 'Authentication plugin' (Native MySQL authentication). The 'Database for user account' section has two checked options: 'Create database with same name and grant all privileges' and 'Grant all privileges on wildcard name (username)_%'. The 'Global privileges' section has a 'Check all' button checked. Below this, there are four columns of checkboxes for 'Data', 'Structure', 'Administration', and 'Resource limits', all of which are checked. The 'Resource limits' section includes input fields for 'MAX QUERIES PER HOUR', 'MAX UPDATES PER HOUR', 'MAX CONNECTIONS PER HOUR', and 'MAX_USER_CONNECTIONS', all set to 0.

Figure 5.16 Create User Account Page

Next, go back to phpMyAdmin page, go to “User accounts” section and “add user account”. In figure 5.15, key in the username and hostname based on the Raspberry Pi IP address. Then check all the boxes and create the user account.

5.3.3 Telegram Bot

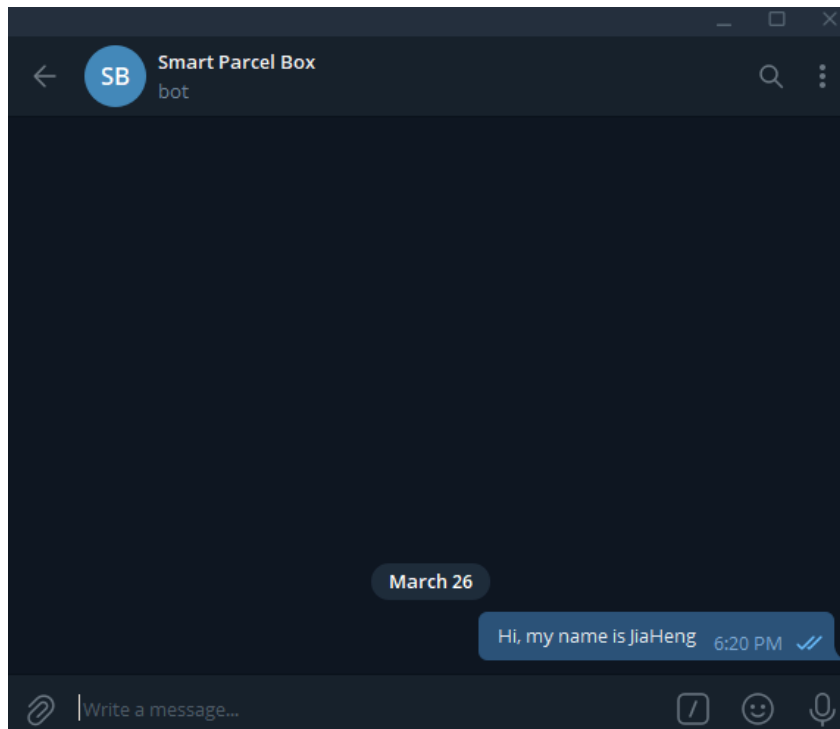


Figure 5.17 Chat with Telegram Bot

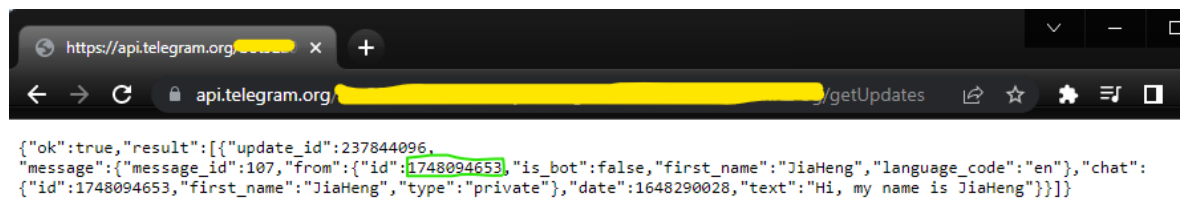


Figure 5.18 Telegram Chat ID

To get the user telegram chat ID, first, the user needs to send a message to the telegram bot, as shown in figure 5.17. Then, use the URL below and replace <TOKEN> with the bot API token from the previous step.

<https://api.telegram.org/bot<TOKEN>/getUpdates>

User chat ID was circled in green. Save the user chat ID in a file, which will be using to send the notification to the user. Next, create a python script as shown in the figure 5.19.

CHAPTER 5

```
telebot.py x
1  #import library
2  import requests
3
4  #To send the notification via telegram bot
5
6  def telegram_msg(bot_msg):
7      #telegram bot api token
8      token_id = "5220081165:AAF5yZsVzAg7-RNoltnieirvw1NmmnRZ-5g"
9
10     #user chat id
11     chat_id = "1748094653"
12     send_text = "https://api.telegram.org/bot" + token_id + "/sendMessage?chat_id=" + chat_id + "&parse_mode=HTML&text=" + bot_msg
13     print("Notification sent")
14     response = requests.get(send_text)
15
16     return response.json()
17
18
19
```

Figure 5.19 Telegram Bot Python Script

5.3.4 Main Python Script

```
1  import cv2
2  import numpy as np
3  from pyzbar.pyzbar import decode
4  import mysql.connector
5  import datetime
6  import telebot
7  import time
8  import RPi.GPIO as GPIO
9
10     #Create a connection to MySQL database
11     mydb = mysql.connector.connect(user='pi', password='', host='192.168.0.107', database='fypdb')
12     #Class that allows Python code to execute MySQL command
13     mycursor = mydb.cursor(buffered=True)
14     #Check if the table was created
15     mycursor.execute("SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA='fypdb' AND TABLE_NAME='tracking';")
16     #Store returned result into table_count
17     table_count = mycursor.fetchall()
18
19     if (mycursor.rowcount == 1):
20         #If it is created, do nothing
21         print("Table Found.")
22     else:
23         #If it is not created, create one
24         print("Table not found.")
25         exit()
26
27     username = 'jiaheng56' #declare username
28     datalist = ''
29     checkpoint = time.time()
30
31     servo_pin = 11 # GPIO Pin where sero is connected
32     GPIO.setmode(GPIO.BOARD) # Define the Pin numbering type.
33     # Defing Servo Pin as output pin
34     GPIO.setup(servo_pin, GPIO.OUT)
35     servo = GPIO.PWM(servo_pin, 50) # PWM channel at 50 Hz frequency
36     servo.start(0) # Zero duty cycle initially
37
38     #function to unlock parcel box
39     def unlock():
40         servo.ChangeDutyCycle(2)
41         print(' [INFO] The Door Unlocked.')
42         print(' [INFO] The Door will be Locked After 10 seconds.')
43         time.sleep(10)
44         servo.ChangeDutyCycle(7)
45         print(' [INFO] The Door Locked.\n')
```

Figure 5.20 Main Python Script (1)

CHAPTER 5

```
47 #retrieve information of parcel
48
49 try:
50     sql_select_query = f"SELECT * FROM tracking WHERE parcel_received_time IS NULL AND username = '{username}'"
51     mycursor.execute(sql_select_query)
52     records = mycursor.fetchall()
53
54     #store the tracking number in a list
55     for col in records:
56         dataList = dataList + col[1] + '\n'
57
58 finally:
59     mydb.commit()
60
61 def convertToBinaryData(filename):
62     #convert digital data to binary format
63     with open(filename, 'rb') as file:
64         binaryData = file.read()
65     return binaryData
66
67 #setting up camera frame
68 cap = cv2.VideoCapture(0)
69 cap.set(3,640)
70 cap.set(4,480)
71
72 while True:
73     success, img = cap.read()
74
75     #when a qr code detected
76     for barcode in decode(img):
77         data = barcode.data.decode('utf-8')
78         print("\n [INFO] QR Detected: " + data) #show detected qr data
79         if data in dataList:
80             check = time.time() #current time
81             #to avoid notification spam to the receiver
82             if (check - checkpoint > 15):
83                 out = cv2.imwrite(f'{data}.jpg',img) #save photo
84                 photo = convertToBinaryData(f"/home/pi/{data}.jpg")
85                 print(" [INFO] Tracking Number Matched.")
86                 dt = datetime.datetime.now() #get current timestamp
87                 now = dt.strftime("%Y-%m-%d %H:%M:%S")
88                 unlock()
89                 try: #update database
90                     sql_update_query = "UPDATE tracking SET parcel_received_time=%s, photo=%s WHERE tracking_num=%s"
91                     update_data = (now, photo, data)
```

Figure 5.21 Main Python Script (2)

```
91         update_data = (now, photo, data)
92         mycursor.execute(sql_update_query, update_data)
93         mydb.commit()
94         #send notification
95         telebot.telegram_msg('Hey ' + username + '. Your parcel with the tracking number of ' + data + ' had been received on ' + now)
96         dataList = dataList.replace(data, "") #remove tracking number from the list
97         checkpoint = time.time() #mark checkpoint
98         print(' [INFO] Database Updated.\n')
99     except:
100         print(" [ERROR] Error Updating Data.")
101     else:
102         print(' [WARNING] Time Interval Too Short')
103
104     else:
105         check = time.time()
106         #to avoid notification spam to receiver
107         if (check - checkpoint > 15):
108             print("\n [INFO] Tracking Number Unmatch.")
109             dt = datetime.datetime.now()
110             now = dt.strftime("%Y-%m-%d %H:%M:%S")
111             out = cv2.imwrite(f'{data}.jpg',img) #save photo
112             photo = convertToBinaryData(f"/home/pi/{data}.jpg")
113             try: #update database
114                 sql_insert_query = "INSERT INTO abnormal(username, record_time, qr_info, photo) VALUES (%s, %s, %s, %s)"
115                 insert_data = (username, now, data, photo)
116                 mycursor.execute(sql_insert_query, insert_data)
117                 mydb.commit()
118                 #send notification
119                 telebot.telegram_msg('Hey ' + username + '. The system had detected an abnormal activity on ' + now)
120                 checkpoint = time.time()
121                 print(' [INFO] Abnormal Activity Recorded.\n')
122             except:
123                 print(" [ERROR] Error Updating Data.")
124             else:
125                 print(' [WARNING] Time Interval Too Short')
126
127     cv2.imshow('Result',img)
128     cv2.waitKey(1)
129
130 #close database
131 mycursor.close()
132 mydb.close()
```

Figure 5.22 Main Python Script (3)

Figure 5.20 to figure 5.22 show the main python script. Create a python file named it as “parcel.py” and type in the code as shown above.

CHAPTER 5

```
1 from flask import Flask, render_template_string, request # Importing the Flask modules
2 import RPi.GPIO as GPIO # Importing the GPIO library
3 from time import sleep # Import sleep module from time library
4
5 servo_pin = 11 # GPIO Pin where sero is connected
6 GPIO.setmode(GPIO.BOARD) # Define the Pin numbering type. Here we are using BCM type
7 # Defing Servo Pin as output pin
8 GPIO.setup(servo_pin, GPIO.OUT)
9 p = GPIO.PWM(servo_pin, 50) # PWM channel at 50 Hz frequency
10 p.start(0) # Zero duty cycle initially
11 app = Flask(__name__)
12 #HTML Code
13 TPL = '''
14 <!DOCTYPE html>
15 <html lang="en">
16 <head>
17 <title>Parcel Box Control</title>
18 <meta charset="utf-8">
19 <meta name="viewport" content="width=device-width, initial-scale=1">
20 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
21 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
22 <style>
23 .button {
24 background-color: #4CAF50; /* Green */
25 width: 120px;
26 height: 40px;
27 border-radius: 10px;
28 border: none;
29 color: white;
30 padding: 1px 10px;
31 text-align: center;
32 text-decoration: none;
33 display: inline-block;
34 font-size: 16px;
35 margin: 2px 2px;
36 transition-duration: 0.4s;
37 cursor: pointer;
38 }
39 .button5 {
40 background-color: white;
41 color: black;
42 border: 2px solid #555555;
43 }
44
45 .button5:hover {
```

Figure 5.23 Parcel Box Control Function Script (1)

```
45 .button5:hover {
46 background-color: #555555;
47 color: white;
48 }
49 </style>
50 </head>
51
52 <body>
53 <div class="p-5 bg-dark text-white text-center">
54 <h1>My Smart Parcel Box</h1>
55 </div>
56
57 <div class="container mt-5">
58 <h2> Unlock or Lock Your Parcel Box</h2><br>
59 <form method="POST" action="test">
60 <button type="submit" value="2" class="button button5" name="slider">Unlock</button>
61 <button type="submit" value="7" class="button button5" name="slider">Lock</button>
62 </form>
63 </div>
64 </body>
65 '''
66 @app.route("/")
67 def home():
68 return render_template_string(TPL)
69 @app.route("/test", methods=["POST"])
70 def test():
71 # Get slider Values
72 slider = request.form["slider"]
73 # Change duty cycle
74 p.ChangeDutyCycle(float(slider))
75 sleep(1)
76 # Pause the servo
77 p.ChangeDutyCycle(0)
78 return render_template_string(TPL)
79 # Run the app on the local development server
80 if __name__ == "__main__":
81 app.run(host='0.0.0.0', debug=False)
82
```

Figure 5.24 Parcel Box Control Function (2)

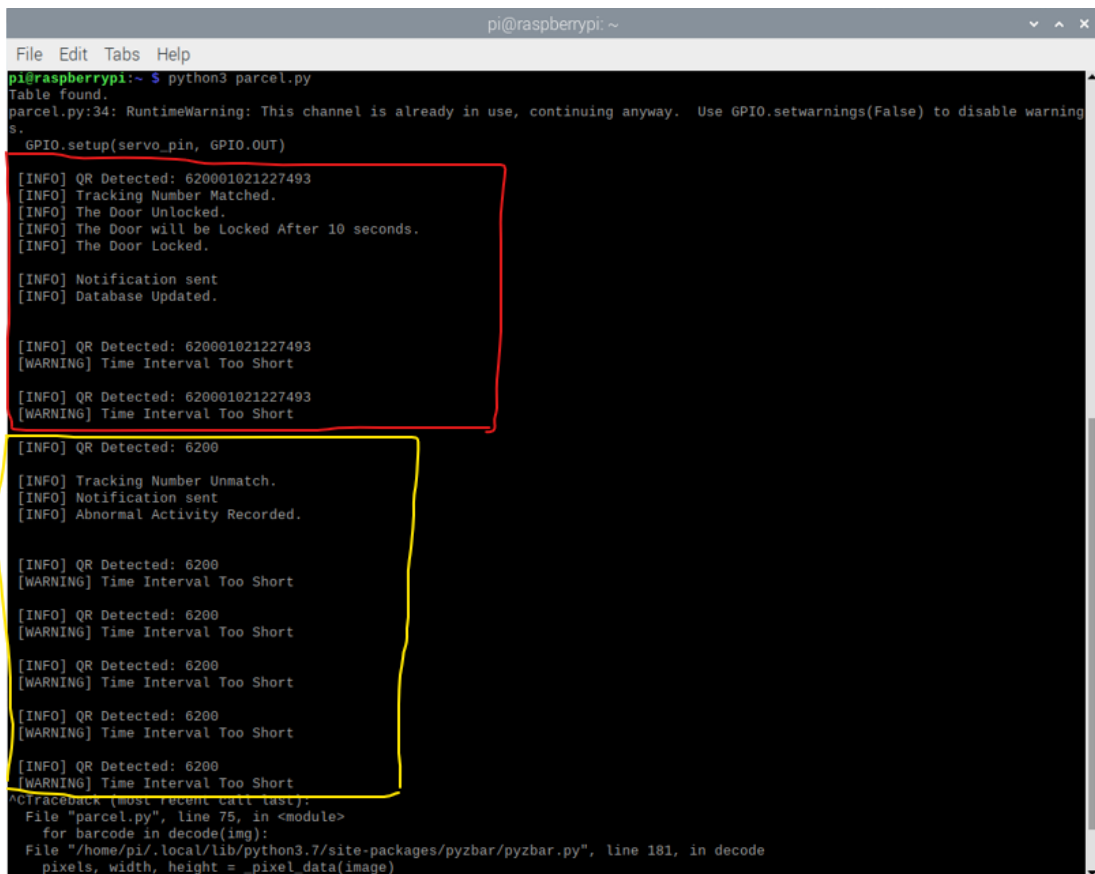
CHAPTER 5

Figure 5.23 and figure 5.24 show the python script of the parcel box control function. Create a python file name “parcel_control.py” and copy the code into the file.

5.4 System Operation

To make the system to function, three things need to run. The first thing is the WAMP Server, run the WAMP Server and wait until the icon turn green, which means that the server is online, and the database can be accessed by the Raspberry Pi. The next thing is the python script, using the command: `python3 parcel.py` in the terminal to run the main function of the system. After that, there is another python script need to be run, which is the parcel box control function script, using the command: `python3 control_parcel.py` to start the function.

5.4.1 Main Function



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ python3 parcel.py
Table found.
parcel.py:34: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(servo_pin, GPIO.OUT)

[INFO] QR Detected: 620001021227493
[INFO] Tracking Number Matched.
[INFO] The Door Unlocked.
[INFO] The Door will be Locked After 10 seconds.
[INFO] The Door Locked.

[INFO] Notification sent
[INFO] Database Updated.

[INFO] QR Detected: 620001021227493
[WARNING] Time Interval Too Short

[INFO] QR Detected: 620001021227493
[WARNING] Time Interval Too Short

[INFO] QR Detected: 6200
[INFO] Tracking Number Unmatch.
[INFO] Notification sent
[INFO] Abnormal Activity Recorded.

[INFO] QR Detected: 6200
[WARNING] Time Interval Too Short

[INFO] QR Detected: 6200
[WARNING] Time Interval Too Short

[INFO] QR Detected: 6200
[WARNING] Time Interval Too Short

[INFO] QR Detected: 6200
[WARNING] Time Interval Too Short

[INFO] QR Detected: 6200
[WARNING] Time Interval Too Short

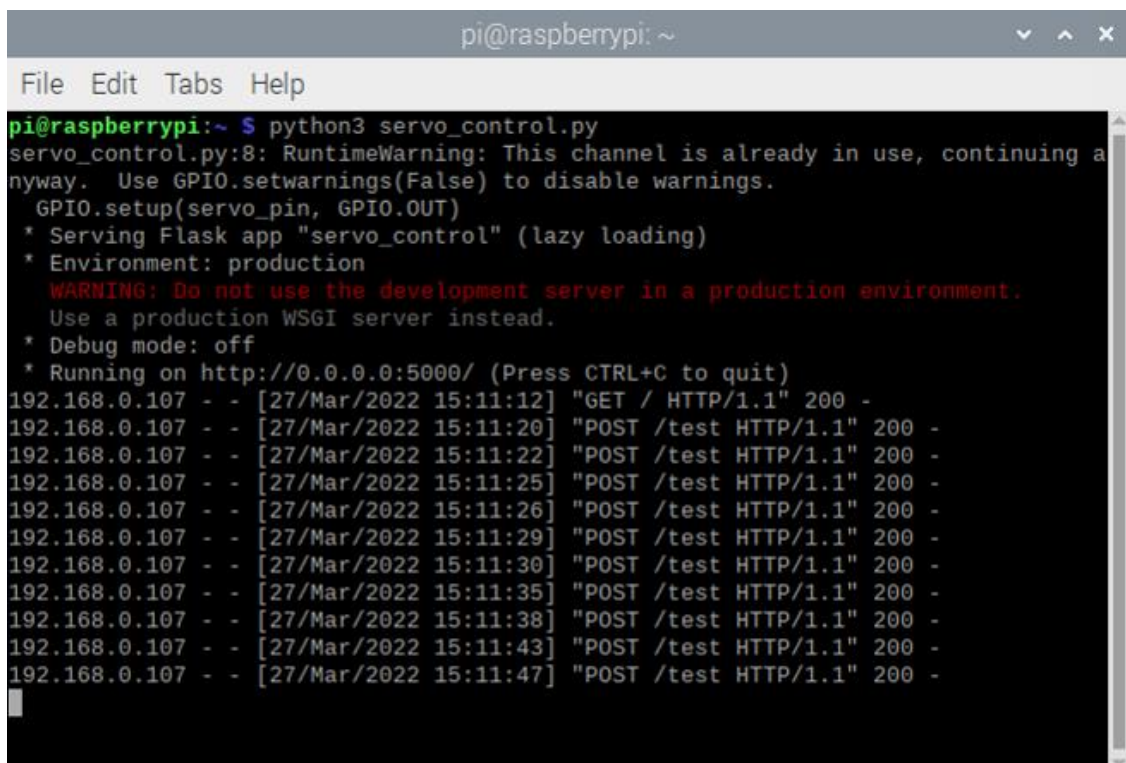
^C
Traceback (most recent call last):
  File "parcel.py", line 75, in <module>
    for barcode in decode(img):
  File "/home/pi/.local/lib/python3.7/site-packages/pyzbar/pyzbar.py", line 181, in decode
    pixels, width, height = _pixel_data(image)
```

Figure 5.25 Result of the Main Function

From figure 5.25, the result shows the process of QR verification, once the QR code is detected on the Raspberry Pi camera module, the system will analyse the data of the

QR code. The result in the red zone is using a valid QR code, once the QR is validated, the parcel box will be unlocked for 10 seconds and then immediately locked. After that, a notification will be sent to the receiver and the database will be updated. Next, the result in the yellow zone is using an invalid QR code, when the QR code is detected as an invalid code, the door will not be unlocked, and a notification will send to alert the receiver. Then, the abnormal activity will be recorded and inserted into the database. In both results, there are warning shown in the terminal. The purpose of this function is to avoid notification spam to the receiver as the system needs some buffer time to process.

5.4.2 Parcel Box Control Function



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3 servo_control.py
servo_control.py:8: RuntimeWarning: This channel is already in use, continuing a
nyway. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(servo_pin, GPIO.OUT)
* Serving Flask app "servo_control" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
192.168.0.107 - - [27/Mar/2022 15:11:12] "GET / HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:20] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:22] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:25] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:26] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:29] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:30] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:35] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:38] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:43] "POST /test HTTP/1.1" 200 -
192.168.0.107 - - [27/Mar/2022 15:11:47] "POST /test HTTP/1.1" 200 -

```

Figure 5.26 Result of Parcel Box Control Function

In figure 5.26, it shows the result of parcel box control function. The Raspberry Pi will be hosting a website at the address of https://<raspberrypi_ipaddress>:5000, replace the *<raspberrypi_ipaddress>* with the Raspberry Pi's IP address. Next, the result shows that a machine with the IP address of 192.168.0.107 had connected to the Raspberry Pi and controlled the parcel box remotely.

5.4.3 Telegram Bot Notification

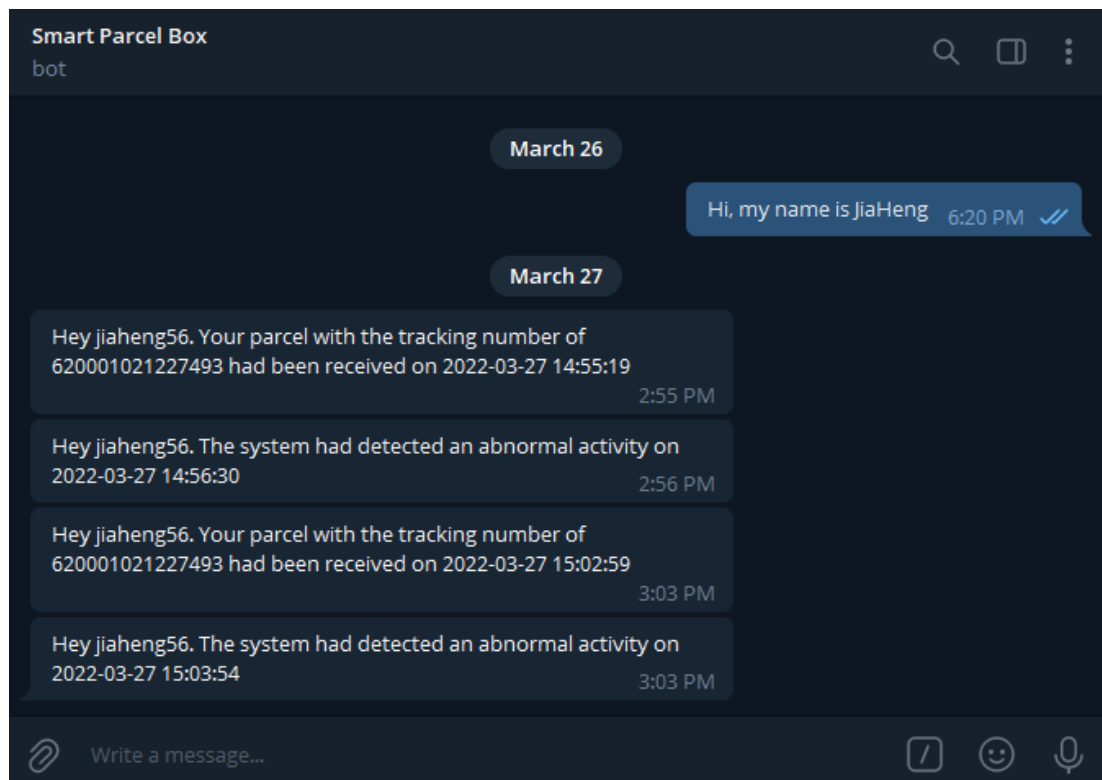


Figure 5.27 Telegram Bot Result

Figure 5.27 shows the result of the telegram bot. The result indicates that the notifications are successfully sent to the receiver.

5.5 concluding Remark

In summary, Chapter Five discussed the system implementation. Hardware setup and some wiring setups are shown in this chapter. Next, software installation and setup are also presented, as well as the configuration of the software. Moreover, the system operation also explained about how the process is being carried out, and the result of the operation is also explained in detail.

CHAPTER 6

SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

In the system testing process, the accuracy and connectivity are focused on as this system is involved in the communication between applications wirelessly and remotely. To test the accuracy of the system, it will be testing on the accuracy of the Raspberry Pi camera module for the QR code reading and how accurate the parcel box is to unlock after the QR verification. Besides, the accuracy of the data update or insert into the database, the accuracy of the Telegram notification sent to the receiver, and the accuracy of the control of the smart parcel box remotely will be testing as well. All tests will be carried out for 10 times.

As for the connectivity testing for the system, it will be testing on the connectivity of the Raspberry Pi camera module to scan the QR code, and the connectivity of the servo to unlock the parcel box. Next, there will also be testing on the connectivity of the parcel box control remotely and the Telegram notification sent to the receiver. All tests will be carried out for 10 times.

6.2 Testing Setup and Result

6.2.1 Testing of the Raspberry Pi Camera Module

To test the accuracy of the Raspberry Pi camera module, there will be several scenarios to test the result under different environment settings, for example when the environment is very dark, the QR code gets wet from water, and there are some scratches or flaws on the QR code. For the connectivity testing, the time detected and verification process will be recorded, and see how long the system takes to verify the QR code.

Scenario 1 – Normal Light (Test 10 times)

Table 6.1 Result of System Testing (1)

	Accuracy	Connectivity (second)
Valid QR Code	10	0.23
Invalid QR Code	10	0.20

Scenario 2 – Little Light (Test 10 times)

Table 6.2 Result of System Testing (2)

	Accuracy	Connectivity
Valid QR Code	10	0.25
Invalid QR Code	10	0.22

Scenario 3 – No Light (Test 10 times)

Table 6.3 Result of System Testing (3)

	Accuracy	Connectivity
Valid QR Code	0	-
Invalid QR Code	0	-

Scenario 4 – QR Get Wet from Water (Test 10 times)

Table 6.4 Result of System Testing (4)

		Accuracy	Connectivity
Little Water	Valid QR Code	10	0.21
	Invalid QR Code	10	0.24
More Water	Valid QR Code	10	0.31
	Invalid QR Code	10	0.28

Scenario 5 – Flaws on QR (Test 10 times)

Table 6.5 Result of System Testing (5)

	Accuracy	Connectivity
Valid QR Code	6	0.26
Invalid QR Code	4	0.29

CHAPTER 6

Results above show that with the presence of light (normal or little), the accuracy is 100%, instead, when there is no light in the environment, the QR is not able to detect hence the accuracy is 0%. None of the QR codes is able to detect when it is no light, this is because of the limitation of the Pi Camera which is missing the night vision function. When there are flaws in the QR code, the accuracy is 50%, as it is depending on the condition of the flaw, like figure 6.1 and figure 6.2 shown. The connectivity results also state that the system needs around 0.2 to 0.3 seconds to verify the QR code.

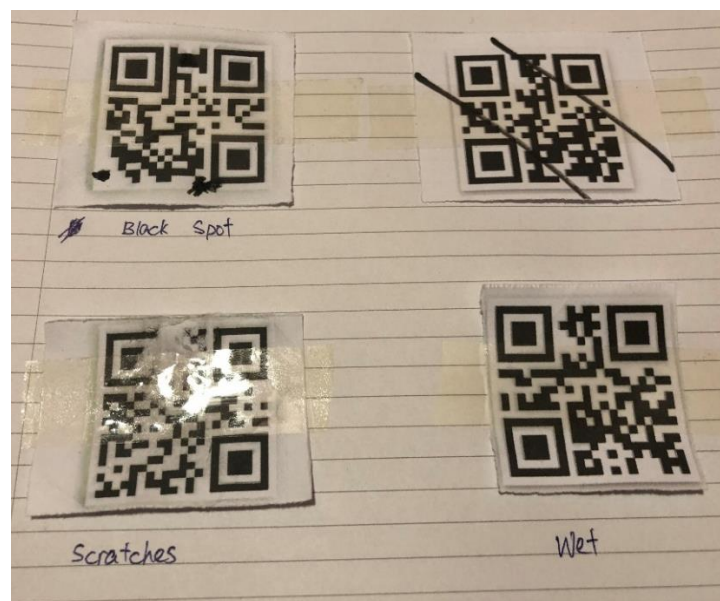


Figure 6.1 Flaws on QR Code (1)



Figure 6.2 Flaws on QR Code (2)

6.2.2 Testing of the Servo

There are two testings done for the servo motor, which are the system control after QR verification and receiver control using the webpage. The accuracy test will be testing on the accuracy of the servo based on the result of QR verification or receiver's control. As for the connectivity test, it will be testing on the time taken for the servo to function after the control signal is given.

Scenario 1 – System Control (Test 10 times)

Table 6.6 Result of System Testing (6)

	Accuracy	Connectivity
Valid QR Code	10	0.64
Invalid QR Code	10	0.79

Scenario 2 – Receiver Control (Test 10 times)

Table 6.7 Result of System Testing (7)

	Accuracy	Connectivity
Unlock	10	1.04
Lock	10	0.96

The results above state that, the servo motor has 100% accuracy and an average of 0.64 to 0.79 seconds of response time after the QR verification. While for the receiver control scenario, it has 100% accuracy and an average of 0.96 to 1.04 second of response time after receiver click on the button on the webpage. The reason that connectivity of the servo in scenario 2 (receiver control) is longer than in scenario 1 (system control) is that when the receiver is controlling from the website, the signal needs to travel through the Internet to reach the Raspberry Pi.

6.2.3 Testing of the Database

There will only be accuracy testing for the database as the time taken for the database to update the data is unable to obtain.

Table 6.8 Result of System Testing (8)

	Accuracy
Received Parcel	10
Abnormal Activity	10
Insert New Tracking Number	10

The accuracy testing for the database is 100% as all the information updated or insert is correct as it is meant to be.

6.2.4 Testing of the Telegram Bot

For the Telegram bot testing, the accuracy test will be based on the content of the message, either is notification for received parcel or alert for abnormal activity. Then, the connectivity is to test on the delay (in seconds) of the message received after the system has sent the message.

Table 6.9 Result of System Testing (9)

	Accuracy	Connectivity (seconds)
Valid QR Code (Received Parcel)	10	6
Invalid QR Code (Abnormal Activity)	10	5

The results show that the Telegram bot has an accuracy of 100% and an average of 5.5 seconds delay for the receiver to receive the message.

6.3 Project Challenges

The challenge of project implementation is the limited resources. Since the project is developed by one person. That person needs to play the roles of a system developer and a client or user. This has limited the feedback and opinion on developing the project because the developer was developing the system and testing by himself, all the feedback he can get is very subjective. Therefore, the finalised system might not be able to fit the user requirement in the market.

In addition, there are some difficulties in the setup of the hardware and software environment. The Raspberry Pi model 1 A+ only has 512MB of RAM, which make it need a longer time to start the system and run the program.

6.4 Objectives Evaluation

The project objectives to achieve are:

- a. To develop a prototype smart parcel box using Raspberry Pi.

This objective was achieved, and the system prototype that fully functions was built and is controlled by the Raspberry Pi.

- b. To implement a verification process to unlock the smart parcel box.

This objective had achieved with the function of the QR verification process. The SG90 Servo is using to lock or unlock the smart parcel box after the system had verified the detected QR code from the Raspberry Pi camera module. The results of system testing also proved that the system has 100% of accuracy to verify and unlock the parcel box.

- c. To implement a module informing the user that the parcel is received.

This objective is also achieved. The Telegram bot had created for the purpose of sending a notification to receiver after the parcel is received. The result of system testing for the Telegram bot can prove that the accuracy and connectivity of the notification module can function well.

6.5 Concluding Remark

This chapter discussed on system testing and performance metrics. The testing on the Raspberry Pi camera module, servo motor, database, and Telegram bot is carried out. The testing setup and result are presented in this chapter as well. This chapter also included with the discussion on the challenges of developing this project. Finally, the evaluation on the objective is also discussed in this chapter.

CHAPTER 7

Conclusion

7.1 Conclusion

Smart Parcel Box is a parcel box that will help parcel recipients to receive and keep the parcel safe inside the box. It has the function to check and verify the incoming parcels that belong to the recipient, and the front door will only be opened once the parcel is verified. The entire process of parcel receiving is run automatically, it does not require any help or action from a human.

The aim of the project is to solve the problem encountered by the parcel recipient and deliveryman in the current delivery process. The issue with the current delivery process is the waste of time for both recipient and deliveryman. For example, the parcel recipient must wait at home to receive the parcel, but what if they had something important that they need to go out. Besides, the deliveryman must wait for the recipient to answer the door to deliver the parcel to the recipient, but what if there is no one in the house to receive the parcel. Moreover, this project encouraged the practice of zero-contact delivery to avoid any direct contact between humans to prevent the spread of the covid-19 virus.

This project has proposed a solution by letting the smart parcel box to play the role of a recipient to receive the parcel. Since the smart parcel box is installed outside of the house, therefore the deliveryman can just interact with the smart parcel box to deliver the parcel. It solves the problem of the absence of the recipient in the delivery process so that the recipient and the deliveryman do not need to wait for each other to complete the delivery process.

The smart parcel box is not just a time-saving solution to the current issue in the delivery process. In fact, it also encourages the practice of the zero-contact delivery process

especially when the Covid-19 pandemic is still getting serious in the world. Because any interaction between humans will have a chance of catching the virus of Covid-19, therefore zero contact delivery will also keep the recipient and the deliveryman from the Covid-19 virus.

7.2 Future Work

The finalised project is not yet perfect from the developer's point of view, there are some features that still can be added to the system. First off is the camera live stream function, receiver could see the live stream from the Raspberry Pi camera module through the website, which will be hosted by the Raspberry Pi. With this function, the receiver can make sure that controlling the smart parcel box remotely is necessary by observing what is really happening at that time. For example, the receiver can ask the deliveryman to place the parcel in front of the camera if the QR code is unable to detect by the system.

Next, from the result of the testing for the Raspberry Pi camera module, it shows that the QR code is unable to detect in the environment of the absence of light. Hence, the system could upgrade the camera module to a night vision camera module. The other solution to this problem is to install a LED (Light-Emitted Diode) beside the camera module to provide some light to scan the QR code since the result had shown that even a little light can make sure the 100% accuracy of the QR verification.

Moreover, SG90 Micro Servo using in this project is not the best solution to lock the smart parcel box. A solenoid door lock could be replacing the SG90 Micro Servo because the solenoid door lock is easier to function as it only needs to unlock, to lock the smart parcel box, a person just needs to close the door and the solenoid will automatically lock. Unlike the SG90 Micro Servo need to wait for the system control to lock the door.

BIBLIOGRAPHY

- [1] “Top Reasons Why People Shop Online || Axis eCommerce Enabler - eCommerce Platform Solution Provider | #1 Malaysia,” *Axis*.
<https://axisnet.asia/e-commerce/why-people-purchase-products-online/>.
- [2] L. Ecola, H. Lu, and C. Rohr, *How Is COVID-19 Changing Americans’ Online Shopping Habits?* RAND Corporation, 2020.
- [3] Cambridge Dictionary, “mailbox,” @*CambridgeWords*, Mar. 23, 2022.
<https://dictionary.cambridge.org/dictionary/english/mailbox> (accessed Mar. 28, 2022).
- [4] “How to Prevent Package Theft,” *Parcel Pending*, Sep. 25, 2017.
<https://www.parcelpending.com/blog/top-5-scary-facts-package-theft-prevent/> (accessed Mar. 28, 2022).
- [5] A. Yeoh, “Not home? Get your parcels delivered to a locker or cafe near you | The Star,” *The Star*, Oct. 04, 2018. <https://www.thestar.com.my/tech/tech-news/2018/10/04/check-out-these-local-parcel-delivery-services>.
- [6] Gareth Halfacree, *The official Raspberry Pi beginner’s guide : how to use your new computer*. Cambridge: Raspberry Pi Trading Ltd, 2020.
- [7] Arduino, “Arduino - Introduction,” *Arduino.cc*, 2019.
<https://www.arduino.cc/en/guide/introduction>.
- [8] “Raspberry PI vs Arduino | Top 10 Useful Differences You Need To Learn,” *EDUCBA*, Aug. 16, 2018. <https://www.educba.com/raspberry-pi-vs-arduino/>.
- [9] “Arduino vs Raspberry Pi: Which Board is Best for Your Project,” *The CoderPedia*, Nov. 07, 2020. <https://www.thecoderpedia.com/blog/arduino-vs-raspberry-pi/> (accessed Mar. 28, 2022).
- [10] “What Information Does a BarCode Hold, BarCode Lookup,” *ByteScout*, Jun. 20, 2020. <https://bytescout.com/blog/barcode-information.html#3> (accessed Mar. 28, 2022).
- [11] “Guide to Barcodes vs. QR Codes: Analysis and Comparison - MPC,” *Metalphoto Of Cincinnati*, Aug. 24, 2017.
<https://www.mpofcinci.com/blog/barcode-vs-qr-code/>.

BIBLIOGRAPHY

- [12] “QR Codes Vs. Barcodes: Which is best for asset tracking and inventory?,” *itemit*, Nov. 26, 2019. <https://itemit.com/qr-codes-vs-barcodes-which-is-best/> (accessed Mar. 28, 2022).
- [13] “Survey shows that 37% of U.S. consumers don’t trust crowdsourced, peer-to-peer delivery services,” *PRWeb*.
<https://www.prweb.com/releases/2017/03/prweb14168655.htm> (accessed Mar. 28, 2022).
- [14] R. Panko, “Package Theft Statistics: How to Prevent Lost or Stolen Packages | Clutch.co,” *clutch.co*, 2019. <https://clutch.co/logistics/resources/package-theft-statistics-prevent-lost-stolen-packages>.
- [15] “2020 Package Theft Statistics Report,” *C+R Research*.
<https://www.crresearch.com/blog/2020-package-theft-statistics-report>.
- [16] T. Nonthaputha, M. Kumngern, J. Phookwantong, and S. Keawwang, “Arduino Based Smart Box for Receiving Parcel Posts,” *2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE)*, Nov. 2020, doi: 10.1109/ictke50349.2020.9289909.
- [17] S. S. Kumar, D. Hemanth, S. Dwneeth, K. Dilip, and A. Divyatej, “Automated Package Delivery Accepting System - Smart Freight Box,” *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, May 2019, doi: 10.1109/rteict46194.2019.9016912.
- [18] “SDLC- Waterfall Model,” *The Tech Platform*, Feb. 18, 2021.
<https://www.thetechplatform.com/post/sdlc-waterfall-model> (accessed Apr. 14, 2022).
- [19] Rajkumar, “Spiral Model in Software Development Life Cycle,” *Software Testing Material*, Apr. 11, 2016.
<https://www.softwaretestingmaterial.com/spiral-model-in-sdlc/>
- [20] “SDLC Prototype Model,” *www.w3schools.in*.
<https://www.w3schools.in/sdlc/prototype-model> (accessed Apr. 14, 2022).
- [21] “Introduction to software development life cycle (SDLC),” *www.subjectcoach.com*.
<https://www.subjectcoach.com/tutorials/detail/contents/introduction-to-software-development-life-cycle-sdlc/chapter/prototype-model-of-sdlc>

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Year 3 Sem 1	Study week no.: 2
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Identify problem statement and project objective.

2. WORK TO BE DONE

- Install Raspberry Pi OS in the virtual machine.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- Looking forward to working on the project.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Year 3 Sem 1	Study week no.: 4
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Successfully installed Raspberry Pi OS and updated the environment.

2. WORK TO BE DONE

- Write a python script to generate QR code
- Write a python script to read data from QR code

3. PROBLEMS ENCOUNTERED

- The update and upgrade of the Raspberry Pi OS take a longer time due to the connectivity of the Internet.

4. SELF EVALUATION OF THE PROGRESS

- Learning some basic operations and commands in Raspberry Pi OS.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Year 3 Sem 1	Study week no.: 6
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Python script of QR generator and QR reader had written.

2. WORK TO BE DONE

- Configuring WAMP Server and create a database

3. PROBLEMS ENCOUNTERED

- As a beginner in Python, writing a script takes some time to familiarize with the Python language.

4. SELF EVALUATION OF THE PROGRESS

- Learning Python language, and some python library to work with QR code like OpenCV.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Year 3 Sem 1	Study week no.: 8
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Configured WAMP Server
- Create a database to store parcel tracking numbers.

2. WORK TO BE DONE

- Build a simple website for user login and insert new parcel tracking numbers using PHP.

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- Learning basic configuration of WAMP Server and MySQL.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Year 3 Sem 1	Study week no.: 10
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- A simple website for user login and insert new parcel tracking numbers are built using PHP.

2. WORK TO BE DONE

- Buy the hardware and build the prototype.

3. PROBLEMS ENCOUNTERED

- The website is too simple and need to enhance with appropriate CSS or bootstrap.

4. SELF EVALUATION OF THE PROGRESS

- Overall, the progress of the project is as scheduled.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Year 3 Sem 1	Study week no.: 12
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- The prototype of the smart parcel box is built and functional.

2. WORK TO BE DONE

- None

3. PROBLEMS ENCOUNTERED

- The hardware is more complicated to put together than I have imagined.

4. SELF EVALUATION OF THE PROGRESS

- Although the hardware is complicated and harder to put together, but I managed to build a functional prototype.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	Study week no.: 2
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Modify the python script from FYP1.
- Planning and designing on the functionality of the whole system.

2. WORK TO BE DONE

- Configure Telegram Bot for the notification module.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- Trying to pick up from the work of FYP1 and able to familiar with the system easily.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	Study week no.: 4
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Successfully configured the Telegram Bot notification module.

2. WORK TO BE DONE

- Plan and design on the website.

3. PROBLEMS ENCOUNTERED

- None.

4. SELF EVALUATION OF THE PROGRESS

- The configuring of the Telegram Bot takes some time to complete, but integrating the Telegram Bot into the python script is easy and working fine.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	Study week no.: 6
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Wireframed the layout of website.
- 50% of the development of website.

2. WORK TO BE DONE

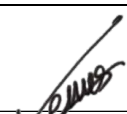
- Complete the development of the website.

3. PROBLEMS ENCOUNTERED


- The design process of the website takes longer time as it involved with UI and UX design.

4. SELF EVALUATION OF THE PROGRESS

- Designing of the website require the knowledge in HTML, CSS, and Bootstrap. I need to take some time to study the knowledges.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	Study week no.: 8
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Complete the development of the website.
- System prototype.

2. WORK TO BE DONE

- Testing on the system prototype.

3. PROBLEMS ENCOUNTERED

- Prototyping the system process is very slow as the Raspberry Pi has low memory to operate efficiently.

4. SELF EVALUATION OF THE PROGRESS

- Project progress is as scheduled, despite the system prototyping takes longer time.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	Study week no.: 10
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- System prototype.
- System testing.

2. WORK TO BE DONE

- Finalize system prototype.
- Finalize report.

3. PROBLEMS ENCOUNTERED

- System testing taking longer time as the Raspberry Pi has low memory to operate efficiently.

4. SELF EVALUATION OF THE PROGRESS

- Progress run very smoothly, everything goes as planned. The prototype is working fine.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	Study week no.: 12
Student Name & ID: Lee Jia Heng 18ACB02515	
Supervisor: Ms Tan Lyk Yin	
Project Title: A PROTOTYPE OF SMART PARCEL BOX	

1. WORK DONE

- Finalize system prototype.
- Finalize report.

2. WORK TO BE DONE


- Presentation.

3. PROBLEMS ENCOUNTERED

- No issue on the finalization of project.

4. SELF EVALUATION OF THE PROGRESS

- The project completed as scheduled.




Supervisor's signature



Student's signature

POSTER



UNIVERSITI TUNKU ABDUL RAHMAN

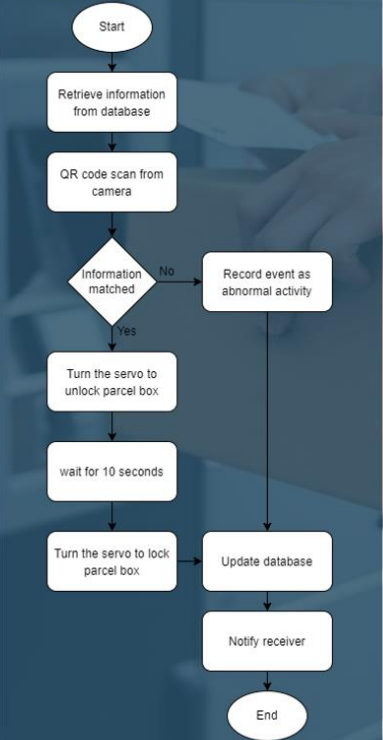
A Prototype of Smart Parcel Box

Developer: Lee Jia Heng
Supervisor: Ms. Tan Lyk Yin

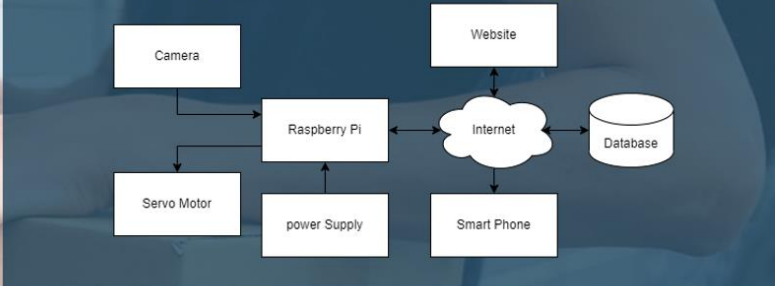
Introduction

With the number of online shoppers increasing, delivery service is getting important. Especially the guarantee of receiving the correct parcels safely is one of the main targets that advertised by the delivery service companies. Hence, the proposed smart parcel box can assist in increasing the security, reducing the delivery delay, and reducing the delivery cost during the delivery process.

System Flow Design



System Architecture Design



Project Objective

The aim of the project is to introduce an alternate way for people to receive their parcel. The three main objectives of this project including:

- To design a prototype smart parcel box to receive a parcel using Raspberry Pi.
- To implement a verification process to unlock the smart parcel box.
- To implement a module informing the user that the parcel is received.

Conclusion

Smart Parcel Box will help receiver to receive and keep the parcel safe inside the parcel box. It has the function to verify the incoming parcel and the front door will only be opened once the parcel is verified. A notification will send to receiver as well.

The proposed system also come with a website that allow receiver to monitor and keep track of their parcel. Functionality of website include:

- Track incoming parcel
- Acknowledge received parcel
- Add new tracking number
- View history record
- Control parcel box remotely

PLAGIARISM CHECK RESULT

PLAGIARISM CHECK RESULT

Turnitin Originality Report

Processed on: 15-Apr-2022 16:12 +08
ID: 1808919584
Word Count: 13747
Submitted: 2

FYP2_turnitin By Jia Heng LEE

Similarity by Source	
Similarity Index	
Internet Sources:	5%
Publications:	1%
Student Papers:	7%

exclude quoted exclude bibliography exclude small matches	mode: <input type="text" value="quickview (classic) report"/>	<input type="button" value="Change mode"/>	print
download			
1% match (Internet from 20-Mar-2022) http://eprints.utar.edu.my			
<1% match (Internet from 15-Oct-2021) http://eprints.utar.edu.my			
<1% match (Internet from 22-Jul-2020) http://eprints.utar.edu.my			
<1% match (Internet from 28-Nov-2020) http://eprints.utar.edu.my			
<1% match (Internet from 27-Jul-2021) http://eprints.utar.edu.my			
<1% match (Internet from 27-Jul-2021) http://eprints.utar.edu.my			
<1% match (Internet from 14-Jan-2022) http://eprints.utar.edu.my			
<1% match (Internet from 15-Apr-2020) http://eprints.utar.edu.my			

PLAGIARISM CHECK RESULT

Match Overview		
10%		
<		>
1	eprints.utar.edu.my Internet Source	3% >
2	Submitted to Universiti ... Student Paper	1% >
3	Submitted to Universiti ... Student Paper	1% >
4	au.rs-online.com Internet Source	<1% >
5	Submitted to Laureate ... Student Paper	<1% >
6	Submitted to University... Student Paper	<1% >
7	Submitted to Multimed... Student Paper	<1% >
8	Thanat Nonthaputha, ... Publication	<1% >
9	Submitted to National ... Student Paper	<1% >
10	Submitted to Asia Paci... Student Paper	<1% >
11	Submitted to Universiti ... Student Paper	<1% >
12	www.educba.com Internet Source	<1% >
13	Submitted to Institut...	<1% >

PLAGIARISM CHECK RESULT

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LEE JIA HENG
ID Number(s)	18ACB02515
Programme / Course	BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMMUNICATIONS AND NETWORKING
Title of Final Year Project	A PROTOTYPE OF SMART PARCEL BOX

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)
Overall similarity index: <u>10</u> % Similarity by source Internet Sources: <u>5</u> % Publications: <u>1</u> % Student Papers: <u>7</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required, and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



 Signature of Supervisor

 Name: TAN LYK YIN

 Date: 20 APRIL 2022

 Signature of Co-Supervisor

 Name: _____

 Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB02515
Student Name	Lee Jia Heng
Supervisor Name	Ms. Tan Lyk Yin

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

I, the author, have checked and confirmed all the items listed in the table are included in my report.



 (LEE JIA HENG)

Date: 20 April 2022