

**RUBBISH MONITORING SYSTEM
(R-MS)**

NABILA BINTI MOHD NIZAM

**BACHELOR OF COMPUTER SCIENCE
(COMPUTER SYSTEMS & NETWORKING)
WITH HONOURS**

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : NABILA BINTI MOHD NIZAM

Date of Birth :

Title : RUBBISH MONITORING SYSTEM (R-MS)

Academic Session : 2022/2023

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

NABILA BINTI MOHD NIZAM

Date: 20 JANUARY 2023



(Supervisor's Signature)

WAN ISNI SOFIAH WAN DIN

Name of Supervisor

Date: 4/07/2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name NABILA BINTI MOHD NIZAM
Thesis Title RUBBISH MONITORING SYSTEM (R-MS)

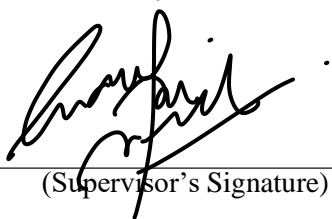
Reasons (i)

 (ii)

 (iii)

Thank you.

Yours faithfully,



(Supervisor's Signature)

Date: 04/07/2023

Stamp: DR. WAN ISMI SOFIAH BINTI WAN DIN
SENIOR LECTURER
FACULTY OF COMPUTING
COLLEGE OF COMPUTING & APPLIED SCIENCES
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARÜL MAKMUR
TEL : 09-424 4726 FAX : 09-424 4666

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (computer System and Networking).

A handwritten signature in black ink, appearing to read 'wan isni', is written over a horizontal line.

(Supervisor's Signature)

Full Name : DR WAN ISNI SOFIAH BINTI WAN DIN

Position : LECTURER

Date : 04/07/2023

(Co-supervisor's Signature)

Full Name :

Position :

Date :



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : NABILA BINTI MOHD NIZAM

ID Number : CA20009

Date : 27 JUNE 2023

RUBBISH MONITORING SYSTEM
(R-MS)

NABILA BINTI MOHD NIZAM

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Computer Science (Computer System & Networking) With Honours

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

“In the Name of ALLAH, Most Gracious, Most Merciful and big Gratitude to Prophet Muhammad SAW”

Thanks to Allah, praise and thanks to His Almighty and His greatest blessings, I was able to complete the project on time. First of all, I would like to express my thanks to my supervisor, Dr. Wan Isnii Sofiah binti Wan Din, for her guidance, knowledge, ideas, and patience as well as for her motivated advice, suggestions for improvement, time commitment, and moral support as I struggled to complete this project.

A particular thanks also goes out to my parents, my entire family, and especially my mother Zunaidah binti Ibrahim and my father Mohd Nizam bin Mustafa for their understanding, emotional support, financial support, and always prayers for my success in pursuing this course of study.

Last but not least, I want to express my gratitude to all of my professors and friends who participated in discussions, shared knowledge and ideas, brainstormed together, and helped and supported me while I completed each assignment for this project.

May Allah S.W.T bless all of you.

Thank you.

ABSTRAK

Sistem pemantauan sampah pintar ialah projek berasaskan IoT yang boleh mengukur tahap sampah bagi setiap tong. Dengan gabungan beberapa peranti perkakasan seperti Raspberry Pi dan penderia ultrasonik ia dapat mengukuhkan lagi sistem dalam projek ini. Tujuan utama menggunakan Raspberry Pi ialah ia sangat mudah digunakan dan sambungan Wi-Fi adalah berterusan. Sebab utama sistem pemantauan sampah pintar diwujudkan adalah kerana demi menjaga kebersihan sekitar fakulti dan kebajikan pengguna di fakulti yang menggunakan tong sampah untuk membuang sampah. Dengan penciptaan sistem pemantauan sampah pintar ini, ia boleh membantu dan memudahkan pemungut sampah untuk mengutip sampah secara teratur dan tepat pada masanya. Masalah kebiasaan yang dihadapi ialah jumlah sampah yang terlalu banyak menyebabkan pencemaran bau berlaku di kawasan tong sampah. Oleh itu, matlamat yang ingin dicapai adalah untuk mewujudkan sistem pemantauan sampah yang boleh menguruskan sampah dengan lebih teratur dan berkesan. Penderia ultrasonik akan diletakkan di bahagian atas tong sampah untuk mengesan isipadu sampah. Sekiranya sampah di dalam tong sampah penuh, maka pemberitahuan akan dihantar kepada pengguna melalui aplikasi Blynk. Bahasa yang digunakan ialah C++. Metodologi dalam prototaip adalah kaedah yang mesra pengguna. Oleh itu, pengwujudan projek ini dapat membantu pihak fakulti mengekalkan persekitaran yang bersih dan menarik serta dapat memberi keselesaan kepada pelajar dan pekerja.

ABSTRACT

The smart rubbish monitoring system is an IoT-based project that can measure the level of rubbish for each bin. With the combination of several hardware devices such as Raspberry Pi and ultrasonic sensors it can further strengthen the system in this project. The main purpose of using the Raspberry Pi is that it is very easy to use and the Wi-Fi connection is continuous. The main reason of smart rubbish monitoring system created is because for the sake of maintaining cleanliness around the faculty and the welfare of users in faculty that use rubbish bins to dispose of rubbish. With the creation of this smart rubbish monitoring system, it can approximately help and facilitate the rubbish pickers to pick up the rubbish in an organized and timely manner. The habitual problem faced is that the amount of garbage that is too much has caused the smell pollution to occur in the rubbish bin area. Therefore, the goal to be achieved is to create a rubbish monitoring system that can manage rubbish in a more organized and effective manner. Ultrasonic sensor will be placed on the top of the rubbish bin to detect the volume of rubbish. If the rubbish in the bin is full, then a notification will be sent to the user via the Blynk application. The language used is C++. The methodology in the prototype is a user-friendly method. Therefore, the creation of this project can help the faculty to maintain a clean and attractive environment and can provide comfort to students and workers.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1 INTRODUCTION	1
1.1 Project Background	1
1.2 Problem Statement	4
1.3 Goal and Objective	5
1.3.1 Goal	5
1.3.2 Objective	5
1.4 Scope	5
1.5 Significance	6
1.6 Thesis Organization	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Existing Work	7

2.2.1	Smart Garbage Monitoring System	7
2.2.2	Internet of Things based Wireless Garbage Monitoring System	12
2.2.3	IoT Based Dustbin Monitoring With Dumpster Alert System	16
2.3	Comparison of Existing System	19
2.3.1	Relevance of Comparison With Project Title	21
2.4	System Development Methodology	21
2.4.1	Waterfall Development Methodology	21
2.4.2	Agile Development Methodology	23
2.4.3	Rapid Application Development Methodology	24
2.5	Comparison of Methodology	25
2.6	Summary	27
CHAPTER 3 METHODOLOGY		29
3.1	Introduction	29
3.2	Model For Proposed Project	29
3.2.1	Requirement Phase	30
3.2.2	Design Phase	30
3.2.3	Development Phase	31
3.2.4	Testing Phase	31
3.2.5	Deployment Phase	31
3.2.6	Review Phase	31
3.3	User Requirement	33
3.3.1	Functional Requirement	33
3.3.2	Non-functional Requirement	33
3.3.3	Limitations	34
3.3.4	Constrains	34

3.4	Project Overview	35
3.5	Design	37
3.5.1	Use Case Diagram	38
3.5.2	Flowchart	39
3.5.3	Context Diagram	43
3.5.4	R-MS System Application Storyboard	44
3.5.5	R-MS Physical Design Project	47
3.5.6	Design Of The Rubbish Monitoring System (R-MS)	48
3.5.7	Hardware And Software Specification	48
3.5.8	Hardware Requirement	50
3.5.8.1	Raspberry Pi	50
3.5.8.1	Ultrasonic Sensor	51
3.5.9	Software Requirement	52
3.6	Gantt Chart	53
3.7	Proposed Solution	54
3.7.1	Prototype	54
3.8	Testing Plan	56
3.9	Summary	56
	CHAPTER 4 RESULTS AND DISCUSSION	55
4.1	Introduction	55
4.2	Development Tools	55
4.3	Implementation Requirements	55
4.3.1	Implementation of Database	56
4.3.2	Implementation of Hardware	57
4.4	Project Testing	59
4.4.1	Development of Mobile Application	59

4.5	Conclusion	55
4.6	Summary	61
CHAPTER 5 CONCLUSION AND RECOMMENDATION		62
5.1	Introduction	62
5.2	Limitation of The Project	62
5.3	Recommendation	62
5.4	Conclusion	63
REFERENCES		57
APPENDIX A GANTT CHART		65
APPENDIX B CODING		67
APPENDIX C PROTOTYPE RUBBISH MONITORING SYSTEM		75
APPENDIX D DATASET RUBBISH MONITORING SYSTEM		76

LIST OF TABLES

Table 1.1	Problem Statement of Rubbish Monitoring System	4
Table 2.1	Comparison Of The Projects	20
Table 2.2	Comparison Of Methodology	26
Table 3.1	Proposed Research Framework	35
Table 3.2	List of Hardware	49
Table 3.3	List Of Software	52
Table 4.1	Tools used to develop R-MS	55

LIST OF FIGURES

Figure 1.1	Rubbish bins in FKOM	1
Figure 1.2	Faculty of computer	2
Figure 2.1	Diagram of connection of the components	8
Figure 2.2	System architecture	9
Figure 2.3	Webpage men hostel block	10
Figure 2.4	Webpage women hostel block	11
Figure 2.5	Result the garbage bins	11
Figure 2.6	Hardware based on Arduino	13
Figure 2.7	System block diagram	13
Figure 2.8	View of garbage monitoring	15
Figure 2.9	View of monitoring result	15
Figure 2.10	Waste collection system	18
Figure 2.11	Phase of waterfall model	22
Figure 2.12	Phase of agile model	24
Figure 2.13	Phase of RAD model	25
Figure 3.1	Agile model phase	30
Figure 3.2	Use case diagram	38
Figure 3.3	Flowchart of the R-MS	40
Figure 3.4	Flowchart of the R-MS login system	41
Figure 3.5	Flowchart of the system application	42
Figure 3.6	Context diagram of the R-MS	43
Figure 3.7	R-MS application system design	46
Figure 3.8	Physical design of rubbish monitoring system	47
Figure 3.9	Design of rubbish monitoring system	48
Figure 3.10	Raspberry Pi	51
Figure 3.11	Ultrasonic sensor	51
Figure 3.12	Project Gantt chart	53
Figure 3.13	Front view of rubbish bin	54
Figure 3.14	Side view of rubbish bin	55
Figure 3.15	Back view of rubbish bin	55
Figure 4.1	Blynk website	56
Figure 4.2	RMS website page in Blynk	57
Figure 4.3	Blynk dataset for RMS	57

Figure 4.4	Rubbish Monitoring connected circuit	58
Figure 4.5	Output measure of ultrasonic sensor	59
Figure 4.6	Blynk website interface	60
Figure 4.7	Widgets for interface	60

LIST OF ABBREVIATIONS

Wi-Fi	Wireless Fidelity
AC	Alternating current
DC	Direct current
IoT	Internet of Things
UV	Ultraviolet
HC-SR04	Ultrasonic distance sensor
IDE	Integrated development environment
TCP/IP	Transmission control protocol/Internet protocol
ARM	Advanced RISC machine
GSM	Global system for mobile communication
RFID	Radio-frequency identification
LCD	Liquid-crystal display
MQ135	Air quality sensor
MySQL	Structured query language
R-MS	Rubbish monitoring system
App	Application
SDLC	Software development life cycle
RAD	Rapid application development
QA	Quality assurance
IT	Information technology

CHAPTER 1

INTRODUCTION

1.1 Project Background

In this era of globalization, we already know that every day humans will produce a large amount of rubbish (Tarmudi et al., 2009). The rubbish produced is considered a useless material and does not bring any benefit, in fact it can cause harm if stored. The accumulation of too much rubbish in the rubbish bin will cause the arrival of epidemics of dangerous diseases such as dengue fever, leptospirosis, and skin diseases (Fmipa, n.d.). To preserve the environment, the level of rubbish needs to be reduced as much as possible. Therefore, a rubbish monitoring system is needed to perform in this task.



Figure 1.1 Rubbish bins in FKOM

The rubbish industry also needs solutions that make event-driven rubbish collection possible. Rubbish management organizations must stop focusing on outdated rubbish trends and patterns that do not really fit our current way of life. To ensure that rubbish containers are only picked up, when necessary, the whole rubbish sector needs a

system that utilizes real-time information and notifications to alert users. The rubbish monitoring system plays a role in this problem.

The location of this research will be doing at Faculty of Computer (FKOM). The main focus for this project is on the small area as a location for testing plan. It also can be implemented in large areas as well. Because of the fact that the faculty of computer (FKOM) is still employs a manual system in which the rubbish will be collected and picked up in each class and lab each working days by the cleaners. This technique is time-consuming and ineffective since it demands a lot of labor from humans, which raises expenses.



Figure 1.2 Faculty of computer

In Faculty Computer (FKOM) also, each class and lab of the faculty is provided with one rubbish bin in their class and lab. In the existing system, the employees will monitor and collect the rubbish every day except Saturday and Sunday. The employees have to pick up the rubbish to every floor level of the FKOM building.

The rubbish monitoring system project is a cutting-edge system that will support keeping a clean environment around the faculty. The system keeps an eye on the rubbish bins and notifies users of how much rubbish has been collected there. Basically, this project idea is based on the Internet of Things (IoT) and has something to do with smart cities (A & Ranjana, 2019). Since living a sensible lifestyle requires cleanliness, cleaning starts with the rubbish bin.

The rubbish monitoring system also attempts to improve the working environment for rubbish collectors and drivers. By taking care of the bins that need maintenance, rubbish collectors may use their time more efficiently. Rubbish disposal becomes more important when rubbish collectors view their work as productive. In addition, efficient rubbish management reduces illegal dumping and overflowing rubbish, which is necessary to make the workplace of rubbish collectors cleaner and healthier.

To do this, ultrasonic sensors will be installed over the rubbish bin to measure and detect the volume of the rubbish and compare it to the depth of the rubbish bin facing the solid rubbish. As the amount of rubbish increases, so does the distance between the ultrasonic and the rubbish. The Raspberry Pi will receive this real-time data. Our Raspberry Pi then processes the data and sends it to the Wi-Fi receiver to receive notification to empty the bins.

1.2 Problem Statement

One of the problems in Faculty of computer (FKOM) is the quantity of solid waste. Rubbish bins collection is generally performed using ineffective, time-consuming, and labor-intensive methods. This will bother the other students because the smell from the overflowing rubbish bins is unpleasant.

One more problem is the improper management of solid waste at Faculty of Computer (FKOM), where there is a lack of understanding of how to manage rubbish collection on every floor in FKOM. As a result, the faculty continued to use of a conventional system, co-workers are unable to accurately predict how much solid rubbish will be put out for collection each day. As a result, the environment will become unclean, and bad smells will develop that could possibly lead.

Table 1.1 Problem Statement of Rubbish Monitoring System

No.	Problem	Description	Effect
1.	The amount of solid rubbish	Current rubbish collection is inefficient, waste time, and requires a significant amount of human energy.	It will make society uneasy because overflowing bins cause foul odors in the environment.
2.	Mishandling of solid rubbish	In FKOM buildings, there is a lack of knowledge about how to handle rubbish collection on each floor level.	It will produce an unclean environment for the immediate area and a foul smell that can cause the spread of some dangerous diseases.

1.3 Goal and objective

1.3.1 Goal

The project goal of the project is to create a real-time Rubbish Monitoring System for Faculty of computer (FKOM) so that they can manage their rubbish collection more efficiently and effectively.

1.3.2 Objective

The objectives of this project are:

- i. To study the existing system in rubbish monitoring system.
- ii. To develop a rubbish monitoring system using IoT, the main components are Raspberry Pi and ultrasonic sensors.
- iii. To evaluate the rubbish monitoring system by send the notification of the rubbish into a mobile phone via Blynk application, as well as to receive a notification to empty the bins.

1.4 Scope

The scopes of the project are:

- i. To concentrate on the detection of quantity waste in rubbish bins, and the data is transmitted to a mobile phone via Wi-Fi receiver to receive a notification.
- ii. To concentrate on the installation of Raspberry Pi and ultrasonic sensor module.
- iii. To concentrate to the system under test so that it can determine the amount rubbish in the bin and send the information to the user via notification.

1.5 Significance

The significance of this project is:

- i. To make it easier for environmental management to improve the environment of faculty.
- ii. To upgrade the earlier project by including a new functionality.

1.6 Thesis Organization

This thesis for the rubbish monitoring system project has five chapters. The problem statement, it shows how it contributes to the creation of a solution, is discussed in Chapter 1 together with the introduction of the project. Following that will be the project goal and objectives. The scope and thesis organization of this project come last.

The literature review for the project is covered in Chapter 2. The literature review will discuss about the current system project that has been developed. This system involves the introduction of projects, methodologies, and technologies that can be employed in this project. The explanation of the various learning strategies is also included in this chapter.

The methodology that will be involved in developing this project is covered in Chapter 3. The technology or programming that will be utilized to implement this project will also be covered in this chapter. Additionally. The project design was covered.

Results, testing, and implementation are covered in chapter 4. In this chapter will explains about how the system was put into place in accordance with the specifications. It also covers the tests performed on this project to make sure it is efficient and error-free. The test outcome is stated.

Chapter 5 will finally bring the rubbish monitoring system project to a conclusion. This chapter includes an introduction that explains about the work that has been done, lists the challenges encountered, and discusses how the project might be improved in the upcoming.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the discusses and describes selected books, articles, and previous journals that from similar topics that can be referenced for Rubbish Monitoring Systems. This chapter also provides research based on similar existing systems, similar sensors used and IOT boards that can be guidelines or samples for upgrading or developing new systems that are better than existing systems.

2.2 Existing work

Existing system refers to a different project created by others. Regardless of technology, it can be any commercially available products that are connected to this project. It is used to compare the existing product to the one that will be developed in the future. The product benefits and goals are considered in the comparison.

2.2.1 Smart Garbage Monitoring System

This system was proposed by Priya G, Ronit Chaudhuri, Pritthish Chattopadhyay, Sreyam Dasgupta (Chaudhuri et al., n.d.). In this journal article, it discusses the methods used to handle rubbish problems on any school, college, or other location campus. Because as population is increasing, more waste that is generated, which requires the placement of more garbage bins in a given area to maintain a clean and sanitary environment. As we know, it becomes very challenging for the garbage collector workers to constantly check to determine if all the bins are either full or empty. If the garbage bins are overloaded, they need to transfer the garbage into a dump site.

The goal of this project is to keep the place at campus of the VIT University clean and healthy. It probably offers a smart way to use Wi-Fi connections around the campus university to efficiently check the status of all bins, regardless of how many garbage bins that has in that campus. The status of the garbage bins is being shown on a website. Other than, they want to include a payment module that will allow customers to make online payments on as many trips as they make to the smart garbage bin. This will lighten the workload of those who work for public employees and are responsible with collecting payments from customers who regularly dispose of their garbage in the dumpsters.

In figure 2.1, this project use three main components hardware. It is Arduino UNO, an ultrasonic sensor, and a Wi-Fi ESP module. The Arduino board is connected to the laptop, Wi-Fi modem, and ultrasonic sensor. So that, a centralized web server will provide the status of each garbage bins on that campus, including whether it is empty or full. The Arduino UNO is a microcontroller board. The ATmega chip serves as the base for Arduino, an open-source, programmable microcontroller, and software. The Arduino software can be used to programmed the Arduino board. This is uses syntax similar to java and C/C++. It comes with everything need to support the microcontroller. Besides that, Arduino is a computer hardware and software that user creates and produces microcontroller-based kits for digital devices that can sense and control objects in the physical world. The Arduino just need USB cable, AC-to-DC adapter, or battery to power it.

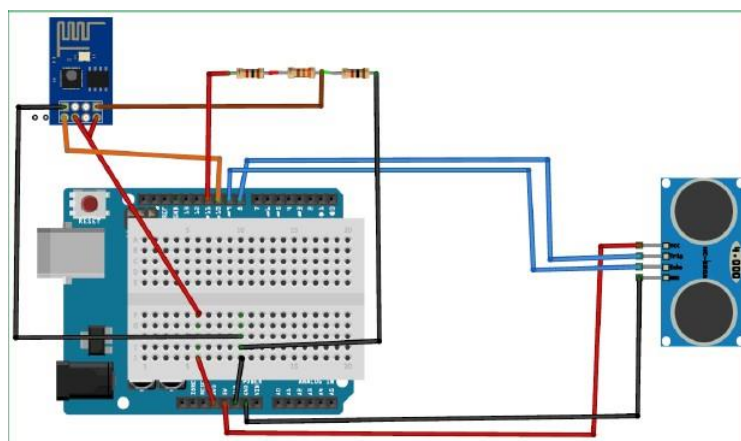


Figure 2.1 Diagram of connection of the components

In addition, the ultrasonic sensor is a device that uses ultrasonic waves to detect a range to an item. An ultrasonic sensor transmits and receives ultrasonic pulse from a

transducer to determine the proximity of an object. For ultrasonic sensors to function, a sound wave above the range of human hearing must be sent out. The sensor transducer server as a microphone for receiving and transmitting ultrasonic sound. Ultrasonic sensors send a pulse and receive an echo using a single transducer. By monitoring the amount of time that passes between delivering and receiving an ultrasonic pulse, the sensor can estimate the distance to a target.

Besides, the function of Wi-Fi ESP module is providing the internet connection project and device. Internet of Things (IoT) projects can be developed with the help of Wi-Fi ESP module. It can direct the project to send data over the internet or transfer data over the internet by using Wi-Fi ESP module. Other than that, by using Wi-Fi ESP module, it can send data over the internet to their project or make it send data over the internet.

After completing the Wi-Fi and Arduino board connections to the laptop and ultrasonic sensor. The Arduino board is used to keep track of the conditions of each garbage bins on the campus. They construct a visually appealing web page for the University VIT so that the administrator can track the condition of every garbage bin on campus and provide an immediate report to the appropriate parties. This system process is shown in figure 2.2.

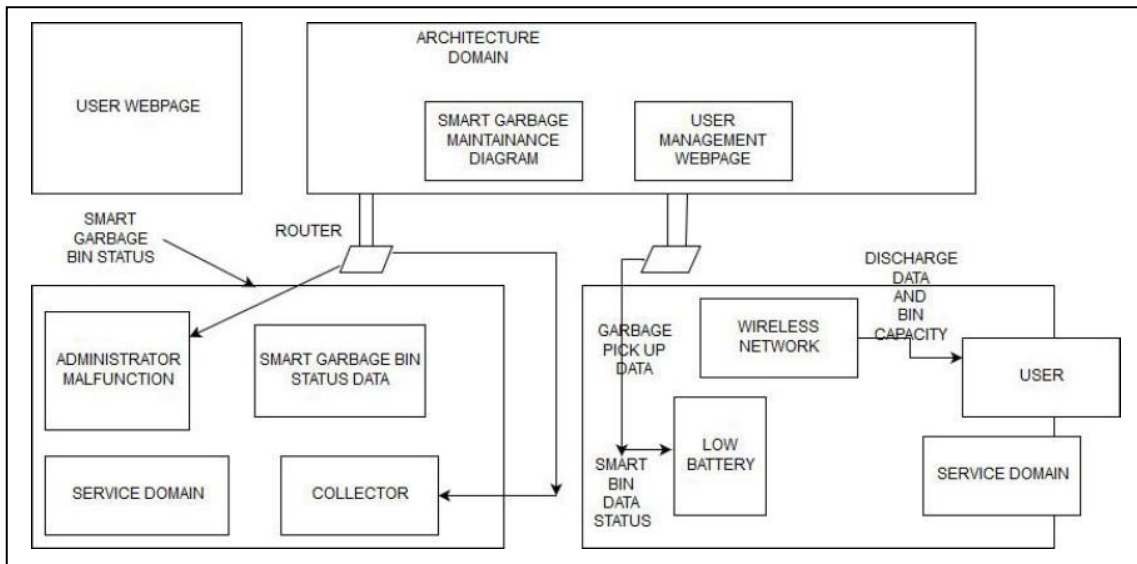


Figure 2.2 System architecture

The Smart Garbage Monitoring System works as follows, if the garbage is less than 5cm empty, the status of the bin will be loaded so otherwise it will be empty. The entire website, which shows the status of each campus building, will be viewable to the

data center. The ultraviolet (UV) sensor is connected to Wi-Fi ESP modules, which display the products. The bin status is displayed as either full or empty, but it is unable to determine how much of bin is actually full. It only displays the extremes of full or empty.

In figure 2.3 shows the application interface displays the webpage demonstrates how to check the garbage bins in the men hostel block. This is expected that each block has a maximum of four garbage bins on each floor. Next, in figure 2.4 it displays the condition of the garbage bins for the various levels and block of the women hostel. In the next figure, it displays the condition of the bins in additional structures. They have assumed there are only four bins on each floor of the buildings throughout the case study. Depending on how many bins are added or remove from the floors, the front end will be updated. In figure 2.5 shows the final image displays of the condition in bins according to whether they are less than 5 cm full or more than 5 cm full when their click the relevant link.

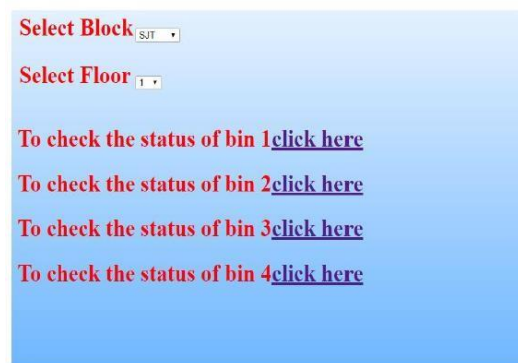


Figure 2.3 Webpage men hostel block

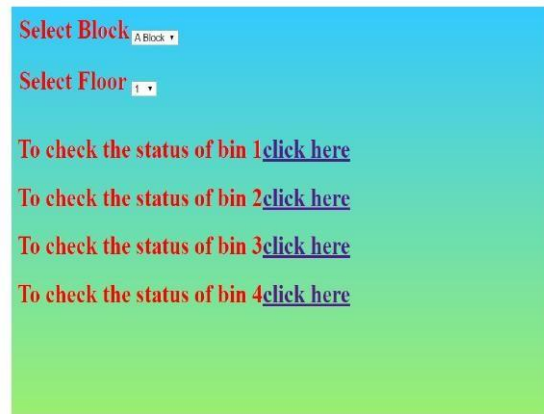


Figure 2.4 Webpage women hostel block



Figure 2.5 Result the garbage bin

Among this monitoring system disadvantages are if the Wi-Fi module is disabled, the device completely ceases functioning. Due to a connectivity issue, loading data into the server can suddenly become challenging. The administrator must have access to a quick and reliable Wi-Fi connection that will allow them to regularly check the status of the garbage bins. Since the data is not yet uploaded to the cloud, despite the best efforts, the administrator cannot verify the status of the bins while it outside the campus.

In conclusion, the creation of Smart Garbage Monitoring System is the one of the good ideas to support the clean and healthy environments. The construct a smart city is the creation of smart garbage bins that incorporate many characteristics like durability, affordability, damage prevention and support concerns. Before being executed on a large scale, innovations like this should be tested in small area. The implementation of payment applications for the monitoring system is the good idea. To make it simpler for the public to access through their mobile phones and detect the closet garbage bin. An android application was created to determine whether the garbage bin is empty or full.

2.2.2 Internet of Things Based Wireless Garbage Monitoring System

This monitoring system was proposed by Yanglu, Asif khan, and Amit Yadav (2019 *Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*., n.d.). In this article it explains about the intelligent waste management that can improve the intensity and efficiency of the garbage monitoring system using the Internet of Things (IoT). The goal for this system is to develop the software system to achieve remote monitoring, that can minimize human intervention and also improve the efficiency of human activities. They also developed a simple prototype of garbage monitoring system to give the platform to build a better design and its implementation.

One of the most important ideas used to create a smart city is waste management based on IoT. The urban designer is the one person who wants achieving intelligent monitoring for the waste management. The implementation of necessary intelligent sensors in pertinent areas is the main problem for the garbage management. They need to make sure that garbage bin project is placed faraway from areas where there may be stolen, vandalism, or accidents. Other than that, the efficient remote monitoring of garbage volume is the main factors to collecting information anytime and anywhere through the use of the internet, including data transmission.

Among the components used in this garbage monitoring system is Arduino, HC-SR04 module, temperature sensor, tilt sensor, and USR-C215. The Arduino acted as the main controller and configured some peripheral components. It is made up of a physical microcontroller, which is a programmable circuit board, as well as an integrated development environment (IDE), a piece of software that runs on a computer and it is used to write and upload the computer coding to the physical board.

Next, the HC-SR04 module is for the detection of the garbage capacity based on the ultrasonic principle, temperature sensor and tilt sensor for the detection as additional function. The tilt sensor is a device that used for measuring the tilt of an object in multiple axes with reference to an absolute level plane. In simple words, it is used to calculate the angles. Last hardware device is USR-C215 is used for the wireless data transmission. It

supports Wi-Fi protocol and TCP/IP protocol. It can realize this device via simple configuration. The figure 2.6 shown about the USR-C215 for wireless data transmission.



Figure 2.6 Hardware based on Arduino

The system is based on the Internet of Things (IoT) of three architecture, which consists of bottom-up sensing, a network and application services. Information gathering is the responsibility of the low-end sensing component, whose control the terminal device is connected to the wireless communication module. The network component is aware of information control and transmission. With the help of the client application, the application service part gathers important data and delivers it to the user. Example of the system flow shown in figure 2.7.



Figure 2.7 System block diagram

The main controller and the sensor usually make up the sensing component. When everything is functioning normally, the status light on the controller is still green while it continuously monitors the ashbin and uses sensors to measure the level of the height and weight. The light becomes red, and the door is programmed to close in order to avoid overthrowing and to wait until sanitation workers remove the waste when the level reaches the intended threshold. In order to verify that task has been completed, workers reset the machine after the controller has sent input to the server.

GSM and Wi-Fi transmission protocol are all options for the master and server share the data. When the garbage bin is full, the following processes will be taken. The master core sends the signals to the server in step 1. For the step 2, the server notifies the sanitation worker of its status. The server also gets the data request from the supervisor in the step 3. In the step 4, after finish cleaning, sanitation workers send respond to the server, and the machine controller then confirms the task completion. Last step is reset the machine to its initial stage.

Further, they are in charge of identifying and emptying out every numbered garbage bin in the vicinity. Supervisors and sanitation workers are among the personnel positions. The supervisor is in charge of keeping an eye on the current situation in the region and overseeing the workers, who respond to alerts and complete the cleaning and verifying tasks. The necessary department or the urban environment monitoring centre can obtain the management data for each region. In a nutshell, different roles have various access privileges to the system.

The main function, registration, and login pages are already created. User in the general and administrative positions made up the system. After successfully logging in, the average user can view all of the current garbage depths in a particular area in the Garbage Monitoring view, which is the default as shown in the figure 2.8. To view the data trend graph of the rubbish amount for almost 24 hours, the status of the garbage bin and its surroundings, including temperature, humidity, garbage depth, tilting or not, on fire or not, as shown in figure 2.9. Just click the choose button to observe.

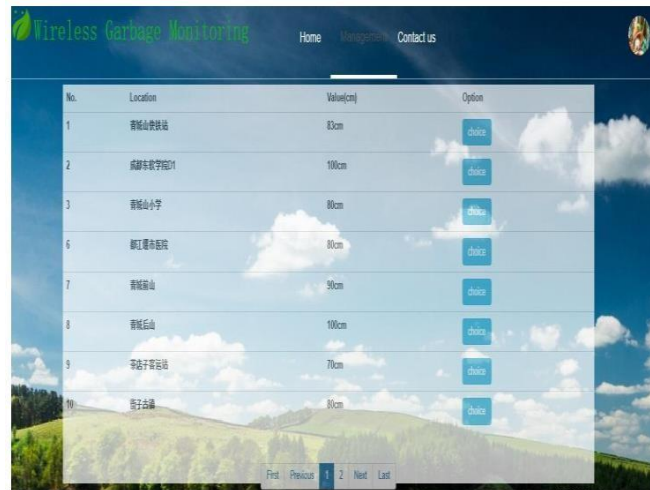


Figure 2.8 View of garbage monitoring



Figure 2.9 View of monitoring result

The method previously employed for garbage collection had several negative consequences on society and the environment. The existing method of managing urban waste collection and management, which primarily relies on a lot of labor and material resources, is not ideal. Whether there is rubbish in the garbage bin or not, fixed sanitation employees are stationed in a small area. The fact that the environment's management, control, and care are fully dependent on human activity causes a great deal of subjective issues. At the same time, harmful phenomena like extended stacking times and even spills frequently occur as a result of inadequate monitoring and a lack of supervisory policies. In addition to impacting the ecosystem, these occurrences make it simple for diseases to spread.

In general, it is also an effective idea to develop the garbage management. The process is more in saving time, effort on management and cost. The system needs to determine the point of the garbage collection. So that, user can easily monitor the status of the garbage anytime and anywhere.

2.2.3 IoT based Dustbin Monitoring with Dumpster Alert System

This article was proposed by Sathishkumar N, Preethi M, Pravinkumar M, and Subash R (Sathishkumar et al., 2022). The abrupt rise in population that has led to an increase in the generation of a lot of waste is one of the issues highlighted in this article. Lack of waste management will have a number of negative repercussions, including odour pollution, the build-up of a lot of waste, and the spread of illness.

The purpose to develop this system is intended to reduce costs and ensure timely waste material collection to avoid the overflow bin. The devised method is very effective at gathering waste, keeping track of the quantity of waste dumped, indicating the smell and poisonous smell produced by the waste, and normalizing the environment.

The hardware that they use in this system is ESP32 Microcontroller, buzzer, RFID reader, LCD display, GSM module, ultrasonic sensor, and MQ135 air quality sensor. The waste collection system uses a load cell with an ultrasonic sensor to monitor the dumpster. An RFID monitoring device will notify the area when the dumpster is full. The issues of waste management can be effectively solved with the help of this technology.

The ESP32 microcontrollers functions are low-cost, low-power microcontroller that were created with Wi-Fi and dual Bluetooth. The ESP32 consists of antenna switch, power management, low-noise amplifier, filters, and power amplifier modules. The ESP32 is a standalone system or as a serial interface to any host microcontroller. It can interact with other devices and offer Wi-Fi and Bluetooth functionality.

A buzzer is an instrument that produces sound by converting audio signals to sound signals. It is frequently used as a sound source in alarm clocks, laptops, printers, and other electronic devices. There are two different categories of buzzer, first is Piezoelectric and second is electromagnetic buzzers. Depending on its design and

intended use, the buzzer can produce a range of sounds, including music, sirens, buzzers, alarms, and electric bells.

An RFID reader is a gadget that gathers information from RFID tags and keeps track of any particular operation. Radio waves are used to convey data from the tag to the reader. It is sufficient to only have the reader's sight and not to directly scan the RFID tags. An RFID tag must be between 3 and 300 feet from an RFID reader in order to be read. With the use of RFID technology, a variety of products can be quickly scanned, and even when an object is surrounded by other products, it can still be quickly identified.

A GSM modem, also known as a GSM module, is a piece of hardware that connects to a distant network using a GSM mobile phone. They are identical to a typical mobile phone in the case of the mobile network, even down to the requirement that the mobile network identify itself to the channel. The majority of the time, they are used in embedded systems.

An ultrasonic sensor is a device that measures a distance to a target using ultrasonic sound waves. An ultrasonic sensor uses a transducer to transmit and receive ultrasonic pulses to provide information about the position of an object. A sound wave louder than human hearing is produced by ultrasonic sensors to carry out their function. The sensor's transducer serves as a microphone, capturing and transmitting ultrasonic sound. The ultrasonic sensors generate a pulse and detect the echo using a single transducer. The sensor calculates the separation from a target by timing the transmission and reception of ultrasonic pulses.

The MQ-135 Gas Sensor can detect all types of gas. The digital pin goes high when the level of this gas in the air exceeds a threshold limit. An on-board potentiometer can be used to set the threshold value. The analog output pin generates an analog voltage that can be used to estimate the concentration of various gases in the atmosphere.

After the project is completed. The first stage will involves letting local know through GSM module where the dumpster may be located. In figure 2.10 shown about the block diagram of the waste collection system. To alert the bin to the vehicle proximity to their home, a fixed alarm buzzer is installed. The buzzer status will be shown on the LCD. In the second stage, a load cell and an ultrasonic sensor are used to check the level of trash in the collecting bin. To track the maximum level of the trash, an ultrasonic sensor

is placed within the bin and a load cell is fixed below the bin. An alert message is issued to the municipal corporation to take prompt action whenever the bin reaches its maximum capacity. An app that was created for the admin and user allows the locals to rate the front-line employee who was in charge of collecting rubbish from that specific region.

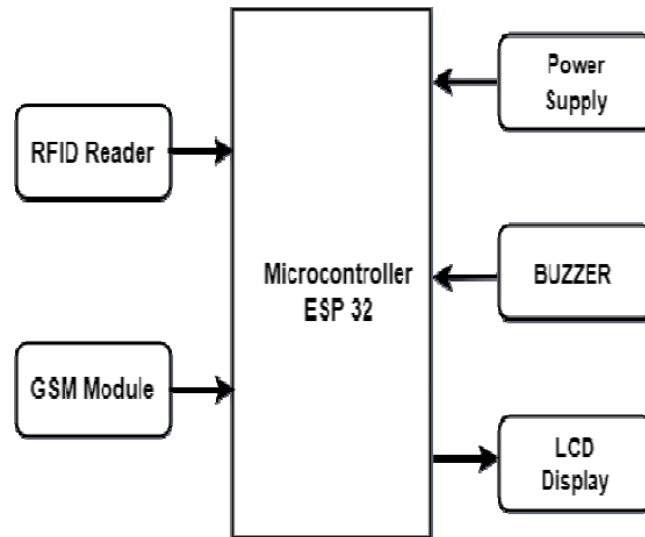


Figure 2.10 Waste collection system

This technique has several advantages, one of which is that it helps to prevent environmental pollution. It is also built to support a town that is always being cleaned. Wet and dry detectors that can be used to distinguish between biodegradable and non-biodegradable garbage when disposing of home garbage are being improved.

In conclusion, this project uses an ultrasonic sensor, a GSM module for mobile communication, RFID tags, and an ESP32 microprocessor to manage waste in any large metropolis. This system is helpful in supplying the right method for waste collection. The city adopts a standardized strategy to clean things up and prevent dumpster overflows in place of the conventional collecting method. The smart waste management system will thereby increase the effectiveness of rubbish collection. It greatly aids in trash management and further cuts down on environmental contamination.

2.3 Comparison of Existing system

The comparison between Smart Garbage Monitoring System, Internet of Things Based Wireless Garbage Monitoring System, and IoT Based Dustbin Monitoring with Dumpster Alert System are listed in the Table 2.1.

Table 2.1 Comparison of The Projects

	Smart Garbage Monitoring System	Internet of Things Based Wireless Garbage Monitoring System	IoT Based Dustbin Monitoring with Dumpster Alert System
Hardware	Arduino UNO, Wi-Fi ESP module, laptop, ATmega chip	Arduino, HC-SR04 module, USR-C215, Wi-Fi ESP module	ESP32 Microcontroller, buzzer, RFID reader, LCD display, GSM module
Database	Java, C/C++	Mysql	Mysql
Sensor	Ultrasonic sensor	Temperature sensor, Tilt sensor	Ultrasonic sensor, MQ135 air quality sensor
Area cover	University campus	Smart city	Town
Operating system	Webpage application	Mobile application	Mobile application
Internet dependency	Yes	Yes	Yes

Advantages	<ul style="list-style-type: none"> - Efficiently to determine whether the bin is filled or empty. - Easy to make online payment. 	<ul style="list-style-type: none"> - Can notify the notification using Mobile application. - Saving time. - Easily monitor the status of the bin anytime and anywhere. 	<ul style="list-style-type: none"> - Help to prevent environmental pollution. - Easily to detect the biodegradable and non-biodegradable garbage.
Goal	<ul style="list-style-type: none"> - Focus on the cleanliness of each bin on the campus. - Added payment module of the user that use the smart bin. - Keep the place at campus university clean and healthy. 	<ul style="list-style-type: none"> - To reduce human involvement. - Increase the effectiveness of human activity - To developed software system for remote monitoring. 	<ul style="list-style-type: none"> - To reduce cost. - Ensure timely waste material collection.

2.3.1 Relevance of comparison with project title

Based on the comparisons that has been made in the Table 2.1. The update and improvement that will be made in Rubbish Monitoring System (R-MS) is to replace the use of Arduino with Raspberry Pi. This is because the Rubbish Monitoring System (R-MS) project need to have an internet connection. Raspberry Pi already has an internet connection, Bluetooth, and ethernet. Here, cost saving in buying extra sensor can be reduced because in the Arduino itself, it is necessary to add a Wi-Fi module if want to have the internet connection in the project. In addition, Raspberry Pi can use the commonly used python language and can use C/C++ language in the project. Meanwhile, Arduino uses the Arduino language and C/C++. Apart from that, the Rubbish Monitoring System (R-MS) will apply the use of system applications such as Blynk App. It works to control and monitor a project. It is also compatible for android and iOS users. Blynk App also needs to have an internet connection to work.

2.4 System Development Methodology

The SDLC, or Software Development Life Cycle, is a method for creating high-quality software as quickly and inexpensively as feasible. An organization may produce higher-quality, thoroughly tested software that is ready for production usage more quickly thanks to the well-organized phase flow provided by the SDLC. Analysis, planning, design, programming, testing, and deployment are the next steps in creating the systems. SLDC helps save unnecessary rework and after-the-fact corrections by foreseeing costly errors like forgetting to ask the end-user or customer for feedback (Scroggins, 2014). The Waterfall model, the Agile model, and the Rapid Application Development (RAD) Model are well-known software development models. In this chapter, we'll go into more detail on the three SDLC models.

2.4.1 Waterfall Development Methodology

As a means of ensuring project success, the waterfall methodology was the first SDLC model to be adopted extensively in software engineering. In the waterfall technique, each step of the software development process is separated into its own sub phase. The waterfall model stages are depicted in figure 2.11. In this waterfall model, the

output from one phase is frequently used as the input for the following step in a sequential manner.

Requirement analysis, system design, implementation, testing, deployment, and maintenance are the seven steps that make up the Waterfall methodology. In the initial stage, a requirement specification document is created and contains all potential system needs. The required specifications from the first phase are examined and the system design is created in the second step. The defining of the overall system architecture and the specification of the hardware and system requirements are both aided by this system design. Third, throughout the execution stage. The system is initially constructed in discrete programs, known as units, based on the system design inputs, and is then combined in the following phase. Building and testing each item individually for functionality is known as unit testing. Fourth, when conducting tests Each item generated during the implementation phase is tested before being added to the system. Following integration, the entire system is examined for flaws and failures. After functional and non-functional testing is finished, the product is then deployed in the customer environment or released to the market. There are a few difficulties that come up in the client environment, as a last point. To fix these problems, patches are published. Newer versions of the product are launched to enhance it. To effect these changes in the client environment, maintenance is carried out. As a result, each of these phases is interconnected, and development is viewed as gradually descending through the phases (like a waterfall). The term "Waterfall Model" refers to a process where each phase is completed and approved before moving on to the next. This model avoids phase overlap (Andrei, n.d.).

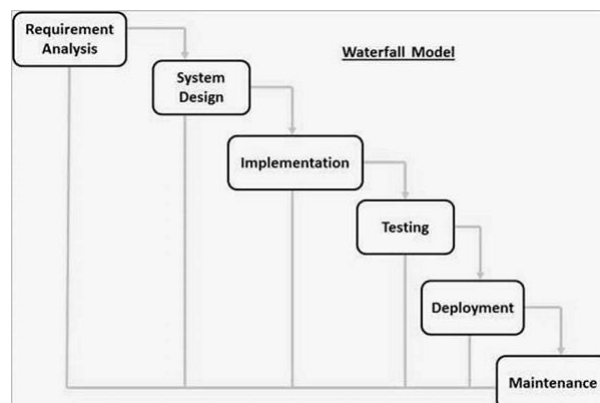


Figure 2.11 Phase of waterfall model

2.4.2 Agile Development Methodology

Agile is a method of software development that considers the requirement for flexibility and applies pragmatism to the delivery of the finished product. Agile software development requires a cultural shift in many businesses since it emphasizes the clean delivery of specific software components rather than the full solution. One of the advantages of Agile is its capacity to support teams in a changing environment while keeping a laser-like focus on the effective delivery of business value. Agile promotes a cooperative culture that boosts productivity across the board as teams work together and recognize their individual roles in the process. Finally, businesses who employ Agile software development can be sure they are providing a high-quality product because testing is done continuously throughout the development process.

This enables adjustments to be made as necessary and notifies teams of any possible problems. Figure 2.12 illustrates the six phases of the Agile software development cycle: concept, iteration, release, production, and retirement. The first step, concept, comprises determining business opportunities in each proposed project as well as estimating the amount of time and labor required to finish the project. This information can then be used to rank projects according to their technical and financial viability. The team is assembled, money is obtained, and the customer's initial requirements are negotiated during the second step, referred to as inception. Additionally, a schedule that details the various roles played by teams and specifies exactly when work is to be finished for each sprint should be developed. A sprint is a set period of time during which a certain amount of work must be finished and prepared for review. Iteration/construction, the third step, is when teams start creating functional software based on specifications and ongoing feedback. The iterations, or single development cycles, that make up the Agile software development cycle are built upon one another and serve as a springboard for the following stage of the overall development process until the project is finished. Each iteration has a predetermined end date and normally lasts two to four weeks. At the conclusion of each iteration, a usable product should be ready for release. Final QA testing, defect resolving, system and user documentation completion, and eventually the deployment of the last version into production make up the fourth step, release. The fifth step, production, focuses on the continuing maintenance needed to keep the software working after the release. The software's development teams must make sure it functions properly and

instruct users on how to do it. Up until the product is planned for retirement or support is discontinued, the production phase is ongoing. All end-of-life actions, including customer notice and final migration, are part of the retirement phase. It is necessary to remove the system release from production. Usually, this is done when a system needs to be replaced by a new version, when it becomes outdated, superfluous, or when it starts to go against the business strategy (Laurie, 2007).

Agile software development cycle

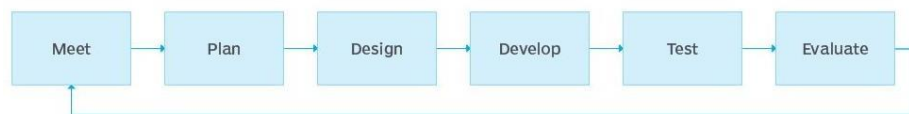


Figure 2.12 Phase of agile model

2.4.3 Rapid application development methodology

The rapid application development process is one that prioritizes quick prototyping over in-depth planning while developing software. A prototype is a working model of a product component that is functionally identical. In the RAD paradigm, functional modules are created concurrently as prototypes, then combined to build the entire product enabling quicker product delivery (Daud et al., 2010). The lack of thorough preparation makes it simpler to absorb changes into the development process. In RAD projects, small teams of developers, subject matter experts, client representatives, and other IT resources work on their component or prototype over time in an incremental and iterative manner. The ability of the developed prototypes to be reused is crucial to the success of this concept. Figure 2.13 depicts the RAD model's phases.

For achieving a deadline, the rapid application development is ideal. A RAD framework might be the ideal option if you need to develop anything rapidly. Rapid application development software is the ideal option if you don't have enough time to plan and build your requirements. Rapid application development is a flexible strategy that permits quick adjustments. Rapid application development speeds up the process by which businesses may test their products in the real world. Along with the focus on sprints and iterative work that RAD emphasizes, software developers can show development. This technique enables enterprises to view the goods right away, in contrast to conventional systems like a waterfall, which forced them to wait until the last stage to see the outcomes. User feedback is given top priority in rapid application development to guarantee that ideas

are updated frequently to satisfy client needs.

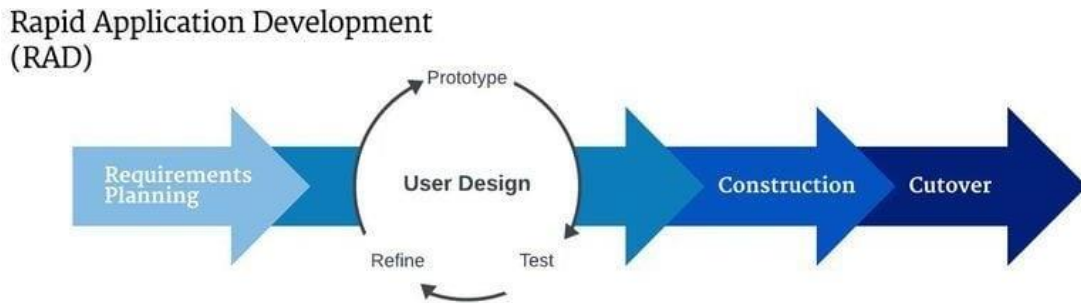


Figure 2.13 Phase of RAD model

2.5 Comparison of methodology

The Table 2.2 shows the comparison of the methodology that suitable to used in develop a project. It is consistent with the techniques that it uses in each phase of model, the advantages and limitation of the model, and the usage of the methodology.

Table 2.2 Comparison of Methodology

Methodology	Waterfall model	Agile model	Rapid application development model
Organizational structure	Linear	Iterative	Iterative
Techniques	The next phase begins only after the preceding phases established set of goals has been meet and approved.	To provide specific characteristics for a deployment, tasks in agile are broken into time boxes, or brief time periods.	A rapid application development cycle has four stages. Rapid development through multiple iterations and continuous feedback.

Advantages	<ul style="list-style-type: none"> - Easily understandable and explainable. - Error detection at an early stage. 	<ul style="list-style-type: none"> - Lower risk - Error detection earlier - Software is delivered on a regular basis. 	<ul style="list-style-type: none"> - Replaceable at any time. - Reviews are quick. - User can be more productive with fewer workers.
Limitations	<ul style="list-style-type: none"> - Expensive in term of money and effort. - Difficult to maintain. - Unsuitable for complex projects. 	<ul style="list-style-type: none"> - Inadequate resource planning. - Limited documentation. 	<ul style="list-style-type: none"> - User requirements are required throughout the production cycle. - More complex
Usage	<ul style="list-style-type: none"> - The requirements are clear and unchanging. - The project is simple, and the cast is small. 	<ul style="list-style-type: none"> - To create a new feature - A brief and adaptable timetable. - Budget adaptability. 	<ul style="list-style-type: none"> - It is possible to modify the systems. - System modifications are possible.

2.6 Summary

In conclusion, there are several advantages and disadvantages in the three existing systems, namely Smart Garbage Monitoring System, Internet of Things Based Wireless Garbage Monitoring System, and IoT based Dustbin Monitoring with Dumpster Alert System.

In the Smart Garbage Monitoring System, it proposes a method for keeping track of each bin's condition on a campus of any college, school, or other location. Here, an Arduino board is used, together with an ultrasonic sensor, a Wi-Fi module, and a system that is wired to every single bin on the campus. The lid of the trash can is the place ultrasonic sensor is pointed, and if there is less than 5 cm of vacant space there, the state of the bin will be displayed as full; otherwise, it will be displayed as empty. The technology will be put into place on the VIT University campus. A centralized server will check the state of the bins on a regular basis and alert the sweepers as necessary. The Wi-Fi module will actually transmit the data to the closest network, and it is anticipated that every time, the data will travel through the routers and be sent correctly to the server. A centralized server will check the state of the bins on a regular basis and alert the sweepers as necessary. The Wi-Fi module will actually transmit the data to the closest network, and it is anticipated that every time, the data will travel through the routers and be sent correctly to the server.

Next, the Internet of Things Based Wireless Garbage Monitoring System is a wireless waste monitoring system with an IoT-based underpinning architecture is put out in this article. With the help of the appropriate software, they may implement remote monitoring using this method, thus reducing human intervention, and enhancing the effectiveness of human efforts. Meanwhile, we have put out a technical approach to create a straightforward hardware and software garbage monitoring system.

Furthermore, the IoT based Dustbin Monitoring with Dumpster Alert System a smart garbage collection system to assist municipal governments. By doing so, problems with IoT-based trash management and waste collection for smart cities are resolved. The devised method is very effective at gathering waste, keeping track of the quantity of waste dumped, indicating the scent and poisonous smell produced by the waste, and

normalizing the environment. The system's key advantage is that it gathers waste promptly, preventing bin overflow and assisting in waste collection.

Lastly, the purpose of this project to make easier for the people to recognize the level of garbage bin using the mobile application. It also can reduce the costing in very particular part. It also can minimize human intervention and improve the efficiency of human activity. Therefore, the amount of waste can detect by using notification or message.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, it will cover about the methodology that used in this project to achieve the objectives of the proposed project which is to establish a Rubbish monitoring system through using the Blynk application. The agile methodology has been chosen as the project development methodology approach that which consist of requirements, design, development, testing, deployment, and review phase. Other than that, it will explain about on how to design the prototype and the suitable hardware and software is suitable to use.

3.2 Model for proposed project

The agile method was chosen to be the best appropriate technique for this suggested project, based on the research article. The agile model is the most appropriate approach for planning, structuring, and developing this project in compared to other software development models. This is due to the agile model's flexibility in terms of timeframe and money, as well as the rapid creation and demonstration of capabilities. To reduce risk, the trash monitoring system can be modified and improved during the Agile model's development phase. The figure 3.1 below depicts the agile phase.

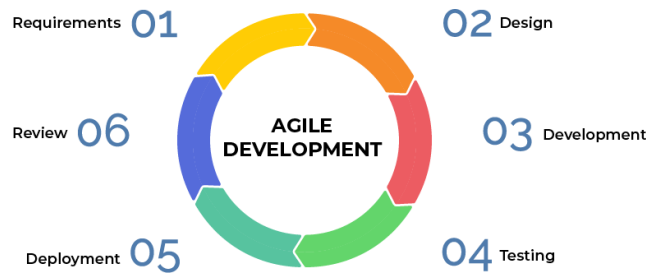


Figure 3.1 Agile model phase

3.2.1 Requirements Phase

During the requirements phase, the proposed project for rubbish monitoring system is looking to improve technology by replacing the use of Arduino Uno with Raspberry Pi technology and to detect the level of rubbish in the bin using the ultrasonic sensor. Besides that, the system will send the information in the form of notifications through the phone using Blynk application. This project uses one types of notifications because the Blynk application needs the internet connection to send notifications. If the user does not have internet access, the notification cannot be sent to the user. With the production of the application, it will make it easier for the admin to identify the level of each bin at faculty. As a result, the admin will get to arrange the rubbish management more flexibly and efficiently compared to the previous technology test project.

3.2.2 Design Phase

In the design phase, this project will begin with understanding the flowchart, use case diagrams, and context diagrams for the system interactions with end user. The flow chart depicts the procedure of the rubbish monitoring system works. The use case diagram depicts the user interaction with the system. The admin of rubbish collectors are the users of this system. User also known as the actors that may conduct the tasks such as viewing the level of the bin, notify the staff collector, and so on. The context diagram is intended to depict data flow from the Raspberry Pi to the system, as well as data flow from the system to the user or the other way around. After that, a prototype of the project and interface design will be developed, with the storyboard approach being utilized to depict the navigation in the interface of the rubbish monitoring system. Prototype design

will begin with planning through sketches until the project production. During this step, the best building method will be chosen.

3.2.3 Development Phase

The system development will be prioritized during the construction part. On the proposed system, stages such as development would be implemented. Writing code for rubbish monitoring system is a method of putting the proposed system into action. The Raspberry Pi will be served as the project primary hardware platform and will also be used to build the code for how the raspberry pi works, while using the Blynk for developing the rubbish monitoring system. C, C++ and python are the three types of the programming languages that can be used in this project. In addition, any system updates or improvements will be implemented throughout this time. Development process and coding are the among tasks that will be completed.

3.2.4 Testing Phase

After the project system has been designed and developed, the system will be tested. The students will through away the rubbish into the smart rubbish bin. When the rubbish bin already full, the admin can get the notification through their smartphone. The admin also can check the level of the bin whether it is full or empty. After getting the notification, admin can let the staff collector know that which bin need to collect. The system project is sent back to the development phase for the correction if any problems of the project are detected. When the problems were resolved, admin users will test it again and do the testing for the whole systems again.

3.2.5 Deployment Phase

The system is in the deployment phase when it has completed development and is ready to be prepared for end users. However, the user can utilize the system, and the prototype system will be tested in this project.

3.2.6 Review Phase

The review phase is the phase that the user can review back starting from the requirement phase until the deployment phase where the user can identify each of the process and the error that will be occurring when the project is power on. In the

requirements phase, replacing the Arduino device with a Raspberry Pi will make a slight difference to the project. If there has problems when running the project, the user can make improvements in the design phase and development phase. In the R-MS project, Blynk App is used to send the notifications to users. The notification delivery process will be carried out in the testing phase.

3.3 User requirement

In the user requirement, will explain the requirements required by the user for the system. This project will also get to know how the process of this system interacts with users. This system has an application that will display the rubbish bin data to the user. Here too, based on requirement it can improve the user needs based on the response given by the user after using the system.

3.3.1 Functional requirement

- i. Admin should be able to get the notification of rubbish bins.
- ii. Admin should be able view the level of the rubbish bins.
- iii. Retrieving information from database for the level rubbish bin which fulfils the rubbish collection.
- iv. The application can generate immediately the data request.
- v. Only the admin can view the information of the bins.
- vi. The application should have to provide on demand services.
- vii. Interface and compilation issues must not exist in the application.

3.3.2 Non-functional requirement

- i. The application needs a user interface for tracking and manually controlling the prompt and effective removal of rubbish from the chosen rubbish bin.
- ii. The system should be user friendly and easy to use.
- iii. Operational

- The system should be able to integrate with the application system in Blynk.
- iv. Security
- User are required to create a strong password when registered an account for this system application. The password must have 8 characters or above with the combination alphabets, number, and symbol.
 - Only the admin can see the data of bins.
- v. Accessibility
- The system needs to use any icon that easier for user to analyze the level of bin.

3.3.3 Limitations

- i. The raspberry pi needs a power supply to ensure that the bin continues to function.
- ii. The project requires an internet connection to send notifications to the admin.
- iii. The application can only view and manage for admin.
- iv. Data collection is not accurate if there are interruptions to the internet and electricity.

3.3.4 Constrains

- i. Requires huge costing for developing this project.
- ii. Ultrasonic sensor is less reliable and handling the sensor need to be done with extreme care.

3.4 Project overview

The table 3.1 shows the proposed project overview of the rubbish monitoring system (R-MS). In this project overview, there are several phases that explain the processes carried out as well as the objectives that need to be achieved in the R-MS project.

Table 3.1 Proposed Research Framework

<p>Phase 1:</p> <ul style="list-style-type: none"> • Identify the problems that exist in Faculty Computer (FKOM). • Lists the rubbish bin problem that often occurs at Faculty Computer (FKOM). • Setting goals and objectives that need to be achieved in the R-MS project to avoid the problem of rubbish bin. • Identify the scope and the significance of the project system. 	
<p>Phase 2:</p> <ul style="list-style-type: none"> • Doing research on several project articles that are approximately comparable to the project being carried out. • Make comparisons between several projects to get the idea for making improvement in the R-MS project. • To identify the suitable methodologies model that can be implemented in the R-MS project. 	<p>Objective number 1</p>

<p>Phase 3:</p> <ul style="list-style-type: none"> • To operate the rubbish monitoring system project using the SDLC agile methodology concept. • Identifying devices and sensors that are suitable to be used for the purpose of improvement in the project. • Identify the suitable software to be used in R-MS project. • To create use case diagrams, flowcharts, and context diagrams for the project. • To sketching the rubbish bin project, graphical user interface (GUI), and storyboard for the R-MS project. • To planning the time period using the Gantt chart to complete the R-MS project. 	
<p>Phase 4:</p> <ul style="list-style-type: none"> • To build a prototype for the R-MS project. • To applying the raspberry pi connection and the ultrasonic sensor that used. • Installing the raspberry pi and ultrasonic sensor on the rubbish bin. • Make a configuration for raspberry pi. • Make a configuration for the ultrasonic sensor. 	<p>Objective number 2</p>
<p>Phase 5:</p> <ul style="list-style-type: none"> • To identify the appropriate application that need to use. • To learning about using of the Blynk application. 	

<ul style="list-style-type: none"> • To make a sketching of the application for the R-MS project. • To developed and configured the R-MS application using Blynk. 	
<p>Phase 6:</p> <ul style="list-style-type: none"> • To testing the R-MS project. • To testing the Blynk application. • To request the data from the bin to the application through Blynk application via mobile phone. • To ensure the Blynk application can receive notifications. 	<p>Objective number 3</p>
<p>Phase 7:</p> <ul style="list-style-type: none"> • To do the second test in the R-MS project. • To identify the deficiencies and weaknesses in the R-MS project. • To making improvements in the R-MS project. • To make a conclusion about the R-MS project. 	

3.5 Design

The project design is important because it allows to identify the appropriate hardware and software required to assist in defining the overall project architecture. Use case diagrams, flowcharts, context diagrams, and data project design are all included in the project design. This system design is important in serving as a roadmap for gaining a deeper understanding of the entire project.

3.5.1 Use case diagram

Use case diagram is a type of behavioral diagram defined by and created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases. Figure 3.2 show the interactions between the system, the actors, and the use case. The use case refers to the actions that the actors might be doing. The admin is responsible to manage the changes of system, monitor the level bins, manage user data, manage notification, and need also manage the rubbish collection if needed. Meanwhile, the worker collector is also can access to the system, getting the notification and manage the rubbish collection.

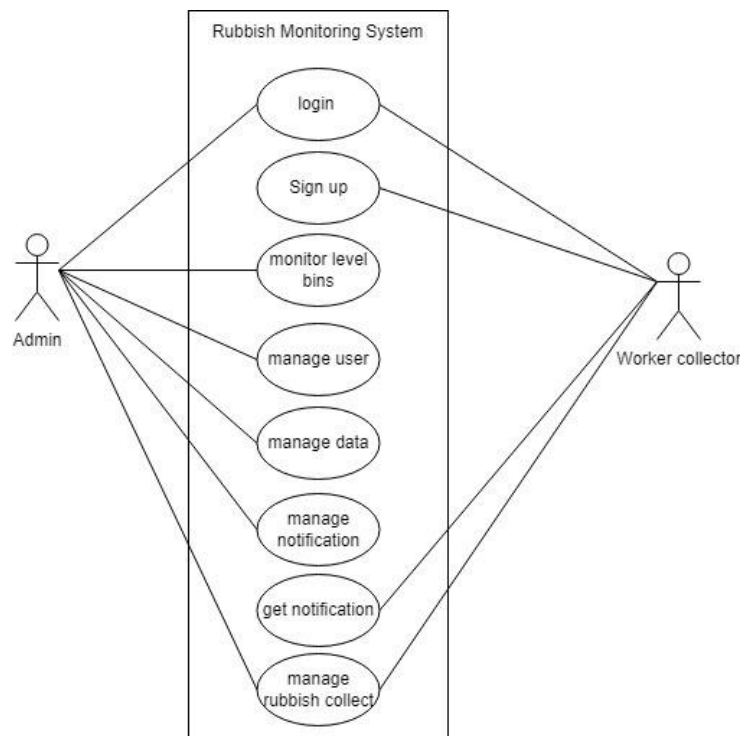


Figure 3.2 Use case diagram

3.5.2 Flowchart

Figure 3.3 Shows the actual flow system of rubbish monitoring system (R-MS). The steps also show for R-MS from starting power on the power supply until the worker collector get the notifications. The IoT based rubbish monitoring system (R-MS) is a quite creative solution that will aid in maintaining a safe and healthy faculty environment. In the figure 3.4 shows the flowchart process for the login and sign-up process. Only the admin user can sign up the new worker collector as new user. When the admin user successfully registered their information, the worker collector can get the notification of the rubbish bin. On the figure 3.5 shows the administration is informed via the system on the amount of rubbish level is being generated in the rubbish bins located all around the faculty. The system will measure the volume of the rubbish bin through the ultrasonic sensor that were implemented at the top or side of the rubbish bin. If the rubbish are on the same level of the sensor, the application system will update the information of the rubbish bin is full. If the volume of the rubbish is not the same level with the ultrasonic sensor, the application system will update the information of the rubbish bin not full. Else, there is no rubbish that were detected by the ultrasonic sensor the application system will update the information of the rubbish bin empty. The worker collector will be doing their work after getting the notifications from the system.

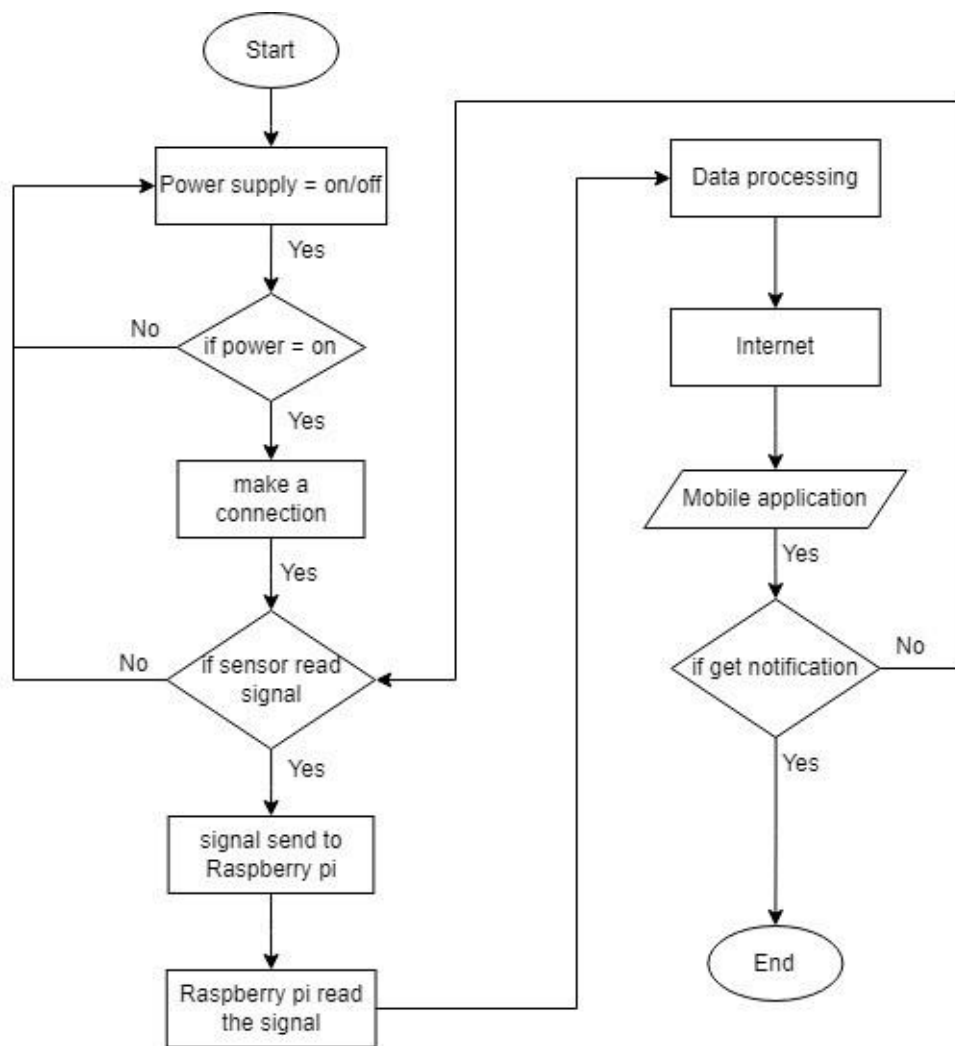


Figure 3.3 Flowchart of the R-MS

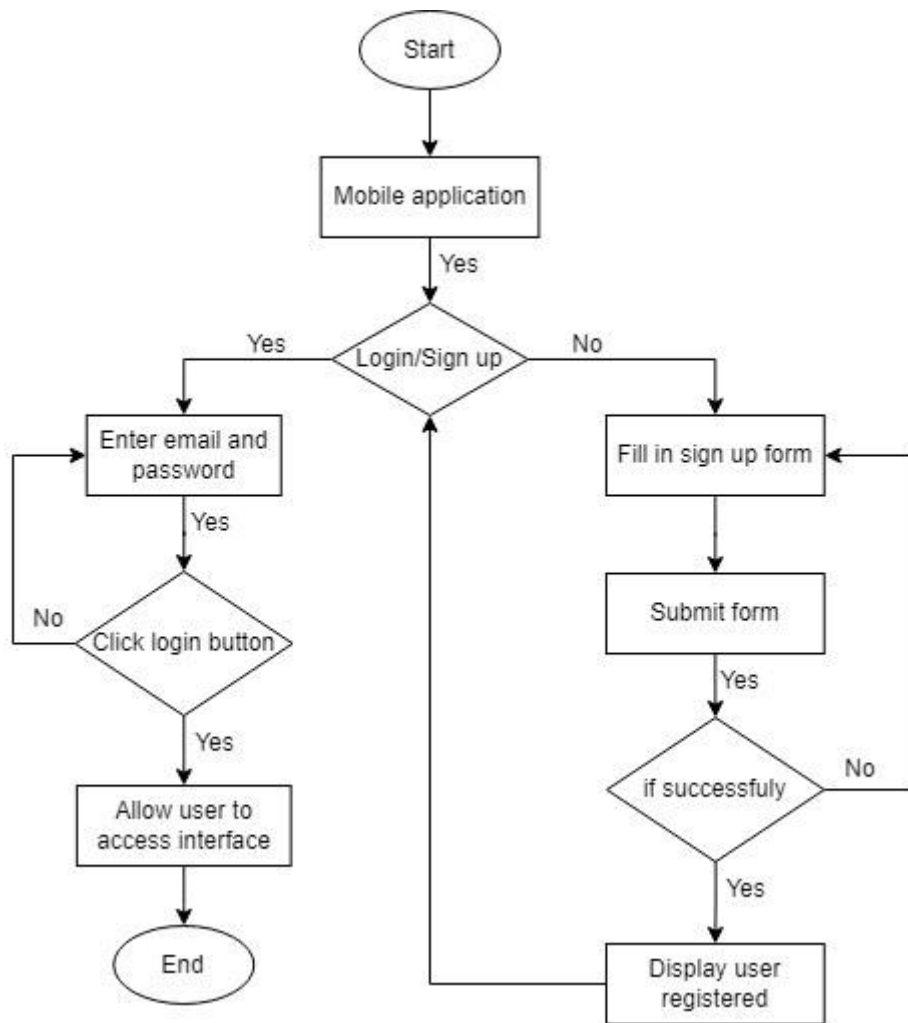


Figure 3.4 Flowchart of the R-MS login system

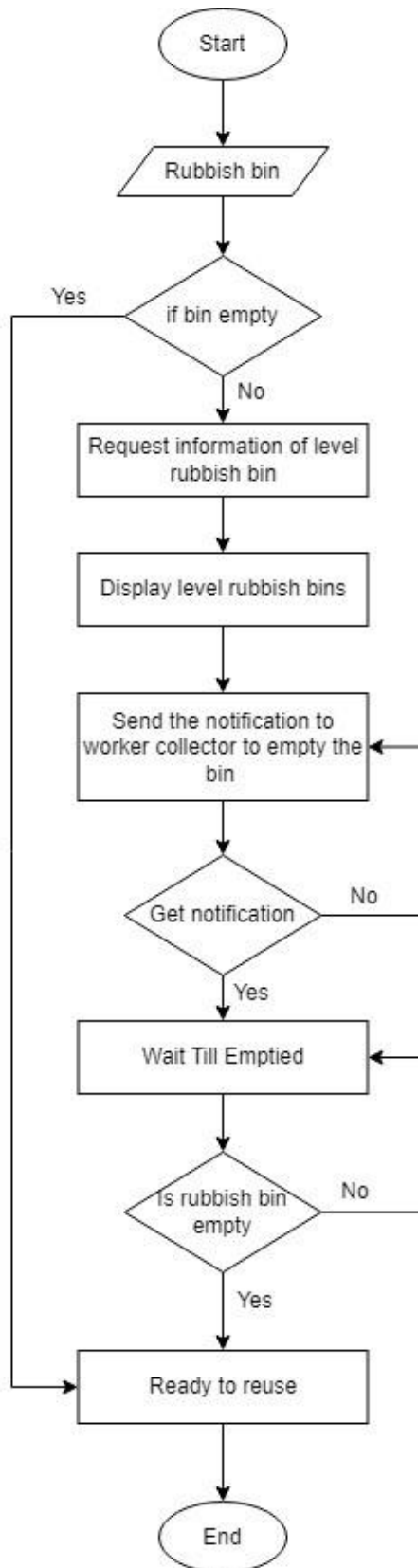


Figure 3.5 Flowchart of the system application

3.5.3 Context diagram

The context diagram defines the system context and limitations, as well as the relationship between the system and external entities by displaying the flows of information between the system and external entities. The context diagram of the proposed system is shown in figure 3.6. Admin user and rubbish bin are examples of external entities. The arrows are a process of a proposed system.

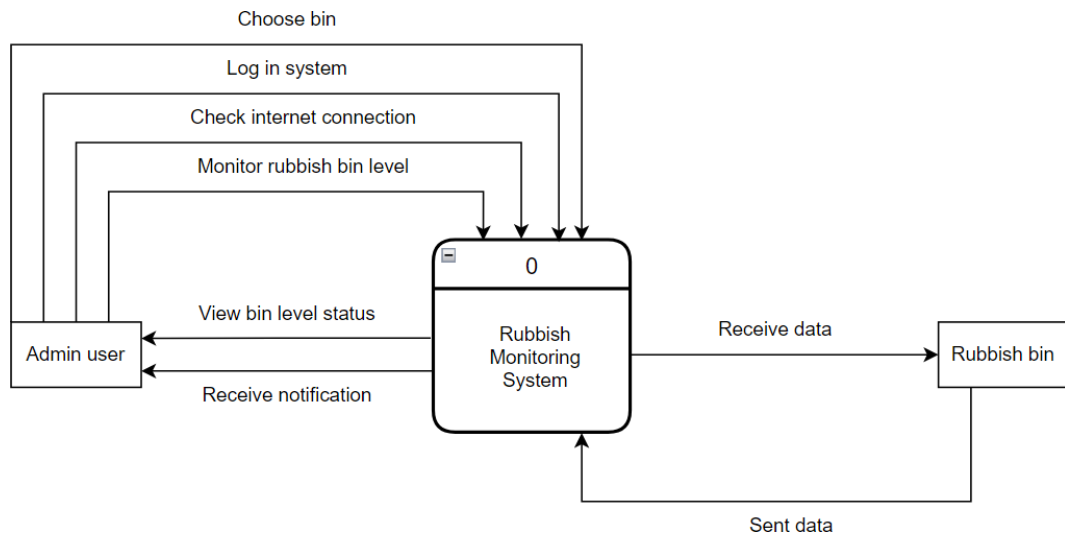
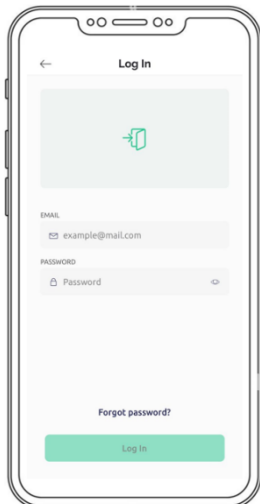


Figure 3.6 Context diagram of the R-MS

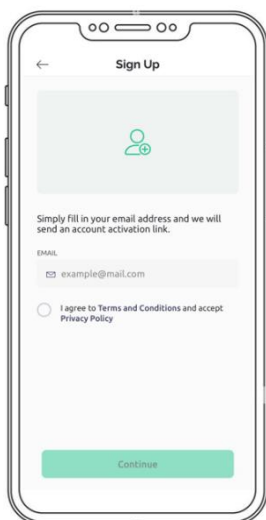
3.5.4 R-MS system application storyboard



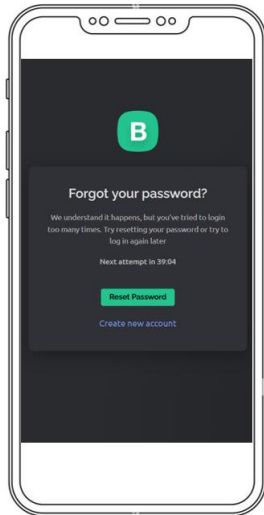
The user needs to enter the application Blynk and log in.



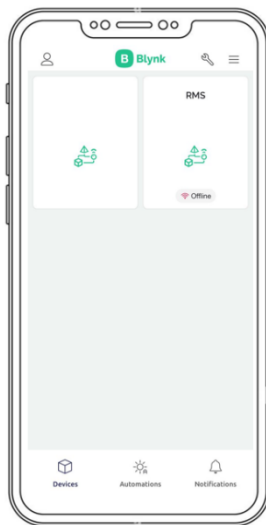
The user needs to enter username and password before log in.



For the new user, only the admin user can register their information. The admin user will insert the email address and click on agree to terms and conditions. After that, click "Continue".



If the user repeatedly key in wrong email address and password it will go to the “Forgot your password?” page. The user can reset password for their account or create new account.



This is homepage of application system. It will show the lists of bins that already register in this project.



After click button “RMS”, it will go to this page. Users need to monitor the rubbish bin to update and get the latest information of that bin. The output measures will appear.

Figure 3.7 R-MS application system design

3.5.5 R-MS physical design project

Figure 3.8 shows the physical project design for rubbish monitoring system based on the Internet of Things (IoT). In this project, just only need a rubbish bin, raspberry pi, ultrasonic sensor, adapter, and jumper wires. An ultrasonic sensor will be connected with the raspberry pi by using a jumper wire (female). After that, the raspberry pi and ultrasonic sensor will be installed in the rubbish bin. After the installation and configuration process has been successfully completed, the ultrasonic sensor that has been installed will check the level in the rubbish bin. The data that has been collected from the ultrasonic sensor will be sent to the raspberry pi. The raspberry pi will send the received data directly to the admin user via mobile phone. After the admin gets the notification, the worker collector will immediately go to empty the rubbish bin.

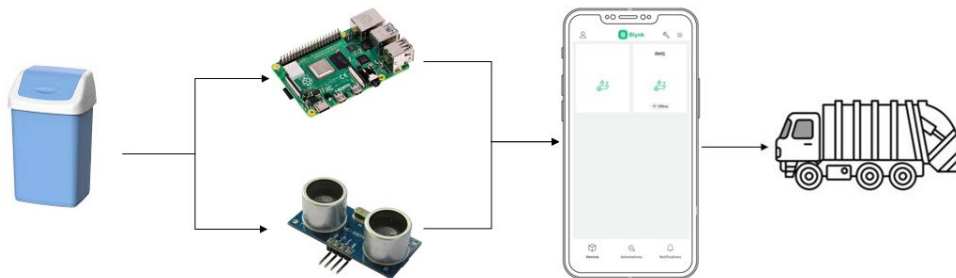


Figure 3.8 Physical design of rubbish monitoring system

3.5.6 Design of the rubbish monitoring system (R-MS)

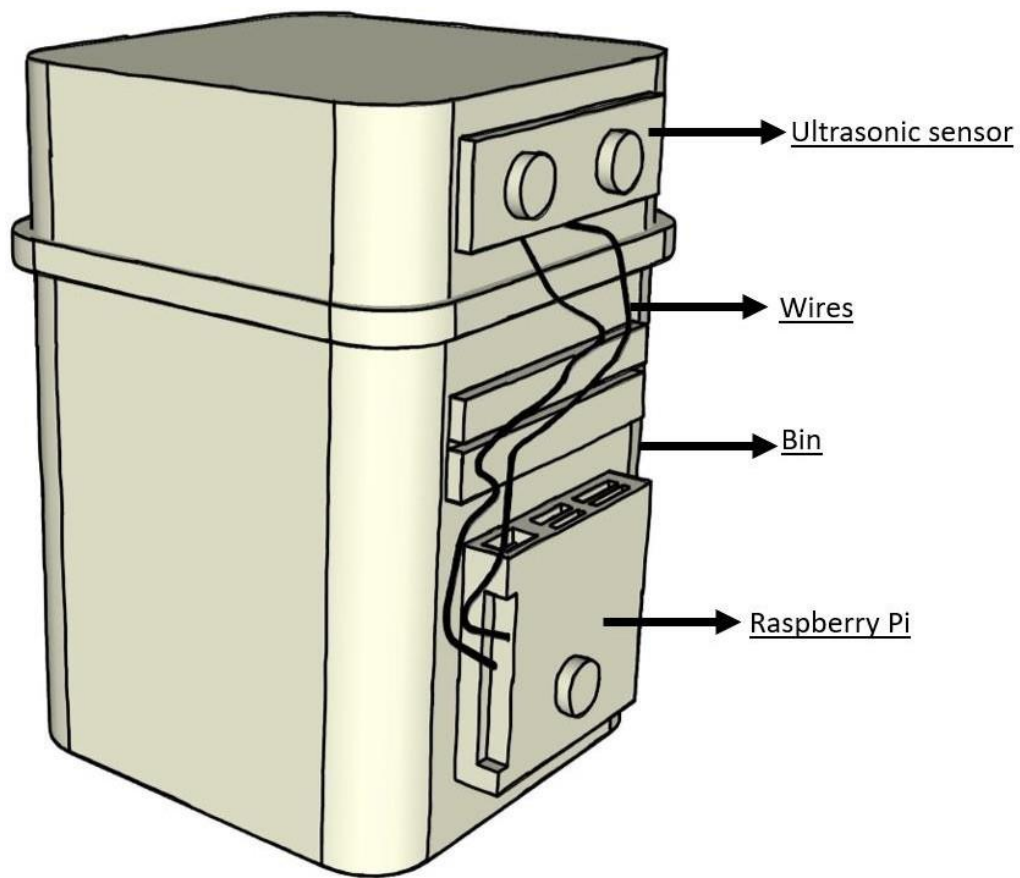


Figure 3.9 Design of rubbish monitoring system

3.5.7 Hardware and software specification

This section provides a brief explanation of the software and hardware used in this project, along with its cost and specifications. Hardware and software are the most essential requirements for this project. A raspberry pi can serve as the project hardware, and the output device can be an ultrasonic sensor for measuring the level. But without software, hardware cannot function on its own. Hardware and software are requiring developing this project.

In order to use tools and developed applications successfully, hardware requirements are important and essential. The most recent hardware is being used for this project to make it effective. The price and specification lists for the hardware used are shown in table 3.2.

Table 3.2 List of Hardware

Hardware	Specification	Function
Laptop	<ul style="list-style-type: none"> • Brand: Asus • Processor: Intel Core i5 • Windows 10 • 16GB RAM • Type of system: 64-bit operating system, x64-based processor 	<p>The laptop is required to create the application system, it also works as the medium for code development, software system installation, system design and project design.</p>
Phone	<ul style="list-style-type: none"> • Brand: iPhone • iOS version: 16.0 • capacity: 128GB 	<p>Its function is to link the project system to application system using Blynk app.</p>
Raspberry Pi	<ul style="list-style-type: none"> • Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz • Memory: 4GB • USB ports: 4 • Storage: MicroSDHC slot • Wi-Fi: 802.11n and Bluetooth 4.1 	<p>It functions as a controller between the sensors and the internet.</p>

Sensor	<ul style="list-style-type: none"> • Ultrasonic sensor HC-SR04 • Power supply: DC 5V • Measurement range: 2cm - 400cm • I/O pins needed: 4 pins • Operating current: 15mA 	A sensor that measures distances or level through ultrasonic sound waves.
Breadboard		It is a construction base for electronic circuit prototype.
Resistor		A passive electrical component that creates resistance in the flow of electric current.

3.5.8 Hardware requirement

3.5.8.1 Raspberry Pi

In this project, Raspberry Pi 4 is a small single board computer that acts as a mini personal computer. It is shown on figure 3.10 Raspberry Pi can be used as a component in a rubbish monitoring system to gather data on the amount of rubbish in a particular location, such as a faculty. The raspberry pi can be connected to sensors that measure the amount of rubbish in a bin, and the data can be used to optimize the rubbish collection routes.

Furthermore, Raspberry Pi is about the size of a credit and has a variety of ports and connectivity options, including HDMI, Ethernet, USB, and 3.5mm audio jack. It runs on a version of the Linux operating system, and it was programmed and run applications using programming languages such as Python and C++.

Lastly, the Raspberry Pi can be programmed to collect data from the ultrasonic sensors at regular intervals, and the data can be stored on the Raspberry Pi or transmitted to a central server for analysis. The Raspberry Pi can also be programmed to send alerts

when the ultrasonic sensors detect that a bin is full. So that, it can be emptied in a timely manner.



Figure 3.10 Raspberry Pi

3.5.8.2 Ultrasonic sensor

The ultrasonic sensor is commonly used in rubbish monitoring system to measure the level of rubbish in a bin. The sensor works by emitting ultrasonic waves and measuring the time it takes for the waves to bounce back after hitting an object. By measuring the time, it takes for the wave to return, the sensor can determine the distance to the object and calculate the level of waste in the bin.

In a rubbish monitoring system, an ultrasonic sensor is typically mounted on the top of the bin, facing down towards the waste inside. As the level of waste in the bin increases, the sensor will detect the rising level and send a signal to the system control panel, which can then trigger an alert or initiate a waste collection process.

Ultrasonic sensor is a popular choice for waste monitoring systems because it is non-invasive, meaning we does not need to come into contact with the waste to measure its level. This makes them hygienic and easy to maintain. We are also relatively inexpensive and can operate over a wide range of temperatures, making them suitable for use in a variety of environments. The figure 3.11 shown the ultrasonic sensor.



Figure 3.11 Ultrasonic sensor

3.5.9 Software requirement.

The software that was utilized to generate this project is important for ensuring that the rubbish monitoring system is operating properly and that the input and output are properly programmed. Table 3.3 Shows the list of software that has been used.

Table 3.3 List of Software

Software	Function / Specification
C++ Programming Language	A general-purpose programming language is C++. It offers facilities for low-level memory manipulation in addition to imperative, object-oriented, and generic programming features.
Python	Python is a programming language that is frequently employed in the fields of scientific computing, data analysis, and web development. Another object-oriented language that lets us define and work with objects in our programming is Python. Additionally, it has a sizable standard library that offers a broad range of tasks and features, making it a powerful and versatile language.

Blynk	<p>Blynk is an IoT platform that allows users to remotely control and monitor devices from their smartphones. It provides a set of tools and services that make it easy for users to build connected applications for a wide range of devices. Blynk also provides a mobile app that users can use to create a visual interface for IoT projects, as well as a server component that handles the communication between the devices and the app. Blynk can control a wide range of devices, including sensors and more.</p>
-------	--

3.6 Gantt chart

The task name, start date, finish date, and length are all listed on this project Gantt chart. It is a significant guideline that the developer must follow. The project Gantt chart is shown in the figure 3.12.

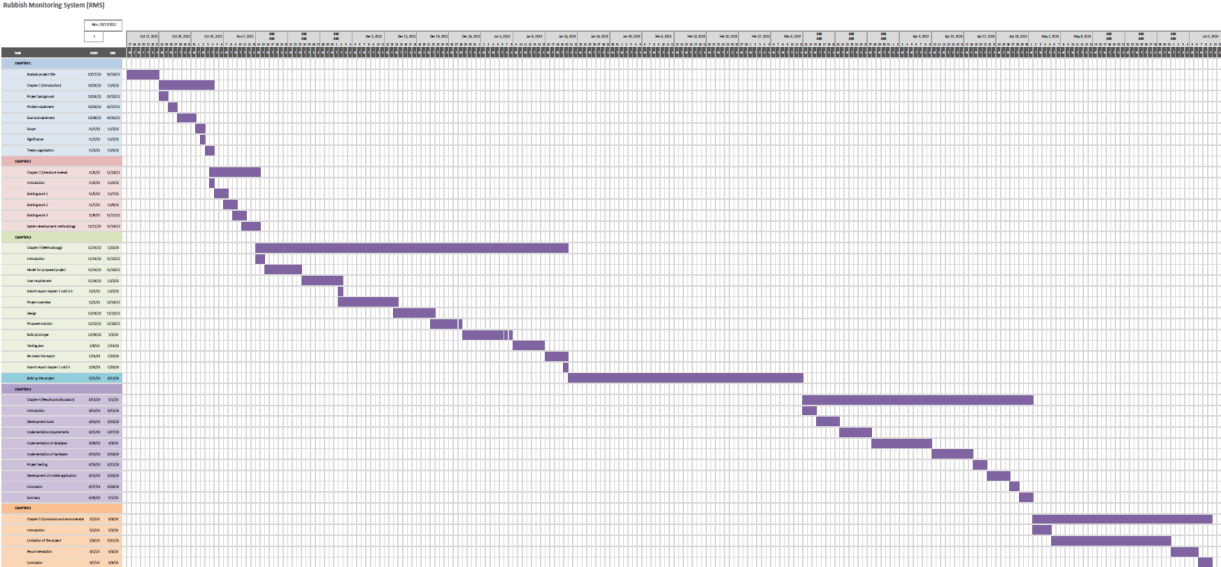


Figure 3.12 Project Gantt chart

3.7 Proposed Solution

3.7.1 Prototype

A prototype is a pre-production sample, model, or version of a product created to test a theory or procedure or to serve as a model for replication or learning. It is a phrase that is employed in a number of fields, such as semantics, design, electronics, and software development. System analysts and users typically utilize a prototype to assess a new design to increase precision. For the Rubbish Monitoring System with Raspberry Pi in IoT, the prototype has been made to see the result of this project.



Figure 3.13 A front view of a rubbish bin



Figure 3.14 A side view of a rubbish bin



Figure 3.15 A back view of a rubbish bin

3.8 Testing Plan

The importance part is the testing phase. After the project has been developed, testing is required to make sure every function operates exactly as intended. Additionally, to make sure the system satisfies the goal and objectives when performing functional and non-functional testing. Verifying whether the system contains error or bugs during this phase is important. If there were any other, the procedure will return to the first phase and implement corrective actions. All system components that are involved in testing are done so. The project is ready for users to operate after the problem has been fixed.

- i. Testing the functionality of R-MS using raspberry pi.
- ii. Testing the functionality of raspberry pi with the ultrasonic sensor.
- iii. Monitoring to see if the ultrasonic sensor is functioning properly.
- iv. Testing the mobile application functionality before it is used in the R-MS.
- v. Testing to ensure the mobile application will be able to detect the rubbish parameter in the rubbish bin whether it is full, not full or empty.
- vi. The testing target is tested.

3.9 Summary

The research methodology applied to this project, an agile methodology model that includes phases for requirements and analysis, design phase, development phase, testing phase, deployment phase, and review phase, has been discussed in this chapter. Each phase has a different purpose and responsibility, and it helps as a developer guidance when making the improvement in this project.

CHAPTER 4

IMPLEMENTATION, RESULT AND DISCUSSION

4.1 Introduction

In this chapter, it will discuss the functionality of the whole system that has been developed. The result and findings of the system are gained after all the functionality testing is conducted. This chapter analyses the results obtained from the testing phase of the Rubbish Monitoring System along with the application and raspberry pi, which included ultrasonic sensor that been used in this project. This discussion also includes the results and finding that had been documented during the development and testing phase.

4.2 Development Tools

Table 4.1 shows the tools that are used to develop the Rubbish Monitoring System (R-MS).

No.	Tool	Purpose
1.	Blynk	To create the mobile application system.
2.	Thonny	To provide a user-friendly and simplified environment for writing, executing, and debugging Python programs.
3.	Raspbian OS	As an operating system.

4.3 Implementation Requirements

In this phase, there are several stages that need to be complete before this project will be successful. It will acknowledge the existing user or else user must sign up to use Application system. In the nutshell, if user forget their password, there will be interaction between applications for user to get their account back. Furthermore, developing this project requires proper software and hardware installation where configuration also must be correct to ensure project can run without any problem.

First step is to setup the Raspberry Pi, and Ultrasonic Sensor HC-SR04 before starting the programming for the first time. The ultrasonic sensor trigger pin is connected to GPIO 17 of the Raspberry Pi through a level shifter, while the echo pin is connected to GPIO 27. The 5V and GND pins of the sensor are connected to the 5V and GND pins of the Pi. After that, to setup the Blynk application for measures. Lastly, the developing Mobile Application for the Rubbish Monitoring System by using the Blynk.

4.3.1 Implementation of Database

To start the development of Mobile Applications, database is needed as important to store all the user data. For this project, database of Blynk had been chosen as web server. Blynk provides a cloud-based database feature called Blynk cloud database. With Blynk Cloud database, the user can store and retrieve data from a project. The database feature allows the user to store and manage data generated by sensors, devices, or any other sources connected to Blynk project.

Blynk must be registered to use its database. Figure 4.1 show the Blynk Get Started web page on the website. Users need to key in their username and password. When the button “log in” clicked, it will redirect to homepage of Blynk. As for system database, Blynk cloud database is used as the database server of the system in order to store the data from the system. For figure 4.2 shows the Blynk website application that has a registered project that already connected to the Blynk. It is important to note that Blynk cloud was provides a limited amount of free storage for any project data. If the user requires more storage capacity, user need to self-hosting the Blynk server, which is gives user more control over the database infrastructure. Meanwhile for figure 4.3, it shows the dataset of the monitoring system to the Blynk application. This database is importance to admin for monitor and manage the schedule for the worker collector.

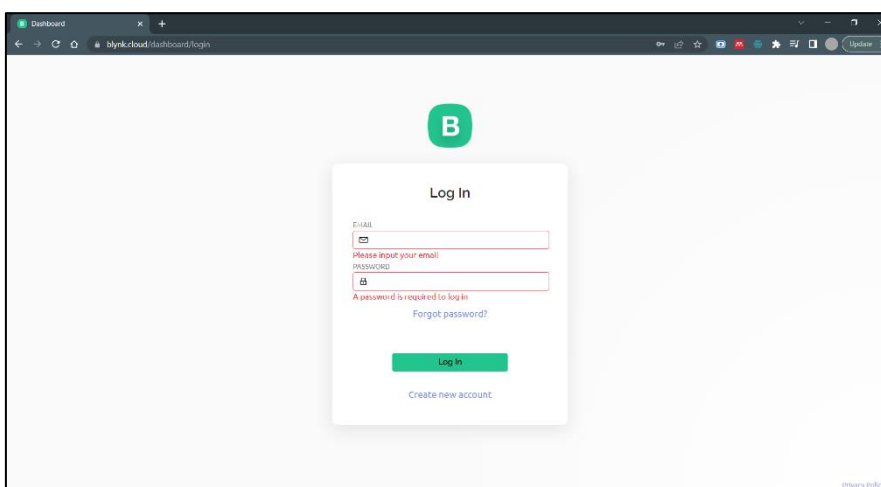


Figure 4.1 Blynk website

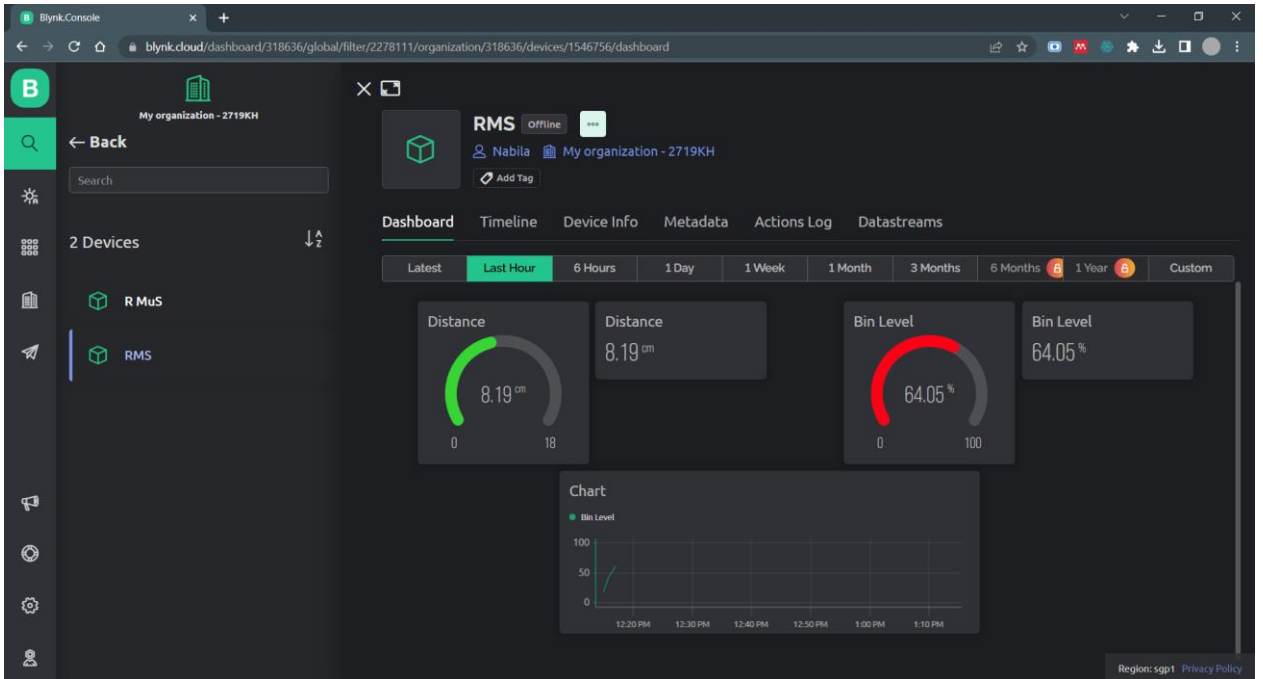


Figure 4.2 RMS website page in Blynk

Time	Event Type	Name	Description	Distance	Bin Level
06/19/23 11:51:28 AM	OFFLINE	Offline		4.878032207	97.74862187
06/19/23 11:51:03 AM	WARNING	Bin_is_Full	Rubbish bin is full	14.63757974	31.06071943
06/19/23 11:50:02 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.896436	15.02545716
06/19/23 11:49:02 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.80604838	15.67108303
06/19/23 11:48:01 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.91121628	14.91988374
06/19/23 11:47:01 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90006554	14.99953185
06/19/23 11:46:00 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90806357	14.94240309
06/19/23 11:45:00 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.86368323	15.25940549
06/19/23 11:43:59 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.89279318	15.0514773
06/19/23 11:42:59 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90327558	14.97660302
06/19/23 11:41:58 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.9098528	14.99296227
06/19/23 11:40:58 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90990446	14.92925384
06/19/23 11:39:57 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.7868155	15.8084607
06/19/23 11:25:45 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.93089174	14.77934469
06/19/23 11:23:10 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.65969789	16.71644364
06/19/23 11:22:10 AM	WARNING	Bin_is_Full	Rubbish bin is full	15.23843288	26.62637141
06/19/23 11:17:45 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.06702954	20.94978903
06/19/23 11:07:12 AM	ONLINE	Online		7.485459	80.52130373
06/19/23 11:07:10 AM	OFFLINE	Offline		10.00920317	63.37390256
06/19/23 11:02:43 AM	WARNING	Bin_is_Full	Rubbish bin is full	8.42243284	75.55405114
06/19/23 10:58:29 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.457182073	82.44869948
06/19/23 10:58:23 AM	ONLINE	Online		7.445701957	82.53070031
06/19/23 10:58:18 AM	OFFLINE	Offline		7.903948426	79.25751124
06/19/23 10:57:28 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.361344496	82.65640256
06/19/23 10:56:26 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.930640307	79.06685495
06/19/23 10:54:58 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.897157967	79.30601452
06/19/23 10:53:56 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.869455814	79.50388704
06/19/23 10:52:55 AM	WARNING	Bin_is_Full	Rubbish bin is full	9.06192151	70.98627493

Figure 4.3 Blynk dataset for RMS

4.3.2 Implementation of Hardware

This project required several hardware to be done. It is Raspberry Pi, Ultrasonic sensor HC-SR04 and jumper wires. Firstly, it needs to connect the power supply to the Raspberry Pi. The power supply should be rated for at least at 5V. It also can connect using micro-USB port on the Raspberry Pi. Ultrasonic sensor was connected to the Raspberry Pi using jumper wires. The ultrasonic sensor has four pins which is VCC, GND, TRIG, and ECHO. The VCC pin of the ultrasonic sensor is connected to a 3.3v pin on the Raspberry Pi using a red jumper wire. Next, the GND pin is connected to a ground pin on the Raspberry Pi using a green jumper wire and the TRIG pin of ultrasonic sensor to a GPIO pin on the Raspberry Pi using a orange jumper wire. This will be the pin that sends out the signal to trigger the ultrasonic sensor. Meanwhile, the ECHO pin of the ultrasonic sensor will connect to another GPIO pin on the Raspberry Pi using a yellow jumper wire. This will be the pin that receives the signal back from the ultrasonic sensor. All the connections must be make sure are secure in the correct pins.

Figure 4.3 shows the connection between all the hardware component used in the Rubbish Monitoring System with Raspberry Pi using the mobile application. In this project, it is using the TightVNC server to be coding and compile to program the Raspberry Pi between the ultrasonic sensor HC-SR04 by connected to the Firebase database via Wi-Fi.

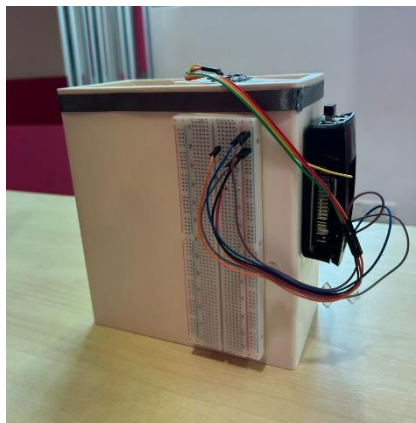


Figure 4.4 Rubbish Monitoring connected circuit

On the previous chapter, it was clearly stated the platform to build the program is using Raspberry Pi. The Raspbian OS is the one of popular operating system because it is specifically designed to run on the Raspberry Pi, and it also include support for the hardware components such as the GPIO pins and useful pre-installed software package. Beside that, there are other operating system that can be installed on the Raspberry Pi such as Ubuntu, Linux, and Windows. Some popular programming languages of Raspberry Pi that use for this project includes Python, C, C++,

and Java. In the development of project there are various integrated development environments (IDEs) that can be used on the Raspberry Pi, such as Visual Studio and Thonny. These IDEs provide tools for code editing, debugging, and project management, and help streamline the development process. Basically, to get starting the programming using the IDEs, the main need to use the Putty for communication between devices and connect the Raspberry Pi to laptop. Besides, the laptop also give power to the Raspberry Pi to make it can interact with Ultrasonic sensor HC-SR04. Figure 4.4 shows the output measure of ultrasonic sensor.

```

1 #libraries
2 import RPi.GPIO as GPIO
3 import time
4
5 #GPIO Mode (BOARD / BCM)
6 GPIO.setmode(GPIO.BCM)
7
8 #set GPIO Pins
9 GPIO_TRIGGER = 17
10 GPIO_ECHO = 27
11
12 #set GPIO direction (IN / OUT)
13 GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
14 GPIO.setup(GPIO_ECHO, GPIO.IN)
15
16 def distance():
17     # set Trigger to HIGH
18     GPIO.output(GPIO_TRIGGER, True)
19
20     # set Trigger after 0.01ms to LOW
21     time.sleep(0.00001)
22     GPIO.output(GPIO_TRIGGER, False)
23
24     StartTime = time.time()
25     StopTime = time.time()
26
27     # save StartTime
28     while GPIO.input(GPIO_ECHO) == 0:

```

```

Shell
Measured Distance = 0.8 cm
Measured Distance = 1.0 cm
Measured Distance = 0.9 cm
Measured Distance = 0.9 cm
Measured Distance = 0.9 cm
Measured Distance = 0.9 cm
Measured Distance = 0.9 cm
Measured Distance = 0.9 cm

```

Figure 4.5 Output measure of ultrasonic sensor.

4.4 PROJECT TESTING

In this part of the chapter 4 will explains about the development of the Mobile Application and testing phase with the hardware and database that been implemented starting from scratch until the end of the development process. It also discusses about the coding along with its programming language and error occurs during the development process.

4.4.1 Development of Mobile Application

The last stage for Rubbish Monitoring System is the development of Mobile application. Firstly, start by installing the Blynk mobile application on the iOS or Android device. User need create a Blynk account by signing up using email address and desired credentials and also obtain

an authentication token that will be used to connect to hardware devices to the Blynk. Once logged in, user can create a new Blynk project by tapping on the “+” icon within the app. Customize the project settings, such as providing a name and selecting the target device for the project.

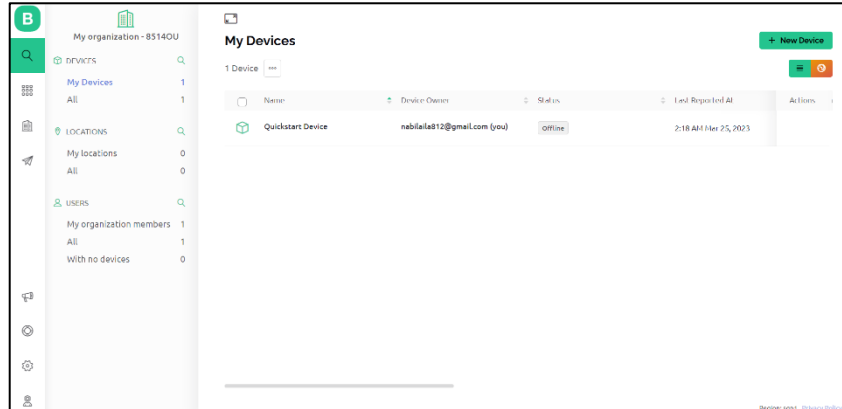


Figure 4.6 Blynk website interface

Secondly, after creating a Blynk project, user will be directed to the project interface within the application. To design user interface, user needs to drag and drop widgets from the widget box into the interface. The widgets can include buttons, label, size and more, that allowing user to create a visually appealing and interactive interface. After that, arrange and organize the widgets on the canvas to create a logical and user-friendly layout.

Thirdly, once user have placed a widget on the interface, user can configure its properties and behaviors. Depending on the widget type, user need define various settings to control its functionality.

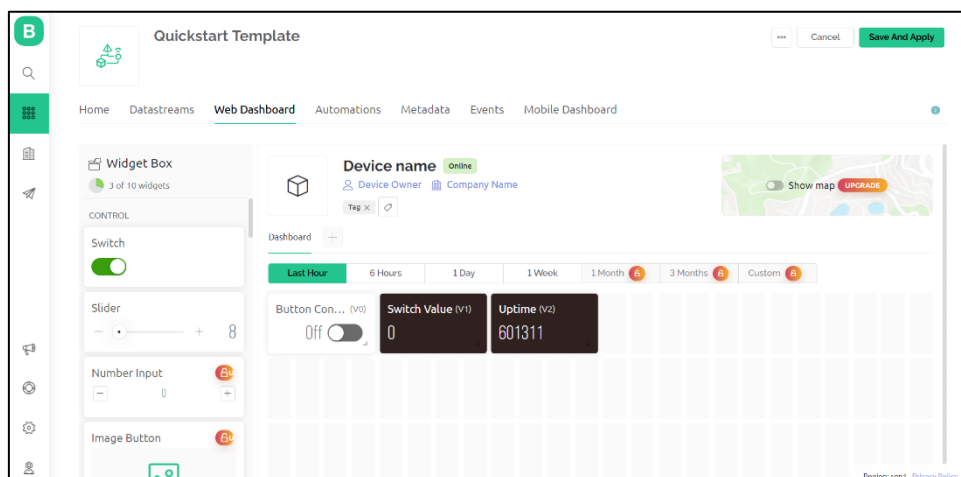


Figure 4.7 Widgets for interface

Fourthly, user need to connect the Blynk app to a hardware component or a virtual pin for example ultrasonic sensor. Hardware pins refer to the physical pin on microcontroller or IoT device, while virtual pins are software-based pins within the Blynk system. In code, user need to set up the appropriate hardware or virtual pin connections to establish communication between the Blynk app and the IoT project. The Blynk library provides a functional or method to link the widgets in user app to the corresponding hardware or virtual pins.

Fifth, user need to ensure the functionality of Blynk mobile applications. Users need to run the applications on the mobile device and interact with the widgets to verify that they perform as intended. After test the communication between the Blynk app and the hardware, user need to ensure that data is transmitted accurately. Lastly, once user is satisfied with the Blynk mobile application, user can deploy it on the other devices or share it with the others.

4.5 CONCLUSION

A Rubbish Monitoring System project using a Raspberry Pi and ultrasonic sensor HC-SR04, along with a mobile application can be a useful and practical solution for managing waste. The Raspberry Pi and ultrasonic sensor can be used to measure the distance to the top of a rubbish bin, allowing the device to monitor the bin. The information of the rubbish can be transmitted to a mobile application, which can provide user with real-time updates on the bin status and alert them when it needs to be emptied. Additionally, the mobile application can allow users to know the bin location and view the bin status whether it is full or not. Therefore, admin must take responsibility to train their worker collector how to use this application and to get familiar or adapt with the system. In addition, a Rubbish Monitoring System also have economic and environmental advantages. By monitoring the fill level of rubbish bin, admin user can optimize their collection routes and schedules so that it can help the worker collector to do their work easier.

4.6 SUMMARY

This chapter has discussed about the project development from the scratch until the end including the problem and error occurs while developing the project. Hardware and software needed to make this project succeeds also has been discussed. This chapter also explains in detail on how to operate each hardware and software involved in.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

This project has developed to grant admin user need by allowing them to measure the level of rubbish bin without needing help from another worker to go checking one by one. From chapter 1 to chapter 4, all has been discussed including the methodology of the project, development, and testing phase. This chapter will be discussing the conclusion, limitation and recommendations for the proposed project.

5.2 LIMITATION OF THE PROJECT

The limitations of this project are:

- This project is experiencing delay problems in sending the correct output measure level of rubbish bin.
- The development of this project using raspberry pi is quite a bit difficult because it needs to have internet connection while configuring.
- The use of the ultrasonic sensor requires the use of 5v power, but it will burn if connected directly through the main plug.

5.3 RECOMMENDATION

In the process of developing this project, three objectives have been achieved according to user needed. But somehow it opens another chance for other developers out there to improve the current project to make it better and be able to compete with another product in the market. Future enhancement of the project could be implementing in the project such as:

- I. Change the rubbish monitoring system Blynk application to the android studio application or flutter application.
 - Add a log in interface for each user. So that, the user can create their own username and password to be more secure.
 - Create a new database that stores all bins data information especially the measure level, the data of worker collector, and bin data.
- II. Replacing the power plug to the solar panel with a power bank combination.
 - Add solar panel to reduce the use of electric power. All the energy will be store in the power bank.
- III. Added a servo motor controller and IR sensor that can open and close lid.
 - The IR sensor will send the impulse to the servo motor controller to open and close bin lid.

5.4 CONCLUSION

As a conclusion, this project was successfully developed and had met its aim and objectives for making the prototype of rubbish monitoring system. The objective of this project is to design and develop a project that allows the admin user and worker collector to manage their work schedules by monitoring the measures of rubbish bin via the phone. Through the development of this prototype, all the objectives have been achieved. Based on chapter 3, the process of analysis, design and development, testing have provided a lot of data and information that will lead to the success of this project knowing that this project is worth rather than investing in a complex and expensive system. From this project development, the developer has gained a whole new knowledge about raspberry pi, ultrasonic sensor and Blynk application that are used in this project. In developing the project, the sensors and its circuit has opened the doors to new opportunities in the future as making an innovative project is currently in demand in the industry.

REFERENCES

- 2019 *Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. (n.d.). IEEE.
- A, A. B., & Ranjana, P. (2019). *Design Of A Monitoring System For Waste Management Using IoT; Design Of A Monitoring System For Waste Management Using IoT*.
- Andrei, B.-A. (n.d.). A STUDY ON USING WATERFALL AND AGILE METHODS IN SOFTWARE PROJECT MANAGEMENT. In *JOURNAL OF INFORMATION SYSTEMS & OPERATIONS MANAGEMENT*.
- Chaudhuri, R., Chattopadhyay, P., & Dasgupta, S. (n.d.). IJERT-Smart Garbage Monitoring System Smart Garbage Monitoring System. In *IJERT Journal International Journal of Engineering Research and Technology*. www.ijert.org
- Daud, N. M. N., Bakar, N. A. A. A., & Rusli, H. M. (2010). Implementing Rapid Application Development (RAD) methodology in developing practical training application system. *Proceedings 2010 International Symposium on Information Technology - System Development and Application and Knowledge Society, ITSIM '10*, 3, 1664–1667. <https://doi.org/10.1109/ITSIM.2010.5561634>
- Fmipa. (n.d.). *Dosen Jurdik Biologi F MIPA UNY*.
- Laurie. (2007). *A Survey of Agile Development Methodologies*.
- Sathishkumar, N., Pravinkumar, M., Preethi, M., & Subash, R. (2022). IoT based Dustbin Monitoring with Dumpster Alert System. *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*, 1814–1818. <https://doi.org/10.1109/ICACCS54159.2022.9785286>
- Scroggins, R. (2014). *SDLC and Development Methodologies*. <http://www.agiledata.org>
- Tarmudi, Z., Abdullah, L., Osman, A., & Tap, M. D. (2009). AN OVERVIEW OF MUNICIPAL SOLID WASTES GENERATION IN MALAYSIA. In *Jurnal Teknologi* (Vol. 51).

APPENDIX B

CODING

Code 1:

```
import RPi.GPIO as GPIO
import time

TRIGGER_PIN = 26
ECHO_PIN = 27
EMPTY_DISTANCE = 25
FULL_DISTANCE = 5

GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIGGER_PIN, GPIO.OUT)
GPIO.setup(ECHO_PIN, GPIO.IN)

def measure_distance():
    GPIO.output(TRIGGER_PIN, True)
    time.sleep(0.00001)
    GPIO.output(TRIGGER_PIN, False)

    pulse_start = time.time()
    pulse_end = time.time()

    while GPIO.input(ECHO_PIN) == 0:
        pulse_start = time.time()

    while GPIO.input(ECHO_PIN) == 1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150
```

```

return distance

def calculate_fill_level(distance):
    fill_level_percentage = 100 - ((distance - FULL_DISTANCE) / (EMPTY_DISTANCE -
FULL_DISTANCE)) * 100
    fill_level_percentage = max(0, min(fill_level_percentage, 100))
    return fill_level_percentage

def main():
    previous_fill_level_percentage = -1
    GPIO.output(TRIGGER_PIN, False)
    print("Ultrasonic sensor example")

    try:
        while True:
            distance = measure_distance()
            fill_level_percentage = calculate_fill_level(distance)

            if fill_level_percentage != previous_fill_level_percentage:
                print("Trash can fill level: {}".format(fill_level_percentage))
                print("Distance: ",distance)
                previous_fill_level_percentage = fill_level_percentage

            if fill_level_percentage >= 100:
                print("The trash can is full. Please empty it.")
                # Add code here to perform any necessary actions when the trash can is full

            time.sleep(0.5)

    except KeyboardInterrupt:
        print("Exiting...")
        GPIO.cleanup()

```

```
if _name_ == "_main_":  
    main()
```

Code 2:

```
import RPi.GPIO as GPIO
import time
from blynktimer import Timer
from blynklib import Blynk

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

TRIGGER_PIN = 26
ECHO_PIN = 27
EMPTY_DISTANCE = 25
FULL_DISTANCE = 5

# Initialize Blynk
BLYNK_AUTH = 'Vk5RUPLGSj7x6zbUDvdm07QlnAdutYpT'
blynk = Blynk(BLYNK_AUTH)

# Create BlynkTimer Instance
# timer = BlynkTimer()
blynk_timer = Timer()

GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIGGER_PIN, GPIO.OUT)
GPIO.setup(ECHO_PIN, GPIO.IN)

def setup():
    GPIO.setup(TRIGGER_PIN, GPIO.OUT)
    GPIO.setup(ECHO_PIN, GPIO.IN)

def measure_distance():
```



```

GPIO.output(TRIGGER_PIN, True)
time.sleep(0.00001)
GPIO.output(TRIGGER_PIN, False)

start_time = time.time()
while GPIO.input(ECHO_PIN) == 0:
    start_time = time.time()

while GPIO.input(ECHO_PIN) == 1:
    end_time = time.time()

duration = end_time - start_time
distance = (duration * 71500) / 2

return distance

@blynk_timer.register(interval=2)
def garbage_monitoring():
    distance = measure_distance()
    bin_level = calculate_bin_level(distance)

    blynk.virtual_write(0, distance)
    blynk.virtual_write(1, bin_level)
    print("Distance: %.2f cm" % distance)
    print("Bin Level: %d%%" % (bin_level * 100))

def calculate_bin_level(distance):
    if distance <= 5:
        return 1 # High bin level
    elif distance >= 25:
        return 0 # Low bin level

```

```

else:
    # Calculate the bin level as a percentage between empty and full distances
    bin_level = 1 - (distance - 5) / (25 - 5)
    return bin_level

@blynk.handle_event("connected")
def blynk_connected():
    print("Hi, You have Connected to New Blynk 2.0")
    print(".....")
    print("..... By NAbila Nizam .....")
    time.sleep(2)

def calculate_fill_level(distance):
    fill_level_percentage = 100 - ((distance - FULL_DISTANCE) / (EMPTY_DISTANCE -
FULL_DISTANCE)) * 100
    fill_level_percentage = max(0, min(fill_level_percentage, 100))
    return fill_level_percentage

def main():
    previous_fill_level_percentage = -1
    GPIO.output(TRIGGER_PIN, False)
    print("Ultrasonic sensor example")

blynk_timer = Timer()

#blynk_timer.setInterval(2, garbage_monitoring)

try:
    setup()
    #blynk_timer = Timer() # Create BlynkTimer instance

```

```

while True:
    blynk.run()

    #blynk_timer.set_interval(2, garbage_monitoring) # Schedule the garbage_monitoring
function to run every 2 seconds

    blynk_timer.run()
except KeyboardInterrupt:
    GPIO.cleanup()

try:
    while True:
        distance = measure_distance()
        fill_level_percentage = calculate_fill_level(distance)

        if fill_level_percentage != previous_fill_level_percentage:
            print("Trash can fill level: {}".format(fill_level_percentage))
            previous_fill_level_percentage = fill_level_percentage

        if fill_level_percentage >= 100:
            print("The trash can is full. Please empty it.")
            # Add code here to perform any necessary actions when the trash can is full

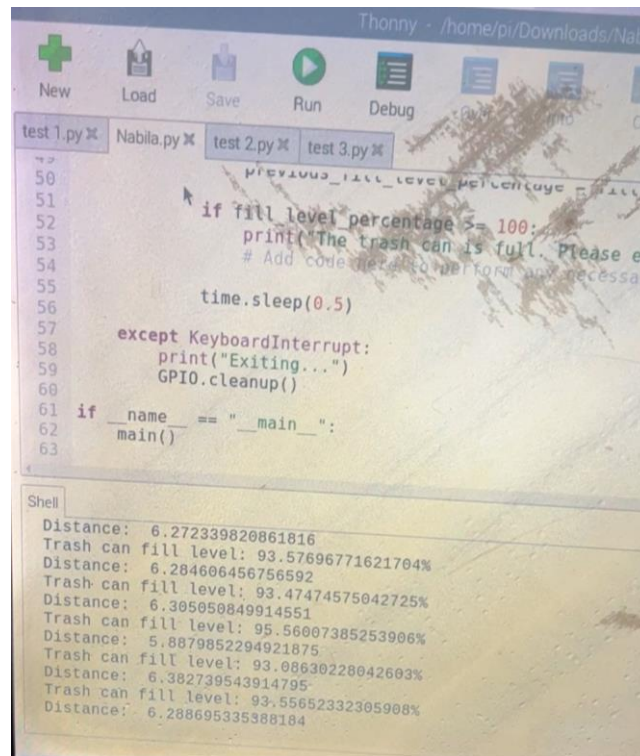
        time.sleep(0.5)

except KeyboardInterrupt:
    print("Exiting...")
    GPIO.cleanup()

if __name__ == "__main__":
    main()

```

Output:



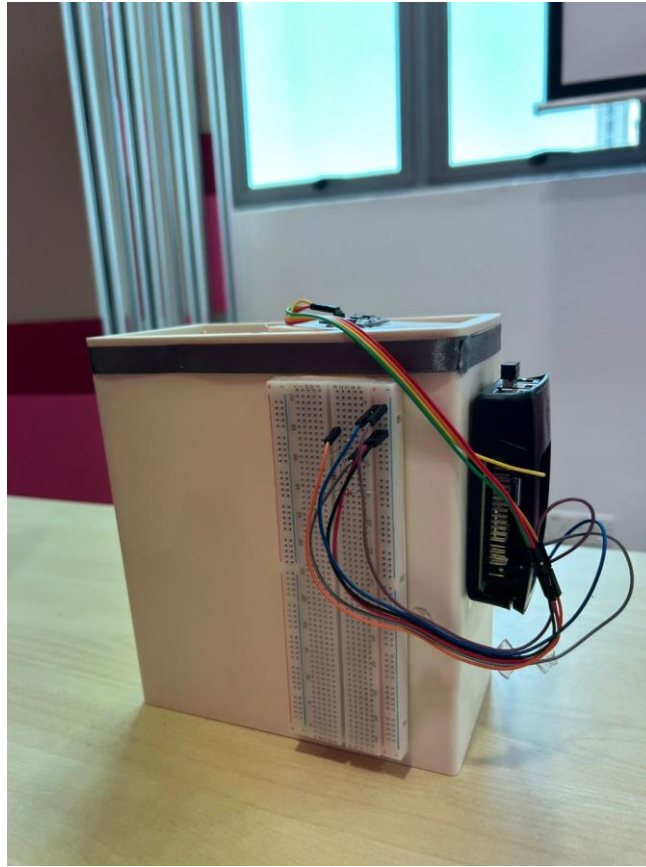
The image shows a screenshot of the Thonny IDE interface. The top menu bar includes 'New', 'Load', 'Save', 'Run', and 'Debug'. Below the menu, there are several tabs for Python files: 'test 1.py', 'Nabila.py', 'test 2.py', and 'test 3.py'. The main editor window displays Python code with line numbers 50 through 63. The code includes a loop that checks if the fill level percentage is greater than or equal to 100. If it is, it prints a message and sleeps for 0.5 seconds. An exception handler catches KeyboardInterrupt, printing 'Exiting...' and calling GPIO.cleanup(). The code is wrapped in a main function.

```
50
51
52     if fill_level_percentage >= 100:
53         print("The trash can is full. Please em
54             # Add code here to perform any necessar
55
56         time.sleep(0.5)
57
58     except KeyboardInterrupt:
59         print("Exiting...")
60         GPIO.cleanup()
61
62 if __name__ == "__main__":
63     main()
```

Below the code editor is a 'Shell' terminal window showing the output of the program. It displays a series of distance and fill level measurements:

```
Distance: 6.272339820861816
Trash can fill level: 93.57696771621704%
Distance: 6.284606456756592
Trash can fill level: 93.47474575042725%
Distance: 6.305050849914551
Trash can fill level: 95.56007385253906%
Distance: 5.8879852294921875
Trash can fill level: 93.08630228042603%
Distance: 6.382739543914795
Trash can fill level: 93.55652332305908%
Distance: 6.288695335388184
```

APPENDIX C
PROTOTYPE RUBBISH MONITORING SYSTEM



APPENDIX D

DATASET RUBBISH MONITORING SYSTEM

1	Time	Event Type	Name	Description	Distance	Bin Level	G	H	I	J	K	L	M	N
2	06/19/23 11:51:28 AM	OFFLINE	Offline		4.878032207	97.74862187								
3	06/19/23 11:51:03 AM	WARNING	Bin_is_Full	Rubbish bin is full	14.63757974	31.06071943								
4	06/19/23 11:50:02 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.896436	15.02545716								
5	06/19/23 11:49:02 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.80604838	15.67108303								
6	06/19/23 11:48:01 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.91121628	14.91988374								
7	06/19/23 11:47:01 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90006554	14.99953185								
8	06/19/23 11:46:00 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90806357	14.94240309								
9	06/19/23 11:45:00 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.86368323	15.25940549								
10	06/19/23 11:43:59 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.89279318	15.0514773								
11	06/19/23 11:42:59 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90327558	14.97660302								
12	06/19/23 11:41:58 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90098528	14.99296227								
13	06/19/23 11:40:58 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.90990446	14.92925384								
14	06/19/23 11:39:57 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.7868155	15.8084607								
15	06/19/23 11:25:45 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.93089174	14.77934469								
16	06/19/23 11:23:10 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.65969789	16.71644364								
17	06/19/23 11:22:10 AM	WARNING	Bin_is_Full	Rubbish bin is full	15.23843288	26.62637141								
18	06/19/23 11:17:45 AM	WARNING	Bin_is_Full	Rubbish bin is full	16.06702954	20.94978903								
19	06/19/23 11:07:12 AM	ONLINE	Online		7.485459	80.52130373								
20	06/19/23 11:07:10 AM	OFFLINE	Offline		10.00920317	63.37390256								
21	06/19/23 11:02:43 AM	WARNING	Bin_is_Full	Rubbish bin is full	8.42243284	75.55405114								
22	06/19/23 10:58:29 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.457182073	82.44869948								
23	06/19/23 10:58:23 AM	ONLINE	Online		7.445701957	82.53070031								
24	06/19/23 10:58:18 AM	OFFLINE	Offline		7.903948426	79.25751124								
25	06/19/23 10:57:28 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.361344496	82.65640256								
26	06/19/23 10:56:26 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.930640307	79.06685495								
27	06/19/23 10:54:58 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.897157967	79.30601452								
28	06/19/23 10:53:56 AM	WARNING	Bin_is_Full	Rubbish bin is full	7.869455814	79.50388704								
29	06/19/23 10:52:55 AM	WARNING	Bin_is_Full	Rubbish bin is full	9.06192151	70.98627493								