



Faculty of Computer Science and Information Technology

***SMART RAINWATER HARVESTING SOLUTION
FOR IRRIGATION IN OPEN SPACES***

Muhammad Fathullah Bin Zahari

**Bachelor of Software Engineering with Honours
2025**

SMART RAINWATER HARVESTING SOLUTION FOR IRRIGATION IN OPEN SPACES

MUHAMMAD FATHULLAH BIN ZAHARI

A dissertation submitted in partial fulfilment
of the requirement for the degree of
Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

PENYELESAIAN PENGUMPULAN AIR HUJAN PINTAR UNTUK IRIGASI DI RUANG TERBUKA

MUHAMMAD FATHULLAH BIN ZAHARI

Projek ini merupakan salah satu
keperluan untuk
Ijazah Sarjana Muda Kejuruteraan Perisian

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE: SMART RAINWATER HARVESTING SOLUTION FOR IRRIGATION
IN OPEN SPACES

ACADEMIC SESSION: 2 - 2024/2025

MUHAMMAD FATHULLAH BIN ZAHARI
(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (✓)

☐

CONFIDENTIAL

(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)

☐

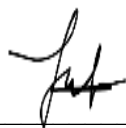
RESTRICTED

(Contains restricted information as dictated by the body or organization where the research was conducted)

☒

UNRESTRICTED

Validated by

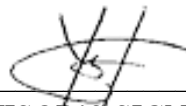


(AUTHOR'S SIGNATURE)

Permanent Address

Lot 2405, Lorong Tanjung 2, Perumahan
Dato' Shahbandar, 26600 Pekan Pahang

Date: 28 JULY 2025



.....(SUPERVISOR'S SIGNATURE).....

Noor Alamshah bin Bolhassan

Associate Professor

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

Note *Thesis refers to PhD, Master, and Bachelor Degree

**For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I hereby declare that this project titled "Smart Rainwater Harvesting Solution for Irrigation in Open Spaces" is my original work and has been conducted under the guidance and supervision of **Associate Professor Dr. Noor Alamshah Bin Bolhassan**. According to academic standards, all information, data, and references used in this project have been properly acknowledged and cited. I certify that all work done is truthful and accurate to the best of my ability, and I accept full responsibility for the information and conclusions presented in this project.



.....
Muhammad Fathullah Bin Zahari

17 January 2025

Faculty Of Computer Science and Information Technology

Universiti Malaysia Sarawak

ACKNOWLEDGEMENT

With this opportunity, I would like to express my deepest gratitude to who has contributed to successfully completing my Final Year Project. Firstly, I want to thankful to Almighty God, Allah for giving me strength and guidance throughout this journey. This achievement is impossible without His grace. Next, I would like to give my heartfelt appreciation to my supervisor **Associate Professor Dr. Noor Alamshah Bin Bolhassan** for his guidance, feedback and support. His expertise and advice have helped me a lot in my Final Year Project. Lastly, honourable mention to my family and my friends for always supporting me and giving me motivation to keep moving on. I could not finish it without their encouragement and wisdom. Finally, before I end my speech, I would like to acknowledge all the individuals or organizations that have contributed to my project. Your assistance has been invaluable in bringing this vision to life. May this work contribute to the betterment of agriculture in future endeavours.

ABSTRACT

As water scarcity and sustainability water management continues to be the global challenges, novel solutions are needed specifically for areas that rely on irrigation for agriculture. Hence, this project, Smart Rainwater Harvesting Solution is designed for irrigation in open spaces. By integrating the IoT technology in this project, it will help to enhance water efficiency and support environmental sustainability. The system is designed to collect and store the rainwater also monitoring the system through a user-friendly mobile application. The implementation of this project is improving water resources management for public parks, gardens and agriculture land. Through the smart technologies, it indicates that the project contributes to a greener and more sustainable future.

ABSTRAK

Masalah kekurangan air dan pengurusan air secara efektif terus menjadi cabaran global, dalam mencari penyelesaian baharu khususnya untuk kawasan yang bergantung kepada pengairan dengan tujuan aktiviti pertanian. Oleh itu, projek ini, Smart Rainwater Harvesting Solution, direka untuk pengairan di kawasan terbuka. Dengan mengintegrasikan teknologi IoT dalam projek ini, ia akan membantu meningkatkan kecekapan penggunaan air dan menyokong kelestarian alam sekitar. Sistem ini direka untuk mengumpul dan menyimpan air hujan serta memantau sistem melalui aplikasi mudah alih yang mesra pengguna. Pelaksanaan projek ini bertujuan meningkatkan pengurusan sumber air untuk taman awam, taman-taman bunga, dan tanah pertanian. Melalui teknologi pintar, ia menunjukkan bahawa projek ini menyumbang kepada masa depan yang lebih hijau dan mampan.

TABLE OF CONTENTS

DECLARATION	i
ACKNOWLEDGEMENT	ii
<i>ABSTRACT</i>	iii
<i>ABSTRAK</i>	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES.....	ix
LIST OF FIGURES	xi
CHAPTER 1: INTRODUCTION	1
1.1 Introduction.....	1
1.2 Problem Statement.....	2
1.3 Objectives	2
1.4 Methodology	3
1.5 Project Scope	3
1.6 Significance Project	4
1.7 Project Schedule.....	5
1.8 Expected Outcomes	8
1.9 Project Outline	8
1.10 Summary	10
CHAPTER 2: BACKGROUND STUDY.....	11
2.1 Introduction.....	11

2.2	Review on Existing System	11
2.2.1	Smart Kit Rainwater Harvesting System	11
2.2.2	IoT-Based Rainwater Harvesting System	14
2.2.3	Automatic Rainwater Harvesting System	16
2.3	Comparison between Existing System	19
2.4	Comparison between the Proposed System and the Existing System	24
2.5	Tools and Technology Used in the Development of the Proposed Systems	28
2.5.1	Software	28
2.5.2	Hardware	30
2.6	Summary	31
CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN		32
3.1	Introduction	32
3.2	Requirement Analysis	32
3.2.1	Questionnaire via Google Form	32
3.3	System Design	47
3.3.1	System Architecture	47
3.3.2	Conceptual Model	49
3.3.3	Functional Requirements	50
3.3.4	Non-functional Requirements	51
3.3.5	Flowchart	52
3.3.6	Use Case Diagram	53

3.3.7	Activity Diagram	54
3.3.8	Sequence Diagram	55
3.3.9	Class Diagram.....	56
3.3.10	User Interfaces	57
3.4	Hardware and Software Requirements	63
3.4.1	Hardware Requirements.....	63
3.4.2	Software Requirements	65
3.5	Summary	66
CHAPTER 4: IMPLEMENTATION		67
4.1	Introduction.....	67
4.2	System Overview	67
4.3	Hardware Implementation	68
4.3.1	Circuit Diagram	68
4.3.2	Harvesting Model.....	69
4.4	Software Implementation.....	72
4.4.1	ESP32 Coding's.....	72
4.4.2	“SmartHydro” Applications.....	75
4.5	Summary	80
CHAPTER 5: HARDWARE AND SOFTWARE TESTING		81
5.1	Introduction.....	81
5.2	Hardware Testing.....	82

5.2.1	Testing for Harvesting Model.....	82
5.2.2	Electronic Testing	84
5.3	Software Testing	87
5.3.1	Functional Testing	87
5.3.2	Non-Functional Testing	102
5.3.2.1	Reliability.....	102
5.3.2.2	Usability.....	105
5.3.2.3	Efficiency.....	109
5.4	Summary	112
CHAPTER 6: CONCLUSION AND FUTURE WORK		113
6.1	Introduction.....	113
6.2	Achievements.....	113
6.3	Limitations	114
6.4	Improvement work.....	114
6.5	Future Work.....	115
REFERENCES		116

LIST OF TABLES

Table 1.1 Project schedule for FYP 1.	5
Table 1.2 Project schedule for FYP 2.	7
Table 2.1 The component used in Automatic RWH System.	18
Table 2.2 The comparison table for the three existing systems.	19
Table 2.3 The comparison table for selected existing system and proposed system.	24
Table 2.4 The comparison table for ESP8266 and ESP32.....	25
Table 2.5 List of software used.....	28
Table 2.6 List of hardware used.....	30
Table 3.1 Respondents' suggestion for features.	42
Table 3.2 Responses on difficulties during implementation.....	46
Table 3.3 Functional requirements for the system.....	50
Table 3.4 Non-functional requirements for the system.	51
Table 3.5 Hardware specifications.....	63
Table 3.6 Software specifications.	65
Table 4.1 Pin connection for sensors and ESP32.	69
Table 4.2 Material used for the harvesting model.	70
Table 5.1 Modules to be tested in this project.	81
Table 5.2 Testing for authentication module.	83
Table 5.3 Testing for electronic module.	84
Table 5.4 Testing for authentication module.	88
Table 5.5 Testing for homepage module.	92
Table 5.6 Testing for report page module.....	94
Table 5.7 Testing for setting page.....	96

Table 5.8 Testing for threshold module.	98
Table 5.9 Testing for notification module.	100
Table 5.10 Testing for reliability module.	103
Table 5.11 Testing for usability module.	105
Table 5.12 Testing for efficiency module.	109
Table 5.13 Test summary.....	112

LIST OF FIGURES

Figure 1.1 Gantt Chart from 14 October 2024 until 13 December 2024.	6
Figure 1.2 Gantt Chart from 16 December 2024 until 26 January 2025.	6
Figure 2.1 Design of the first flush diverter. (Sunar et al, 2018)	13
Figure 2.2 Complete design of the system. (Sunar et al, 2018)	13
Figure 2.3 Designated storage tank.....	14
Figure 2.4 The process of the system.....	15
Figure 2.5 the prototype model for the system. (Lasya et al., 2020)	17
Figure 2.6 Show the block diagram for Automatic RWH System. (Lasya et al., 2020). ..	17
Figure 3.1 The percentage of respondents' gender.	33
Figure 3.2 The percentage of respondents' age.	33
Figure 3.3 The education level for every respondent.	33
Figure 3.4 The respondents' profession that taking this questionnaire.	34
Figure 3.5 The level of knowledge about rainwater harvesting system.	34
Figure 3.6 The level of understanding about agriculture practices.....	34
Figure 3.7 Show the interested level in using rainwater harvesting system.	37
Figure 3.8 Characteristic of the system.....	37
Figure 3.9 Area to use the system.	37
Figure 3.10 Budget for the system.	38
Figure 3.11 Level of automated for the system.	38
Figure 3.12 Level of importance for sustainability.....	38
Figure 3.13 Level of importance for user-friendliness.	39
Figure 3.14 Preferences type of system.	39
Figure 3.15 Suggested element for the system.	39

Figure 3.16 Suggested design for the system.....	43
Figure 3.17 Type of energy source for the system.....	43
Figure 3.18 Data type that should system provide.....	43
Figure 3.19 Preferred way for user to interact with the system.	44
Figure 3.20 Difficulties that would present.	44
Figure 3.21 System architecture for the system.....	48
Figure 3.22 Harvesting model for this project.	49
Figure 3.23 Flowchart of the system.....	52
Figure 3.24 Use case diagram for the system.	53
Figure 3.25 Activity diagram for the system.	54
Figure 3.26 Sequence diagram for the system.	55
Figure 3.27 Class diagram for the system.....	56
Figure 3.28 Login page for mobile app.	57
Figure 3.29 Sign up page for mobile app.....	58
Figure 3.30 Main dashboard page for mobile app.	59
Figure 3.31 Notification page for mobile app.....	60
Figure 3.32 Update profile page for mobile app.....	61
Figure 3.33 History report page for mobile app.	62
Figure 4.1 Show the system overview of this project.....	67
Figure 4.2 Circuit diagram for this project.	68
Figure 4.3 Side view of harvesting model.	71
Figure 4.4 Front view of harvesting model.....	71
Figure 4.5 Wi-Fi and Firebase Initialization in ESP32.....	72
Figure 4.6 Pin assignment for each sensor and time upload to the firebase.	73

Figure 4.7 Sensors reading code section.....	73
Figure 4.8 Update sensor values to RTDB.	74
Figure 4.9 Store sensor values to Firestore.....	75
Figure 4.10 Login page of SmartHydro.....	75
Figure 4.11 Signup page of SmartHydro.	76
Figure 4.12 Email verification of SmartHydro.....	76
Figure 4.13 Forgot password page of SmartHydro.....	77
Figure 4.14 Homepage of SmartHydro.....	77
Figure 4.15 Report page of SmartHydro.....	78
Figure 4.16 Threshold page of SmartHydro.	78
Figure 4.17 Setting page of SmartHydro.	79
Figure 4.18 Notification page of SmartHydro.	79
Figure 5.1 Graph for test summary.....	112

CHAPTER 1:INTRODUCTION

1.1 Introduction

Since time immemorial, water is nature's gift for good purpose in our life. As stated in the Al-Quran as in Sura Al-Anbiya verse 30, mentioned that the source of life is water. Water is a vital resource for all living things, including humans and plants, to exist. As time goes on, the total of population will increase as well high demanding percentages of water. Thus, water scarcity will become a major issue to meet daily needs. Meanwhile, climate change also can lead to water scarcity for example, drought. A natural phenomenon known as drought occurs when specific regions experience dry weather and a prolonged absence of precipitation, including rain, snow, and sleet. Climate change and rising global temperatures are changing rainfall patterns, which in turn affect the quality and geographical distribution of the world's water resources, even though rainfall can naturally change between different locations and times of year. (Lai, 2022)

To overcome the issue, water can be obtained from the rainwater as it is a free source. By incorporating a rainwater harvesting system, it can help to reduce water scarcity by collect all the water for the rainwater. This is an alternative way of agriculture activity as it is the most water used compared to other industries. The aim of this project is to design and implement Smart Rainwater Harvesting Solution for Irrigation in Open Spaces. The objective is to store the water in open spaces and use rainwater efficiently and sustainably. The system will consist of IoT (Internet of Things) technology to monitoring water levels, conductivity, turbidity, pH value in real time. The system will automatically run or can be controlled by the user through a dashboard or control system where it will display the data captured by the sensors. A model will develop in purpose to collect, store and distribute rainwater for irrigation purpose.

1.2 Problem Statement

Agriculture industry is the most water used industry as it needs the water for irrigation purposes. According to Smoor (2024), agriculture uses a lot of water, it accounts for nearly 70% of all water used globally. It shows how great the value of water used globally only for agriculture. However, this value can increase if the water is manually controlled by the human as it is including human error factor. This is because manual control of the water distribution is imprecise and inevitably lead to water waste.

Irrigation water may become contaminated as it cannot control or monitor by the human. It is including the water becoming too acidic for the plant make the plant die. This shows how important to monitor the contamination of the water so that it can prevent disaster happen. Depending on direct water supply is impossible to track and monitor the water level, conductivity and turbidity in real-time. This is the important requirement to ensure the water quality for irrigation purpose.

1.3 Objectives

The project aims to develop a smart rainwater harvesting solution using the Internet of Things (IoT). Based on the previous problem statements, followings are the project's objectives.

1. To design and develop a smart rainwater harvesting system utilizing advanced IoT technologies to monitor and collect rainwater for irrigation.
2. To implement a system that can monitor water level, pH value and turbidity in real-time by integrating IoT sensors.

3. To reduce dependency on domestic water systems by using the gathered rainwater for irrigation.

1.4 Methodology

The selection of software methodology is a crucial step in the development process. The Agile methodology is chosen for its flexibility, iterative nature, and ability to accommodate changes throughout the project lifecycle.

1. Requirement Analysis: Analysis the system requirements for the software and prioritized it.
2. Planning: Develop a clear plan to deliver the software including the features.
3. Implementation: Implement the complete system. Integrate the sensors, IoT and software.
4. Testing: Validate the system functions by testing it to ensure meets the specified requirements.
5. Deployment: Deploy the real system and put it into use.
6. Maintenance: Maintain the current system to ensure it continuously meet the expectations.

1.5 Project Scope

The following are the main priorities of this project development:

1. The development location for this project will be at Universiti Malaysia Sarawak, Kota Samarahan and data will be collect within in Universiti Malaysia Sarawak.
2. The project will include in developing a smart rainwater harvesting system with sensor to track water level, pH value and turbidity in real-time.

3. The project will include in develop a system that, in response to monitoring data, and control the system.
4. The project will include in implementing a mobile application which is a user interface to monitor and control the system in real-time.

1.6 Significance Project

Finding sustainable ways to manage and conserve water resources has become crucial as environmental pressures and population growth increase. By collecting and using rainwater, this project tackles this pressing issue and lessens dependency on conventional water sources like groundwater and municipal water supplies. By integrating real-time monitoring of water quality parameters like pH, turbidity, and water levels, this project offers substantial benefits beyond rainwater harvesting.

By enabling users to monitor vital information, this feature reduces the risks associated with low water quality and guarantees that the water satisfies requirements for irrigation. In addition, the system helps to prevent possible agricultural losses from contaminated water by offering real-time insights. It is a comprehensive approach to water management and conservation because of its capacity to monitor water quality, adjust to changes in the environment, and lessen user workload. By putting such a system in place, we can help ensure that water resources are used responsibly in the future, meeting both environmental sustainability and human needs.

1.7 Project Schedule

Table 1.1 Project schedule for FYP 1.

No.	Tasks	Start	Finish	Working Days
1.	Approach Lecture for Supervising the FYP.	14/10/2024	15/10/2024	2 days
2.	Finalising/Review Brief Proposal	16/10/2024	19/10/2024	4 days
3.	Submission of the Approved Brief Proposal	20/10/2024	20/10/2024	1 day
4.	Feedback and Comment from Reviewer/Examiner	28/10/2024	1/11/2024	5 days
5.	Finalising/Review Full Proposal	4/11/2024	12/11/2024	7 days
6.	Submission of Full Proposal	14/11/2024	14/11/2024	1 day
7.	Finalising/ Review Chapter 1	15/11/2024	20/11/2024	4 days
8.	Submission of Chapter 1	21/11/2024	21/11/2024	1 day
9.	Finalising/ Review Chapter 2	25/11/2024	12/12/2024	14 days
10.	Submission of Chapter 2	13/12/2024	13/12/2024	1 day
11.	Finalising/ Review Chapter 3	16/12/2024	2/1/2025	14 days
12.	Submission of Chapter 3	5/1/2025	5/1/2025	1 day
13.	Finalising/ Review of FYP 1 Final Report and Paper for Assessment	6/1/2025	15/1/2025	8 days

14.	Final Submission of FYP 1 Final Report and Paper for Assessment	17/1/2025	17/1/2025	1 day
15.	Reserved for Presentations if any	24/1/2025	26/1/2025	2 days

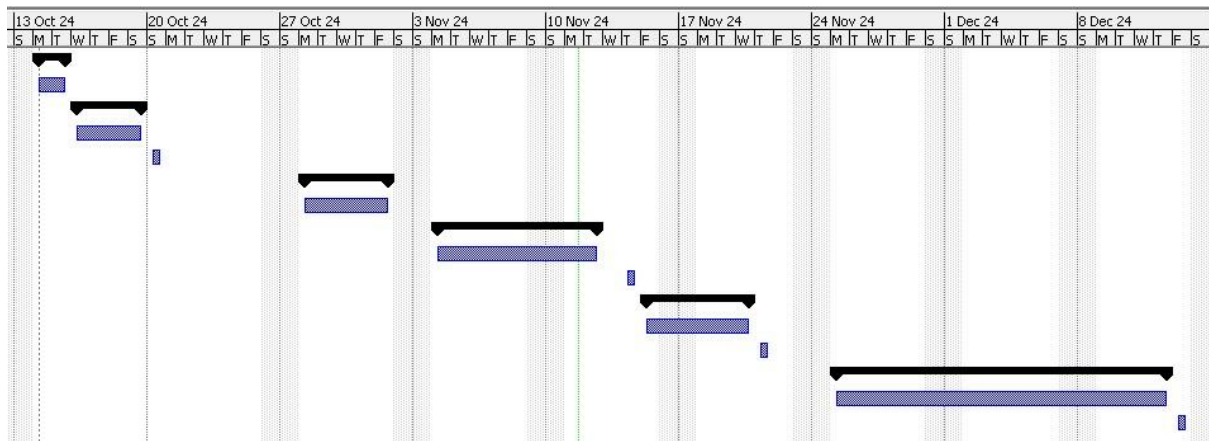


Figure 1.1 Gantt Chart from 14 October 2024 until 13 December 2024.

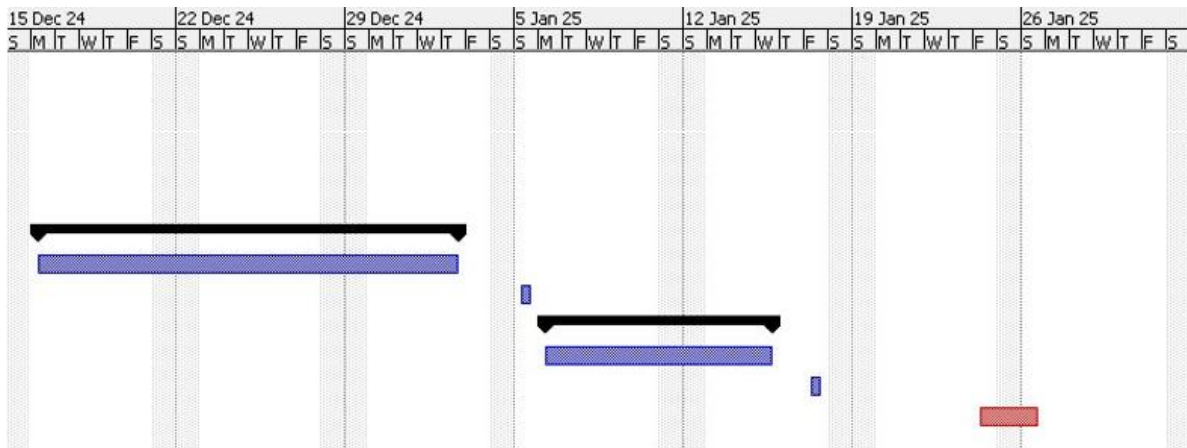


Figure 1.2 Gantt Chart from 16 December 2024 until 26 January 2025.

Table 1.2 Project schedule for FYP 2.

No.	Tasks	Start	Finish	Working Days
1.	Submission of the revised structure of FYP report.	1/4/2025	1/4/2025	1 day
2.	Finalising/Review of Chapter 4	13/5/2025	13/5/2025	2 days
3.	Submission of Chapter 4.	15/5/2025	15/5/2025	1 day
4.	Finalising/Review of Chapter 5, 6 and Abstract for Paper.	27/5/2025	29/5/2025	2 days
5.	Submission of Chapter 5, 6 and Abstract for Paper.	31/5/2025	31/5/2025	1 day
6.	Finalising/Review of FYP 2 Full Report and Paper.	7/6/2025	9/6/2025	2 days
7.	Submission of FYP 2 Full Report and Paper.	10/6/2025	10/6/2025	1 day
8.	Submission of FYP 2 Final Report (source code, installation kits, user manual) and Paper for Assessment	23/6/2025	23/6/2025	1 day

1.8 Expected Outcomes

The system is anticipated to accomplish a few important goals by the project's conclusion, which will greatly advance sustainable water management techniques. Effective rainwater harvesting for irrigation will be one of the main accomplishments, lowering dependency on traditional water sources like domestic tap water. This strategy is an economical option for open-space and agricultural irrigation since it reduces the expenses related to water procurement. Additionally, the system will be able to track vital parameters like water level, conductivity, turbidity, and pH value in real-time with the integration of modern IoT technology. Users will be able to control the system from almost anywhere thanks to the user interface's real-time insights. This will lessen their physical workload and save them time. Furthermore, the system's adaptability and scalability are demonstrated by the fact that it can be used in a variety of open areas, such as parks, farms, and public landscapes.

1.9 Project Outline

1. Introduction

In this chapter, it will introduce the reader about the project by providing an overview of the issue being addressed, for example its objectives, the project scopes and the significance of the project. It will describe why water scarcity is a serious problem, especially in agriculture and how rainwater harvesting system can be a long-term fix. The project's goals and expected impact will conclude in the chapter's conclusion.

2. Literature Review

In this chapter, it will review the existing research and system related to the rainwater harvesting system. It identifies the knowledge gaps and highlights how the project can build with different solutions from existing solutions. The sources are published material such as journal, articles and conference paper. End of the chapter, it will establish a strong foundation for the project.

3. Requirement Analysis and Design

This chapter includes the methods and techniques used to develop the system. It will identify the project's approach obtained the detailed requirements specification for the system. This chapter will discuss the tools and technologies required for the project. The chapter also will include how data will be collected and analysed for the system performance.

4. Implementation

This chapter will explain the hardware and software implementation for this project. It will consist of system overview, harvesting model, circuit diagrams, sensors coding and mobile application. The objective is to give an understanding of the process implementation in this project.

5. Hardware and Software Testing

This chapter will highlight the testing that have been done to the hardware and software after the implementation. This section is to prove all the functionality implemented works properly.

6. Limitations and Future Work

This chapter will include the limitations or challenges faced in implementing this project. It also will include the future work can be done to improve the project.

1.10 Summary

This chapter describes how population growth and climate change are contributing to the worldwide problem of water scarcity. It highlights the importance of water as a scarce resource and the potential of rainwater harvesting as a sustainable solution for the world's largest water-consuming industry, agriculture. The project's objective, which is to use the Internet of Things to monitor and harvesting rainwater for irrigation, is introduced in this chapter. The system will use a mobile application to monitor parameters like pH, turbidity, and water level.

CHAPTER 2:BACKGROUND STUDY

2.1 Introduction

In this chapter, three existing rainwater harvesting systems will be studied and reviewed. These systems will be similar to the proposed system. The review will cover the functionality, advantages, and disadvantages of each system. Additionally, the tools and technologies used in these systems will also be examined in this chapter.

2.2 Review on Existing System

Three existing systems similar to the proposed rainwater harvesting system will be examined to identify the knowledge gaps.

2.2.1 Smart Kit Rainwater Harvesting System

Smart Kit Rainwater Harvesting System (RWH) focus on providing the user-friendly, easy to install and affordable solution to harvesting rainwater. The system is designed for household-use, hence benefiting the rural communities in accessing clean water. The designed system applies simple yet efficient components, making it appropriate for a wide range of households and agricultural purposes.

The system is consisting of three main components, water storage tank, first flush diverter and conveyance system. The rectangular design perfectly fit for the system to ensure space saving and easy installation. In addition, the water tank is designed with inflow and outflow pipe. The inflow pipe is to collect the rainwater and outflow pipe to remove the excess water. The tank size has been used in this system is determined by using the formula from Gulf Island Water Connection in 2015. The formula uses rainfall data and the roof area. Before do the calculation, the monthly precipitation is needed to be recorded. Hence, the calculation obtained in the article are:

1 inch of Rain Falling On,

1 Square Foot of Surface = 0.52 gallons

Therefore,

(7.8 inch × 438 square foot) × 0.52 = 1776.528 gallons

To account for the appropriate runoff,

coefficeint below include it in the equation (used 95%):

(7.38 × 438) × 0.95 × 0.52 = 1687.7 gallons ≈ 1700 gallons

One month of water collection is 1700 gallons ≈ 6435.2 Litre

Assume one day of water collection,

$$\frac{1700}{30} = 56.6 \text{ gallons} \approx 60 \text{ gallons @ 230 liter}$$

Hence, the best size of water tank is 60 gallons.

Next, is first flush diverter where the purpose is to prevent the initial contamination from entering the water tank. The first flush diverter function is to replace the rainwater filter. This component will divert the first flow which may contain any heavy sediment and dirt from the roof to ensure cleaner water in rainwater storage. The features are consisting of a floating ball mechanism to close the pipe when the diverter is full and allows the clean water flows into the tank. To maintain the diverter, a screw cap is attached at the bottom of the pipe that can be manually. It can be open to discard all the heavy sediments and dirt from the pipe. To determine the capacity of the divert needed, a calculation will be performed based on the roof size and pollution factor.

Roof Area × Pollution Factor = Liters to be diverted

Therefore, 40.656 × 0.2 L = 8.13 L

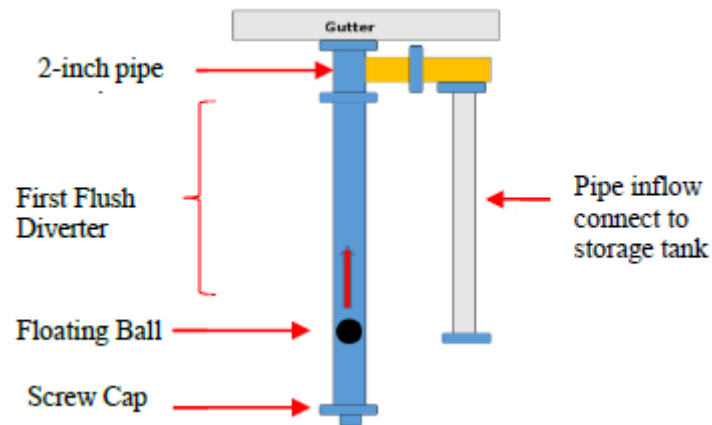


Figure 2.1 Design of the first flush diverter. (Sunar et al, 2018)

Lastly, is conveyance system where it includes gutters, downpipes, and connectors made of UPVC material. It is designed to increase the efficiency of harvesting rainwater from the roof catchment area to storage tank. Therefore, the installation of this Smart Kit Rainwater Harvesting System only requires the installation of piping work and storage water tank. It is easy to assemble with limited tools, hence allowing users to install it without skilled labour.

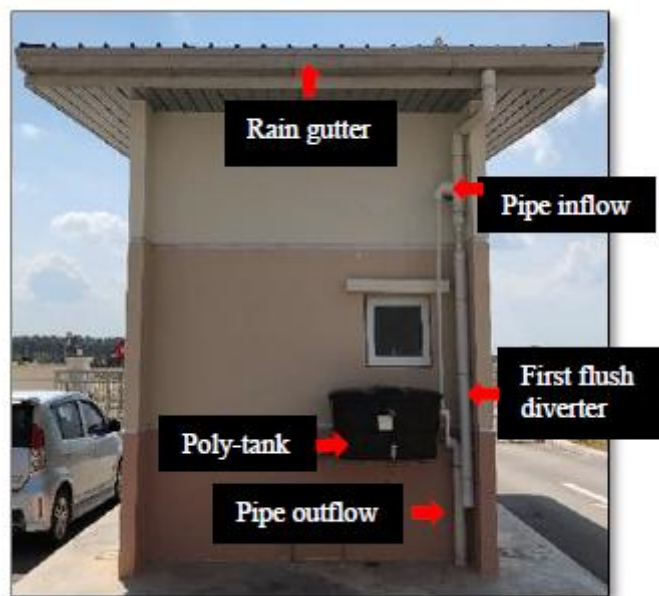


Figure 2.2 Complete design of the system. (Sunar et al, 2018)

2.2.2 IoT-Based Rainwater Harvesting System

IoT-Based Rainwater Harvesting System focusing in preventing the water scarcity issues by integrating IoT-based user interaction, real-time quality monitoring, and rainwater collection. The system highlights the importance of ensuring water quality which is a significant limitation in traditional rainwater harvesting methods.

The key features of the system that need to be focused firstly is, rainwater segregation based on pH value of the water. The system will measure the pH value of the rainwater using pH sensor to determine as potable ($\text{pH} \geq 5$) or acidic ($\text{pH} < 5$). The potable and acidic water will be stored in two different tanks to preventing the water contaminated. The system also implies the IoT integration via NodeMCU (IoT platform) to monitor and control the system remotely. All the data such as pH level, rainfall status and water distribution will be uploaded to a webpage, hence it will show the real-time data to the user.

The system highlights the automation of water distribution where a servo motor will control the water flow to separate rainwater based on its quality. Meanwhile, the system uses a rain sensor to detect rainfall intensity and ultrasonic sensor to measure tank water levels. The two-tank structure allows the system to store two different types of water which are potable water and another for acidic water. This design is to make sure the effectiveness during the segregation process and utilization of collected rainwater.

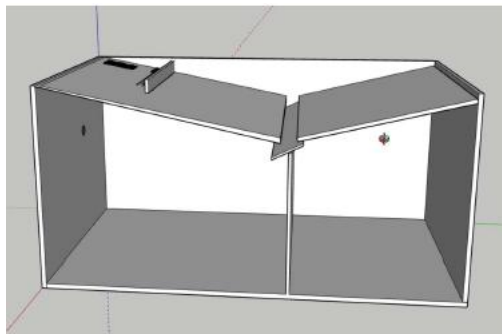


Figure 2.3 Designated storage tank.

Moreover, the components used in the system is pH sensor where the function is to measure the pH value of the rainwater to determine the suitability for potable use. This component is important in segregating water based on its quality. Next is, rain sensor where it detects the rainfall and measure its intensity to allowing the system to start collecting water automatically. Then, the ultrasonic sensor where it measures the water level in the tank using ultrasonic waves. The transmitted ultrasonic wave will be reflected from the target hence it will receive the wave back. Next is, NodeMCU where it will act as the control unit for the system. All the sensor and motors will integrate via this component to enable IoT functionality. Lastly, is servo motor where it controls the hinge mechanism that directs rainwater to suitable tank based on the pH reading from the pH sensor.

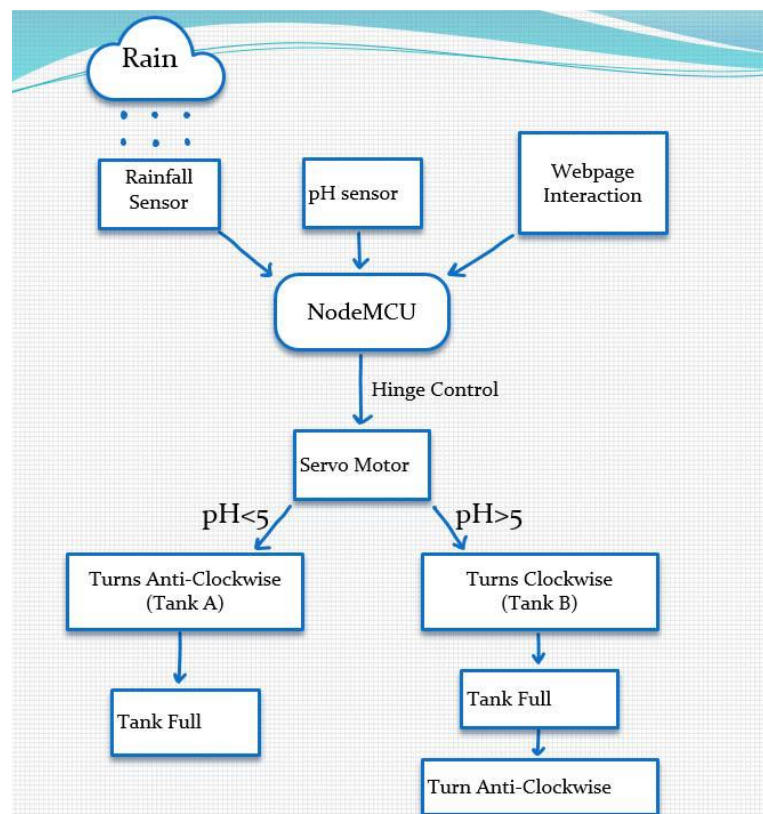


Figure 2.4 The process of the system.

2.2.3 Automatic Rainwater Harvesting System

Automatic Rainwater Harvesting System focuses on developing a model that fully automated rainwater harvesting system by integrating the IoT technology such as pH, conductivity, and turbidity sensor to examine the water quality. The system aims to address the limitations older system by offering automation and efficiency. The system has improved the existing model by integrating more sensors to the model such as pH sensor, water conductivity sensor and water turbidity sensor. It is to ensure a good quality of water is stored.

The key features that need to be focused on the system firstly is, automatic water collection where the system will automatically collect the rainwater when the tank reach a certain level. Sensor will detect the water levels to activate the filtration process automatically. The system prioritizes the water quality monitoring by includes sensors to measure the water quality such as pH, turbidity and conductivity. This monitoring phase ensure the water is suitable for used in future. The system also highlights the IoT integration where the data from the sensors will be sent to a web interface via ESP8266 Wi-Fi module. This allows the user to monitor the water quality by real-time data.

In addition, the system also improves the existing model by adding a filtration system where it is uses a reverse osmosis filter to purify the collected rainwater hence, it enhances the water quality for variety purposes used. In addition, two storage tanks enable the system to store the clean and contaminated water at the same time. As the contaminated water reach a certain water level, the filtration process will be automatically start. The contaminated water will go through the filtration process and stored into the clean water tank. While this process happens, the contaminated water can still be stored in the tank. Lastly, the system uses ThingSpeak as the monitoring platform. The reading from sensors will be sent to the platform, thus it can be monitored from there.

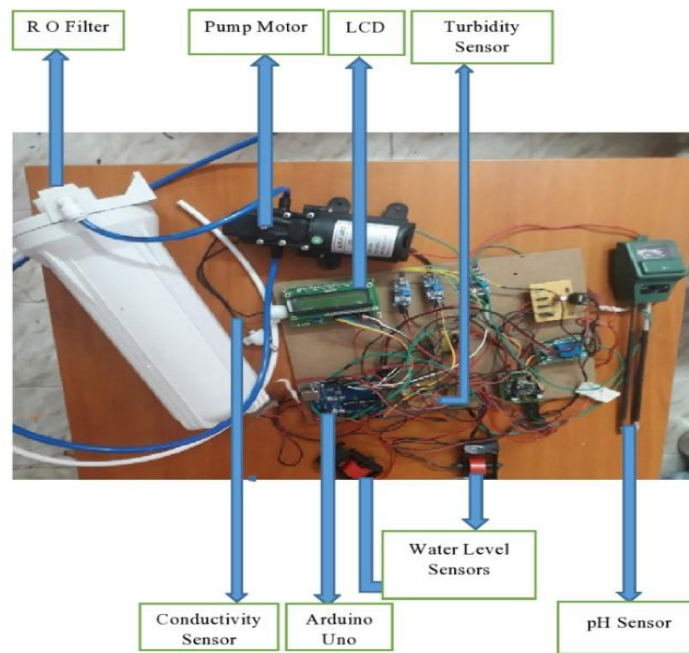


Figure 2.5 the prototype model for the system. (Lasya et al., 2020)

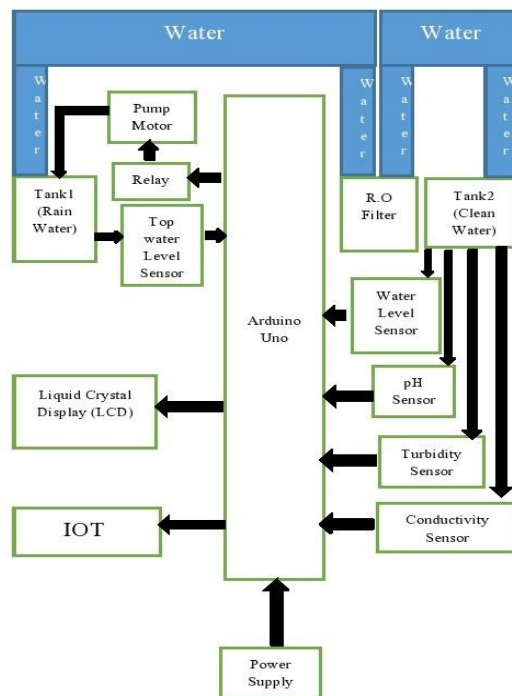


Figure 2.6 Show the block diagram for Automatic RWH System. (Lasya et al., 2020).

Table 2.1 The component used in Automatic RWH System.

Components	Functionality
Microcontroller (Arduino Uno)	The central controller that processes sensor data and controls the system's functions.
Water Level Sensors	Used to detect the water levels in the tanks (Tank1 for collection and Tank2 for storage).
pH Sensor	Measures the acidity or alkalinity of the water.
Turbidity Sensor	Measures the clarity of the water by detecting suspended particles.
Conductivity Sensor	Measures the water's ability to conduct electricity, which is related to the ion concentration in the water.
Reverse Osmosis Filter	Purifies the water by removing dissolved salts, bacteria, and other impurities.
Pump Motor	Moves water from Tank1 to the filtration system and then to Tank2.
Relay Switch	Controls the pump motor based on sensor readings.
LCD	To monitor the readings from sensors.
ESP8266 Wi-Fi Module	Enables the system to send data to a website, allowing users to remotely monitor the system.

2.3 Comparison between Existing System

Based on the three existing systems above, hence the comparison is shown below to investigate the knowledge gap between system.

Table 2.2 The comparison table for the three existing systems.

Features	Smart Kit RWH System	IoT-Based RWH System	Automatic RWH System
IoT Integration	No IoT	Basic IoT via ESP8266 module.	Full IoT integration with a webpage.
Water Quality	Stores untreated water.	Stores water by pH value.	Stores purify water through reverse osmosis filtration.
Automation	Manual cleaning for first flush diverter required.	Automated with IoT monitoring.	Fully automatic filtration and storage.
Scalability	Limited to small installations.	Scalable with IoT capabilities.	Suitable for small to medium setups.
Affordability	High	Moderate	Low
Purpose	Non-potable uses.	Potable and non-potable uses.	Potable and non-potable uses.
Monitoring system	No	No	Yes. Monitor through ThingSpeak platform.

Based on the table above, the three existing systems has been compared by specific features to analyse the differences and knowledge gap for the purposed system. The first feature is IoT integration whereas shown in the table the Smart Kit RWH System is lack with IoT integration. There is no any IoT technology used or implement in the system. This is because this module is focusing more on the affordability and simplicity to implement the system. Meanwhile, the IoT-Based RWH System employ basic IoT functionality in the system and Automatic RWH System come out with its real-time monitoring and user interface. According to Gitlin (2023) IoT integration allows user to catch up with real-time data from device, besides increasing the efficiency of the system, reducing cost and enhance the decision-making process. By implementing, the IoT devices in the system, it can enable the automation and predict the risk that may come up in the future.

Next is, water quality management where the Smart Kit RWH System is using the first flush diverter mechanism for the system. This mechanism only prevents the sediment and dirt from the roof to enter the tank and does not treat the water for potable use. For IoT-Based RWH System, its use a pH sensor to segregate potable and non-potable water, emphasizing drinking water safety. The Automatic RWH System have used pH, turbidity and conductivity sensors to ensure the water is highly suitable for diverse application such as household use. According to Caburao (2024) water quality management is important to ensure the water quality meet the standards for specific uses such as irrigation and drinking water. The good water quality management can improve the operational efficiency and reduce the need for maintenance and can maintain good health and safety to the water consumer. According to the World Health Organization, water is one of the easiest mediums for bacteria and virus to spread among the human hence, it is important to take care of good water quality management.

In addition, automation is one of the key features in the system where the Smart Kit RWH System relies on manual intervention for flushing and filtration hence the maintenance of the system needs humans work handle while IoT-Based RWH System uses automation to differentiate the acidic and alkaline water and provide real-time updates, enhancing the convenience. The Automatic RWH System is fully automated via microcontroller managing the water flow and filtration process. Automation gives option to the user to easily handle the system and control it through their mobile or web application. According to the research Coyne et al. (2019) it proved that about 50 percent of tasks in our daily life can be automatable with the current technology and the rest 50 percent is non automatable tasks such as decision-making steps, interactions and handoffs. Therefore, to take the advantage of convenience with the current technology the automation should be apply in the system to make life easier.

The scalability for three existing systems is different and have been shown in the table. The scalability is important to evaluate how effectively the system can in adapt in capacity and functionality without effecting the performance. The key that needs to be considered in measuring the scalability is performance of the system. The system ability needs to meet the water demand as it scaled for. When supply is not constrained by catchment area or tank volume and the supply of water from rainfall closely equates with the demand for water from the harvesting system, performance will typically be higher. Hence, it is important to determine the purpose of uses in harvesting the rainwater. As stated in the table above Smart Kit RWH System is suitable for limited to small installations only as it only uses fitting mechanism to harvest the rainwater.

The IoT-Based RWH System is scalable with IoT capabilities where adding more IoT devices will not affect the system performance. Therefore, it become easier and more effective to implement it in a larger coverage area. The Automatic RWH System is suitable for small to medium setups only as it involves a limited number of components and area of operation. Hence, it shows the scalability level for every system in the table. Affordability is also important features to analyse in the system as it effects the system to keep sustain in the future. The key aspect needs to be focused in evaluating the affordability is initial cost. The process in determining the initial cost is important as it create a budget to build the system. Maintenance cost also important key in determine the affordability of the system as it deals with the unexpected damage occurred.

The third key aspect is the accessibility where it is economical for individuals and communities to build it. The Smart Kit RWH System offered low budget model as it is not using any IoT technology and just using modular components besides the maintenance cost for this system is low as it only need human skilled to open and clean the first flush diverter by user itself. The system also affordable for individual or community with limited budget. IoT-Based RWH System use moderate budget because it is using a few basic IoT devices in the system such as pH sensor to measure the pH value of water. Even though, the system offered a moderate budget, but it still considered affordable as it also came up with good functionality in the system. Meanwhile, the Automatic RWH System use high budget to implement the system as it is including a lot of IoT devices and tool such as pH sensor, turbidity, water conductivity and reverse osmosis. The maintenance cost may be high in the future as it need to cover the cost to clean the filtration.

In addition, the key feature needs to be analysed is the purpose of the system in harvesting the rainwater. Knowing the purpose of the system is important to evaluate how well the system meets its intended goals and objectives. The purpose of the Smart Kit RWH System is to harvest rainwater for non-potable uses, such as irrigation. The system achieves this objective through a first flush diverter, which prevents sediment and dirt from entering the tank. Since the harvested water is intended for non-potable purposes, there is no need for additional water treatment, unlike other systems designed for potable use. Next is the purpose of the IoT-Based RWH System is for potable uses and non-potable uses. According to the model, the system only differentiates the water based on its pH value.

Therefore, assuming the water is suitable for potable use is inappropriate, as it requires specific treatment to ensure its cleanliness. However, for non-potable uses, the water is very suitable. Lastly, the purpose of the Automatic RWH System is for potable and non-potable uses. According to the model, the water will go through the filtration process where it uses reverse osmosis mechanism to ensure the water is clean. Hence, the water is suitable for potable use such as washing, cooking and drinking. For non-potable uses, the water is very suitable. Lastly, is monitoring system. The Smart Kit RWH system did not have any monitoring system due to no IoT involved in the system. Meanwhile, IoT Based RWH System also do not have any monitoring system even though it involved IoT technology. This is because the devices only scan the pH value does not store the data. For Automatic RWH System, it does have monitoring system by using ThingSpeak platform to show the readings from the sensors. It can only monitor the readings but cannot control the system.

In conclusion, based on the explanation and justification for each system outlined above, the Automatic RWH System stands out as a high-tech solution that integrates IoT devices, such as pH, turbidity, and conductivity sensors, for effective water quality management. The system also features an advanced mechanism that ensures clean water for both potable and non-potable uses. It provides real-time data and a user-friendly web interface to keep users updated. Although the budget is relatively high, the system offers exceptional functionality, ensuring an efficient and effective rainwater harvesting process.

2.4 Comparison between the Proposed System and the Existing System

Based on the selected existing system above, hence the comparison is shown below to investigate the knowledge gap between the proposed system.

Table 2.3 The comparison table for selected existing system and proposed system.

Features	Automatic RWH System	Smart RWH System (Proposed System)
IoT Integration	Full IoT integration with a webpage. (ESP8266 module)	Full IoT integration. (ESP32 module)
Water Quality	Stores purify water through reverse osmosis filtration.	Stores untreated water.
Automation	Fully automatic filtration and storage.	Fully automatic.
Scalability	Suitable for small to medium setups.	Scalable with IoT capabilities.

Affordability	High	Moderate
Purpose	Potable and non-potable uses.	Non-potable uses.
Monitoring system	Monitor through ThingSpeak platform.	Monitor and control through mobile app.

Table above show the comparison between the existing system with the proposed system. Both systems utilize the IoT technology and offers an intuitive user interface that makes it convenient and accessible for end users. This is a clever, contemporary approach to water conservation since it enables users to monitor the system from any location. However, the proposed system uses ESP32 and Automatic RWH System used ESP8266 as the microcontroller of the system.

Table 2.4 The comparison table for ESP8266 and ESP32.

Features	ESP8266	ESP32
Processing Power	Single core, 80-160 MHz	Dual-core, 160-240 MHz
RAM	50 KB SRAM	520 KB SRAM
Connectivity	Wi-Fi only	Wi-Fi and Bluetooth
I/O Pins	17 Pins	34 Pins
Analog Inputs	1 (10-bit ADC)	18 (12-bit ADC)
Power Consumption	Higher power consumption	Energy efficiency
Application	Simple IoT	Complex IoT

These two models have a different performance based on the table above. The processing power for ESP32 is much faster than ESP8266 where it provides dual-core processors with a clock speed up to 240 MHz. This make the ESP32 is capable of handling demanding applications compared to ESP8266. Next is the RAM where ESP32 provided more RAM which enables better performance for advanced projects. For connectivity, ESP32 have more versatility as it supports both Wi-Fi and Bluetooth compared to ESP8266 supports only Wi-Fi. ESP32 offered more input and outputs pins and flexibility for connecting multiple sensors and peripherals. Higher analogue inputs allow the better resolution and making it ideal for projects requiring multiple analogue sensors where it can see from the table ESP32 have more analogue input. For power consumption, ESP32 is better due to its energy-saving features. Lastly is application where ESP32 is suitable for complex IoT compared to ESP8266 it is limited to simpler tasks.

Although storing untreated water might seem like a drawback, the proposed Smart RWH System is made especially for non-potable applications like irrigation. This promotes sustainability and resource conservation by lowering the demand on treated municipal water supplies and guaranteeing that clean water resources are set aside for cooking and drinking. By automating non-potable water storage for tasks like irrigation, the proposed Smart RWH System excels in helping users save money, time, and effort while encouraging environmentally friendly habits in daily life. The modular IoT integration of the proposed Smart RWH System makes it highly adaptable. This implies that it can be readily extended to accommodate larger structures, such as community-based rainwater harvesting systems. Larger communities will benefit in the long run from the system's ability to scale in tandem with rising user demands.

Because of its affordability, the proposed Smart RWH System is a desirable choice for communities, small businesses, and homes looking for economical water harvesting solutions. By concentrating on non-potable water uses, costly filtration systems are not as necessary, guaranteeing a balance between affordability and usefulness. The non-potable applications of the proposed Smart RWH System include irrigation. Although this might seem insufficient, it effectively tackles the problem of water scarcity by drastically lowering reliance on treated water for applications where purified water is not required. This specific goal minimizes resource waste while enabling the system to effectively support sustainable water practices. Lastly, Smart RWH System use monitoring system thorough mobile app. The mobile app can monitor the readings from the sensors also control the sensors whether to turn it on or off. Compare to the Automatic RWH System it can only monitor the readings but cannot having control through the platform.

2.5 Tools and Technology Used in the Development of the Proposed Systems

This section will review the tools and technologies used in the development of the proposed system.

2.5.1 Software

This section will review the software used in the development of the Smart Rainwater Harvesting System.

Table 2.5 List of software used.

Software	Description
Arduino IDE	The Arduino Integrated Development Environment (IDE) is open-source software used to write, compile, and upload code to microcontroller such as ESP32. It provides user-friendly platform for students and professionals that working on embedded systems and IoT projects.
Figma	Figma is an online design and prototype development tool for collaborative design projects, user interfaces (UI), and user experiences (UX). Because of its simplicity of use, real-time collaboration capabilities, and capacity to optimize the design-to-development process, it is highly favoured by developers.
Android Studio	The official Integrated Development Environment (IDE) for creating Android apps is called Android Studio. It is built on top of the robust Java IDE IntelliJ IDEA and is created by Google. Android Studio gives developers a full suite of tools to effectively design, develop, test, and debug Android apps.

Firestore	A cloud-based platform developed by Google that provides tools and backend services to assist developers building mobile and web efficiently. Developers can create great user experience without worrying to set up and maintain servers.
-----------	--

2.5.2 Hardware

This section will review the hardware used in the development of the Smart Rainwater Harvesting System.

Table 2.6 List of hardware used.

Devices	Functionality
ESP32	- Serves as the system's main processing unit. - Makes Wi-Fi connectivity possible for Internet of Things functions like data transfer to a mobile app and real-time monitoring. - Handles sensor data processing and manages the automated features of the system.
Ultrasonic Sensor	- To determine the distance needed to check the water level in tanks.
Connectors	- Connect parts such as sensors, power supplies, and the microcontroller (ESP32).
pH Sensor	- Measures the acidity or alkalinity of the rainwater.
Rain Sensor	- Detects the presence and intensity of rainfall.
Water Turbidity Sensor	- Measures the cloudiness of the water by detecting suspended particles.
Breadboard	- Provides a platform for solderless circuit component testing and connection.

2.6 Summary

This chapter concludes with a review of the three existing systems. To come up with fresh concepts to incorporate into the suggested system, the features of the current system are compared. Additionally, the technology and tools utilized to create this proposed system are reviewed and studied. There aren't many comparable coaching systems on the market that coach players via video processing. As a result, the suggested system is crucial to the application of rainwater harvesting system in the future.

CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

This chapter will focus on the analysing the requirements and designing the system for the Smart Rainwater Harvesting Solution for Irrigation in Open Spaces. The analysis process is important to ensure that the system meet the requirement from the user and align with the project's objectives. This chapter will include the system architecture, designed model and system functionality to describe the foundation of an effective and efficient system. Additionally, this chapter also specifies the visual models such as data flow diagrams (DFD), use case diagram and entity relationships diagram ensuring the flow of the system presentable and deliverable to the user.

3.2 Requirement Analysis

The requirement analysis is to identify the functionalities, features, and constraints of the Smart Rainwater Harvesting Solution for Irrigation in Open Spaces. It ensures the system align with the user needs and project objectives to prevent user dissatisfaction and low adoption. The analysis was conducted through a questionnaire using Google Form tool to collect input on desired system behaviour and performance.

3.2.1 Questionnaire via Google Form

Questionnaire is one of data collection instrument that commonly used during the requirement analysis phase. In this project, open-ended and closed-ended questions has been implemented in the questionnaire to gather responses from the respondents. There are three sections included in the questionnaire which are demographic information represents the respondent background, user preferences represented the respondents needs, preferences and expectations about the proposed system and recommendations for the proposed system.

Section A: Demographic Information

Gender
43 responses

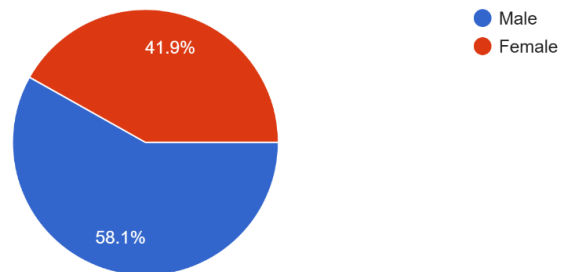


Figure 3.1 The percentage of respondents' gender.

Age
43 responses

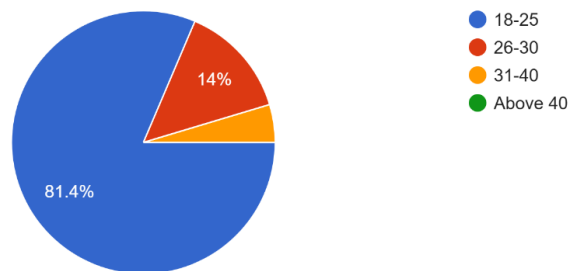


Figure 3.2 The percentage of respondents' age.

Education Level
43 responses

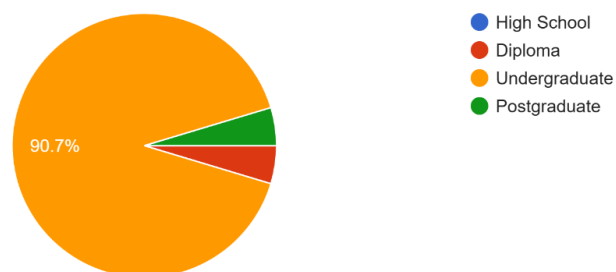


Figure 3.3 The education level for every respondent.

Profession
43 responses

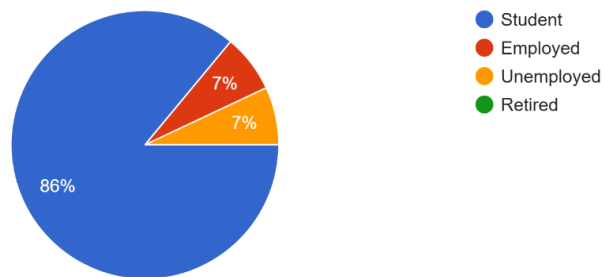


Figure 3.4 The respondents' profession that taking this questionnaire.

Knowledge about Rainwater Harvesting Systems
43 responses

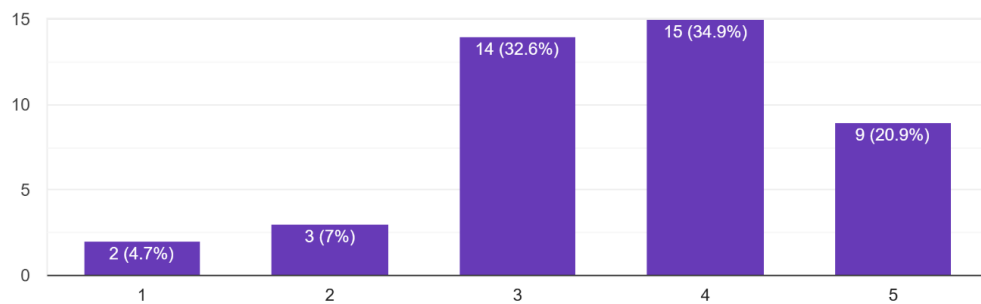


Figure 3.5 The level of knowledge about rainwater harvesting system.

Understanding of Agriculture Practices
43 responses

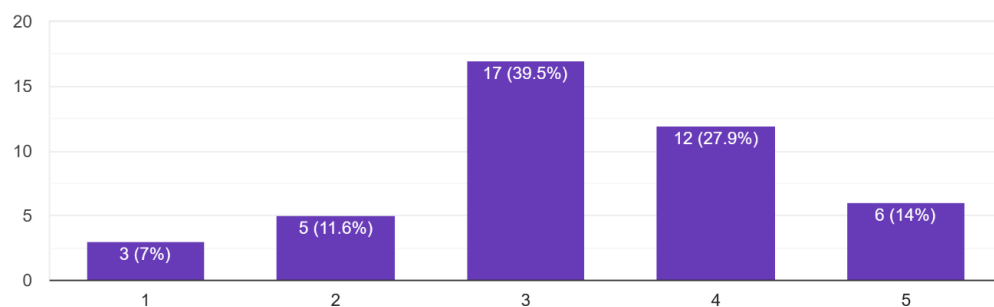


Figure 3.6 The level of understanding about agriculture practices.

Based on the figure above, the questionnaire starts with gathering the background information from the respondents. The question began with the gender of the respondent. According to the pie chart, it shows that 58.1% of 43 responses were male while 41.9% were female. The percentage shows most respondents are male. The second question was about age. The chart interprets that 81.4% of the responses were 18 to 25 years old, 14% were 26-30 years old and 4.7% were 31 to 40 years old. It indicates that most respondents come from Generation Z (Gen Z). Taking the opinion of Gen Z is important as they believe everyone should do their part to prevent climate change from worsening and support greater equity for all. The third question was about the profession of each respondent. The result shows that 86% of them were students, 7% were employed and the other 7% were currently unemployed.

The next question was about the level of knowledge of respondents about the rainwater harvesting system. The question was on a Likert scale where the respondents need to answer it based on the scale. The scale was like (1- Very Familiar, 2- Familiar, 3- Neutral, 4- Not Familiar, 5- Do Not Familiar at All). The graph shows that 55.8% of the respondents have rated as not familiar and do not familiar at all about their knowledge in rainwater harvesting systems. Meanwhile, the other respondents rated it as neutral, familiar, and very familiar with the system. It can be concluded that most of the respondents have minimal knowledge of the system. This may occur due to several factors such as lack of awareness and education. For example, there may not be enough public awareness campaigns or educational programs about the system and its benefits.

The last question was about the level of understanding of any agriculture practices. It refers to the acknowledgment of an individual to the relationship between agriculture and the environment. The question also was on the Likert scale where the respondents need to answer it based on the scale. The scale was like (1- Very Understand, 2- Understand, 3- Neutral, 4- Not Understand, 5- Do Not Understand at All). The results show that 17 respondents (39.5%) rated as neutral, 18 respondents (41.9%) rated as not understand and do not understand at all, while 8 respondents (18.6%) rated as very understand and understand. Based on the results, show that most of the respondents have a low level of understanding of any agriculture practices. This is due to the lack of training in agriculture and limited access to agricultural equipment and workshops. Hence, it will reduce learning opportunities for the community in agriculture practices.

Section B: User Preferences

How interested are you in using a rainwater harvesting system for irrigation purposes?

43 responses

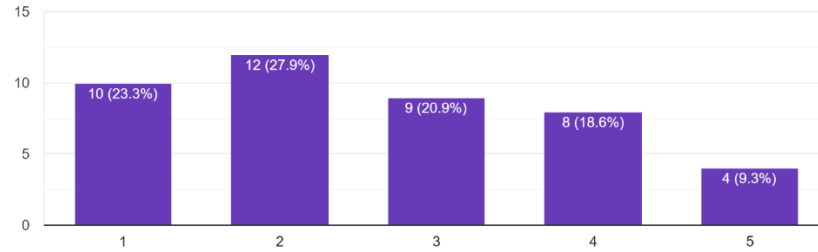


Figure 3.7 Show the interested level in using rainwater harvesting system.

Which characteristics of a smart rainwater harvesting system do you think are needed? (Select all that apply.)

43 responses

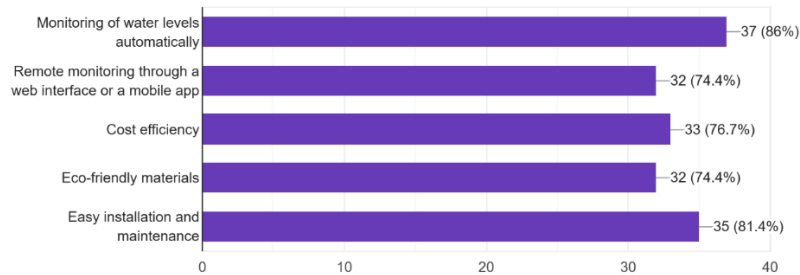


Figure 3.8 Characteristic of the system.

What kind of area would the rainwater harvesting system be used in? (Select all that apply)

43 responses

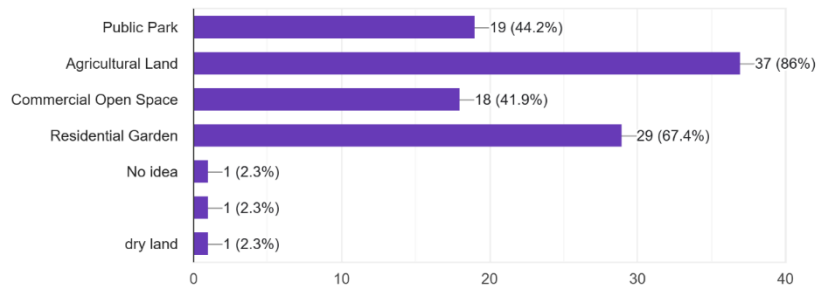


Figure 3.9 Area to use the system.

How much are you willing to spend on a smart rainwater harvesting system for irrigation?
43 responses

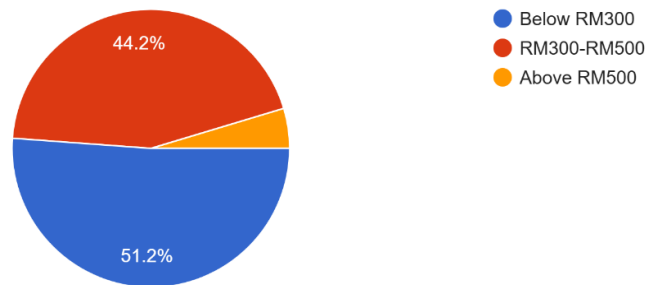


Figure 3.10 Budget for the system.

To what extent would you like the smart rainwater harvesting system to be automated?
43 responses

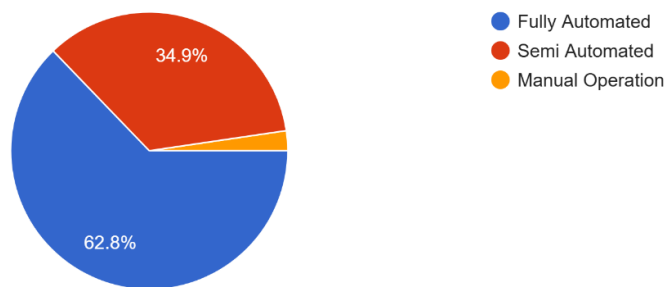


Figure 3.11 Level of automated for the system.

How important is sustainability when finding a system for collecting rainwater?
43 responses

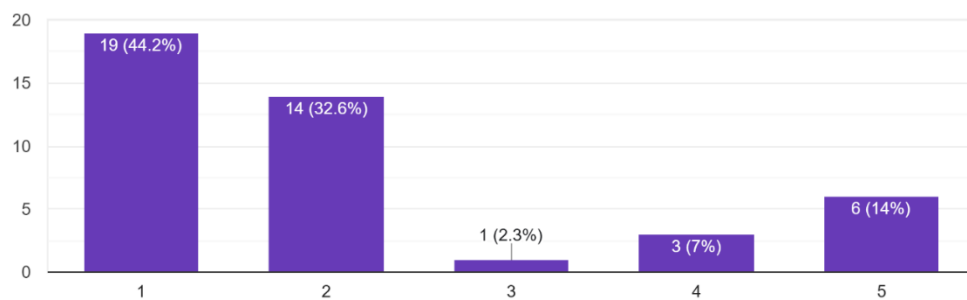


Figure 3.12 Level of importance for sustainability.

How important is user-friendliness for conducting the system?

43 responses

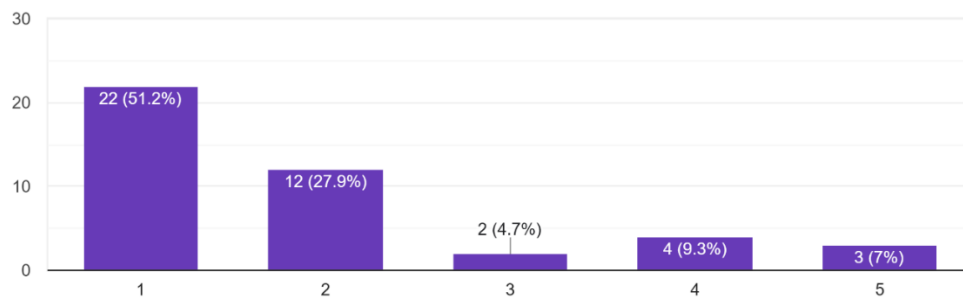


Figure 3.13 Level of importance for user-friendliness.

Would you prefer a smart rainwater harvesting system over a conventional rainwater harvesting system?

43 responses

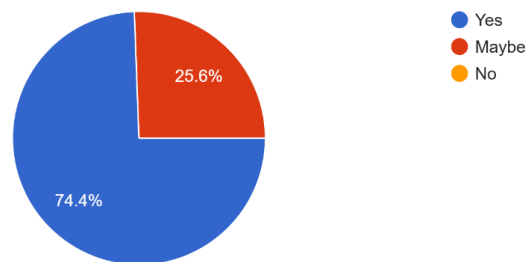


Figure 3.14 Preferences type of system.

What other elements or features are you hoping the proposed system will have?

43 responses

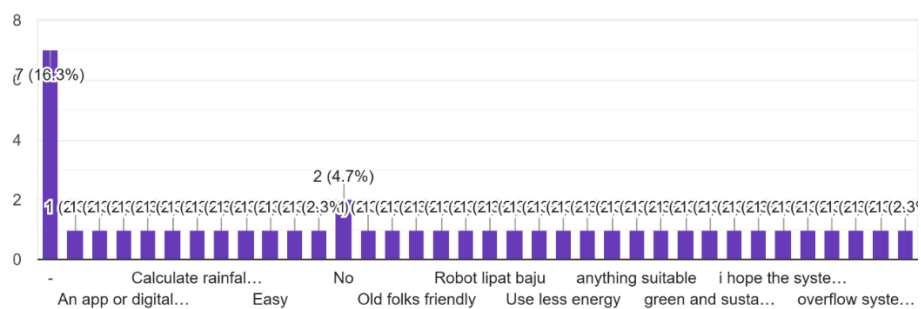


Figure 3.15 Suggested element for the system.

The next section is user preferences which refers to the needs, desires, and expectations of the respondents regarding the system. This survey will help guide the design, functionality, and features of the systems to align with what users value the most. The first question in this section is about the respondents' interest in using the system for irrigation purposes. The respondents must answer it based on the Likert scale provided in the questions. The Likert scale starts with (1- Very Interested, 2- Interested, 3- Neutral, 4- Not Interested, and 5- Not Interested at All). Most respondents were interested in using the system for irrigation purposes as 22 respondents (51.2%) rated very interested and interested. The other respondents rated as neutral where the value was 20.9% and not interested and not interested at all was 27.9%. It indicates high interest among the respondents, and the system has potential adoption if it follows the user needs.

The next question was about the characteristics that respondents think it is necessary to the system. Based on the graph, shows the characteristics that respondents chose were monitoring water levels automatically with 86% of choices, easy installation and maintenance with 81.4% of choices, cost-efficiency with 76.7% of choices, remote monitoring through a web interface or a mobile app with 74.4% choices and eco-friendly materials with 74.4% choices. Hence, it can be concluded that the respondents want the system focused on sustainability, low maintenance, and compatibility while leveraging IoT technology for monitoring. The third question was what kind of areas the system should be used. The majority chose agricultural land as the area that the system should be used with 86% choices. The other areas were residential gardens with 67.4% choice, public parks with 44.2% choice, commercial open space with 41.9% choice, and dry land with 2.3% choice. Based on the responses to this question, it can be assumed that the respondents preferred to develop a flexible system suitable for both large-scale public and smaller private spaces.

The fourth question was about budget willingness. It refers to how much respondents want to spend on the system. More than half of the respondents chose to spend below RM300 for this project while other 44.2% chose to spend RM300 to RM500 and 4.7% chose to spend above RM500. The result shows that the system must focus on cost-efficient designs while maintaining quality and performance. The fifth question was about the level of automation for the system. 27 respondents wanted the system to be fully automated, 15 respondents wanted the system to be semi-automated and 1 respondent wanted it to be manual operation. So, it can be concluded that most of the respondents choose the system operate in fully automated. Next, is about the importance of sustainability in the system. The respondents must answer it based on the Likert scale provided in the questions. The Likert scale starts with (1- Very Important, 2- Important, 3- Neutral, 4- Not Important, 5- Not Important at All). 33 respondents decided that sustainability was important as they rated it very important and important. The other 9 respondents rated it as not important and not important at all while 1 respondent rated it neutral to the importance of sustainability for the system. It highlighted that eco-friendly materials and energy efficiency are important in the system.

Next is the question about user-friendliness for conducting the system. This question has the same Likert scale as the previous question. Most of the respondents thought that user-friendliness in conducting the system was very important as they rated it 1. It is to ensure that the system is simple to install and operate with easy instruction and support for non-technical users. The next question was about which type of system respondents preferred the smart one over the conventional one. Based on the chart shows, 74.4% of respondents preferred the smart rainwater harvesting system while others chose to answer maybe they preferred the smart one. The last question is about the suggestions on features that respondents hope the system will have. Hence, the table below shows some of the collected answers to the question.

Table 3.1 Respondents' suggestion for features.

Respondents	Suggestions
1	Green and sustainable features.
2	The system is in mobile.
3	A feature to redirect excess water during heavy rain to avoid wastage or flooding.
4	Smart Sensors
5	Notifications about rain
6	An app or digital monitoring system so users can monitor it anywhere, anytime
7	Can filter acidic in the rainwater
8	Old folks friendly
9	Can be monitored by phone
10	Overflow system and filtration for the harvested rainwater

Section C: Recommendations for the Proposed System

What would you suggest for the design of the rainwater harvesting system?

43 responses

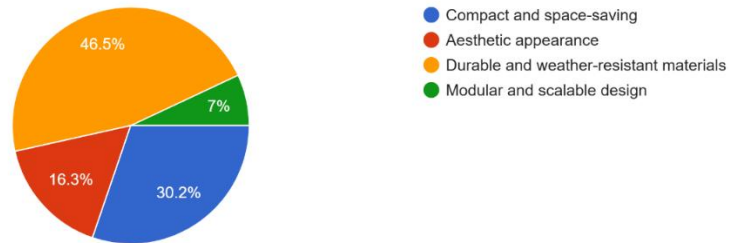


Figure 3.16 Suggested design for the system.

What energy source would you suggest for powering the system?

43 responses

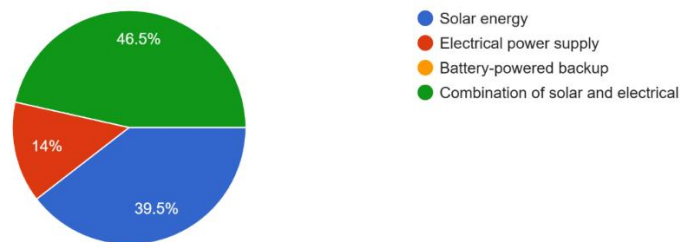


Figure 3.17 Type of energy source for the system.

What type of data should the system provide to users? (Select all that apply.)

43 responses

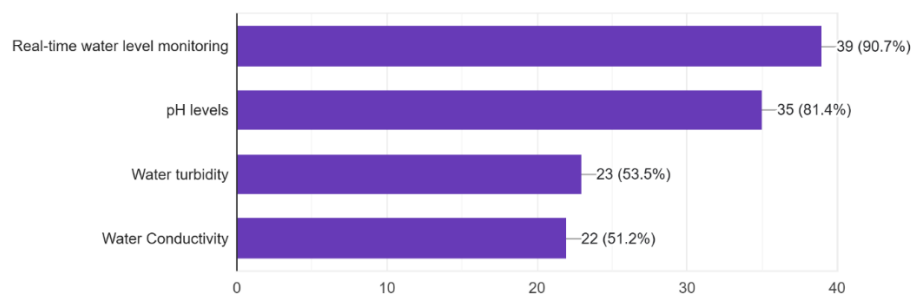


Figure 3.18 Data type that should system provide.

How would you prefer to interact with the system?
43 responses

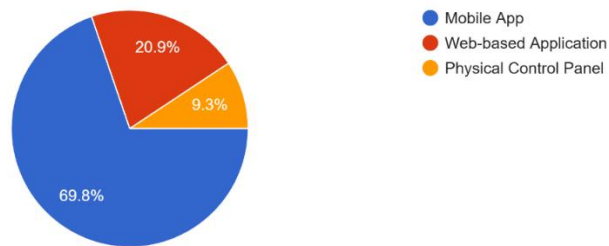


Figure 3.19 Preferred way for user to interact with the system.

What difficulties do you think the suggested system would present, and how would you suggest resolving them?
43 responses

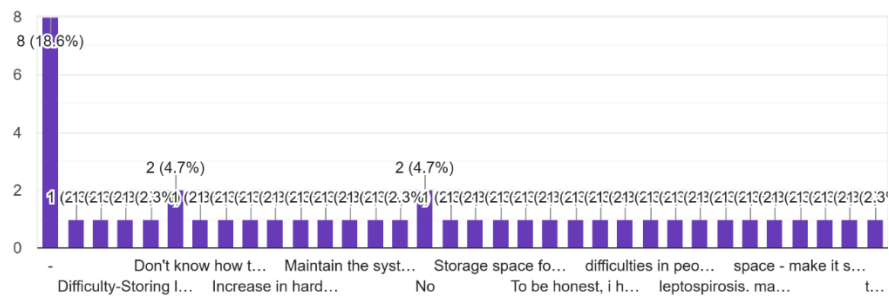


Figure 3.20 Difficulties that would present.

The last section was the recommendation for the proposed system. In this section, the respondents will give their suggestions about the design, the energy source, and the type of data that should be included in the system. The first question is about the suggestion for the design of the system. Based on the percentage shown, most of the respondents chose a design with durable and weather-resistant materials with 46.5%. The second highest choices were compact and space-saving with 30.2%, the third highest choices were aesthetic appearance with 16.3%, and lastly with 7% modular and scalable design. It shows that the respondents wanted a model that is durable and resistant to weather conditions. The next question was about the energy source that respondents would suggest for the system. The highest choice with 46.5% was a combination of solar and electrical, the second highest was solar energy with 39.5%, and electrical power supply with 14%. The result indicates that the respondents pay attention to the save energy features for the system as it mostly used the solar energy. The third question was about the type of data should the system provide. Most of the respondents chose real-time water level monitoring at 90.7%, pH levels at 81.4%, water turbidity at 53.5%, and water conductivity at 51.2%.

All these features will assist the user in monitoring the system and the water condition for irrigation purposes. The fourth question was how respondents wanted to interact with the system. The mobile app was the most chosen way to interact with 69.8% while web-based applications only 20.9% and physical control panel with 9.3%. It shows that the respondents tend to use more mobile apps rather than web-based applications. Lastly, was about the difficulties that respondents think may present during implementation. Hence, the table below shows some of the collection answers to this question.

Table 3.2 Responses on difficulties during implementation.

Respondents	Suggestions
1	Difficulties in people handling the system especially for those unfamiliar with high tech.
2	Rural or remote areas might lack reliable internet for smart features.
3	System may be less effective during prolonged dry periods or excessive rainfall.
4	Connection problem
5	Ensuring the system is compact without compromising functionality might be challenging.
6	Increase in hardware maintenance due to 24 hours operation
7	To educate the user using the apps. As the age range might be unidentical.
8	Foreign objects in the tank.

9	Bug, inaccuracies in details.
10	Storage space for water.

3.3 System Design

System design is a process to define the architecture, components, modules, interfaces, and data for the system. It outlines how the system will work, how to interact with the components and follow the user requirements. This process is important as it ensure feasibility of the system and optimize the system functionality. It will ensure the system is structured and efficient for the user.

3.3.1 System Architecture

The main objective of system architectures is to differentiate layers and components such as hardware, software, data and user interact to establish the system. System architectures refer to the conceptual design and structure of the system. The framework from the architecture will describe how the sensor, microcontroller, software and user interface will work together. It will include input layer, processing layer, data storage layer, output layer, and communication layer. Based on the architecture, the project will be focusing on creating a smart rainwater harvesting system to collect the rainwater.

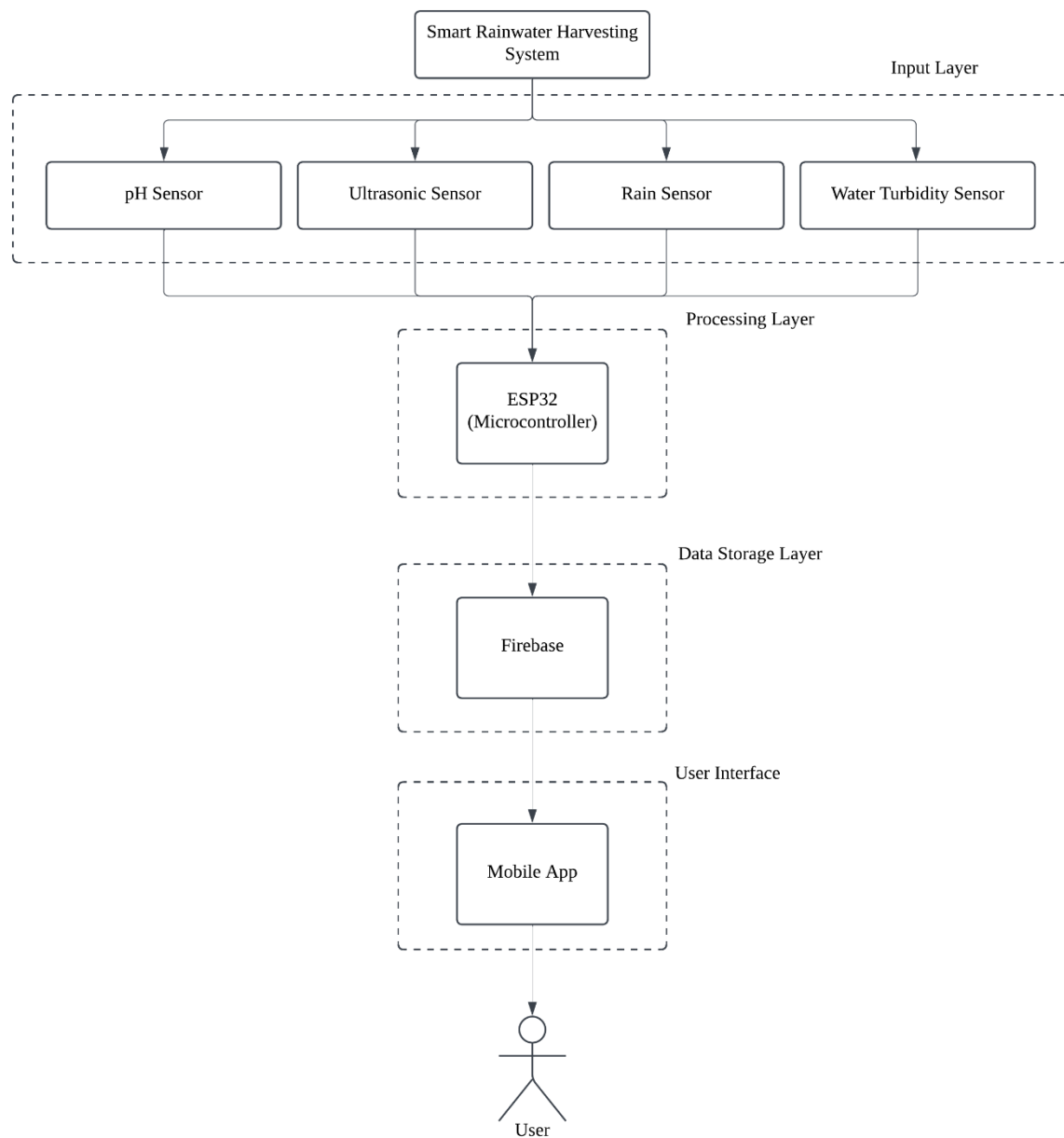


Figure 3.21 System architecture for the system.

3.3.2 Conceptual Model

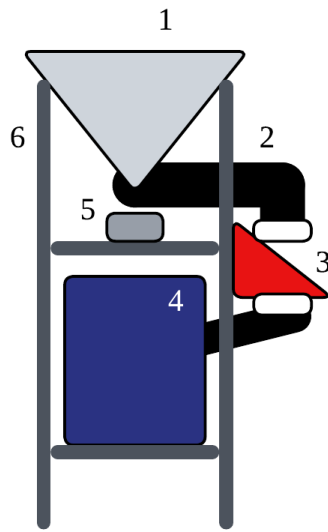


Figure 3.22 Harvesting model for this project.

Indicator:

1. Catchment area: Act as a tool to collect rainwater based on the total surface area of the roof. The roof will be channelling the water to collection system like gutter.
2. Gutter: It is a channel made of metal or plastic installed along the edges of the roof to collect and redirect rainwater into the tank.
3. Gutter Trap: Filter system. It prevents any sediment, dirt or dry leaves from entering the water tank.
4. Water tank: To store the collected water for irrigation purpose.
5. Control box: The main controller for the whole system such as microcontroller.
6. Building structure: The structure made from PVC pipes. It helps the structure to keep steady and support the tank.

3.3.3 Functional Requirements

Functional requirements define the specific features that the system must exhibit to meet and satisfy the user requirements and achieve project objectives. Its fundamentally describe what the system should do. The importance of functional requirements is to provide clear instructions to the development team besides to ensure the user satisfaction. There are a few key aspects that need to be considered in the functional requirements such as the features must achievable, measurable, relevant, specific and verifiable through testing procedures.

Table 3.3 Functional requirements for the system.

ID	Requirements
FR1	The system must collect real-time data using sensors that included.
FR2	The system must allow the user to monitor the system through a mobile app.
FR3	The system must measure the water level in the water tank.
FR4	The system must measure the water turbidity in the water tank.
FR5	The system must measure the pH level in the water tank.
FR6	The system must notify users through a mobile app about the system status.

3.3.4 Non-functional Requirements

Non-functional requirements are crucial aspects as it define how well will the system operates. It focuses on the quality attributes, constraints and operational guidelines for the system. To ensure the system execute well, reliably and maintaining user satisfaction the non-functional requirements will be listed.

Table 3.4 Non-functional requirements for the system.

Attributes	Requirements
Efficiency	The user interface for the system must effective and responsive providing the up-to-date data for the user. For example, the user interface must load the real-time data within 10 seconds under normal network connections.
Usability	The user interface must be user-friendly, accessible, and easy to interact with. For example, an intuitive design should be implemented to ensure ease of use for non-technical users.
Security	The system must include account security features, requiring users to log in with a unique username and a password that meets stringent criteria. This ensures data safety and maintains confidentiality.
Reliability	The system must free from failures or errors to execute its intended operations efficiently. For example, the system should have a mean time between failures (MTBF) of at least 5,000 hours.

3.3.5 Flowchart

This flowchart represents the flow of the system. The system will start with the detection of rainfall. It will start to collect the rainwater using the harvesting infrastructure. After ensuring the sensor pairing successfully, it will start to collect the data from sensor. Then, the data will be sent to the ESP32. The data will be saved to firebase for future references. To monitoring the system, the data will be retrieved from firebase, and a notification will be sent to the user interface if it is triggered. For user to monitoring the system, user must log in first to have access to the data. Then, user ready to view data through mobile app and monitor the system.

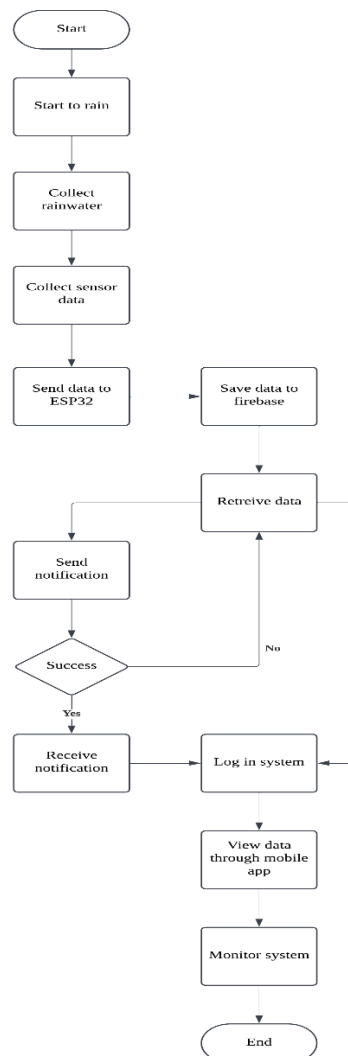


Figure 3.23 Flowchart of the system.

3.3.6 Use Case Diagram

A use case diagram is to visually explain the interactions between system and its user. It includes a high-level overview of how a system works by showing the functionalities and roles of actors involved. The other purpose of use case diagram is to defining system scope by determine what the system will and not do. Beside that it also serves as a tool to refine system requirements with stakeholders. The key elements of use case diagram are actors, use cases, relationships, and system boundary.



Figure 3.24 Use case diagram for the system.

3.3.7 Activity Diagram

An activity diagram represents the workflow of activities in the system. It provides view of how processes are executed, and it flow from one activity to another. The purpose of an activity diagram is to breakdown complex processes into easy-to-follow diagram and assist in system design and analysis. The key elements of activity diagram are start node, activity, decision node, arrow, end node and swimlanes.

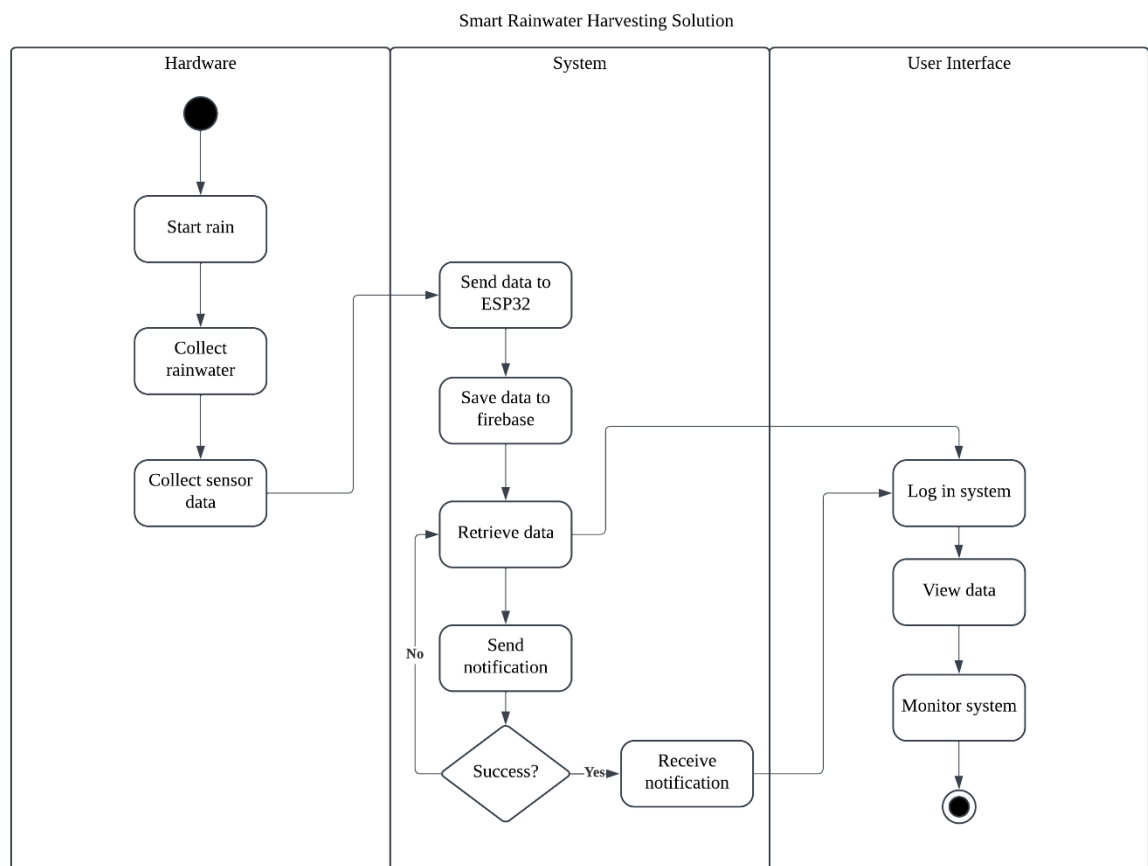


Figure 3.25 Activity diagram for the system.

3.3.8 Sequence Diagram

A sequence diagram illustrates how objects in a system interact with each other in specific sequence of events. It focuses on showing the flow of information or control. The other purpose of a sequence diagram is to explain how specific function work step by step. Besides that, it also helps the developer to design the system and debugging process. The key elements of sequence diagram are actors, objects, lifelines, messages, and activation bars.

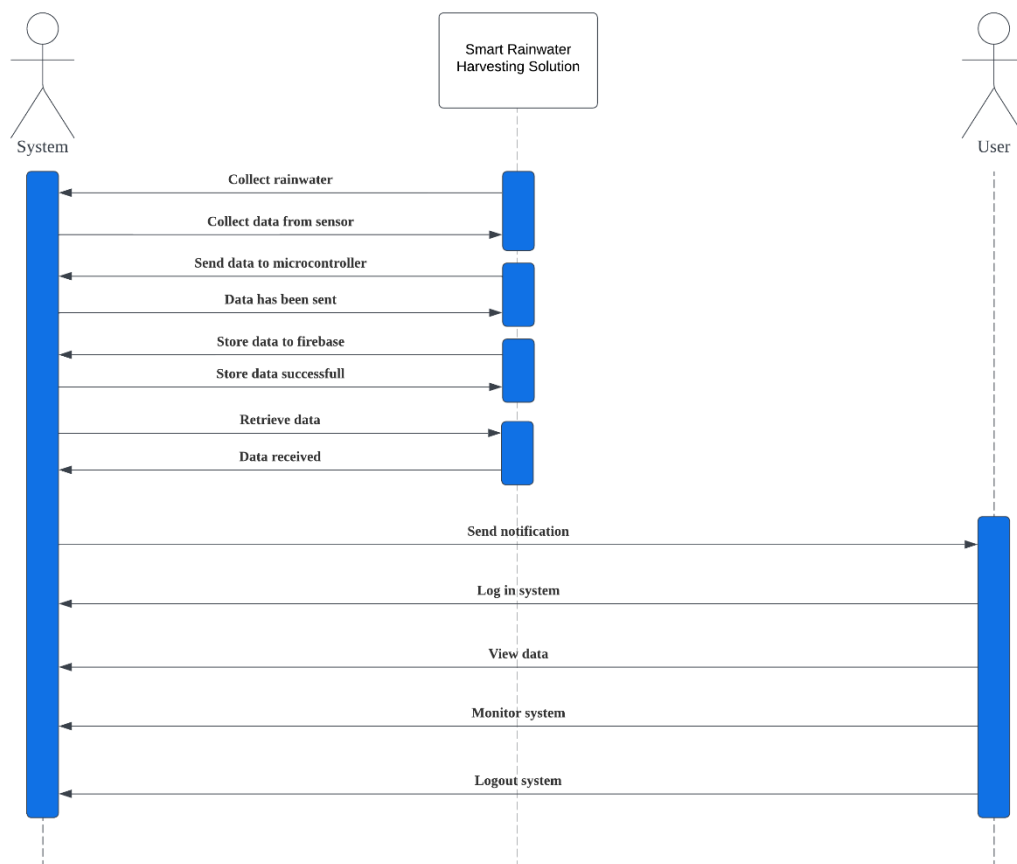


Figure 3.26 Sequence diagram for the system.

3.3.9 Class Diagram

A class diagram visually represents the system by modelling its classes, attributes, methods, and relationships. The purpose of class diagram is to define relationship by clarifying how the components interact each other, guide the implementation process by showing the system's structure. The key elements of a class diagram are class, attributes, methods, relationships and visibility.

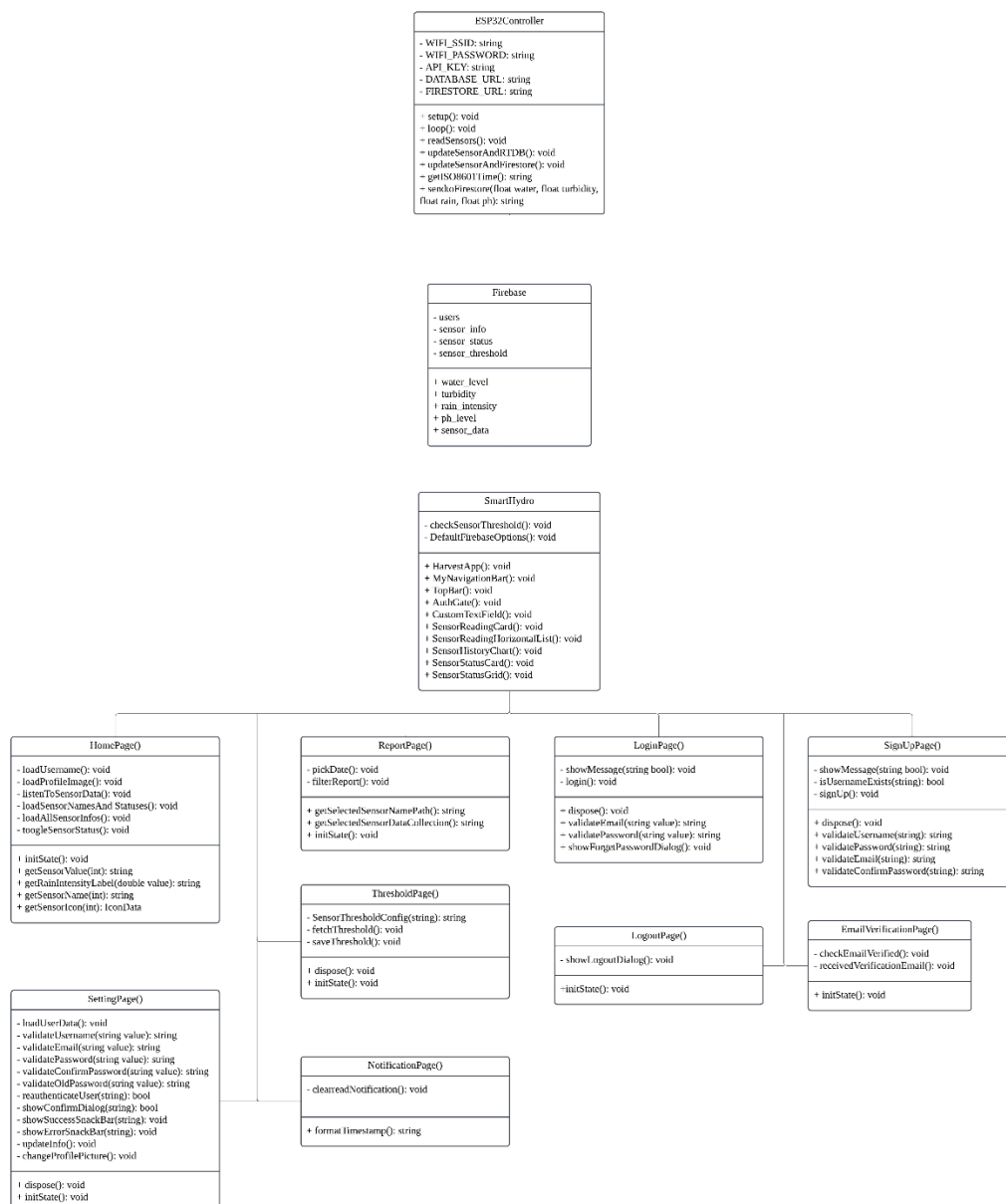


Figure 3.27 Class diagram for the system.

3.3.10 User Interfaces

This is the user interface of login page for the system. The user will input their username and password to have access into the system. It is important in any system as it is requiring user authentication ensuring that only authorized user can gain access to the information and functionality of the system.

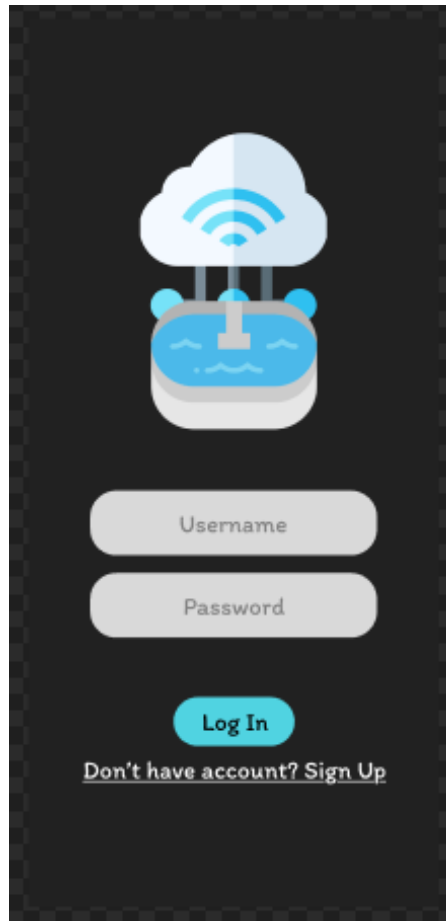


Figure 3.28 Login page for mobile app.

This is the user interface of sign-up page for the system. It requires the user to fill in their username, password, email address and phone number. To gain access to the system, user must sign up first to create an account.

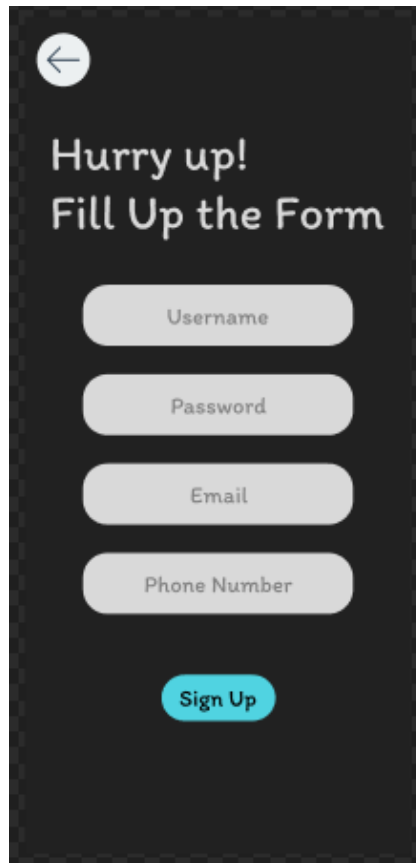
The image shows a mobile app sign-up page with a dark background. At the top left is a white back arrow icon. Below it, the text "Hurry up! Fill Up the Form" is displayed in a white, sans-serif font. There are four light gray rounded rectangular input fields stacked vertically, each containing a placeholder label: "Username", "Password", "Email", and "Phone Number". At the bottom of the form is a bright green rounded rectangular button with the text "Sign Up" in white.

Figure 3.29 Sign up page for mobile app.

This is the user interface of main dashboard page for the system. It will show to the user after they successfully login into the system. In this page, user can monitor the reading from the sensors and control the sensors whether to turn on or off.

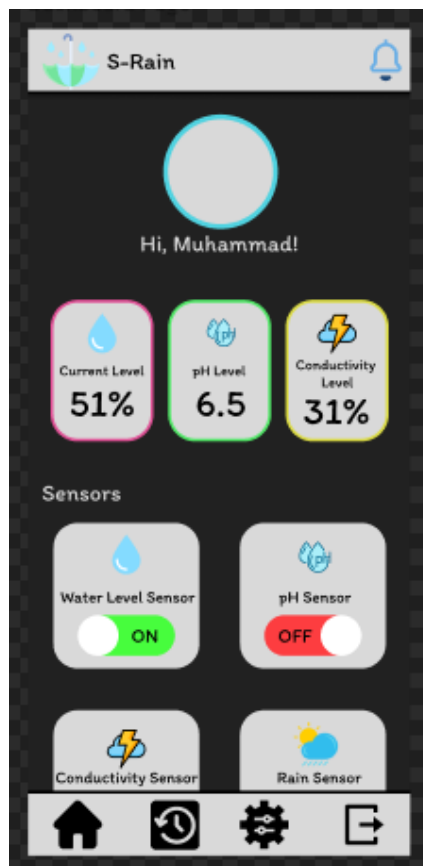


Figure 3.30 Main dashboard page for mobile app.

This is the user interface of notification page for the system. This page will show to the user the list of notification they get. The system will send the user a notification if any changes happen to keep user up to date to the system. The sended notification will store at this page for user to review it.

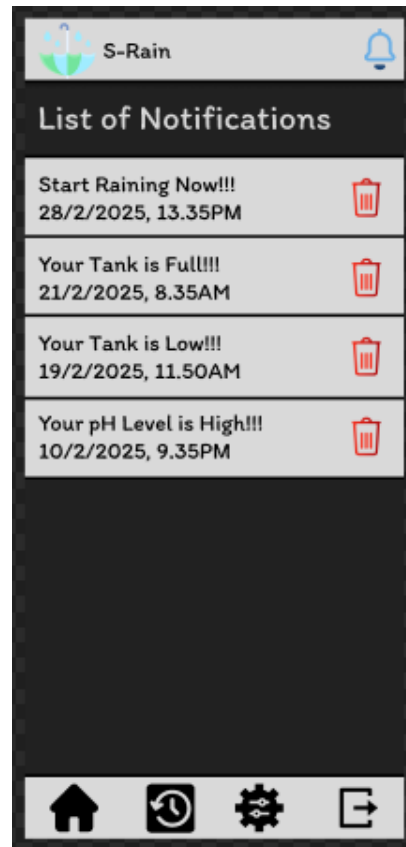


Figure 3.31 Notification page for mobile app.

This is the user interface of update profile page for the system. This page is for the user to update information about their profile. User can just input their new username, new password, new email address and new phone number and click save for the update.

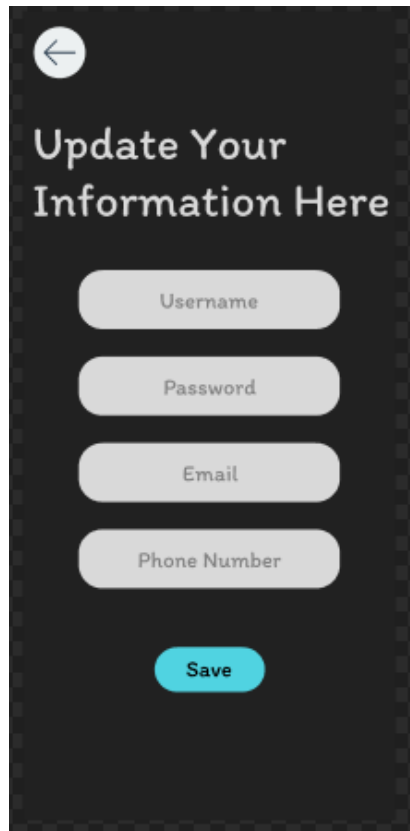
The image shows a mobile application screen for updating a user's profile. At the top left, there is a white circular button with a left-pointing arrow. Below this, the text "Update Your Information Here" is displayed in a white, sans-serif font. Underneath the title, there are four light gray, rounded rectangular input fields stacked vertically. The first field is labeled "Username", the second "Password", the third "Email", and the fourth "Phone Number". At the bottom of the screen, there is a bright blue, rounded rectangular button labeled "Save". The entire interface is set against a dark background with a subtle checkered pattern.

Figure 3.32 Update profile page for mobile app.

This is the user interface of history report page for the system. This page will show the history of data reading from the sensor to the user. User can filter the report by month and year to generate the report. The report can be downloaded by the user to view it.

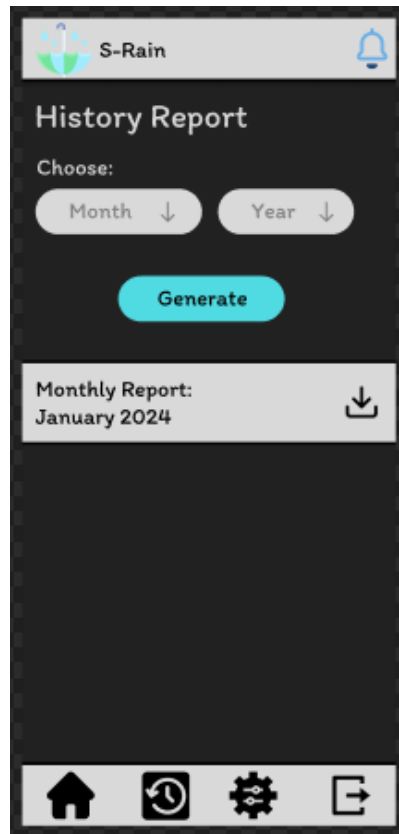


Figure 3.33 History report page for mobile app.

3.4 Hardware and Software Requirements

3.4.1 Hardware Requirements

The hardware and the requirement that needed in this project will be listed to allow the system to operate well. This is important as it ensure the system perform the core operations and maintain the system performance.

Table 3.5 Hardware specifications

Devices	Specifications
ESP32	- Processor: Dual-core, 160-240 MHz - Memory: 520 KB SRAM, 4 MB Flash - Connectivity: Wi-Fi and Bluetooth - Operating voltage: 3.0V – 3.6V
Ultrasonic Sensor	- Size: 45 mm x 20 mm x 15 mm - Measurement range: 2 cm to 400 cm - Echo output signal: Pulse width proportional to distance - Operating voltage: 5V DC
Connectors	- Type: Female-to-male, male-to-male, or female-to-female - Material: Copper with PVC insulation - Voltage rating: Up to 250 V - Length: 20cm to 30cm
pH Sensor	- Type: Glass electrode/ Analog - Measurement range: pH 0 to pH 14 - Size: 12-15 mm in diameter - Power supply: 3.3V - 5V DC
Rain Sensor	- Sensing material: Conductive material

		- Sensing area: 2.5 cm x 2.5 cm - Size: 50 mm x 40 mm x 20 mm - Operating voltage: 5V DC
Water Sensor	Turbidity	- Sensitivity: Sensitive to solids things - Sensing element: Infrared light emitter - Size: 60 mm x 30 mm x 20 mm - Measurement range: 0 to 1000 NTU - Operating voltage: 5V DC
Breadboard		- Type: Half-size, and mini breadboards - Size: 400 points to 170 points - Material: Plastic - Voltage rating: 12V to 24V DC

3.4.2 Software Requirements

Software requirements refer to the software or tools that needed to implement the system. It ensures the application run efficiently and effectively.

Table 3.6 Software specifications.

Software	Specifications
Arduino IDE	- Supported platforms: Windows, macOS, Linux - Programming language: C and C++ - Supported boards: Over 100 boards such as ESP32 - Code editing: Syntax highlighting, code completion, auto-indentation, and error checking. - License: Open source
Android Studio	- Supported platforms: Windows, macOS, Linux - Programming language: Java or Dart - Platform type: IDE for android app development - Emulator: Built-in Android Emulator
Figma	- Platform type: Cloud-based design and prototyping tool - Supported OS: Web-based: Chrome, Firefox, Safari, Edge Desktop App: Windows 7+, macOS 10.12+, Linux - Purpose: UI/UX design, wireframing, prototyping
Firebase	- Android API Level: 21 or higher - Gradle version: 7.3.0 or higher - SDK version: 28 or higher - Required Google account.

3.5 Summary

In conclusion, the requirement analysis and system design for the Smart Rainwater Harvesting Solution are thoroughly covered in this chapter. According to the information acquired from the questionnaire, there is a great demand for user-friendly design, a strong interest in the system, and an emphasis on sustainability. The system architecture, hardware and software specifications, and functional and non-functional requirements provide a solid foundation for the successful creation and application of this novel solution. The project intends to promote sustainable irrigation practices, especially in open spaces, by attending to user needs and preferences.

CHAPTER 4:IMPLEMENTATION

4.1 Introduction

This chapter will explain the implementation of the Smart Rainwater Harvesting Solution for Irrigation in Open Spaces. It will highlight two important section which are hardware and software implementations. The purpose of this chapter is to demonstrate how the proposed system was realized step-by-step.

4.2 System Overview

Before starting, importance to understand the flow of the system. The system will consist of various sensors such as pH sensor, ultrasonic sensor, raindrop sensor and water turbidity sensor while ESP 32 as the microcontroller. The sensors' reading will transfer to the Realtime Database in Firebase to store the data. From the Firebase, the data will retrieve and review by the user from the application. In order to achieve the objectives of this project, the system must run accordingly to the diagram below.

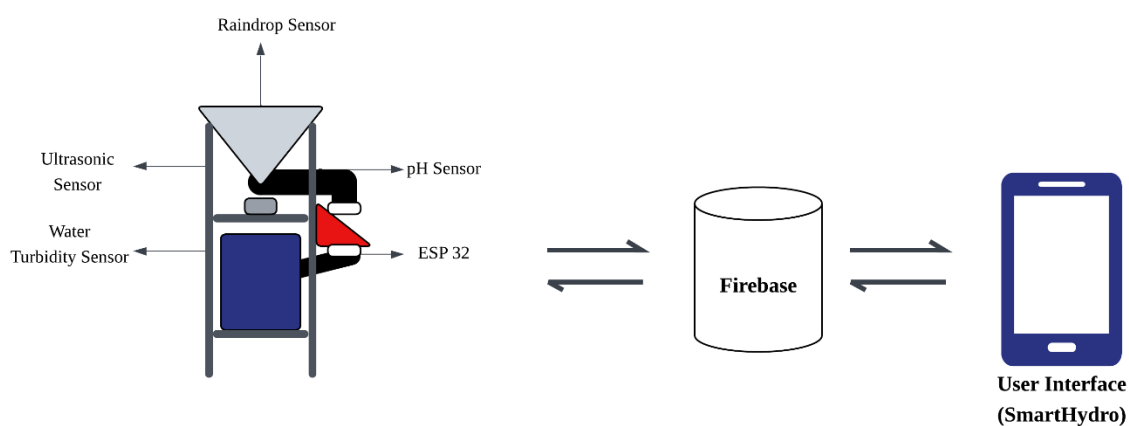


Figure 4.1 Show the system overview of this project.

4.3 Hardware Implementation

This section highlights the steps how the hardware and components will be connected and set up. Its importance to give the understanding and to view the design of the model. It provides the actual models and circuit design used for simulated rainwater collection.

4.3.1 Circuit Diagram

The circuit diagram shows how all sensors are connected to microcontroller, ESP32. The design circuit is to ensure that all sensors are connected and working properly according to system flow. Each component will be tested first by a simple coding to ensure the component is working. All components are powered through the ESP32 and carefully wired to avoid short circuits.

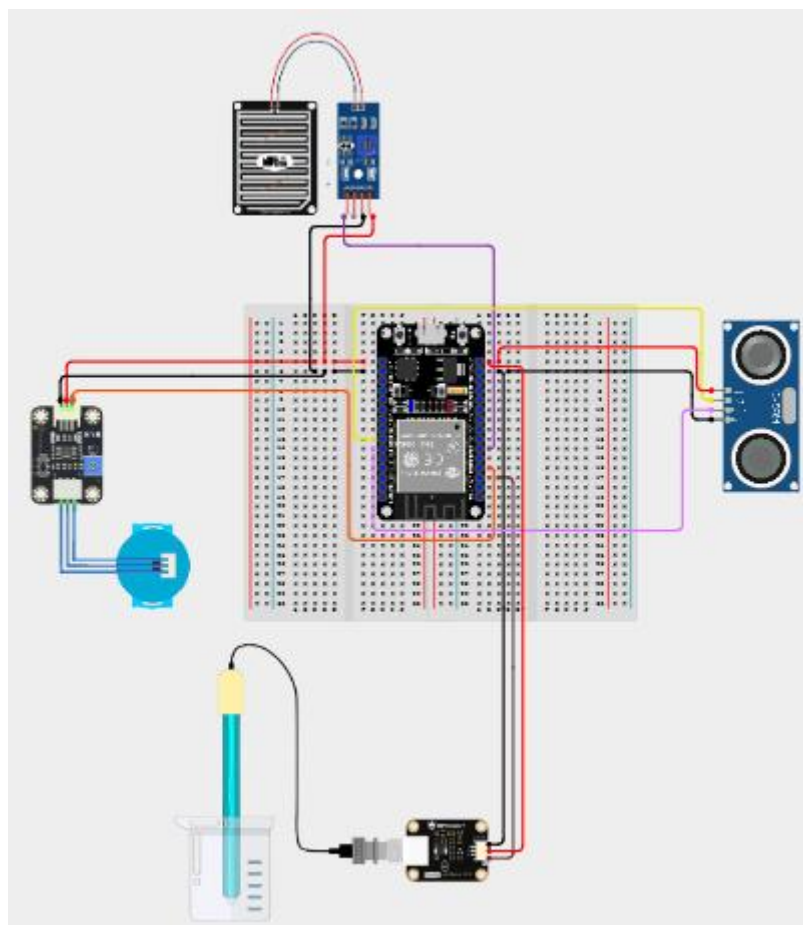


Figure 4.2 Circuit diagram for this project.

Table 4.1 Pin connection for sensors and ESP32.

Sensors/Modules	PIN Connection
Ultrasonic Sensor	VCC → VIN TRIG → GPIO 18 ECHO → GPIO 19 GND → GND
Turbidity Sensor	VCC → 3.3V GND → GND OUT → GPIO 34
Rain Sensor	VCC → 3.3V GND → GND AO → GPIO 32
pH Sensor	V+ → VIN G → GND Po → VN (GPIO39)

4.3.2 Harvesting Model

A small-scale version of real rain harvesting system develop to demonstrate how the system would work in a real outdoor environment. A small container is used to store the rainwater, and the pipe is used to direct water into the tank. The raindrop sensor placed at the top of the roof, to detect the rain and rain intensity. The pH sensor and water turbidity sensor placed in the container to measure the pH level and water turbidity. While the ultrasonic sensor placed under container lid to measure the water level in the tank. The model will be tested first to ensure the rainwater can be collected properly.

Table 4.2 Material used for the harvesting model.

Materials Used	Description
- PVC pipe (20mm) - Elbow pipe - T-joint pipe	Act as the main structure of the harvesting model.
- UPVC pipe (50mm) - Elbow pipe	Act as the gutter where it catches the rain from the roof. (Upper section)
- UPVC pipe (32mm) - Elbow pipe	Act as the pipeline for water go to the tank. (Lower section)
- Polycarbonate Roofing Sheet (20-inch x 15-inch)	Act as the catchment area for the model.
- Plastic Food Container - Plastic net	Act as the gutter leaf trap. It will prevent any sediment or leaf from enter the tank.
- Plastic Pail	Act as the water tank to store the rainwater.
- Plastic Food Container	Act as the main box controller for the electronic part.



Figure 4.3 Side view of harvesting model.



Figure 4.4 Front view of harvesting model.

4.4 Software Implementation

There are two parts for the software implementation including the code that running on the ESP32 and the mobile application named SmartHydro. This section will describes the development of both parts including the functionality and integration with Firebase.

4.4.1 ESP32 Coding's

The microcontroller is responsible for collecting the real-time data from the sensors and upload it to the Firebase using the REST API. This to allow the user to monitor the system by the real-time. Hence, the implementation that related to ESP32 included in this section.

a) Wi-Fi and Firebase Initialization

ESP32 will connect to the Wi-Fi first based on the ID and password that have been set in the code. Then, it will authenticate with Firebase using an API key and database URL. The API key and database URL can get from the firebase console on project settings.

```
// --- Wi-Fi Credentials ---
#define WIFI_SSID "toratoradatanglagi"
#define WIFI_PASSWORD "allahmahabesar"

// --- Firebase RTDB Credentials ---
#define API_KEY "AIzaSyDg9-gcLC9mxCQx6lBNZF4iRsULwMSaywQ"
#define DATABASE_URL "https://hydrosmart-5f1f9-default-rtdb.asia-southeast1.firebaseio.com/"

// --- Firestore REST ---
String FIRESTORE_URL = "https://firestore.googleapis.com/v1/projects/hydrosmart-5f1f9/databases/(default)/documents/sensor_logs?";
                        key=AIzaSyDg9-gcLC9mxCQx6lBNZF4iRsULwMSaywQ";
```

Figure 4.5 Wi-Fi and Firebase Initialization in ESP32.

b) Pin Assignments and Upload Time

Each sensor will assign a specific pin based on the sensor's pin connection to the ESP32. The constant value will be declare first before used in the function. The upload time to the Firestore will be declared in milliseconds. For Realtime Database it will upload for every 5 seconds and for the Firestore it will upload for every 15 minutes.

```

// --- Pin Assignments ---
const int trigPin = 18;
const int echoPin = 19;
const int rainSensorPin = 32;
const int turbidityPin = 34;
const int phPin = 33;

// --- Constants ---
const float tankHeight = 24.0;
const float voltageRef = 3.3;
const int adcResolution = 4095;

// --- Timers ---
unsigned long lastRTDBUpdate = 0;
unsigned long lastFirestoreUpload = 0;

const unsigned long RTDB_INTERVAL = 5000; // RTDB every 5 seconds
const unsigned long FIRESTORE_INTERVAL = 900000; // Firestore every 15 minutes

```

Figure 4.6 Pin assignment for each sensor and time upload to the firebase.

c) Sensors Readings

This code section is to get the readings from the sensors. Each sensor required specific calculations to ensure the readings are accurate.

```

void readSensors(float &water, float &turbidity, float &rain, float &ph) {
// --- Ultrasonic ---
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
long duration = pulseIn(echoPin, HIGH);
float distance = duration * 0.034 / 2;
distance = constrain(distance, 0, tankHeight);
water = ((tankHeight - distance) / tankHeight) * 100.0;

// --- Turbidity ---
int turbidityRaw = analogRead(turbidityPin);
float turbidityVoltage = turbidityRaw * (voltageRef / adcResolution);
turbidity = (3.0 - turbidityVoltage) / (3.0 - 1.0) * 100.0;
turbidity = constrain(turbidity, 0, 100);

// --- Rain ---
int rainRaw = analogRead(rainSensorPin);
float rainVoltage = rainRaw * (voltageRef / adcResolution);
rain = (3.3 - rainVoltage) / (3.3 - 0.5) * 100.0;
rain = constrain(rain, 0, 100);

// --- pH Sensor ---
int phRaw = analogRead(phPin);
float phVoltage = phRaw * (voltageRef / adcResolution);
ph = (phVoltage - 0.5) * (14.0 / 2.5);
ph = constrain(ph, 0, 14);
}

```

Figure 4.7 Sensors reading code section.

- d) Update sensor to RTDB and sendToFirestore().

The code defines two functions that responsible in updating sensor values to the RTDB and Firestore. It will call the readSensor() functions to get the sensor values before updating it. The second functions will update the current value hold in as variables to send it to the sendToFirestore().

```
void updateSensorsAndRTDB() {
    float water, turbidity, rain, ph;
    readSensors(water, turbidity, rain, ph);

    Serial.println("Updating Realtime Database...");
    Firebase.setFloat(fbdo, "/sensor_data/water_level", water);
    Firebase.setFloat(fbdo, "/sensor_data/turbidity", turbidity);
    Firebase.setFloat(fbdo, "/sensor_data/rain_intensity", rain);
    Firebase.setFloat(fbdo, "/sensor_data/ph_level", ph);
}

void updateSensorsAndFirestore() {
    float water, turbidity, rain, ph;
    readSensors(water, turbidity, rain, ph);
    sendToFirestore(water, turbidity, rain, ph);
}
```

Figure 4.8 Update sensor values to RTDB.

- e) Upload sensor values to Firestore.

The variables used in previous functions will use in this functions to store the values to the Firestore. This will loop for every 15 minutes as declared above.

```
void sendToFirestore(float water, float turbidity, float rain, float ph) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String timestamp = getISO8601Time();
        String payload;

        // --- Water Level ---
        http.begin("https://firestore.googleapis.com/v1/projects/hydrosmart-5f1f9/databases/(default)/documents/water_level?key=" + String(API_KEY));
        http.addHeader("Content-Type", "application/json");
        payload = "{\"fields\": {\"timestamp\": {\"timestampValue\": \"\" + timestamp + \"\", \"value\": {\"doubleValue\": \" + String(water) + \"}}}}";
        int code = http.POST(payload);
        Serial.println("Water level + " + String(code));
        Serial.println(http.getString());
        http.end();

        // --- Turbidity ---
        http.begin("https://firestore.googleapis.com/v1/projects/hydrosmart-5f1f9/databases/(default)/documents/turbidity?key=" + String(API_KEY));
        http.addHeader("Content-Type", "application/json");
        payload = "{\"fields\": {\"timestamp\": {\"timestampValue\": \"\" + timestamp + \"\", \"value\": {\"doubleValue\": \" + String(turbidity) + \"}}}}";
        code = http.POST(payload);
        Serial.println("Turbidity + " + String(code));
        Serial.println(http.getString());
        http.end();
    }
}
```

```
// --- Rain Intensity ---
http.begin("https://firestore.googleapis.com/v1/projects/hydrosmart-5f1f9/databases/(default)/documents/rain_intensity?key=" + String(API_KEY));
http.addHeader("Content-Type", "application/json");
payload = "{\"fields\": {\"timestamp\": {\"timestampValue\": \"\" + timestamp + \"\"}, \"value\": {\"doubleValue\": \" + String(rain) + \"}}}\"";
code = http.POST(payload);
Serial.println("Rain Intensity → " + String(code));
Serial.println(http.getString());
http.end();

// --- pH Level ---
http.begin("https://firestore.googleapis.com/v1/projects/hydrosmart-5f1f9/databases/(default)/documents/ph_level?key=" + String(API_KEY));
http.addHeader("Content-Type", "application/json");
payload = "{\"fields\": {\"timestamp\": {\"timestampValue\": \"\" + timestamp + \"\"}, \"value\": {\"doubleValue\": \" + String(ph) + \"}}}\"";
code = http.POST(payload);
Serial.println("pH Level → " + String(code));
Serial.println(http.getString());
http.end();
```

Figure 4.9 Store sensor values to Firestore.

4.4.2 “SmartHydro” Applications

The “SmartHydro” applications is a Flutter-based mobile app that allows the user to view real-time data, manage sensors, receive notifications and adjust threshold value. Hence, this section will include the key pages implemented in the application.

a) User Authentication

Login Page - The user must enter their username and password that has been registered.

The user must enter valid format email address and 8 length passwords.

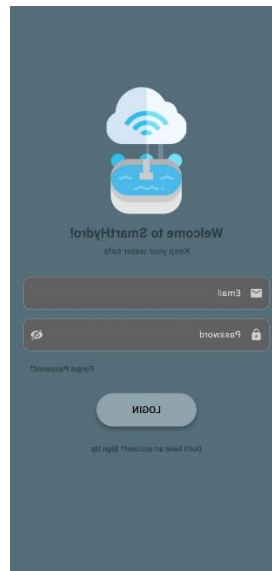


Figure 4.10 Login page of SmartHydro.

Signup Page - If the user is not registered yet, the user can register an account by clicking the sign-up link. User must enter their details in the fields to complete the registrations.

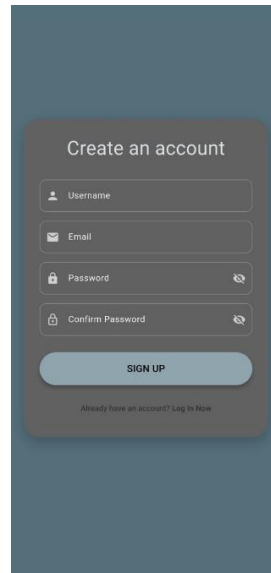
A mobile app interface for creating an account. The background is a solid dark blue. In the center, there is a white rounded rectangle containing the title "Create an account" in a sans-serif font. Below the title are four input fields, each with a small icon on the left: a person icon for "Username", an envelope icon for "Email", a padlock icon for "Password", and another padlock icon for "Confirm Password". To the right of the "Password" and "Confirm Password" fields are small "show/hide" icons. Below the input fields is a large, rounded, light blue button with the text "SIGN UP" in white. At the bottom of the white rectangle, there is a small link that says "Already have an account? Log In Now".

Figure 4.11 Signup page of SmartHydro.

Email Verification Page – If the user already done with the registration, they must verify their email through a link that has been sent to the email address. After verifying, user can login using the account registered.

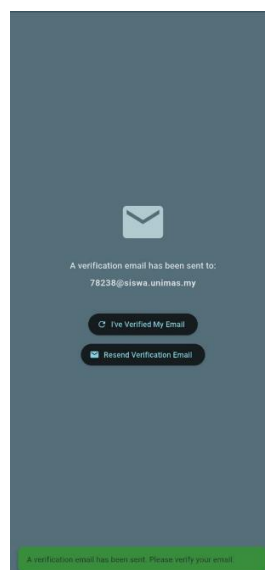
A mobile app interface for email verification. The background is a solid dark blue. In the center, there is a white envelope icon. Below the icon, the text "A verification email has been sent to:" is displayed, followed by the email address "7823B@siswa.unimas.my". Below this text are two buttons: a light blue button with a circular arrow icon and the text "I've Verified My Email", and a dark blue button with an envelope icon and the text "Resend Verification Email". At the very bottom of the screen, there is a green banner with white text that says "A verification email has been sent. Please verify your email."

Figure 4.12 Email verification of SmartHydro.

Forgot Password Page – If the user forgot their password, the user could reset the password by sending a reset link through the email. User must enter their registered email to get the reset link password.

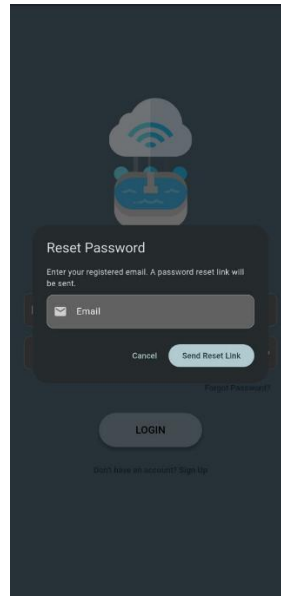


Figure 4.13 Forgot password page of SmartHydro.

- b) Homepage – This page consists of user's profile picture and username. The real- time sensors readings displayed and controlling the sleep mode for every sensor. User can long tapped on the information icon for more information about the readings.

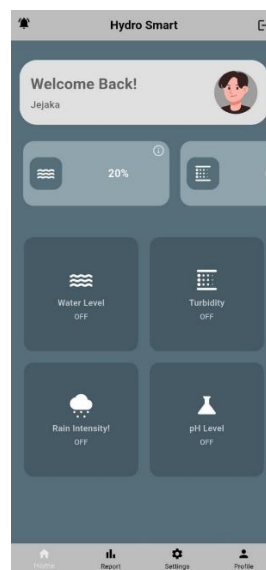


Figure 4.14 Homepage of SmartHydro.

- c) Report Page – This page displayed the user of history data for each sensor in graph form. The user needs to enter the selected sensor and pick date to filter the data that want to be shown.

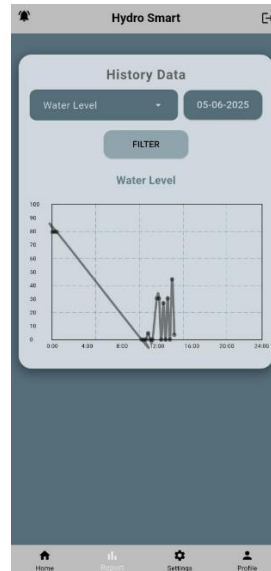


Figure 4.15 Report page of SmartHydro.

- d) Threshold Page – This page allows the user to set and save the threshold value for each sensor. The threshold value will be used to trigger the notification conditions.

The screenshot shows the 'Hydro Smart' app interface. At the top, there's a header with a bell icon, the text 'Hydro Smart', and a share icon. Below this is a 'Threshold Level' section. It contains four rows of input fields, each with a sensor name, a minimum value field, and a maximum value field. The sensors and their values are: Water Level (15.0, 50.0), Turbidity (30.0, 60.0), Rain Intensity (10.0, 60.0), and pH Level (7.0, 14.0). Each row has a small green 'Min' label and a red 'Max' label. A 'SAVE' button is at the bottom of the form. The bottom of the screen has a navigation bar with icons for Home, Report, Settings, and Profile.

Figure 4.16 Threshold page of SmartHydro.

- e) Setting Page – This page allows the user to edit their own information such as username, email, password and profile picture. The updated profile picture will be shown on the homepage.

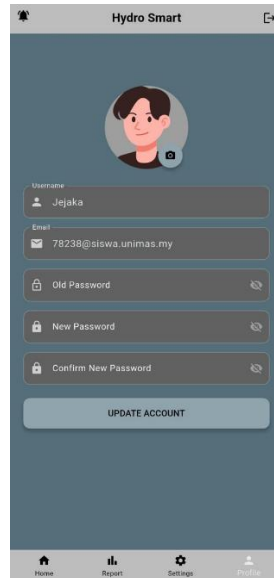


Figure 4.17 Setting page of SmartHydro.

- f) Notification Page – This page will displayed the user list of notifications received. User can mark as read the notification and remove all after reading the notification.

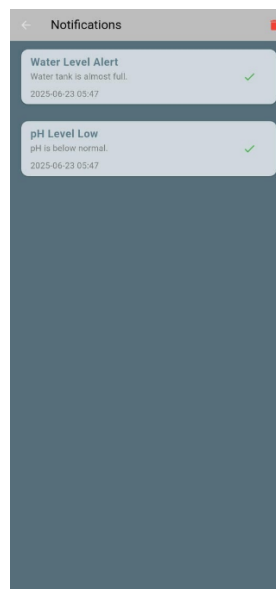


Figure 4.18 Notification page of SmartHydro.

4.5 Summary

This chapter has included both software and hardware implementation of the Smart Rainwater Harvesting Solutions in Open Spaces. It gives understanding of the overall process from creating the model, assemble the circuit, coding the ESP32 and developing the mobile application. Together it provides an integrated solution for monitoring and managing harvested rainwater using modern IoT and mobile technologies.

CHAPTER 5: HARDWARE AND SOFTWARE TESTING

5.1 Introduction

This chapter explains the testing procedures conducted to validating the functionality, reliability, and performance of both components' hardware and software. This procedure is important as it ensures that the system operate well as intended. One of the objectives of testing are to verify all the sensor give the accurate readings, to ensure proper connection between hardware and software and to validate the functionality of both hardware and software. Hence table below show the module tested in this project.

Table 5.1 Modules to be tested in this project.

Type	Modules	Description
Hardware	Harvesting Model	- Model for harvesting the rainwater.
	Electronic Model	- Model for measuring the data and send to the user.
Software (Mobile App)	User Authentication	- Credentials for user to use the app.
	Homepage	- Main dashboard of the app. Display real time data and sensor status.
	Report Page	- View history data for user.
	Setting Page	- Allow the user to change user information.
	Threshold Page	- Allow the set and save their own threshold value.
	Notification Page	- Notify user if changes happen.

5.2 Hardware Testing

Hardware testing is a crucial part of this project as it ensures each sensor and hardware component especially harvesting model functions properly and reliably under difference conditions. This stage verifies the performance, calibration, stability and responsiveness of the components connected to ESP32 microcontroller. This section will divide into two type of testing which are testing for harvesting model and testing for electronic part.

5.2.1 Testing for Harvesting Model

This section tests the functionality of rainwater harvesting model. It is to ensure that all component works well and can efficiently harvesting the rainwater.

Table 5.2 Testing for authentication module.

Module Name: Harvesting Module Test Case Description: To test all the functional component in harvesting model performs well. Testing Objective: To ensure the model can collect the rainwater efficiently. Testing Approach: Scenario based testing. Testing Technique: Requirement-based Testing. Remarks:								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Feedback
HM-01	Harvesting the rainwater.	The model placed at the outside house and leave it for a few days.	None	The rainwater collects successfully.	The rainwater harvest successfully.	Pass	Critical	None

HM-02	Trap dirt or leaves from the gutter.	The leaves and dirt will be put at the catchment area.	Leaves and dirt.	The gutter trap the leaves and dirt prevent it from enter the tank.	No leaves enter the water tank.	Pass	High	None
-------	--------------------------------------	--	------------------	---	---------------------------------	------	------	------

5.2.2 Electronic Testing

This section tests all the electronic part in involve in this project. It is to ensure each part can function properly and measure the data from the harvesting model. The part consists of sensors, microcontrollers, and relay.

Table 5.3Testing for electronic module.

Module Name: Electronic Module Test Case Description: To test all the electronic component performs well. Testing Objective: To ensure the component can measure the data and function according to its coding. Testing Approach: Scenario based testing. Testing Technique: Requirement-based Testing. Remarks:
--

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Feedback
EM-01	Measuring water level.	The ultrasonic sensor placed under the tank lid to measure the water level.	Water level	The ultrasonic sensor gives the current water level in the tank.	The ultrasonic sensor successfully measures the current water level.	Pass	Critical	The sensor works properly.
EM-02	Measure water turbidity.	The water turbidity sensor placed in the tank to measure turbidity level.	Water turbidity	The water turbidity sensor will measures the turbidity level in the tank.	The water turbidity still measures the water turbidity level but inconsistently.	Fail	Critical	Inconsistent reading.
EM-03	Measure rain intensity.	Rain sensor placed on the roof to measure rain intensity.	Rain intensity	The rain sensor will measures the level of rain intensity.	The rain sensor successfully measures the rain intensity.	Pass	Critical	The sensor works properly.

EM-04	Measure pH level.	pH sensor placed in the tank to measure the pH level of the water.	pH level	The pH sensor will give the current pH level water in the tank.	The pH level measured but inconsistently.	Fail	Critical	Inconsistent readings.
EM-05	Send data to Firebase	ESP32 coded to send the data from the sensors to the Firestore and Realtime Database in Firebase.	Sensor readings	The sent data show in the Firestore and Realtime Database.	The data successfully upload to the Firestore and Realtime Database.	Pass	Critical	Real-time data upload successfully.

5.3 Software Testing

This section is to ensure the functional and non-functional part in the app work as intended. It covered the functionality, reliability and usability of the app and its integration with the cloud database. The tested module is user authentication, homepage, report page, settings page, logout page and notification page. Hence the test case for each module provided in this section to identify the result.

5.3.1 Functional Testing

The software starts with the functional testing to test the function that implement in the app. This is to detect and fix functional bugs or logical errors happen during the test.

Table 5.4 Testing for authentication module.

Module Name: User Authentication Module Test Case Description: Test the login and sign-up functionality, include the validation and error handling for different input scenarios. Testing Objective: To verify all the login and sign-up functionality. Testing Approach: Manual black-box testing Testing Technique: Equivalence Partitioning Remarks: Firebase Authentication included.								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
AMF-01	Login with valid credentials.	User input correct email and password.	Valid credentials	Redirect to user homepage.	Show successful message and redirect to the homepage.	Pass	Critical	User successful login.
AMF-02	Login with invalid credentials.	User input wrong email and password	Invalid credentials	Show error message.	Show error message "Login failed. Please check your credentials".	Pass	Critical	Error message successfully shown.

AMF-03	Login with invalid email format.	User input invalid email format.	Invalid email format.	Show error message.	Show error message "Please enter a valid email address".	Pass	High	Error message successfully shown.
AMF-04	Login with empty fields.	User submit without email and password	None	Show error message	Show error message "Email is required" and "Password is required".	Pass	High	Error message successfully shown.
AMF-05	Password visibility for login.	User tap icon "eyes" to enable the password visibility.	None	User's password visible while typing.	User can see their password while typing.	Pass	Low	Password visibility works properly.
AMF-06	Sign-up using valid input.	User enter proper username, valid email, phone number and password in the fields.	None	Show success message and redirect to email verification page.	User successfully sign up and redirect to the email verification page.	Pass	Critical	User successfully signup.

AMF-07	Sign-up with empty fields.	User submits with empty fields.	None	Show error message.	Show error message for every text fields.	Pass	Critical	Error message successfully shown.
AMF-08	Sign up password with only small letter.	User register password using small letter only.	Password	Show error message.	Show error message "Password must include upper/lowercase, number, and special character".	Pass	High	Error message successfully shown.
AMF-09	Sign-up password with numbers only.	User register password using numbers only.	Password	Show error message.	Show error message "Password must include upper/lowercase, number, and special character"	Pass	High	Error message successfully shown.
AMF-10	Sign-up password with only 4 characters.	User register password with only 4 characters.	Password	Show error message.	Show error message "Password must be at least 8 characters!"	Pass	High	Error message successfully shown.

AMF-11	Re-enter password	User re-enter password not same with the previous password field.	Password	Show error message	Show error message “Password do not match!”	Pass	High	Error message successfully shown.
AMF-12	Password visibility for sign-up	User tap icon “eyes” to enable the password visibility.	Password	User’s password visible while typing.	User can see their password while typing.	Pass	Low	Password visibility works properly.
AMF-13	Email verification	User verify the registered email using the verification link sent in the email.	Link verification	User’s email has been verify. Proceed to the login page.	User verify with the received link in the email and redirect to the login page.	Pass	Critical	Email verification works properly.
AMF-14	Send reset password link through email.	User enter the registered email to send reset password link.	Email registered	User receive reset link in the email.	User received the reset link successfully reset the password.	Pass	Critical	Forgot password works properly.

Table 5.5 Testing for homepage module.

Module Name: Homepage Module Test Case Description: Verify real-time display of sensor data on homepage. Testing Objective: To ensure real-time data from the sensor such as water level and sensor status display on the homepage. Testing Approach: Manual black box testing Remarks: Realtime database included.								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
HMF-01	Display profile image.	User change profile image in setting page and back to homepage to see the changes.	User's profile image.	The updated profile image displayed on the homepage.	User profile image updated according to the setting page.	Pass	Medium	Profile image updated.

HMF-02	Display registered username.	User review their username on the homepage.	Username registered	Show the username registered.	User clearly review their username shown in the homepage.	Pass	Medium	Username updated.
HMF-03	Display real-time data from the sensor.	User review the real-time data displayed on the homepage.	Real-time data from the sensor.	Show the real-time data.	User clearly review the real-time data shown.	Pass	Critical	Real-time data updated.
HMF-04	Display sensor status.	User review the sensor status on the homepage.	Sensor status	Show the sensor status is “ON” or “OFF”.	User clearly review the sensor status shown.	Pass	Critical	Status updated.
HMF-05	Info button on sensor.	User tap the info button to view guidance information.	None	Show the guidance information.	User tap the info icon and review it.	Pass	Medium	Info shown successfully.

Table 5.6 Testing for report page module.

Module Name: Report Page Module Test Case Description: Verify sensor data filtering and graph display based on selected data, time and sensor type. Testing Objective: To ensure the report page allow the user to filter sensor data according to the selected sensor, selected date and selected time and display the sensor data in graph form within one hour. Testing Approach: Manual black box testing Remarks: Firestore included.								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
RMF-01	Dropdown sensor list.	User tap the dropdown list to select the wanted sensor.	None	The name of selected sensor will display.	User successfully select a sensor from dropdown list.	Pass	Critical	Dropdown list works properly.
RMF-02	Select data date.	User tap the select date button to select	Selected date	The selected date will display in the button.	User successfully select a date they want.	Pass	Critical	Date picker works properly.

		the date they wanted.						
RMF-03	Filter button	User tap the filter button without select sensor and date.	None	Show error message.	Show error message “No data available for selected date and time.”	Pass	Critical	Error message successfully shown.
RMF-04	Display sensor data in the graph.	User select sensor, date, and tap filter button.	Selected sensor and date.	Display the sensor data in the graph according to filter data.	The successfully displayed according to the filtered.	Pass	Critical	Filter functions work properly.

Table 5.7 Testing for setting page.

Module Name: Setting Page Module Test Case Description: Verify account information update functionally. Testing Objective: To ensure the user can update and view their account information. Testing Approach: Manual black box testing Remarks: Firestore included.								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
SMF-01	Update profile image.	User tap the camera icon to select new image for profile image and tap the update button.	Selected image.	The new profile image displayed.	User's profile image successfully displayed.	Pass	High	Profile image updated.
SMF-02	Update user information.	User update information.	User's information	Show successful message.	User successfully update the	Pass	Critical	User's information

					information and need to login again.			successfully updated.
SMF-03	Email verification	User update valid email.	New email	Send verification link through email registered.	User receive verification email and verify to login.	Pass	Critical	Link verification sent successfully.
SMF-04	Update existing username.	User update username with the existing username.	Existing username	Show error message.	Show error message "Username already exist, please choose another one."	Pass	High	Error message successfully shown.

Table 5.8 Testing for threshold module.

Module Name: Threshold Page Module Test Case Description: Verify threshold page can save the threshold value set by the user. Testing Objective: To ensure threshold page works properly. Testing Approach: Manual black box testing Remarks:								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
TM-01	Enter valid value	User set the threshold value for each sensor.	Threshold value	Successfully save the value.	Successfully save the threshold value.	Pass	Critical	Value successfully stored.
TM-02	Enter empty fields	User saves with empty fields.	None	Show error message	Show error message "Enter max" and "Enter min".	Pass	Critical	Error message successfully shown.

TM-03	Enter min value higher than max.	User enter min value is higher than max value.	Min and max value	Show error message.	Show error message “Min must be less than max”.	Pass	Critical	Error message successfully shown.
TM-04	Enter invalid number	User enter words.	Invalid number	Show error message.	Show error message “Invalid numbers”	Pass	Critical	Error message successfully shown.
TM-05	Enter max pH value is higher than 14.	User enter max pH value is 15.	Max pH value	Show error message	Show error message “Max pH value is 14”	Pass	Critical	Error message successfully shown.

Table 5.9 Testing for notification module.

Module Name: Notification Module Test Case Description: Verify the page can display notification message and remove it. Testing Objective: To ensure notification function work in app if changes happen. Testing Approach: Manual black box testing Remarks: Local notification used.								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
NF-01	Display the list notification message.	User tap the notification icon at the top bar to open notifications page.	None	Display the list of notification message including its details.	User review the notification displayed in the page.	Pass	Critical	Notification displayed successfully.

NF-02	Remove already read notification.	User tap the trash icon at the top bar.	Notifications message.	The notifications have been removed.	User successfully remove the already read notifications.	Pass	Critical	Remove functions works properly.
NF-03	Push notification.	User receive push notification when condition triggered.	Notifications message.	Show push notification in the app and the list in the notification page will increase.	Notification lists increase in the page.	Pass	Critical	Notification received.

5.3.2 Non-Functional Testing

The objective of non-functional testing is to evaluate the overall performance, user experience, and system behaviour of the mobile application. This is to ensure that the user is satisfied with the delivered application and collect feedback from the user for the improvement. This testing also to verifies how well the app work under different conditions. Non-functional testing will covered three types of testing which reliability, usability and efficiency. Each type of testing will bring different value and objective to ensure a quality software delivered.

5.3.2.1 Reliability

The first one is reliability. The main objective of this testing is to ensure hardware and software components operate consistently well overtime without failure. Hence, the test case for this testing is provided below.

Table 5.10 Testing for reliability module.

Module Name: Reliability Module Test Case Description: Verify the application performs consistently well under specified condition. Testing Objective: To evaluate the system is stable, consistent, durable, handle and recover from failure. Testing Approach: Testing based on scenario Remarks:								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
REL-01	Multiple login attempts.	1.Attempts login with valid credentials. 2.Attempts login with invalid credentials 10 times.	Valid and invalid credentials.	Show error message for invalid credentials and redirect to homepage for valid credentials.	The app successfully shows error and successful message after multiple login attempts.	Pass	Critical	The app did not crash.

REL-02	Repeated filtering data	1. Open report page. 2.Change sensor and date for 10 times, then filter it. 3. Observe how app response.	Selected sensor and date.	The app filters each data without failure.	The app successfully filters all the data.	Pass	Critical	The app did not crash.
REL-03	Repeated updating of user information.	1. Update any user information multiple times. 2. Observe the changes happen.	User's information	Each update saves successfully.	The app successfully saves all the data for each update.	Pass	Critical	The app did not crash.

5.3.2.2 Usability

The second is usability testing. The objective is to evaluate the application from user's feedback and determine whether it is intuitive, accessible and easy to interact with. This section will record the user's feedback from the testing procedure.

Table 5.11 Testing for usability module.

Module Name: Usability Module								
Test Case Description: Verify the user can complete a task without confusion. Ensure the system is intuitive, efficient and user-friendly.								
Testing Objective: To evaluate the application from user's feedback and determine whether the app is pleasant and easy to interact.								
Testing Approach: Interview tester.								
Remarks: Gather feedback from the users.								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
USB-01	Fields labels, icon and button clarity.	1. Open each page in app. 2. Observe the fields labels	None	Field labels, icon and button are clear.	All labels, icon and button are clear.	Pass	Critical	Clear and visible.

		icon and button. 3. Ask the user feedback.						
USB-02	Navigation Bar Intuitiveness	1. Tap each icon in navigation bar for redirect to other page. 2. Observe if the user redirect to the correct page.	None	All pages are correctly directed.	All pages are successfully directed.	Pass	Critical	Pages are easy to navigate.
USB-03	Filter sensor data on report page.	1. Ask the user to filter a sensor data according to its date. Then, gather the feedback.	None	Data filters accurately.	All sensor data successfully shown.	Pass	Critical	The flow to select sensor and date are easy to understand.

USB-04	Review real-time sensor reading.	1.Open homepage. 2. Ask the user to review the current reading for each sensor. 3. Ask user's feedback.	Real-time reading.	Data is clearly labelled and readable.	Data is clearly to read and easy to understand.	Pass	Critical	None.
USB-05	Update user information.	1.Open settings page. 2.Update any information. 3.Ask user's feedback.	User's information.	User able to update the information without confusion.	User successfully update the information.	Pass	Critical	The label for the text fields is clearly labelled.

USB-06	Forgot password process.	1. Ask the user to tap forgot password on the login page. 2. Enter the registered email. 3. Check if reset link is received. 4. Ask the user's feedback.	Registered email.	User receives email and successfully changes the password.	User successfully receives email and changes the password.	Pass	Critical	Interactive pop-up for entering the registered email to reset the password.
USB-07	First time user experience.	1. Ask the new user to complete all task without guidance.	None	User complete the task with minimal confusion or assistance.	User successfully complete the task with minimal confusion or assistance.	Pass	Critical	User have minimal confusion but still can continue the task.

5.3.2.3 Efficiency

Lastly is efficiency testing. The objective is to evaluate the system's performance such as response time and processing speed.

Table 5.12 Testing for efficiency module.

Module Name: Efficiency Module Test Case Description: Verify how well the systems performance. Testing Objective: To evaluate the application's speed, resource consumption and scalability. Testing Approach: Record time response. Remarks:								
Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)	Notes
EFF-01	Authentication response time.	1. Enter the valid credentials for login and sign up.	Valid credentials.	Authentication process completes under 3 seconds.	Authentication process completes in 2 seconds.	Pass	High	Response time is acceptable.

		2. Record the time taken to complete it.						
EFF-02	Real-time sensor data loading.	1. Send new sensor data. 2. Record time taken to display the updated data.	Sensor data	Sensor data displayed under 3 seconds.	Sensor data displayed in 2 seconds.	Pass	Critical	None.
EFF-03	Filter report data load time.	1. Filter sensor data on report page. 2. Record time taken to display the data based on the selected sensor and date.	Selected sensor and date.	Filtered data shown under 4 seconds.	Filtered data shown in 2 seconds.	Pass	Critical	Response time is acceptable.

EFF-04	Update user information response time.	1.Update user info on setting page. 2.Record time taken for changes to be saved.	Updated user's information.	Profile update under 3 seconds.	Profile update in 4 seconds.	Fail	High	Profile update did not update under average time.
EFF-05	Push notification delivery time.	1. Trigger a condition to send a notification. 2.Record time taken to notification appear.	Notification message.	Notifications appear under 3 seconds.	Notifications appear in 2 seconds.	Pass	High	Notifications received under average time.

5.4 Summary

A structured testing approach was implement consisting if functional and non-functional testing covered different scenario to ensure both hardware and software components performs well. Hence, test summary table and graph included below to conclude the testing.

Table 5.13 Test summary

Types	Characteristics	Total test cases	Pass	Fail	Success Rate (%)
Hardware	Harvesting Model	2	2	0	100
	Electronic Model	5	3	2	60
Software	Functionality	35	35	0	100
	Reliability	3	3	0	100
	Usability	7	7	0	100
	Efficiency	5	4	1	80

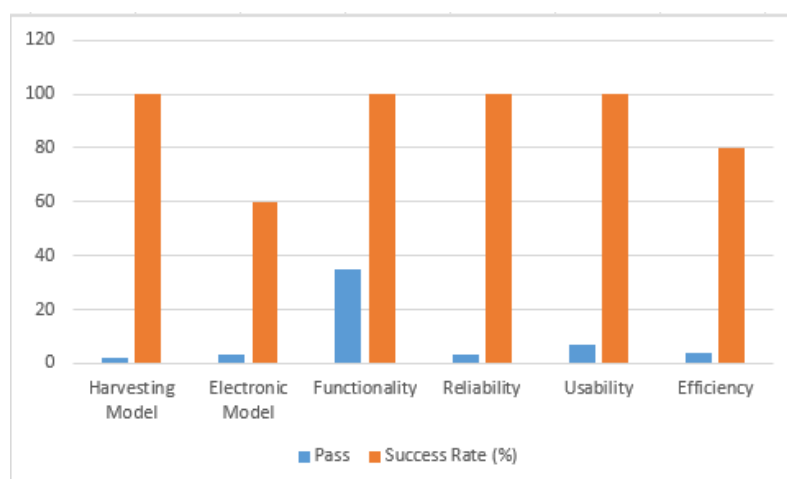


Figure 5.1 Graph for test summary.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Introduction

The Smart Rainwater Harvesting Solution is an IoT-Based System that uses an ESP32 to monitor water level, turbidity, pH and rain intensity. Real-time data sent to Firebase for monitoring and historical data. A mobile app called SmartHydro used to display live sensor data, allows threshold-based alerts and provides user account management. This chapter highlights the challenges and limitations faced during the implementation of Smart Rainwater Harvesting Solution in Open Spaces. It helps to identify areas of improvement for the next phase of development beyond the current prototype.

6.2 Achievements

This section highlights the successful outcome of the Smart Rainwater Harvesting Solution in Open Spaces. Hence, the key objectives achieved listed below.

a) Functional IoT-Based Monitoring System

A complete system using ESP32 allowing real-time data collection and monitoring for multiple sensors. The systems successfully integrate hardware and software including the cloud technology such as Firebase to monitor environmental conditions.

b) Mobile Application (SmartHydro)

Fully functional mobile application was developed using Flutter. It includes Firebase Authentication, real-time dashboard, historical data, notification page, threshold page and settings page.

c) Low-Cost and Scalable Architecture

The use of ESP32, Firebase and Flutter ensures that system is cost effective, and easy to develop in future.

6.3 Limitations

This section explains the constraints that affected the system's performance during the development and testing.

a) Sensor Accuracy Due to Environmental Conditions.

Sensor such as pH and turbidity are very sensitive to the environmental conditions. Various conditions such as excessive exposure to sunlight, debris in water and unstable placement may lead to the inaccurate or inconsistent readings.

b) Power Supply Dependency.

The ESP32 only have 5V and 3.3V source voltage to power all sensors. Connecting all sensors at the same time are not reasonable because it may effect the voltage consistency to the sensors. This will affect the reading consistency of the sensor. Hence, the accuracy rate is reduced.

c) Lack of Waterproof Housing for Sensors

Sensors were exposed to the water during testing making them vulnerable to environmental wear. This affects their longevity in the field development in actual scenarios.

6.4 Improvement work

Based on the limitations above, this section provides suggestions to improve the system in future versions.

a) Industrial-Grade Sensors

Replace current sensors with industrial-grade sensors to improve the measurement accuracy, durability and reliability in outdoor environments.

b) Integrate Solar Power for Standalone Operation

Equip the system with a solar charging circuit to enable long-term operation even in remote areas.

c) Predict Rain Trends using Machine Learning

Implement machine learning algorithms using history data and weather APIs to forecast rain events.

6.5 Future Work

This section discusses potential features that can be added in future development.

a) Add More Sensor Types

Additional sensors such as temperature and humidity to improve monitoring accuracy to measure the climate impact on water quality.

b) Expand System to Cover Multiple Harvesting Sites

Design multiple ESP32 units report to a centralized Firebase structure, allowing the system to be used across multiple sites.

c) Include Water Flow Control Valves

Automate the flow of water to irrigation pipes using solenoid valves to respond to real-time sensor readings or time schedules.

REFERENCES

- Coyne, M., Larson, J., Shieh, J., & Yeon, H. (2019, December 5). *Winning in automation requires a focus on humans*. McKinsey & Company.
https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/winning-in-automation-requires-a-focus-on-humans?utm_source=chatgpt.com#/
- GeeksforGeeks. (2018, July 12). *Agile Software Development Software Engineering*. GeeksforGeeks. https://www.geeksforgeeks.org/software-engineering-agile-software-development/?ref=gcse_outind
- Gitlin, J. (2023, February 8). *IoT integration: here's everything you need to know*. Workato.
<https://www.workato.com/the-connector/iot-integration/>
- Lai, O. (2022, June 26). *Causes and effects of water shortage*. Earth.org.
<https://earth.org/causes-and-effects-of-water-shortage/>
- Lasya, B., Bhavana, Y., Deekshitha, B., & Priya, B. (2020). An Innovative and Effective Electronic Based Automatic Rainwater Harvesting System. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 114–121.
<https://doi.org/10.1109/icssit48917.2020.9214118>

- Rani, K. Pushpa., Srija, K., Jyothianvitha, A., Ashasri, M., Mamatha, I., & Rajesh, G. (2021). Rainwater Harvesting for Smart Water Management Using IoT. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*.
<https://doi.org/10.1109/iciccs51141.2021.9432365>
- Ranjan, V., Reddy, M. V., Irshad, M., & Joshi, N. (2020, March 1). *The Internet of Things (IOT) Based Smart Rainwater Harvesting System*. IEEE Xplore.
<https://doi.org/10.1109/ICSC48311.2020.9182767>
- Razali, M. H. H., Puteh, A. Q., Sulaiman, A. H., & Yatim, M. H. M. (2023, January 18). *Smart Rainwater Harvesting System for Sustainable Agricultural Irrigation and Drainage System*. Wwww.intechopen.com; IntechOpen.
<https://www.intechopen.com/chapters/82188>
- Smoor, T. (2024, August 27). *6 facts you should know about water use in agriculture*. Cargill.com.
<https://www.cargill.com/story/agriculture-water-use#:~:text=Agriculture%20uses%20a%20lot%20of,Nations'%20World%20Water%20Development%20Report>
- Sunar, N. M., Kosat, A. T., Harun, H., Hamid, N. H. A., Khalid, A., Zaidi, E., & Azhar, A. T. S. (2018). Development of Smart Kit Rainwater Harvesting System for Potable Water Purpose. *International Journal of Engineering & Technology*, 7(4.30), 338–341.
<https://doi.org/10.14419/ijet.v7i4.30.22304>