



Faculty of Computer Science and Information Technology

***SOLAR PANEL TRACKING SYSTEM WITH BATTERY AND
LIGHT INTENSITY MONITORING FOR AUTOMATED
IRRIGATION IN POLYBAG FARMING***

Syed Arif Azri Bin Wan Saiful Bahari

Bachelor of Software Engineering with Honours

2025

**SOLAR PANEL TRACKING SYSTEM WITH BATTERY AND LIGHT INTENSITY
MONITORING FOR AUTOMATED IRRIGATION IN POLYBAG FARMING**

SYED ARIF AZRI BIN WAN SAIFUL BAHARI

This project is submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

**SISTEM PENJEJAKAN PANEL SURIA DENGAN PEMANTAUAN BATERI DAN
KEAMATAN CAHAYA UNTUK PENGAIRAN AUTOMATIK DALAM PERTANIAN
POLIBEG.**

SYED ARIF AZRI BIN WAN SAIFUL BAHARI

Projek ini merupakan salah satu keperluan untuk
ijazah Sarjana Muda Kejuruteraan Perisian dengan Kepujian.

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signed,



.....

Name: Syed Arif Azri Bin Wan Saiful Bahari

Matric No.: 78559

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 16 January 2025

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious, the Most Merciful, I begin by expressing my heartfelt gratitude to **Allah (SWT)** for granting me the strength, wisdom, and guidance throughout the course of this project. Without His blessings, this achievement would not have been possible. I am profoundly grateful to my supervisor, **Assoc. Prof. Dr. Noor Alamshah Bin Bolhassan**, whose expertise, continuous support, and valuable feedback have greatly shaped the success of this project. I would also like to extend my deepest thanks to my family for their unwavering love, understanding, and constant encouragement, which has been a source of motivation and strength throughout my academic journey. A special thank you to my friends and colleagues for their assistance in research, discussions, and testing, which has helped refine and improve the quality of this project. I am also thankful to my university for providing the necessary resources, facilities, and a conducive learning environment that have contributed significantly to the completion of this project. Lastly, I express my gratitude to everyone who has been a part of this process and supported me in any way. May this project serve as a small contribution to the fields of Software Engineering and Agriculture, and may it benefit others as much as I have gained from it.

ABSTRACT

This project develops a Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming to support sustainability and improve resource management in small-scale agriculture. The system employs dual-axis solar tracking to enhance sunlight capture throughout the day, while real-time monitoring of battery voltage and light intensity ensures reliable operation and informed irrigation control. An IoT-based communication system using Firebase enables seamless data transfer between the hardware and a Flutter-based mobile application, providing users with live insights and remote system control. By integrating renewable energy with smart monitoring technology, the system offers a practical, low-cost solution tailored to polybag farming, contributing to the broader adoption of green energy and digital agriculture practices.

ABSTRAK

Projek ini membangunkan Sistem Penjejakan Panel Suria dengan Pemantauan Bateri dan Keamatan Cahaya untuk Pengairan Automatik dalam Pertanian Polibeg bagi meningkatkan keberkesanan penggunaan tenaga dan menyokong kelestarian dalam pertanian berskala kecil. Sistem ini menggunakan penjejakan suria dwi-paksi untuk memaksimumkan penyerapan cahaya matahari sepanjang hari, disokong oleh pemantauan voltan bateri dan keamatan cahaya secara masa nyata. Data ini dihantar melalui sambungan IoT menggunakan Firebase, yang membolehkan pengguna memantau dan mengawal sistem secara langsung melalui aplikasi mudah alih berasaskan Flutter. Pendekatan ini menyediakan penyelesaian yang praktikal, kos efektif, dan mudah diakses, sejajar dengan keperluan pertanian polibeg. Dengan menggabungkan tenaga boleh diperbaharui dan teknologi pemantauan pintar, projek ini menyumbang kepada amalan pertanian mampan dan menggalakkan penggunaan teknologi hijau dalam sektor pertanian.

TABLE OF CONTENTS

DECLARATION	I
ACKNOWLEDGEMENT	II
ABSTRACT	III
ABSTRAK	IV
LIST OF TABLES	X
LIST OF FIGURES.....	XI
CHAPTER 1 INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Problem Statement	2
1.3 Project Scope	3
1.4 Aim and Objectives	3
1.5 Brief Methodology.....	3
1.6 Significance of Project.....	5
1.7 Project Outline.....	6
1.7.1 Chapter 1: Introduction.	6
1.7.2 Chapter 2: Literature Review.....	6
1.7.3 Chapter 3: Requirement Analysis and Design.	6
1.8 Summary.....	7
CHAPTER 2 BACKGROUND STUDY	8
2.1 Introduction.....	8
2.2 Review of Existing System	9
2.2.1 IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	9
2.2.2 Automatic Single-Axis Solar Tracking System	13
2.2.3 Solar Tracking System with Renewable Energy Optimization	17
2.3 Comparison between Existing System	21
2.4 The Shortcoming in Existing Systems	24
2.5 Tools and Technology Used in the Development of the Proposed System.....	28
2.5.1 Hardware.....	28
2.5.1.1 Solar Panel (2W).....	28
2.5.1.2 Light-Dependent Resistors (LDRs).....	28
2.5.1.3 Servo Motors.....	28
2.5.1.4 Microcontroller (ESP32)	29
2.5.1.5 Battery (3.7V Li-on)	29

2.5.1.6	Voltage Sensor Module	29
2.5.1.7	Structural Materials	29
2.5.1.8	Miscellaneous Electronics	29
2.5.2	Software	29
2.5.2.1	PlatformIO (Visual Studio Code Extension)	29
2.5.2.2	Firebase (Realtime Database)	30
2.5.2.3	Flutter (Visual Studio Code Extension).....	30
2.5.2.4	Firebase (Cloud Firestore).....	30
2.5.2.5	Figma (Interface Design)	30
2.6	Summary.....	30
CHAPTER 3 REQUIREMENT ANALYSIS AND DESIGN		32
3.1	Introduction.....	32
3.2	Requirement Analysis	33
3.2.1	Method For Gathering Requirements	33
3.2.1.1	Survey Using Google Form	33
3.2.2	List Of Required Requirements for The Proposed System	50
3.2.2.1	Functional Requirements	50
3.2.2.2	Non-Functional Requirements	53
3.2.2.3	Hardware requirements	54
3.2.2.4	Software Requirements	57
3.3	System Design	60
3.3.1	System Architecture.....	60
3.3.2	Flowchart	62
3.3.3	UML Diagram	65
3.3.3.1	Use Case Diagram	65
3.3.3.2	Class Diagram.....	67
3.3.3.3	Activity Diagram	69
3.3.3.4	Sequence Diagram	71
3.3.4	Block Diagram.....	73
3.3.5	Interface Design.....	75
3.4	Summary.....	80
CHAPTER 4 IMPLEMENTATION		82
4.1	Introduction.....	82
4.2	Implementation Tool	84

4.2.1	Software Tools	84
4.2.1.1	Visual Studio Code	84
4.2.1.2	Flutter SDK.....	84
4.2.1.3	PlatformIO	85
4.2.1.4	Firebase	86
4.2.1.5	Figma.....	86
4.2.1.6	Git & GitHub.....	87
4.2.2	Hardware Tools & Components	88
4.3	Solar Panel Tracking System Prototype.....	92
4.3.1	Solar Tracking Module	94
4.3.1.1	Functional Description	95
4.3.1.2	Hardware and Wiring Implementation	96
4.3.1.3	Software/Code Implementation	96
4.3.1.1	System Integration.....	97
4.3.2	Light Intensity Monitoring Module	99
4.3.2.1	Functional Description	99
4.3.2.2	Hardware and Wiring Implementation	100
4.3.2.3	Software/Code Implementation	100
4.3.2.4	System Integration.....	101
4.3.3	Battery Monitoring Module	102
4.3.3.1	Functional Description	102
4.3.3.2	Hardware and Wiring Implementation	103
4.3.3.3	Software/Code Implementation	103
4.3.3.4	System Integration.....	104
4.3.4	IoT Communication Module.....	105
4.3.4.1	Functional Description	106
4.3.4.2	Hardware and Wiring Implementation	107
4.3.4.3	Software/Code Implementation	107
4.3.4.4	System Integration.....	108
4.3.5	Mobile Application Module	110
4.3.5.1	Functional Description	111
4.3.5.2	System Integration.....	111
4.3.5.3	Key Features	112
4.4	Findings and Challenges Faced During Implementation	113

4.5	Summary.....	115
CHAPTER 5 TESTING		116
5.1	Introduction.....	116
5.2	Testing Strategy.....	116
5.2.1	Levels of Testing.....	117
5.2.2	Test Criteria and Pass Conditions.....	117
5.3	Test Environment Setup.....	118
5.3.1	Location & Environment Conditions	118
5.3.2	Test Devices and Diagnostic Tools.....	119
5.3.3	Assumptions and Limitations	119
5.4	Module Testing.....	120
5.4.1	Solar Tracking Module	120
5.4.1.1	Functionality Testing.....	120
5.4.1.2	Reliability Testing.....	122
5.4.1.3	Efficiency Testing.....	124
5.4.2	Light Intensity Monitoring Module	126
5.4.2.1	Functionality Testing.....	126
5.4.2.2	Reliability Testing.....	128
5.4.2.3	Efficiency Testing.....	130
5.4.3	Battery Monitoring Module	132
5.4.3.1	Functionality Testing.....	132
5.4.3.2	Reliability Testing.....	133
5.4.3.3	Efficiency Testing.....	135
5.4.4	IoT Communication Module.....	136
5.4.4.1	Functionality Testing.....	136
5.4.4.2	Reliability Testing.....	138
5.4.4.3	Efficiency Testing.....	140
5.4.5	Mobile Application Module	142
5.4.5.1	Functionality Testing.....	142
5.4.5.2	Reliability Testing.....	144
5.4.5.3	Efficiency Testing.....	145
5.4.5.4	Usability Testing	147
5.5	Summary of Results	149
CHAPTER 6 CONCLUSION AND FUTURE WORK.....		152

6.1	Introduction.....	152
6.2	Project Achievement.....	152
6.3	Project Limitation.....	154
6.4	Future work	155
6.5	summary.....	157
REFERENCES.....		158

LIST OF TABLES

Table 1.1: Schedule for each task for Final Year Project 1	Error! Bookmark not defined.
Table 2.1: Comparison Table of the Existing System	21
Table 2.2: Comparison of the Proposed System and the Existing System	24
Table 3.1: Functional requirements for the proposed system	51
Table 3.2: Non-Functional requirements of the proposed system	53
Table 3.3: Hardware requirements of the proposed system	55
Table 3.4: Software requirements of the proposed system	58
Table 4.1: List of Hardware Tools & Components	88
Table 4.2: Hardware/Wiring Connection for Solar Panel Tracking Model	96
Table 4.3: Hardware/Wiring Connection for Battery Module	103
Table 4.4: Hardware/Wiring Connection for IoT Communication	107
Table 5.1: List of Level of Testing	117
Table 5.2: List of Test Device/Tools	119
Table 5.3: Solar Tacking Module's Functionality Testing	120
Table 5.4: Solar Tacking Module's Reliability Testing	122
Table 5.5: Solar Tacking Module's Efficiency Testing	124
Table 5.6: Light Intensity Monitoring Module's Functionality Testing	126
Table 5.7: Light Intensity Monitoring Module's Reliability Testing	128
Table 5.8: Light Intensity Monitoring Module's Efficiency Testing	130
Table 5.9: Battery Monitoring Module's Functionality Testing	132
Table 5.10: Battery Monitoring Module's Reliability Testing	133
Table 5.11: Battery Monitoring Module's Efficiency Testing	135
Table 5.12: IoT Communication Module's Functionality Testing	136
Table 5.13: IoT Communication Module's Reliability Testing	138
Table 5.14: IoT Communication Module's Efficiency Testing	140
Table 5.15: Mobile Application Module's Functionality Testing	142
Table 5.16: Mobile Application Module's Reliability Testing	144
Table 5.17: Mobile Application Module's Efficiency Testing	145
Table 5.18: Mobile Application Module's Usability Testing	147
Table 5.19: Overall Test Case Result	149
Table 6.1: Objective & Achievement	152

LIST OF FIGURES

Figure 1.1: Agile Methodology Phases	4
Figure 1.2: Gantt Chart for Final Year Project 1	Error! Bookmark not defined.
Figure 2.1: Circuit Diagram - IoT System (Shah et al., 2021)	10
Figure 2.2: Prototype Showing test output during daytime (Shah et al., 2021)	10
Figure 2.3: Prototype of Proposed Model (Shah et al., 2021).....	10
Figure 2.4: Mobile Application to Monitor Harvesting Voltage (Shah et al., 2021)	11
Figure 2.5: Mobile Application to Control Solar Power System (Shah et al., 2021)	11
Figure 2.6: Schematic Diagram of a Solar PV System with an Automatic Single-Axis Solar Tracking System (Reza et al., 2021)	13
Figure 2.7: Flowchart of Single-Axis Solar Tracking System (Reza et al., 2021)	15
Figure 2.8: Solar Photovoltaic Panel Power Output with and without Solar Tracking System (Reza et al., 2021)	15
Figure 2.9: Solar Panel Implementing Single Axis Solar Tracking Device (Talebi & Aly, 2024)	18
Figure 2.10: Diagram of a Single Axis Solar Tracker (Talebi & Aly, 2024)	19
Figure 2.11: Power Output Results are Displayed Graphically against time for both fixed mode and autonomous system tracking. (Talebi & Aly, 2024)	19
Figure 3.1: Familiarity with solar energy systems	36
Figure 3.2: Respondent's accustomed with solar panel energy system	36
Figure 3.3: Respondent's biggest energy concerns.....	38
Figure 3.4: Challenges faced by respondents with current renewable energy solutions	39
Figure 3.5: Respondent's perception toward the importance of tracking how energy is used and stored.	40
Figure 3.6: Respondent's perception on the importance of knowing battery's charge level in real-time.....	42
Figure 3.7: Respondent's preference in monitoring system's battery.	43
Figure 3.8: Features respondents find useful in solar panel system	44
Figure 3.9: How respondents want to access the system.....	45
Figure 3.10: Respondent's interest in notification/reports feature.....	46
Figure 3.11: Respondents budget for a solar energy system.	47
Figure 3.12: System Architecture for Proposed System.....	60
Figure 3.13: Flowchart of the proposed system	62
Figure 3.14: Use Case of the Proposed System.....	65
Figure 3.15: Class Diagram for the Proposed System.....	67
Figure 3.16: Activity Diagram for the proposed system	69
Figure 3.17: Sequence Diagram for the proposed system	71
Figure 3.18: Block Diagram for the proposed system.....	73
Figure 3.19: Login Page of the proposed system.	75
Figure 3.20: Dashboard page for the proposed system.	76
Figure 3.21: Statistics page for the proposed system	77
Figure 3.22: Control Page of the proposed system	78
Figure 3.23: Profile Page of the proposed system.....	79
Figure 4.1: Visual Studio Code	84
Figure 4.2: Flutter SDK	84
Figure 4.3: PlatformIO	85
Figure 4.4: Firebase.....	86
Figure 4.5: Figma.....	86
Figure 4.6: Git & GitHub	87
Figure 4.7: System Hierarchy Diagram.....	93
Figure 4.8: Overview of the Project's Software	94
Figure 4.9: Solar Tracker Module's Code Example	97
Figure 4.10: JSON Payloads for Solar Tracker Module.....	98
Figure 4.11: Mobile App Interface Presentation of the Panel Orientation and Light Distribution.....	98
Figure 4.12: Light Intensity Module's Code Example	100

Figure 4.13: Total Light Intensity Display	101
Figure 4.14: Battery Monitoring Module's Code Example	104
Figure 4.15: Graph for the Battery Monitoring in Mobile App.....	105
Figure 4.16: Firebase Connection Code.....	108
Figure 4.17: Example of Data Payload in Realtime Database	109
Figure 4.18: Example of the Control Panel of the App	110
Figure 4.19: Statistics Page	112
Figure 4.20: Control Page	112
Figure 4.21: Dashboard Page	113

CHAPTER 1

INTRODUCTION

1.1 Introduction

In recent years, the agricultural industry has increasingly relied on sustainable energy solutions to meet the growing need for efficient and ecologically friendly farming operations. Renewable energy sources, including hydropower, biomass, wind, geothermal, and solar, have proven critical to this transformation. According to Khandekar et al. (2023), the most frequent renewable energy sources are hydropower (25%), biomass (48%), wind (19%), geothermal (2%), and solar (5%), with hydropower, wind, and solar all expanding rapidly. Solar energy has emerged as an essential resource for agriculture, providing a clean, renewable alternative to conventional energy sources. Agriculture offers several options for properly utilizing solar energy, so supporting the green economy—a priority highlighted in recent government measures (Elvis Vikram Novaldo et al., 2022).

Solar energy is crucial to improving energy efficiency for farming operations. The fixed orientation of conventional solar panels, however, makes it difficult to maximize the potential of solar energy. This limitation is overcome by a solar tracking system, which dynamically moves solar panels to follow the course of the sun. This maximizes energy absorption throughout the day and increases the energy yield of the system.

For continuous agricultural operations, a stable power supply is essential in addition to energy generation. By continuously monitoring battery charge levels, battery energy monitoring helps to prevent power shortages that can interfere with vital activities. Because of this

monitoring, energy storage can be managed effectively, allowing for reliable system operation even in the absence of sunlight.

By continuously monitoring light intensity in real time, the system ensures accurate tracking of solar exposure and supports context-aware agricultural operations such as automated irrigation. Light data allows the system to respond effectively to environmental changes, adjusting operations based on actual illumination levels. This enhances the precision and reliability of the system in managing crop needs, especially in polybag farming scenarios. The integration of light intensity sensing with solar tracking and battery monitoring offers a targeted solution for improving situational awareness and environmental responsiveness in agricultural settings.

1.2 Problem Statement

Conventional solar energy systems that rely on fixed-position panels are unable to adapt to the sun's changing position throughout the day, resulting in suboptimal solar energy capture. This limitation is especially critical in agricultural settings, where energy reliability directly impacts essential operations such as irrigation. Furthermore, the absence of real-time battery monitoring makes it challenging to assess the availability of stored energy, leading to difficulties in planning and sustaining power-dependent activities. Similarly, without real-time light intensity monitoring, the system lacks environmental awareness, reducing its ability to make informed operational decisions under varying sunlight conditions. These limitations hinder the full potential of solar energy in small-scale agriculture. Addressing them through solar tracking, battery status awareness, and ambient light sensing is essential to improving sustainability, ensuring efficient resource use, and advancing green agricultural practices.

1.3 Project Scope

This project aims to develop, construct, and evaluate a comprehensive solar energy system for small-scale polybag farming. Important elements include choosing the appropriate solar panel size and power to satisfy the farm's energy requirements, particularly for irrigation and other necessary operations. Batteries will be selected based on their ability to store enough energy to meet the farm's needs during low-light hours. Furthermore, A small-scale test configuration will be used to assess the system's performance and reliability. The initiative is limited to small-scale polybag farming using solar as the sole energy source and does not include larger commercial farms or other renewable sources.

1.4 Aim and Objectives

The project aims to improve the current concept of polybag farming. Three objectives have been developed to accomplish this purpose:

- To develop a solar panel tracking system that continuously adjusts the panels' position in response to the sun's movement throughout the day for optimal sunlight capture.
- To implement a real-time battery monitoring system that continuously tracks and reports the batteries' charge levels.
- To enable real-time monitoring of light intensity for adaptive decision-making in agricultural operations.

1.5 Brief Methodology

The development of this solar panel tracking system with battery and light intensity monitoring will use Agile methodology. Agile methodology is a project management framework that

divides projects into numerous dynamic phases, commonly known as sprints. The figure below shows the phases of the Agile Methodology.

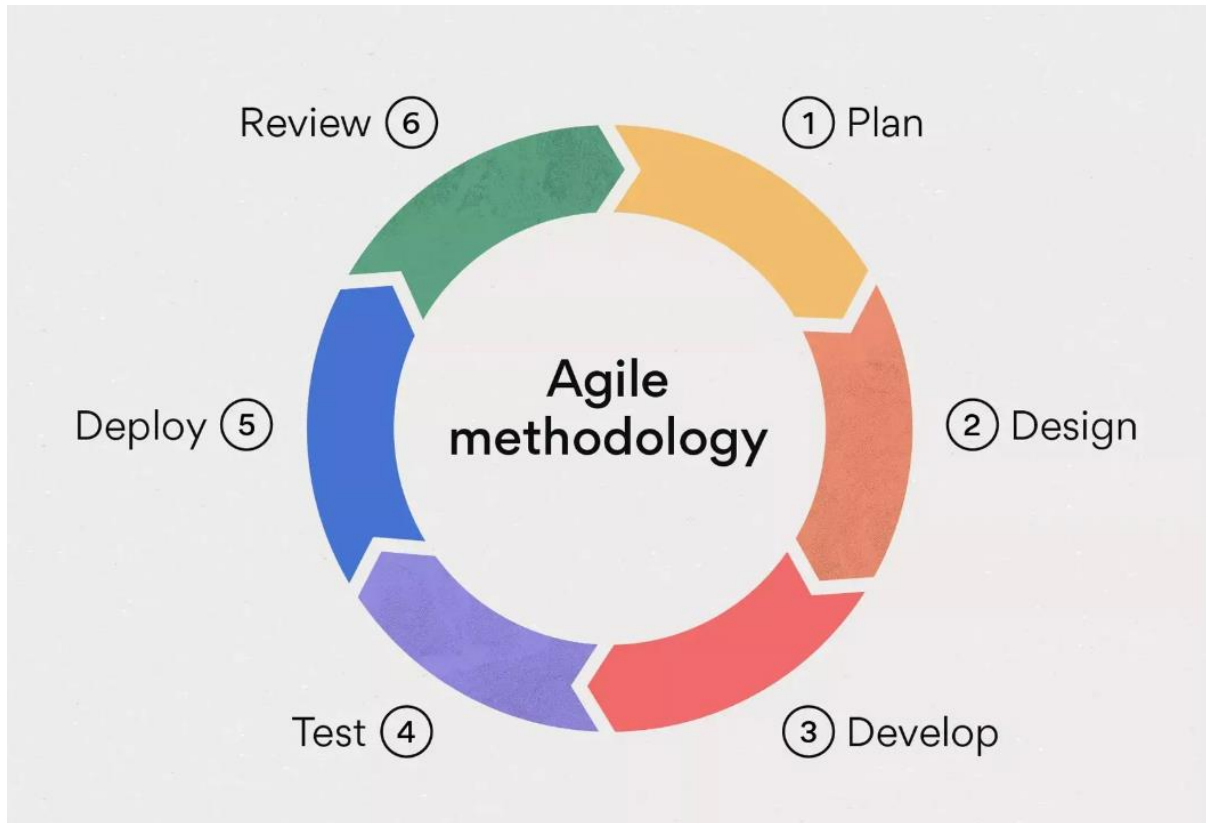


Figure 1.1: Agile Methodology Phases

Agile project management is frequently utilized in software development, including application development. The reason for this is that software is always evolving, and product requirements must also adapt. Linear project management techniques, such as the waterfall approach, are therefore less successful.

Agile procedures in software development offer several advantages, one of which is the flexibility to change tactics quickly without interfering with the project's progress. Changing strategies can be difficult and disrupt the rest of the project plan because phases in the classic waterfall method flow into each other. Project management can be difficult in the traditional

sense because software development is a much more flexible industry. In software development, Agile project management is preferred in part because of this. (Laoyan, 2024).

1.6 Significance of Project

Through the integration of solar energy with dual-axis tracking and intelligent monitoring systems, this project enhances the sustainability and operational efficiency of polybag farming. By overcoming the limitations of fixed-position panels, it implements a solar panel tracking system that dynamically follows the sun's path, thereby maximizing sunlight exposure and improving energy harvesting throughout the day (Khandekar et al., 2023). In addition, the inclusion of a real-time battery monitoring system provides accurate visibility into energy storage levels, enabling more informed energy utilization in farming operations.

A key innovation of this project is the incorporation of real-time light intensity monitoring, which allows the system to adapt its behavior based on actual environmental conditions. This ensures that light-sensitive operations—such as solar tracking and irrigation—are executed more efficiently and in alignment with real-world sunlight availability. With shifting energy demands and an increasing focus on sustainable agricultural practices, this initiative plays a vital role in advancing green technologies within the agricultural sector (Elvis Vikram Novaldo et al., 2022).

Ultimately, the project offers a model for environmentally responsive and resource-efficient agriculture, contributing to broader efforts to promote renewable energy integration and sustainable development in farming systems..

1.7 Project Outline

1.7.1 Chapter 1: Introduction.

Chapter 1 provides an overview of the project, highlighting its objectives, significance, and scope. It introduces the core problem being addressed: the lack of adaptability and environmental awareness in conventional solar energy systems used in small-scale agriculture. To overcome this, the project focuses on developing a solar panel tracking system with dual-axis movement, integrated with real-time battery and light intensity monitoring. This chapter outlines the project's aim to improve agricultural sustainability through IoT-enabled solutions and renewable energy technologies, while addressing the practical challenges faced in resource-constrained farming environments. It also presents the project's objectives, methodology, and overall structure.

1.7.2 Chapter 2: Literature Review

Chapter 2 reviews existing literature and research related to solar panel tracking systems, real-time battery monitoring, and light intensity sensing within the context of agricultural automation. It explores previous studies on dual-axis solar tracking mechanisms, battery monitoring techniques, and the role of light sensors in optimizing farm operations such as irrigation. By analyzing current advancements and limitations in these areas, this chapter identifies key research gaps—particularly the lack of integrated systems combining solar tracking with real-time environmental monitoring—and establishes the rationale for this project. It provides a solid theoretical foundation by synthesizing relevant studies and technologies that inform the system's design and implementation.

1.7.3 Chapter 3: Requirement Analysis and Design.

Chapter 3 delves into the detailed requirements and design for the project. It outlines the functional and non-functional requirements of the solar panel tracking system, battery and light intensity monitoring. The chapter presents the design specifications, including system architecture, hardware and software requirements, and data flow. It describes the technical

approach, tools, and technologies used to develop the system, ensuring the design aligns with the project's goals and objectives. This chapter sets the groundwork for the system's development and implementation phases.

1.8 Summary

This research explores how solar energy can be integrated with intelligent technologies to improve the sustainability and efficiency of small-scale polybag farming. The proposed system features three core components: dual-axis solar panel tracking, real-time battery energy monitoring, and real-time light intensity sensing. Solar tracking ensures maximum sunlight capture by continuously adjusting the panel's orientation throughout the day. Battery monitoring provides accurate insights into energy storage levels, enabling better planning and uninterrupted operation of farm processes. Meanwhile, light intensity monitoring allows the system to respond to varying environmental conditions, ensuring that sunlight-dependent operations—such as irrigation—are executed effectively and only when conditions are optimal.

This project addresses key challenges in traditional systems, such as the inefficiency of fixed-position panels and the absence of real-time environmental feedback. By offering a scalable and responsive system tailored for small-scale agriculture, the research promotes the practical adoption of renewable energy in farming while aligning with broader sustainability and green economy goals. Developed iteratively using the Agile methodology, the project aims to contribute to improved energy reliability, enhanced agricultural productivity, and a model for environmentally responsive farming practices.

CHAPTER 2

BACKGROUND STUDY

2.1 Introduction

Solar energy, a renewable and environmentally friendly alternative to fossil fuels, plays a critical role in advancing sustainable energy solutions. To fully harness its potential, innovative technologies such as solar tracking systems, real-time battery monitoring, and ambient light sensing are increasingly essential. Dual-axis solar tracking enhances energy acquisition by dynamically adjusting panel orientation to follow the sun's movement throughout the day. Simultaneously, battery monitoring offers real-time visibility into energy storage levels, improving system reliability and planning. Light intensity monitoring provides environmental awareness, enabling systems to respond intelligently to fluctuating sunlight conditions—an especially valuable feature for light-dependent operations like irrigation.

These technologies are vital for maintaining a consistent and reliable energy supply in agriculture, where power demands shift based on seasonal cycles and geographic factors. However, many existing renewable energy solutions fall short of addressing the specific challenges faced by small-scale agricultural systems, particularly in polybag farming. These challenges include the lack of real-time environmental monitoring, poor adaptability to varying conditions, and limited integration of complementary technologies.

To overcome these gaps, this study proposes an integrated system that combines dual-axis solar tracking, real-time battery monitoring, and real-time light intensity sensing. Designed with sustainability and resilience in mind, the system targets the unique demands of polybag

farming, offering a practical model for smart, small-scale agricultural innovation that aligns with green economy principles and future-ready farming methods.

2.2 Review of Existing System

2.2.1 IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System

The IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System integrates IoT technology to monitor the power output of solar panels in real-time. The development of this system used Arduino Uno and a Wi-Fi module. Then all the information collected from the solar panel will be displayed on an LCD. The database is generated in the cloud, and a mobile app is constructed for real-time monitoring and system control. The application allows users to operate and control the system remotely. It includes a single axis tracking mechanism to enhance solar energy capture by aligning the panel with the sun's position throughout the day. The system's IoT functionality ensures real-time data acquisition and system responsiveness.

The main key feature of the system is that it leverages IoT devices and sensors to monitor solar panel performance metrics such as voltage in real time. [Refer to Figure 2.1]. Next, it incorporates a **Single-Axis Solar Tracking**, a motorized mechanism to adjust the panel's orientation based on the sun's movement, increasing energy capture. [Refer to Figure 2.2 and Figure 2.3]. All of this is controlled by its **Data Acquisition and Control** feature which remotely accesses the performance data of the system like fault detection. [Refer to Figure 2.4 and Figure 2.5].

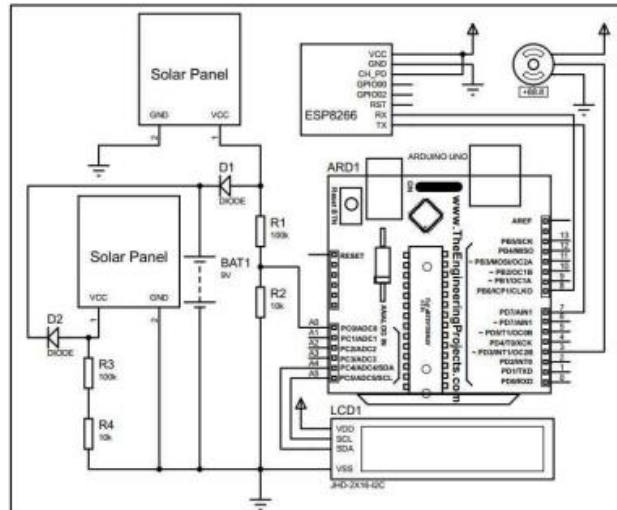


Figure 2.1: Circuit Diagram - IoT System (Shah et al., 2021)

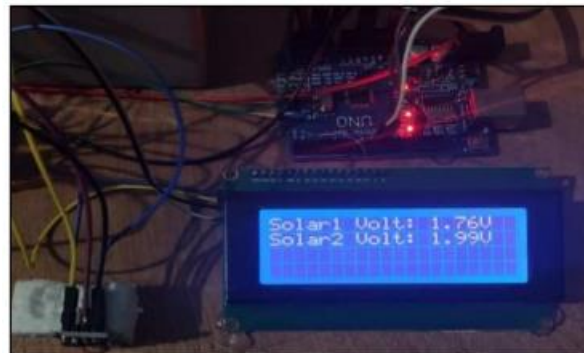


Figure 2.2: Prototype Showing test output during daytime (Shah et al., 2021)

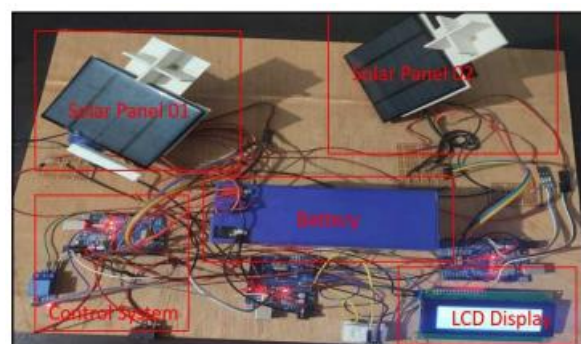


Figure 2.3: Prototype of Proposed Model (Shah et al., 2021)



Figure 2.4: Mobile Application to Monitor Harvesting Voltage (Shah et al., 2021)

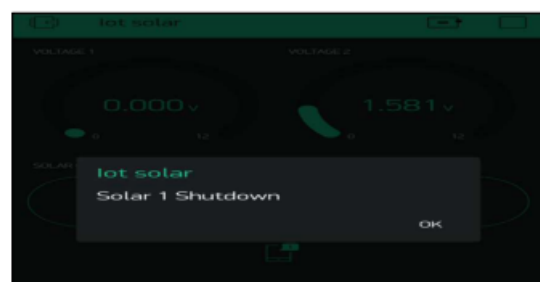


Figure 2.5: Mobile Application to Control Solar Power System (Shah et al., 2021)

This system's strength stems from its **Real-Time Data Monitoring**. IoT sensors provide immediate insights into the system performance like voltage. After that, due to its tracking system, it has enhanced solar energy capture that ensures consistent alignment with the sun. This significantly improves the power output compared to fixed systems. (Shah et al., 2021). In addition, the remote accessibility of the system enables the users to monitor and control the system from remote locations, enhancing convenience and operational efficiency.

However, despite the system's innovative features, the system still has notable limitations. One of them is its lack of application-specific design. The system is designed for general solar energy generation and does not address specific needs like energy allocation for agricultural applications. Other than that, the system has no battery energy monitoring. It lacks real-time tracking of battery storage levels, which is crucial for ensuring energy

availability during low sunlight periods. Next, the system has limited energy prioritization. The system does not incorporate adaptive algorithms to prioritize critical operations, such as irrigation, which is essential for agricultural use.

This system provides valuable insights into integrating IoT technology for real-time monitoring and the benefits of solar tracking. However, it does not address the unique energy demands of agricultural processes. My proposed system builds upon these features by incorporating:

- **Battery Energy Monitoring:** Ensuring operational reliability through real-time insights into storage levels.
- **Dual-Axis Tracking:** Increasing solar energy capture by altering the panel's position in both the horizontal and vertical axes.
- **Adaptive Energy Allocation:** Prioritizing energy use for critical farming operations like irrigation.

The IoT-Based Power Monitoring of Solar Panel Incorporating Tracking system demonstrates the potential of IoT and tracking technologies to enhance solar panel performance. However, its limitations in application-specific design, energy prioritization, and storage monitoring leave significant gaps that my proposed system addresses, particularly for agricultural applications.

2.2.2 Automatic Single-Axis Solar Tracking System

The Automatic Single-Axis Solar Tracking System increases the effectiveness of solar photovoltaic (PV) panels by employing a microcontroller-based tracking mechanism to align the panel with the sun's position throughout the day. The device uses light-dependent resistors (LDRs) to measure sunlight intensity and a motorized tracking system to ensure optimal energy capture. This tracking technology outperforms stationary systems in terms of solar panel power output. Furthermore, the system's price and accessibility make it a feasible renewable energy generation option, particularly in places where solar energy is abundant. [Refer to Figure 2.6 for system design].

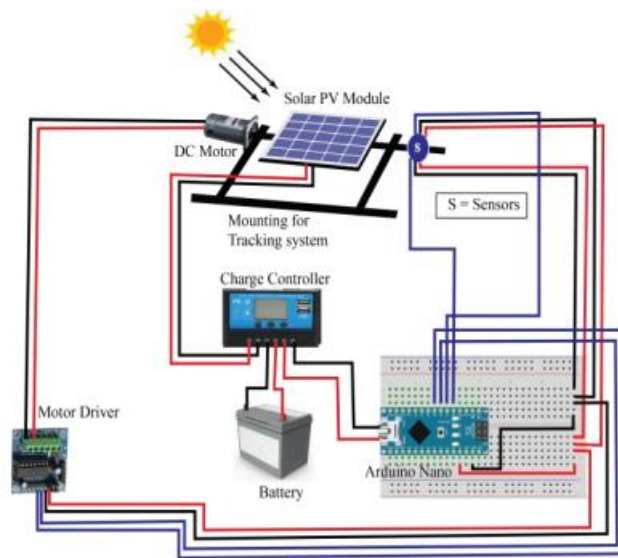


Figure 2.6: Schematic Diagram of a Solar PV System with an Automatic Single-Axis Solar Tracking System (Reza et al., 2021)

One of the system's important characteristics is the Single-Axis Solar Tracking mechanism, which adjusts the panel's orientation in reaction to sunlight detected by the LDR sensors. A low-speed DC motor (10 RPM) turns the panel smoothly, keeping it in alignment with the sun's position. This ensures that the panel receives maximum sunlight exposure,

even as the sun moves during the day, enhancing total efficiency. [Figure 2.7 illustrates the tracking technique]. Another feature is Energy Efficiency Optimization, which improves energy capture during off-peak solar hours by following the sun's motion. Experimental results show higher voltage and current outputs than fixed panels, with significant performance gains in the morning and afternoon. [Figure 2.8 displays performance data].

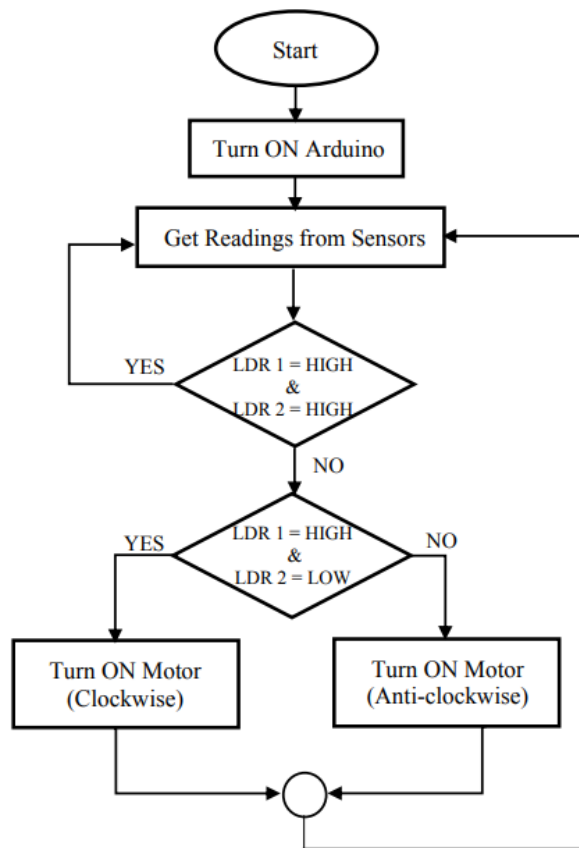


Figure 2.7: Flowchart of Single-Axis Solar Tracking System (Reza et al., 2021)

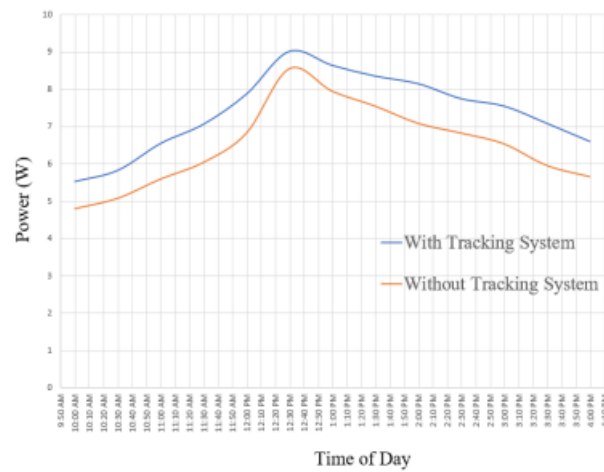


Figure 2.8: Solar Photovoltaic Panel Power Output with and without Solar Tracking System (Reza et al., 2021)

The system's strength rests in its low cost and scalability. The design is simple, utilizing low-cost components such as the Arduino Nano microprocessor and readily available LDR sensors, making it suitable for mass application. Furthermore, the system is very scalable and can be modified to accommodate larger panels or even larger-scale solar power systems with simple modifications. This versatility underlines its suitability for both household and industrial applications. Furthermore, because the microcontroller and motor are powered by the solar panel, the tracking system provides consistent performance without the need for external power.

Despite its merits, the method has some drawbacks. In places with variable solar angles, the Single-Axis Tracking Mechanism may not catch as much solar energy as dual-axis systems. Another disadvantage is that it is not integrated with real-time monitoring systems, which would allow users to track performance indicators, discover errors, and enhance efficiency remotely. Furthermore, the system is primarily built for small-scale solar panels, such as a 10W panel, and requires modifications, such as gear mechanisms, to successfully support medium- and large-scale panels.

This technology sheds light on how solar tracking methods can boost photovoltaic panel efficiency. However, it lacks IoT-based functions for remote monitoring and adaptive energy allocation, both of which are increasingly important in renewable energy applications. My proposed approach enhances these advantages by integrating the following:

- **IoT-Based Monitoring:** Real-time data capture and fault detection to improve system responsiveness and performance tracking.

- **Dual-Axis Tracking:** Increasing solar energy capture by altering the panel's position in both the horizontal and vertical axes.
- **Scalable Design:** Ensuring adaptability to varied panel sizes and uses, including agricultural and industrial use.

The Automatic Single-Axis Solar Tracking System exemplifies the ability of microcontroller-based solutions to increase solar panel efficiency. However, its limits in terms of scalability, real-time monitoring, and advanced tracking mechanisms offer the opportunity for major improvement, which my suggested system seeks to solve, particularly by incorporating IoT and adaptive energy solutions.

2.2.3 Solar Tracking System with Renewable Energy Optimization

Renewable Energy Optimization through Solar Tracking Systems investigates using a single-axis solar tracker to increase the efficiency of solar photovoltaic (PV) panels. This system combats the declining efficiency of stationary solar panels by dynamically aligning them with the sun's position throughout the day. The proposed system significantly improves energy output, increasing power generation by up to 32% compared to fixed systems. [Refer to Figure 2.9 for the implementation setup].

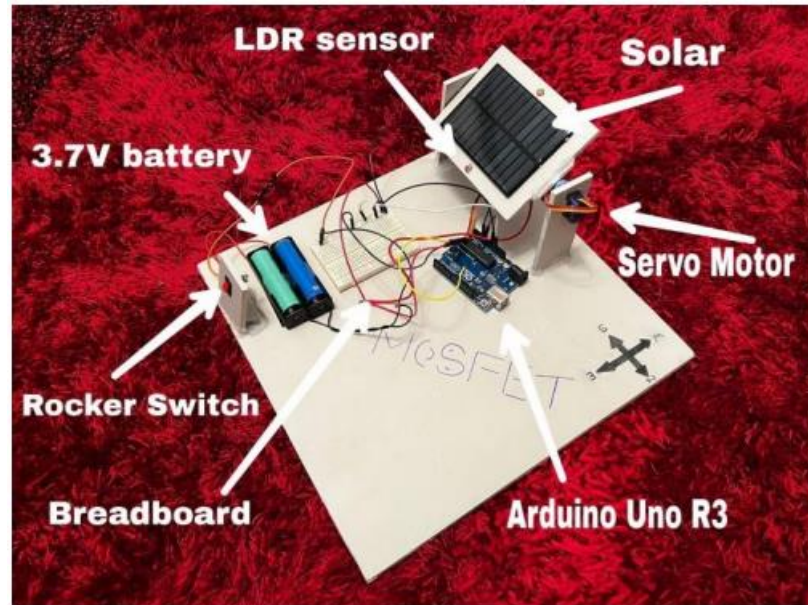


Figure 2.9: Solar Panel Implementing Single Axis Solar Tracking Device (Talebi & Aly, 2024)

The Single-Axis Solar Tracking Mechanism is an important aspect of this system, as it uses light sensors and a motorized control to modify the orientation of the PV panels dependent on the sun's trajectory. This assures maximum solar irradiance absorption, which boosts energy capture and power output. The system also contains Monitoring and Control Features, in which a microcontroller uses sensor input to guide the motorized movement of the panels, maximizing their alignment in real-time. [Figure 2.10 shows the system topology]. Furthermore, the system is intended to function efficiently in a variety of weather circumstances, with realistic simulations indicating performance gains even on cloudy days. [Refer to Figure 2.11 for simulation results].

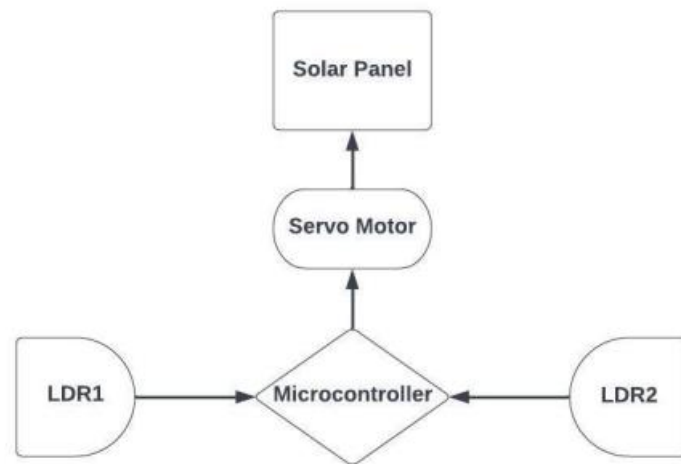


Figure 2.10: Diagram of a Single Axis Solar Tracker (Talebi & Aly, 2024)

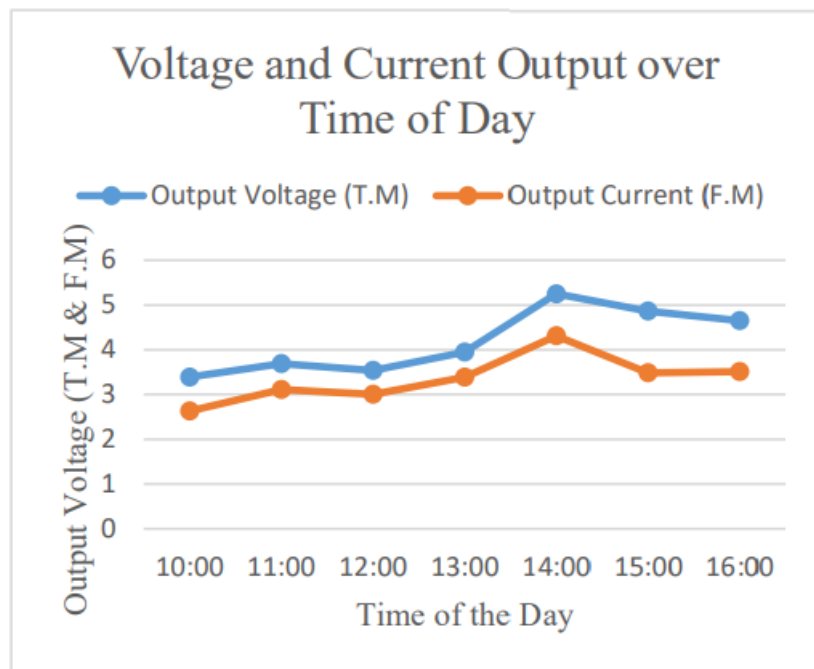


Figure 2.11: Power Output Results are Displayed Graphically against time for both fixed mode and autonomous system tracking. (Talebi & Aly, 2024)

This system's key strength is its capacity to increase solar energy capture while optimizing performance. Tests and calculations show that the system increases power output by 28%-39% over fixed systems, particularly during non-peak hours when the sun's position

varies frequently. Furthermore, the system is inexpensive, utilizing widely available components such as Arduino microcontrollers, LDR sensors, and servo motors, making it an economical alternative for home and industrial solar applications.

Despite these advantages, the system has significant drawbacks. When compared to dual-axis systems, single-axis designs do not fully capture the sun's energy potential, particularly in areas where the sun's azimuth and elevation angles vary greatly. Furthermore, the system lacks battery monitoring and energy prioritization skills, which are essential for assuring energy availability during low-light periods. The lack of complex algorithms for allocating energy for specialized applications, like as agricultural irrigation, reduces its adaptability.

This device demonstrates the utility of sun-tracking technology in improving renewable energy capture. However, it fails to address crucial issues such as energy storage and adaptive energy allocation, both of which are critical for broader applications. My proposed system adds to these features by incorporating:

- **Battery Energy Monitoring:** Gain real-time insight into storage levels to ensure operational reliability.
- **Adaptive Energy Allocation:** Energy is prioritized for vital needs like agricultural irrigation.
- **Dual-Axis Tracking:** This technique maximizes solar energy capture by aligning panels in both horizontal and vertical planes.

The Renewable Energy Optimization using Solar Tracking Systems study reveals that single-axis tracking can improve PV panel performance. However, its limitations in terms of energy storage, adaptive prioritization, and monitoring complexity leave substantial gaps, which my suggested system seeks to fill by including additional functionalities for application-specific requirements.

2.3 Comparison between Existing System

Below is a comparison of the existing system:

Table 2.1: Comparison Table of the Existing System

Existing System Feature	IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	Automatic Single-Axis Solar Tracking System	Solar Tracking System with Renewable Energy Optimization
Tracking Mechanism	Single axis	Single axis	Single axis
Battery Monitoring	IoT-Based	None	IoT-Based
Cost of Implementation	Moderate	Low	High
Complexity of Design	High	Low	High
Panel Size Support	Small to medium panels	Small to medium panels	Small to large panels

Existing System Feature	IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	Automatic Single-Axis Solar Tracking System	Solar Tracking System with Renewable Energy Optimization
Energy Gain (%)	~28% - 35%	~25% - 30%	~30% - 39%
Tracking Accuracy	Moderate	Moderate	High
Power Consumption	Moderate	Low	Low
Ease of Maintenance	Moderate	High	Moderate
Real-Time Data Insights	Yes	No	Yes
Historical Data Storage	Yes	No	Yes
User Interface	Yes (IoT Dashboard)	None	Yes (IoT Dashboard)
Remote Control	Yes	No	Yes
Integration with IoT	Yes	No	Yes
Agriculture-Specific	No	No	No

Table 2.1 above summarizes the features of three existing solar panel tracking and optimization systems: IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System, Automatic Single-Axis Solar Tracking System, and Solar Tracking System with Renewable Energy Optimisation. Each system has distinct advantages and disadvantages, making it appropriate for a variety of applications.

The IoT-Based Power Monitoring of Solar Panels Incorporating Tracking System features a single-axis tracking mechanism as well as IoT-based energy monitoring. While it provides real-time data insights and remote control through an IoT dashboard, it lacks improved tracking precision and application-specific optimisations. Its low cost and adaptability make it ideal for small to medium-sized installations, especially in residential or general renewable energy projects. However, its energy gain (~28-35%) and hazy performance are only minor when compared to more modern devices. (Shah et al., 2021).

The Automatic Single-Axis Solar Tracking System stands out due to its simplicity and inexpensive implementation costs. It is simple to maintain, having components tailored too small to medium-sized panels. However, this system is missing IoT integration, real-time monitoring, and advanced energy optimization. The energy increase (~25-30%) and tracking accuracy are reasonable, making it a suitable alternative for budget-conscious consumers seeking basic solar tracking without further technological complexity. (Reza et al., 2021).

The Solar Tracking System with Renewable Energy Optimization is the most sophisticated of the three systems. It supports real-time monitoring, IoT integration, and a high level of expandability, making it ideal for large-scale applications. With energy gains of ~30-39% and good tracking precision, it outperforms competing systems. However, its high cost and design complexity make it better suited to industrial or large-scale energy

projects than small-scale setups. Despite its advantages, it does not explicitly address agricultural needs, leaving a vacuum in meeting sector-specific energy requirements.

2.4 The Shortcoming in Existing Systems

Existing solar panel tracking and monitoring systems, while offering various benefits as reviewed in the preceding sections, present several key shortcomings that limit their effectiveness, particularly for specialized applications like polybag farming. These limitations highlight critical areas for improvement and serve as the foundational needs that our proposed system aims to address.

The table below summarizes the major shortcomings identified in the reviewed existing systems. These gaps represent opportunities for innovation and directly inform the design requirements for a more comprehensive and tailored solution.

Table 2.2: Shortcomings in Existing Solar Panel Tracking and Monitoring Systems

System Feature /Aspect	IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	Automatic Single-Axis Solar Tracking System	Solar Tracking System with Renewable Energy Optimization	Key Shortcoming Identified
Tracking Mechanism	Single axis	Single axis	Single axis	Limited Sunlight Capture

System Feature /Aspect	IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	Automatic Single-Axis Solar Tracking System	Solar Tracking System with Renewable Energy Optimization	Key Shortcoming Identified
Battery Monitoring	IoT-Based (but lacks real-time tracking of storage levels)	None	IoT-Based (but lacks real-time tracking of storage levels)	Inadequate Battery Monitoring
Light Intensity Monitoring	None explicitly for adaptive decision- making	None	None explicitly for adaptive decision- making	Lack of Environmental Awareness
Application Specificity	Designed for general solar energy generation	Primarily for small-scale solar panels, general use.	General industrial/large- scale energy projects.	Not Agriculture- Specific
Remote Control & UI	Yes (IoT Dashboard)	No	Yes (IoT Dashboard)	Limited Interactive Control
Scalability	Small to Medium Panels	Small to medium panels	Small to large panels	Scalability Constraints

System Feature /Aspect	IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	Automatic Single-Axis Solar Tracking System	Solar Tracking System with Renewable Energy Optimization	Key Shortcoming Identified
Cost of Implementation	Moderate	Low	High	Cost Barriers
Complexity of Design	High	Low	High	Design Complexity
Panel Size Support	Small to medium panels	Small to medium panels	Small to large panels	Limited Scalability for Smaller Farms
Energy Gain (%)	~28% - 35%	~25% - 30%	~30% - 39%	Suboptimal Energy Efficiency
Tracking Accuracy	Moderate	Moderate	High	Accuracy Limitations
Power Consumption	Moderate	Low	Low	Varying Power Efficiency
Ease of Maintenance	Moderate	High	Moderate	Maintenance Challenges

System Feature /Aspect	IoT-Based Power Monitoring of Solar Panel Incorporating Tracking System	Automatic Single-Axis Solar Tracking System	Solar Tracking System with Renewable Energy Optimization	Key Shortcoming Identified
Real-Time Data Insights	Yes	No	Yes	Inconsistent Real-Time Monitoring
Historical Data Storage	Yes	No	Yes	Missing Historical Data
User Interface	Yes (IoT Dashboard)	None	Yes (IoT Dashboard)	Lack of User- Friendly Interfaces
Remote Control	Yes	No	Yes	Limited Remote Control
Integration with IoT	Yes	No	Yes	Incomplete IoT Integration
Agriculture- Specific	No	No	No	General Purpose Design

As evident from Table 2.2, existing systems primarily focus on basic solar tracking and general power monitoring. They often fall short in providing the granular, real-time data and adaptive control necessary for optimizing energy use in dynamic agricultural environments. Specifically, the widespread use of single axis tracking limits energy capture, the absence of detailed battery health monitoring introduces operational risks, and the lack of explicit light intensity sensing for decision-making hinders precision agriculture. Furthermore, the general-purpose design of these systems means they do not fully cater to the specific needs of polybag farming. These identified shortcomings directly inform the functional and non-functional requirements of our proposed system, ensuring it addresses these critical gaps and offers a more robust, intelligent, and agriculture-specific solution.

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

2.5.1 Hardware

2.5.1.1 Solar Panel (2W)

The solar panel is the primary source of renewable energy, generating electricity to power the system and store energy in the battery.

2.5.1.2 Light-Dependent Resistors (LDRs)

LDRs are used to detect the intensity of sunlight from different directions, enabling the system to track the sun's position for optimal energy generation.

2.5.1.3 Servo Motors

Servo motors are responsible for moving the solar panel along its axes (azimuth and elevation) based on input from the LDRs, ensuring precise tracking of the sun.

2.5.1.4 Microcontroller (ESP32)

The microcontroller acts as the brain of the system, processing sensor data, controlling the servo motors, and communicating with the IoT platform for monitoring.

2.5.1.5 Battery (3.7V Li-on)

The battery stores the energy generated by the solar panel, providing power for the irrigation system and ensuring continuous operation even during low sunlight periods.

2.5.1.6 Voltage Sensor Module

These sensors measure the battery's voltage, allowing real-time battery monitoring for the system's operations.

2.5.1.7 Structural Materials

The structural materials provide physical support for the solar panel, motors, and other components, ensuring durability and stability of the tracking system.

2.5.1.8 Miscellaneous Electronics

Additional components like resistors, capacitors, diodes, and wiring are used for connecting and ensuring proper functionality of the system's electrical circuits.

2.5.2 Software

2.5.2.1 PlatformIO (Visual Studio Code Extension)

PlatformIO is an integrated development environment (IDE) extension used to write, debug, and upload the microcontroller's software for controlling the solar panel tracking system.

2.5.2.2 Firebase (Realtime Database)

Firebase Realtime Database can handle connection between microcontroller and flutter app. Therefore, it is used for real-time battery monitoring as well as light intensity monitoring while also enabling remote access and control of the system.

2.5.2.3 Flutter (Visual Studio Code Extension)

Flutter is used to develop a user-friendly mobile application that provides an interface for users to monitor and control the solar tracking and irrigation system.

2.5.2.4 Firebase (Cloud Firestore)

Firebase Cloud Firestore is used to store historical sensor data and system logs, providing insights crucial for data-driven optimization and supporting the core objective of improving energy usage awareness and efficiency in polybag agriculture environments.

2.5.2.5 Figma (Interface Design)

Figma is used to design the mobile application's interface, ensuring an intuitive and visually appealing user experience for system monitoring and control.

2.6 Summary

Solar energy is a key component of sustainable energy solutions, and advancements in solar tracking systems, real-time battery monitoring, and environmental sensing are essential to fully harness its potential. Existing systems—such as the IoT-Based Power Monitoring System, Automatic Single-Axis Solar Tracking System, and Solar Tracking System with Renewable Energy Optimization—offer benefits like real-time data collection, improved energy harvesting, and scalability. However, they often fall short of addressing agriculture-specific

needs such as dual-axis solar tracking, light-intensity-based environmental awareness, and dedicated battery monitoring—features that are crucial in applications like polybag farming.

The proposed solution addresses these limitations by integrating dual axis tracking to maximize solar absorption throughout the day, real-time battery monitoring to provide accurate insights into energy storage levels, and light intensity sensing to adapt operations based on ambient sunlight conditions. Designed with scalability and practical agricultural use in mind, the system ensures efficient energy use and operational reliability. By combining intelligent features with a focus on the specific requirements of small-scale agriculture, the proposed system offers a robust and responsive approach to improving renewable energy utilization in farming environments.

CHAPTER 3

REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

This chapter outlines the requirements and design for the proposed Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming. It ensures the system is aligned with user needs and project objectives, which include enabling real-time environmental monitoring, automated solar tracking, and efficient irrigation control. **Crucially, the requirements analysis presented herein is directly driven by the identified shortcomings in existing solar energy systems, as detailed in Chapter 2. The primary goal of this proposed system is to address these critical gaps by incorporating missing features and functionalities that enhance efficiency, reliability, and applicability for small-scale agricultural contexts.**

The system aims to address challenges in conventional solar setups by dynamically adjusting solar panels through dual-axis tracking, monitoring battery levels in real time, and responding to changes in ambient light intensity. This chapter describes the functional and non-functional requirements, while also focusing on the software and hardware specifications necessary to achieve these objectives and presents a detailed design to realise the proposed solution. Requirement analysis helps identify the system's core functionalities by using surveys to understand user perception and expectations of the system. The design phase then translates these requirements into a concrete, implementable solution using specific tools and methods, such as Lucid Chart for diagramming and Figma for mobile application prototyping. This

ensures the system effectively addresses the identified challenges while providing a user-friendly experience.

This chapter begins with the methods used to gather requirements and a detailed list of hardware and software needs, as well as functional and non-functional requirements. The system design section provides a comprehensive overview of the solution through various diagrams, including a system architecture diagram to present the overall system structure, a flowchart to describe the logical flow of system operations, and UML diagrams—including use case, class, activity, and sequence diagrams—to visualize the software design and behavior. A block diagram illustrates the high-level hardware structure, and an interface diagram showcases the prototype of the proposed mobile application.

Finally, the chapter concludes with a summary that ties together its findings and demonstrates their relevance to the project's objectives. By addressing the limitations of fixed solar systems and lack of real-time environmental feedback, while ensuring intuitive monitoring capabilities, this chapter lays the foundation for the successful implementation of the proposed system.

3.2 Requirement Analysis

3.2.1 Method For Gathering Requirements

3.2.1.1 Survey Using Google Form

In line with the objectives of this Final Year Project (FYP), *"Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming,"* this survey is designed to gather critical insights into the system's real-world relevance, usability, and perceived effectiveness. The core purpose of the survey is to validate how well

the proposed solution aligns with actual agricultural needs, especially for small-scale or polybag farming practices where energy reliability and resource responsiveness are crucial.

The target respondents—or "users"—of this survey include small-scale farmers, agricultural practitioners, and individuals involved in polybag farming, particularly those familiar with renewable energy systems or interested in sustainable agricultural technologies. Their feedback is essential because they represent the intended end-users of the system. Understanding their experiences, challenges, and perspectives ensures the project is not just technically sound, but also practically valuable, adaptable, and user-friendly in real-world farming environments.

This survey aims to:

1. **Evaluate System Effectiveness:** Assess how the dual-axis solar tracking improves sunlight capture and how real-time battery, and light intensity monitoring enhance decision-making in daily operations like irrigation.
2. **Gather User Feedback:** Understand user perspectives on the ease of use, reliability, and practicality of the system's IoT-based features—particularly battery status visibility and adaptive sunlight tracking.
3. **Understand Demographic Influence:** Analyze how the background, experience level, and energy dependency of different users influence their acceptance and perception of the proposed system.

Demographic information plays a key role in contextualizing the survey findings. For example:

- **Energy Dependence:** Helps identify whether users who lack consistent energy access see greater value in a solar-powered, automated solution.
- **User Experience with Technology:** Reveals whether familiarity with mobile apps or IoT tools influences ease of adoption and willingness to engage with the system.

By combining technical validation with user-centered insights, the survey ensures that the system is not only effective in its engineering but also meaningful, accessible, and scalable for its intended users. This human-centered feedback loop is essential for refining the system into a reliable and widely usable tool for sustainable agriculture.

Below is the analysis and the outcome of the survey:

Are you familiar with solar energy systems?

24 responses

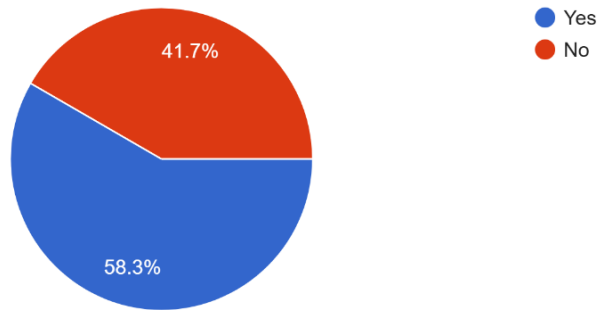


Figure 3.1: Familiarity with solar energy systems

Have you used renewable energy systems like solar panels before?

24 responses

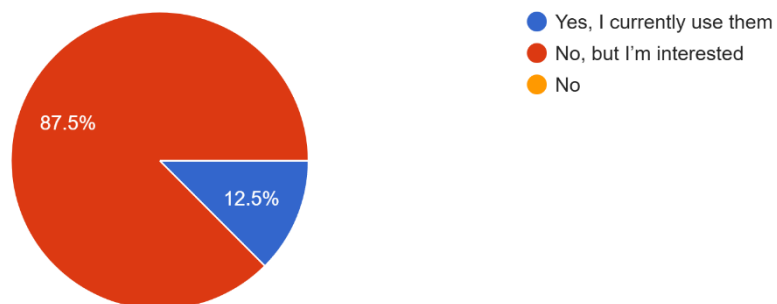


Figure 3.2: Respondent's accustomed with solar panel energy system

Figure 3.1 is a pie chart that depicts the responses from participants regarding how familiar they are with solar energy systems. Majority of the respondents (58.3%) indicated they are familiar with the system, suggesting almost entirely of the user know about solar energy. Meanwhile, 41.7% of the respondents that they do not know of solar energy system indicating that some may not properly being educate with solar energy

system. This could suggest that this system has the potential to educate respondents about the energy system.

Other than that, Figure 3.2 describes respondents accustomed with solar panel energy system. Vast majority of the respondents (87.5%) never use the system but interested with the system. This depicts that majority of the people in my respondents' area of population never have any experience with solar panel energy system. However, they are fascinated by the idea of the system. This suggest that the respondents may adopt the system if its ever release.

Overall, the data from Figures 3.1 and 3.2 highlight both the potential and challenges for the adoption of the proposed **solar-powered tracking system**. The significant familiarity with solar energy systems among 58.3% of respondents indicates a promising foundation for system introduction, while the 41.7% who are unfamiliar underscore the importance of integrating educational components into the system's implementation. Moreover, the overwhelming interest (87.5%) despite the lack of prior usage demonstrates a strong willingness among respondents to embrace the technology. This suggests that the system could address both practical agricultural needs and foster greater awareness of renewable energy benefits, paving the way for broader adoption in the future.

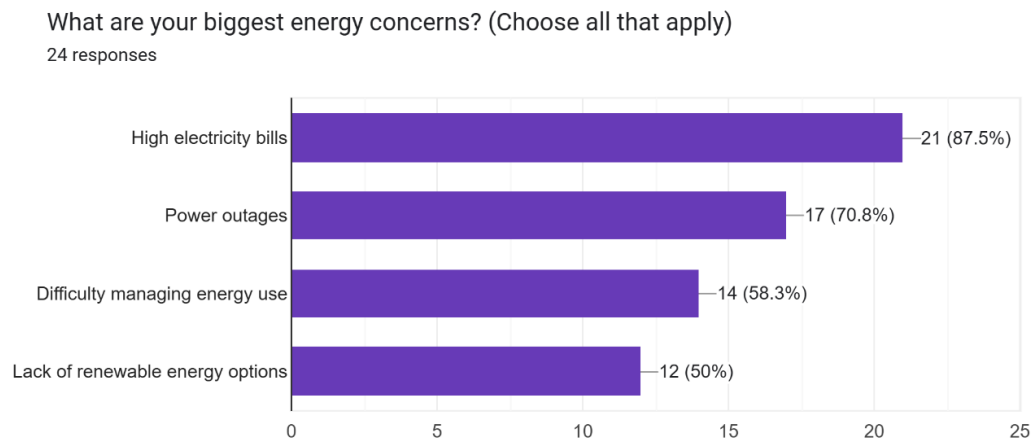


Figure 3.3: Respondent's biggest energy concerns

Figure 3.3 is a bar chart illustrating the key energy concerns of respondents. The most prominent concern, selected by 87.5% of participants, is high electricity bills, emphasizing affordability as a critical issue. This suggests a strong demand for cost-effective energy solutions, such as **solar-powered systems**, that can reduce long-term energy costs. Power outages were the second most significant concern, chosen by 70.8% of respondents, highlighting the need for reliable energy sources that ensure uninterrupted power supply, especially in rural or underserved areas.

Additionally, 58.3% of respondents identified difficulty managing energy use as a concern, indicating the need for systems that simplify energy control. User-friendly energy management tools or automation features, such as the proposed system's real-time monitoring and optimization capabilities, could address this challenge. Finally, 50% of respondents pointed to the lack of renewable energy options, reflecting a moderate interest in sustainable alternatives like solar energy.

Overall, the findings reveal that affordability and reliability are the primary concerns, but there is also significant interest in better energy management and renewable solutions. Addressing these areas through the proposed system would ensure it meets user needs effectively while encouraging the adoption of sustainable energy practices.

What challenges do you face with current renewable energy solutions?

24 responses

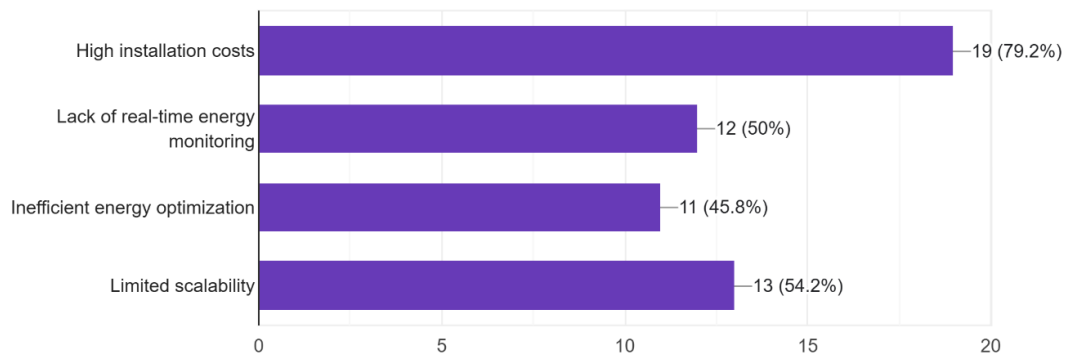


Figure 3.4: Challenges faced by respondents with current renewable energy solutions

The chart in Figure 3.4 highlights the challenges faced by respondents with current renewable energy solutions. The most significant challenge, identified by 79.2% of respondents, is the high installation cost, emphasizing the financial barrier as a major obstacle to adoption. This suggests the need for more affordable and accessible renewable energy systems to encourage broader implementation. The second major concern is limited scalability, reported by 54.2% of respondents, indicating that existing systems may not be adaptable to meet growing or changing energy demands. This highlights the importance of designing solutions that are flexible and scalable to accommodate future needs.

Additionally, 50% of respondents cited the lack of real-time energy monitoring as a challenge, pointing to a desire for systems that provide instant feedback and actionable insights. Integrating real-time monitoring capabilities could enhance user experience and improve energy management. Inefficient energy optimization was identified by 45.8% of respondents, suggesting dissatisfaction with the ability of current systems to maximize energy efficiency. Advanced optimization features could address this issue, enabling better utilization of renewable energy.

Overall, the findings reveal that affordability, scalability, and technological limitations are the primary barriers to the adoption of renewable energy solutions. Addressing these challenges through the proposed system by focusing on cost reduction, scalability, and advanced **monitoring and optimization** tools would ensure it meets user expectations and encourages widespread use.

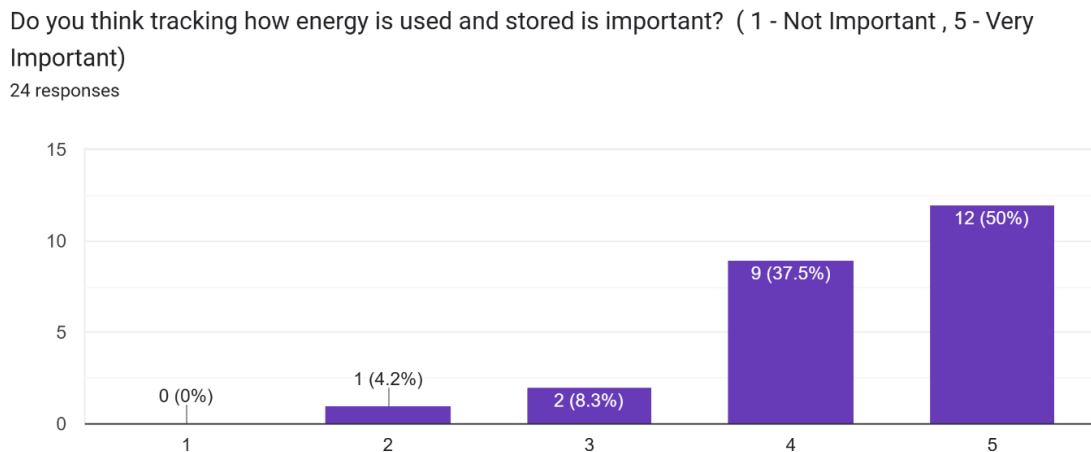


Figure 3.5: Respondent's perception toward the importance of tracking how energy is used and stored.

The chart in Figure 3.5 showcases how respondents perceive the importance of tracking energy usage and storage. A significant majority, 50%, rated this aspect as "very important" (rating 5), while 37.5% rated it as "important" (rating 4). This indicates that 87.5% of respondents believe energy tracking is a crucial factor, demonstrating strong agreement on its importance.

Only a small number of respondents, 8.3%, were neutral about the significance of energy tracking (rating 3), and just 4.2% considered it less important (rating 2). Interestingly, none of the participants rated energy tracking as "not important" (rating 1), which highlights a universal acknowledgment of its relevance.

Based on these findings, it is evident that tracking energy usage and storage is a top priority for users. This aligns perfectly with the objectives of the proposed system, where real-time energy monitoring and optimization are key features. The results reinforce the need to design a system that provides accurate and efficient energy tracking to meet user expectations and enhance the adoption of renewable energy technologies.

How important is it to know your battery's charge level in real-time? (1 - Not Important , 5 - Very Important)

24 responses

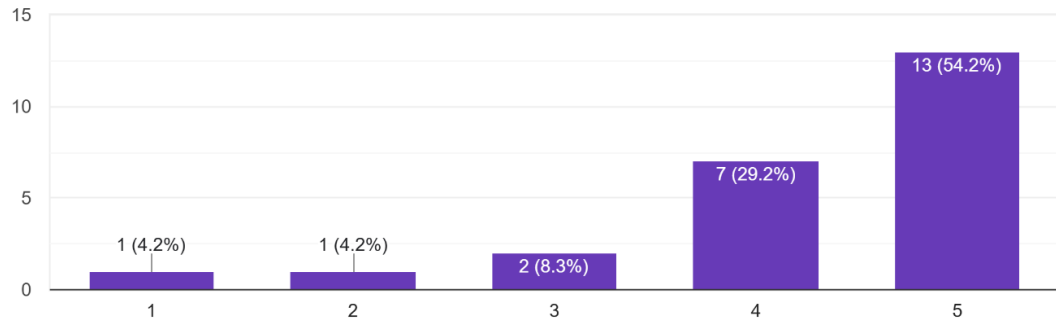


Figure 3.6: Respondent's perception on the importance of knowing battery's charge level in real-time.

The chart in Figure 3.6 highlights respondents' perceptions of the importance of tracking energy usage and storage. A significant majority, 54.2%, rated this feature as "very important" (rating 5), and 29.2% rated it as "important" (rating 4). This means that 87.5% of respondents view real-time monitoring of the battery's charge level as crucial.

A smaller proportion of respondents expressed less urgency for this feature, with 8.3% being neutral, 4.2% considering it "less important," and another 4.2% deeming it "not important." Despite this minority, the data strongly supports the inclusion of real-time energy tracking as an essential component of the proposed system.

These findings underscore the importance of incorporating this feature into the system design to align with user expectations and project objectives. Real-time energy monitoring should be prioritized as one of the key functionalities to ensure the system meets its goals and delivers value to its users.

What would you like to monitor in the system's battery? (Choose all that apply)

24 responses

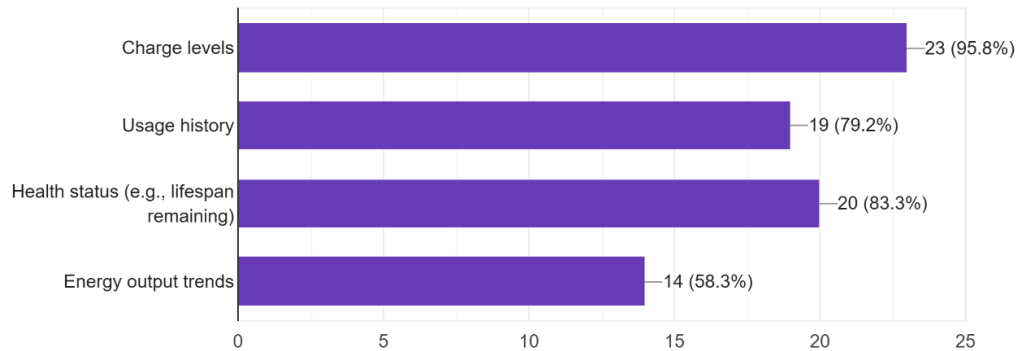


Figure 3.7: Respondent's preference in monitoring system's battery.

The Figure 3.7 reveals respondents' preferences for monitoring features on the system's battery. A majority, 95.8%, prioritized tracking **charge levels**, highlighting its importance as a core feature. Similarly, 83.3% indicated interest in monitoring the **health status** (e.g., remaining lifespan), while 79.2% wanted to track the **usage history** to better understand consumption patterns. Additionally, 58.3% expressed a preference for monitoring **energy output trends**, reflecting an interest in analysing performance over time.

These results underline the need for a comprehensive battery monitoring system. Charge levels and health status should be prioritized as critical functionalities, while usage history and energy trend analysis can provide added value. Incorporating these features ensures the system aligns with user expectations and delivers a robust solution.

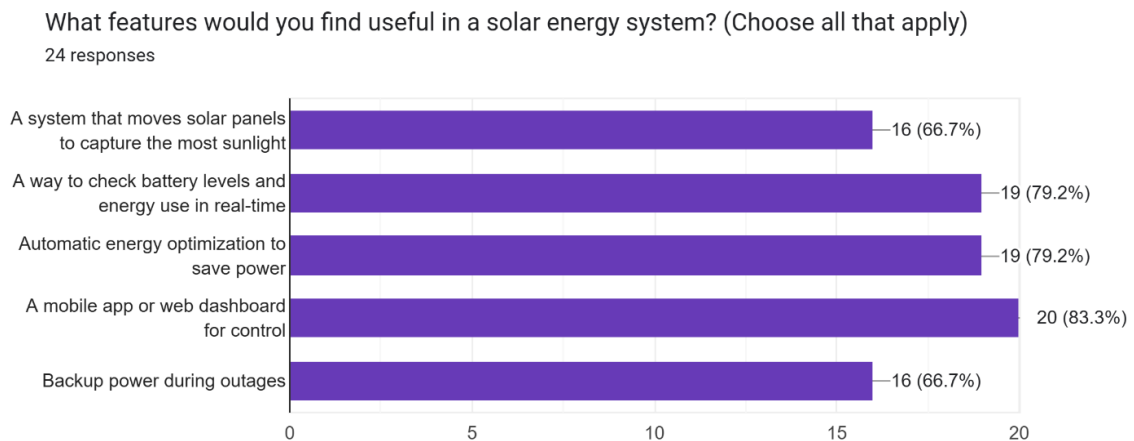


Figure 3.8: Features respondents find useful in solar panel system

The data in Figure 3.8 highlights the top features preferred for a solar panel management system. A majority, **83.3%**, prioritized a **mobile app or web dashboard for control**, emphasizing the need for convenient remote access. This aligns with modern expectations for flexibility and ease of use, enabling users to manage the system seamlessly from anywhere.

Additionally, **79.2%** highlighted the importance of **real-time battery level and energy monitoring**, emphasizing the need for transparency and informed energy management. This feature helps users track usage and battery status, ensuring efficiency and preventing downtimes.

Equally, **79.2%** preferred **automatic energy optimization to save power**, reflecting a strong demand for sustainability and cost-effectiveness. This feature minimizes waste and maximizes efficiency without requiring manual input.

These preferences underscore the need to prioritize these features in the system design. By focusing on remote control, monitoring, and optimization using mobile application, the system will meet user expectations, enhance usability, and support sustainable energy management.

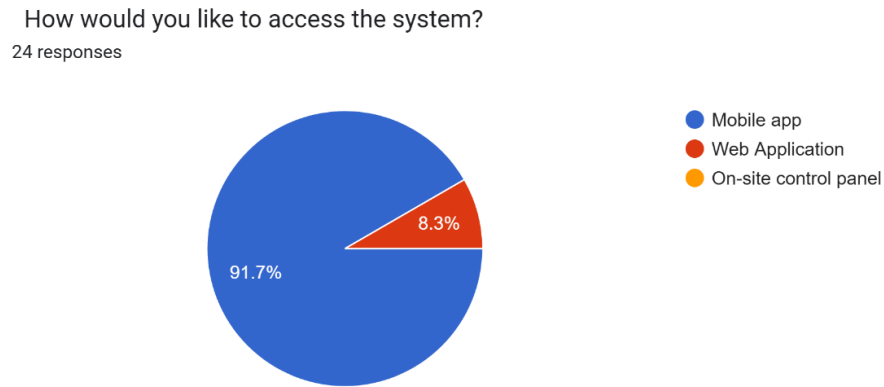


Figure 3.9: How respondents want to access the system.

The pie chart in Figure 3.9 illustrates the preferred methods for accessing the system among respondents. A vast majority, 91.7% of respondents, expressed a preference for accessing the system through a mobile app. This highlights the importance of mobile accessibility, emphasizing user demand for convenient, portable, and user-friendly interfaces.

In contrast, only 8.3% of respondents preferred a web application, and none chose an on-site control panel. The lack of interest in on-site access suggests that users favour remote access over physically interacting with the system.

These findings underscore the critical need to prioritize the development of a robust and intuitive mobile application for my solar panel tracking system. By focusing

on mobile accessibility, the system will meet user expectations and enhance usability. Additionally, while the web application is a secondary preference, it can still serve as a supplementary access method to support a broader range of users.

Would you like notifications or reports on energy usage, battery levels, and system performance?
24 responses

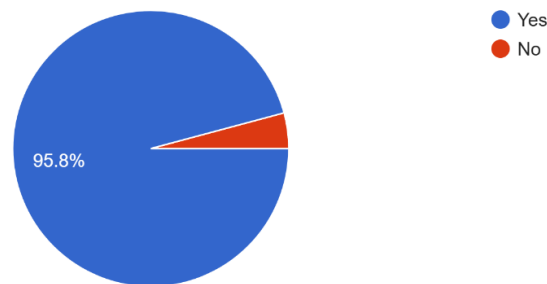


Figure 3.10: Respondent's interest in notification/reports feature

The pie chart in Figure 3.10 illustrates the preference of the respondents in receiving notifications or report on energy usage, battery levels and system performance. An immense majority, 95.8% of respondents, expressed a preference in receiving notifications regarding energy usage, battery levels as well as system performance. This highlights the importance of receiving notifications to the user regarding any activity that happened on the system.

Only 4.2% of respondents preferred not to receive notifications about the data. This low percentage indicates a strong preference among users for receiving notifications rather than none.

These findings undermine the importance of notifications on system that involve remote access feature as one of its mains qualities. By implementing notifications feature,

the user will always be informed of any information happening on site that ensure reliability of the system

What is your budget for a solar energy system?
24 responses

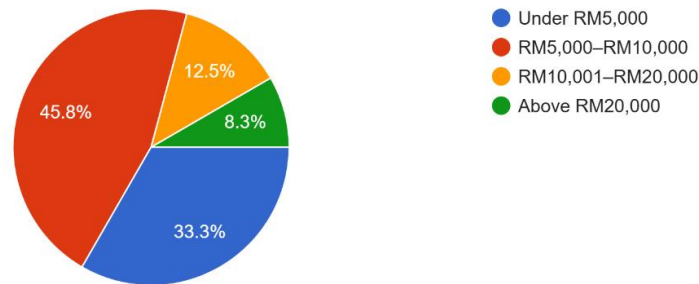


Figure 3.11: Respondents budget for a solar energy system.

The pie chart in Figure 3.11 highlights the budget respondents are willing to allocate for the proposed system. It shows that 45.8% are prepared to spend between RM5,000 and RM10,000, making this the most common range. Additionally, 33.3% have a budget below RM5,000, while 12.5% are willing to spend RM10,001 to RM20,000, and only 8.3% are prepared to invest above RM20,000.

These figures reveal that most respondents (over 79%) prefer to keep their spending below RM10,000. This indicates the need to design a cost-effective system by minimizing unnecessary expenses while maintaining quality. Although affordability is a key concern, the system must not compromise its reliability, effectiveness, or integrity. Balancing cost efficiency with high performance will be essential to meet user expectations and ensure the system's success.

Summary

The survey results provide valuable insights that directly inform the development of the Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming. These findings highlight both user expectations and practical system requirements to ensure its effectiveness in real-world agricultural settings.

Key findings include:

- **Mobile App Accessibility:** A significant majority (91.7%) of respondents prefer accessing the system via a mobile application. This underscores the importance of a user-friendly, portable interface that allows farmers to monitor system data, receive alerts, and respond to conditions in real time—such as manually adjusting solar panel angle when anomalies are detected.
- **Real-Time Notifications:** 95.8% of participants prioritized timely notifications about battery levels, light intensity readings, and system performance. This suggests users require active system monitoring and immediate alerts to take corrective actions, like conserving energy during low battery periods or adjusting irrigation schedules during low sunlight hours.
- **Real-Time Battery Monitoring:** 79.2% identified battery level tracking as a critical feature, aligning with the system's goal of ensuring that energy-dependent tasks, such as irrigation, are only initiated when sufficient power is available—thereby avoiding power outages or system downtime.

- **Light-Based Automation:** Though not previously quantified in the results, the system's real-time light intensity monitoring enables users to understand environmental conditions and act accordingly—such as delaying irrigation during heavily overcast periods or redirecting manual operations when sunlight is insufficient for efficient solar charging.
- **Cost Concerns:** 87.5% of respondents expressed concern over high electricity bills, and 79.2% cited high installation costs as a major barrier. This reinforces the need for a low-cost, solar-based, autonomous system that reduces long-term operational expenses.
- **Budget Preferences:** Over 79% are willing to invest under RM10,000, indicating that the system must be economically viable and justified by its benefits in reducing energy costs and improving farming efficiency.

In conclusion, the survey findings validate the need for a mobile-accessible, real-time responsive system tailored to small-scale farmers. More than just passive monitoring, the mobile app empowers users to act—such as switching between automatic and manual modes, scheduling irrigation based on light and battery data, and diagnosing system performance remotely. These actionable features not only ease immediate operational challenges but also promote long-term sustainability and user satisfaction in agriculture-focused energy systems.

3.2.2 List Of Required Requirements for The Proposed System

3.2.2.1 Functional Requirements

The purpose of functional requirements is to define the specific functions, capabilities and behaviours that a system must exhibit to meet its intended objectives. They serve as a detailed blueprint for how the system will operate, outlining what the system should do in response to various inputs, how it processes data, and the outputs it generates. Functional requirements are crucial because they ensure clarity and alignment among stakeholders, developers and users, reducing the risk of misunderstandings during development. For instance, as shown in Table 3.1, functional requirements might include monitoring battery charge and operating solar panel position based on sunlight. These requirements are essential for verifying that the system meets its goals, guides the development process, and provides a measurable basis for testing and validation. Without clearly defined functional requirements, the project risks inefficiency, incomplete features and failure to meet user needs.

Table 3.1: Functional requirements for the proposed system

Requirement ID	Functionality	Inputs	Processes	Output	Performance Criteria
FR-01	Adjust solar panel angles based on sunlight.	Sunlight intensity data from LDR sensors	Calculate optimal angles and control Servo motors.	Panels aligned for maximum sunlight capture.	Adjustment within 15 minutes of sunlight change.
FR-02	Monitor battery charge levels in real-time.	Battery voltage sensor data	Process sensor data and send to IoT platform.	Battery status displayed on dashboard.	Data updated every 5 seconds.
FR-03	Adjust irrigation based on light intensity	Light intensity data, irrigation schedule	Regulate irrigation timing based on real-time ambient light levels.	Efficient irrigation aligned with optimal sunlight conditions.	Ensures water is used effectively when sunlight is sufficient for plant absorption.

Requirement ID	Functionality	Inputs	Processes	Output	Performance Criteria
FR-04	Provide IoT-based remote monitoring and control.	User input via dashboard.	Retrieve data from IoT database and send user commands to system.	Dashboard shows real-time system stats.	System responds to remote commands in under 2 seconds.
FR-05	Alert users about system errors or low battery.	Sensor or system error logs.	Identify errors and generate alerts.	Notifications sent to the user.	Alerts delivered within 3 seconds of issue detection.
FR-06	Log historical energy usage data.	Energy usage metrics.	Store data in IoT database for later analysis.	Historical data available on the dashboard.	Logs updated continuously without data loss.

3.2.2.2 Non-Functional Requirements

Non-Functional Requirements (NFRs) define the quality attributes, operational constraints and performance expectations of a system, ensuring it operates efficiently, reliably and user-friendly under various conditions. For instance, as shown in Table 3.2, the requirement to adjust solar panels within 15 minutes of detecting sunlight changes ensure that the solar panels tracking system will only adjust if there are significant sun position changes to reduce energy usage. Similarly ensuring the interface is user-friendly for non-technical users is important because farmers, who may lack technical expertise, should be able to interact with the system intuitively. A well-designed interface increases user adoption and reduces the learning curve. By addressing these quality aspects, the NFRs ensure that the proposed system not only meets the functional goals but also delivers a robust, efficient and user-centric solution that supports sustainable and reliable farming practices.

Table 3.2: Non-Functional requirements of the proposed system

Requirement ID	Requirement Description	Priority	Justification
NFR-01	The system should adjust solar panels within 15 minutes of detecting sunlight changes.	High	Ensures maximum energy capture and reduce energy usage for optimized performance.
NFR-02	The interface must be user-friendly for non-technical users.	Medium	Farmers need an intuitive interface to

Requirement ID	Requirement Description	Priority	Justification
			monitor and control the system effectively.
NFR-03	The system must operate reliably in varying weather conditions.	High	Durability is critical to ensure uninterrupted operation in agricultural settings.
NFR-04	Real-time data updates must occur every 5 seconds.	High	Provides accurate and up-to-date insights for battery monitoring and decision-making.

3.2.2.3 Hardware requirements

Hardware requirements are critical as they define the physical components necessary to build and operate a system. Each component serves a specific purpose, ensuring that the system's functionality, efficiency, and reliability. Therefore, in a solar panel tracking system with battery and light intensity monitoring, the hardware components are carefully chosen to meet the system's objectives. For example, as shown in Table 3.1, the ESP32 microcontroller acts as the system's brain and processing data from sensors like LDRs for solar tracking. This interconnected hardware setup ensures seamless data collection, decision-making, and action, aligning with the project's goals of battery monitoring and efficient farming.

Table 3.3: Hardware requirements of the proposed system

Component Name	Specifications	Purpose/Function	Quantity	Estimated Cost	Justification
Solar Panel	5v 2W	Captures solar energy and converts it to electricity.	1	RM 15	Provides the power source for the entire system.
Light-Dependent Resistors (LDRs)	10k Ω	Detects sunlight intensity for solar panel tracking.	4 (1 unit = RM2.00)	RM 8	Essential for determining the optimal angle for panels.
Battery	3.7V, 4400 mAh	Stores energy to power the system during low sunlight.	1	RM 304.00	Ensures continuous operation of the system.
Servo Motors (MG90s)	4.8V, torque: 1.8 kg.cm	Adjusts the solar panel position for dual-axis tracking.	2 (1 unit = RM14.00)	RM 28.00	Provides precise movements for tracking the sun.
Microcontroller	ESP32 with Wi-Fi support	Acts as the brain of the system, processing inputs and controlling components.	1	RM 20.00	Coordinates all hardware operations, including tracking and monitoring.

Component Name	Specifications	Purpose/Function	Quantity	Estimated Cost	Justification
Voltage Sensor Module	Voltage sensors	Measures voltage value for battery monitoring	1	RM 6.00	Critical for real-time battery monitoring.
Structural Materials	Aluminium frame	Provides support and stability for the solar panels.	As needed	RM XX	Ensures durability and safety of the tracking setup.
Miscellaneous Electronics	Wires, connectors, resistors, etc.	Connects and integrates all components.	As needed	RM XX	Ensures proper electrical connections and stability.

3.2.2.4 Software Requirements

Software requirements is to define the tools, platforms, libraries, and systems necessary for designing, developing, and operating a project effectively. They provide a clear roadmap for selecting software resources that support the project's functional and operational goals. In the context of a solar panel tracking system with battery and light intensity monitoring, software requirements ensure that each component of the system works seamlessly. For example, as shown in Table 3.2, the Visual Studio Code with PlatformIO extension facilitates programming the ESP32 microcontroller, and Flutter act as a platform to provides a user-friendly experience for real-time monitoring and control while firebase established two-way communication for both of it. Software requirements are essential as they ensure compatibility between hardware and software, improve development efficiency, minimize errors through reliable tools and enable future scalability.

Table 3.4: Software requirements of the proposed system

Software Name	Purpose/Function	Version/Type	Platform	Justification
Visual Studio Code with PlatformIO Extension	Programming the microcontroller (ESP32)	Latest (v3.3.4)	Windows	Provides an integrated development environment (IDE) with a lightweight editor and Extensions for embedded development.
Firebase (Real Time Database)	Stores real-time sensor data (e.g., light intensity, battery voltage) and system logs; enables two-way communication between devices and applications.	Managed BaaS (PaaS)	Cloud	Facilitates real-time monitoring and control of solar tracking and battery system data through seamless, synchronized communication between the ESP32 and the mobile application.

Software Name	Purpose/Function	Version/Type	Platform	Justification
Visual Studio Code with Flutter SDK	Developing the mobile application for IoT control and monitoring.	Latest (v3.102.0)	Windows	Flutter supports cross-platform app development for Android/iOS, making it ideal for an IoT app UI.
Firebase Cloud Firestore	Stores historical sensor data, system logs, and user profile information.	Managed NoSQL Document Database	Cloud	Provides a highly scalable, real-time synchronized database for IoT data, enabling efficient storage, flexible querying, and instant updates for the mobile application to monitor trends and system performance.
User Interface Design Tool (Figma)	Designing the user interface for the mobile app and system dashboards.	Latest (v16.13.3)	Website	A collaborative tool for creating high-fidelity wireframes and design prototypes.

3.3 System Design

3.3.1 System Architecture

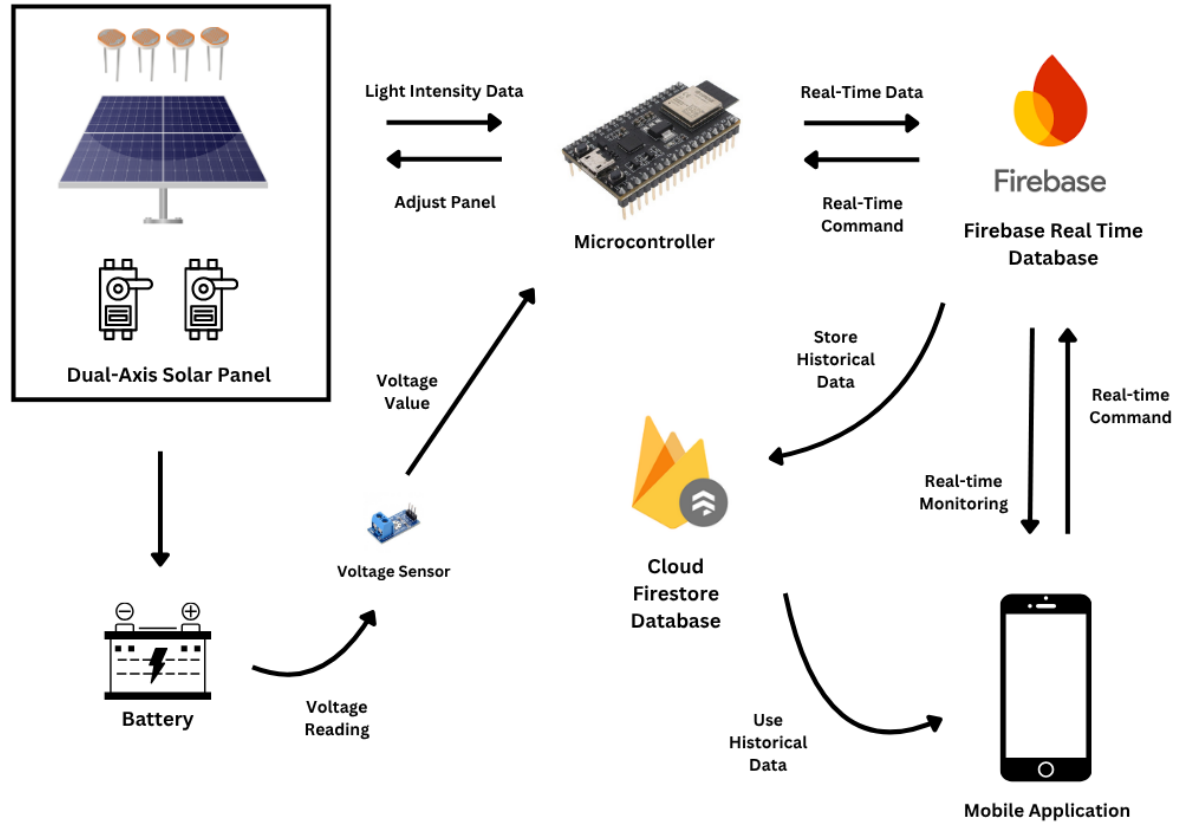


Figure 3.12: System Architecture for Proposed System

This system architecture illustrates a comprehensive Solar Panel Tracking System developed to enhance sunlight capture and support real-time environmental monitoring for automated irrigation in polybag farming. At the core of the system is a dual-axis solar panel equipped with four Light-Dependent Resistors (LDRs), which continuously detect light intensity from multiple directions. These LDR readings are processed by the ESP32 microcontroller, which calculates the optimal orientation of the panel to ensure maximum sunlight exposure throughout the day. Two servo motors are controlled accordingly to adjust the panel's azimuth (horizontal) and elevation (vertical) angles based on calculated light intensity differentials.

The harvested solar energy is stored in a battery, and a voltage sensor connected to the ESP32 continuously monitors the battery's charge status. This real-time battery data ensures safe energy management and helps avoid over-discharge or inefficient energy usage.

To enable smart interaction and remote control, the system is fully IoT-enabled using **Firestore Realtime Database** as the communication bridge between the ESP32 microcontroller and a Flutter-based mobile application. Sensor data—including light intensity, panel orientation, and battery voltage—is sent to Firestore by the ESP32. In turn, control commands and configuration changes made by the user in the mobile app (such as triggering manual tracking or checking sensor logs) are written to Firestore and read by the ESP32, creating a two-way communication loop.

All collected data is stored in **Firestore Cloud Firestore** for historical logging and future performance analysis. This architecture not only ensures intelligent solar tracking and accurate monitoring of environmental conditions but also provides an intuitive, remotely accessible control system. By leveraging Firestore's real-time capabilities, the solution enhances usability, reliability, and operational efficiency—meeting the practical needs of small-scale, sustainable agriculture.

3.3.2 Flowchart

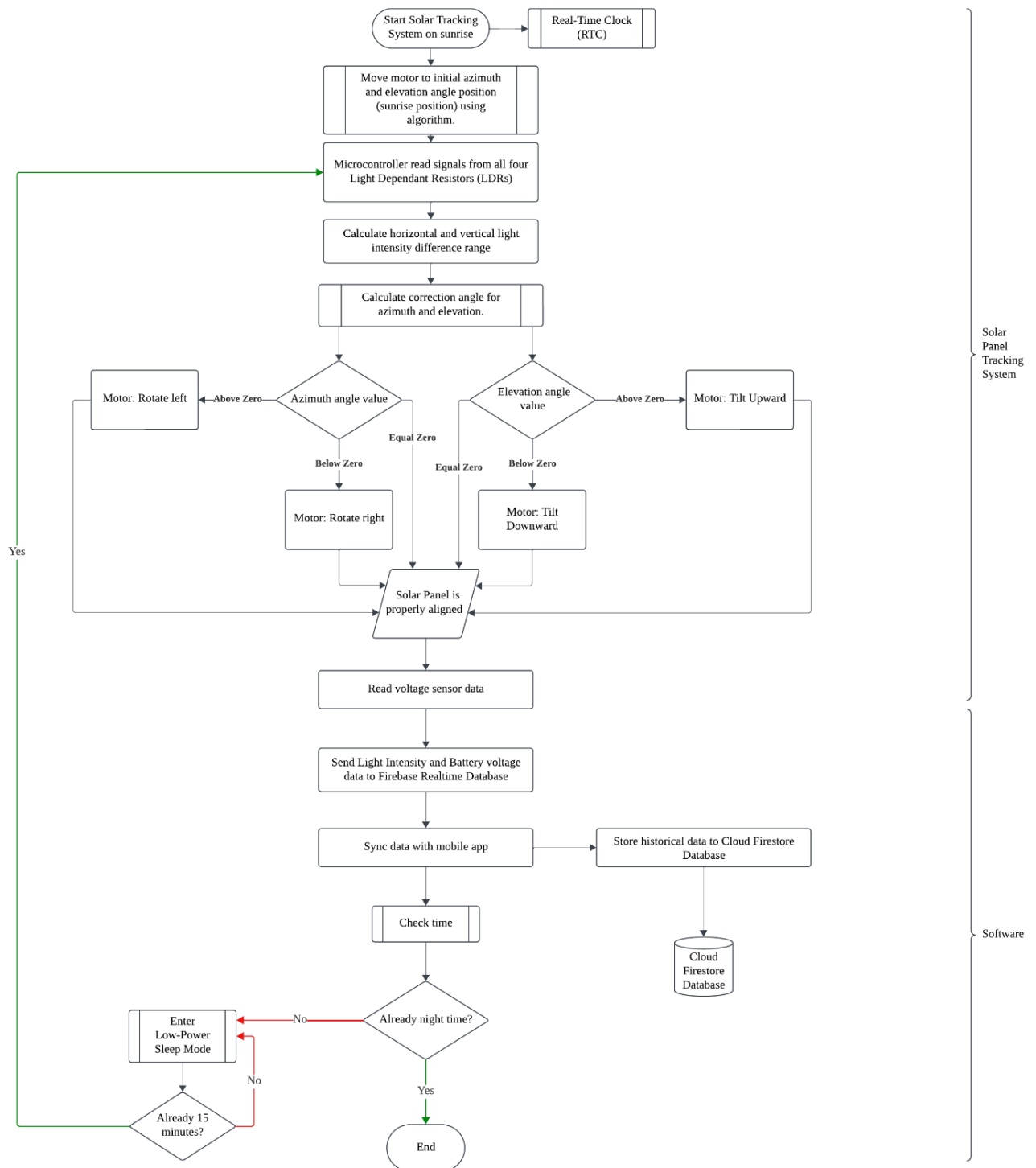


Figure 3.13: Flowchart of the proposed system

The flowchart illustrates the detailed operation of a Solar Panel Tracking System integrated with real-time battery and light intensity monitoring for automated irrigation in polybag farming. The system begins operation at sunrise, triggered either by the Real-Time Clock (RTC) or sufficient ambient light detected by the sensors. Initially, the dual-axis tracking mechanism sets the panel to a pre-defined morning orientation using servo motors—adjusting both azimuth (horizontal) and elevation (vertical) angles.

To continuously optimize sunlight capture, the ESP32 microcontroller reads input from four Light-Dependent Resistors (LDRs) mounted at strategic angles on the panel. These LDRs measure sunlight intensity from multiple directions, allowing the system to compute horizontal and vertical light intensity differences. Based on these values, correction angles are calculated to realign the solar panel with the brightest light source. The dual-axis servo motors then adjust the panel's orientation accordingly, maintaining optimal exposure to sunlight throughout the day.

Simultaneously, the system monitors the battery's voltage using a voltage sensor connected to the ESP32. This data, along with light intensity readings and panel position, is transmitted to **Firestore Realtime Database**, enabling two-way communication with a Flutter-based mobile application. Users can view real-time updates, monitor battery levels, track light intensity trends, and manually control or configure the system via the app interface.

All operational data is also stored in **Firestore Cloud Firestore** for historical analysis, aiding future improvements and performance tracking. To improve energy efficiency, the system enters a **light-duty sleep mode** if no significant adjustments are needed for over 15 minutes. At night, when LDRs detect minimal light, the system transitions into **deep sleep mode**, conserving power until the next operational cycle begins at sunrise.

This intelligent workflow ensures efficient energy harvesting, environmental responsiveness, and user-friendly remote interaction—delivering a sustainable, real-time monitoring system tailored for small-scale agricultural use.

3.3.3 UML Diagram

3.3.3.1 Use Case Diagram

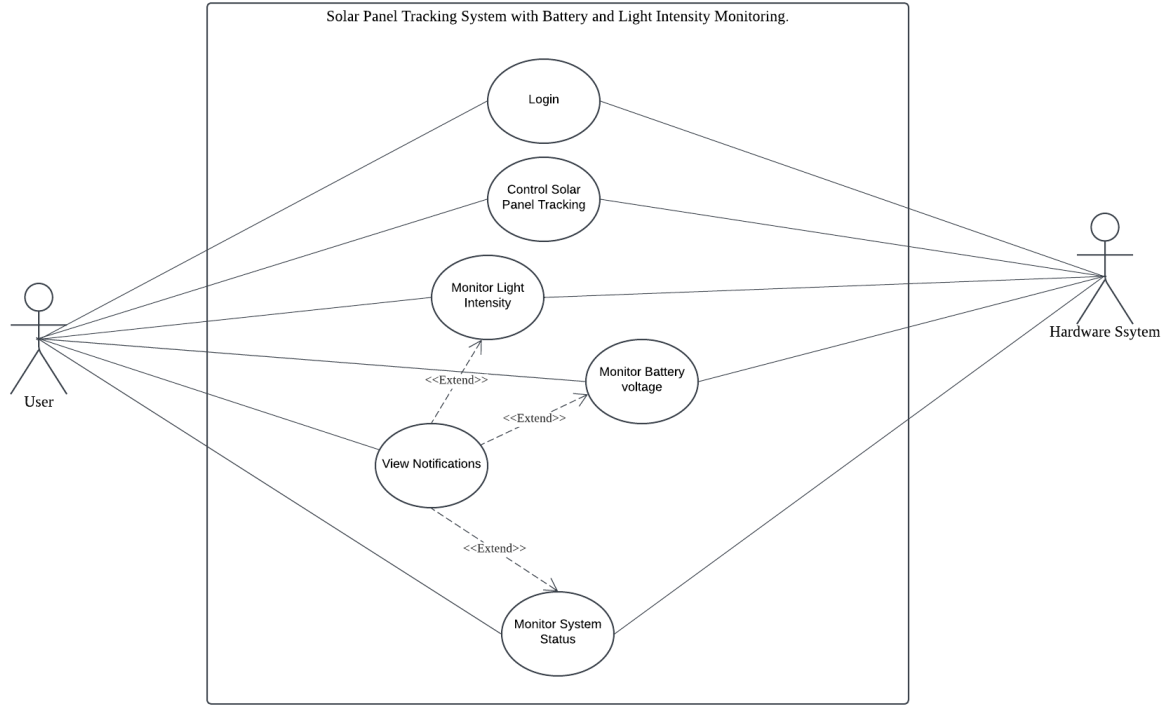


Figure 3.14: Use Case of the Proposed System

The use case diagram serves as a functional blueprint for the Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming. It defines the key interactions between the user, the system's software, and its hardware components, ensuring all functional requirements are clearly captured. The primary purpose of this diagram is to illustrate how the system enables users to interact with and manage solar tracking, monitor environmental and battery conditions, and oversee automated irrigation through a centralized and user-friendly mobile interface.

In this project, the use cases highlight how users can securely log in, control the solar panel's dual-axis tracking mechanism, monitor real-time battery voltage, and

observe ambient light intensity conditions. These real-time monitoring features allow users to understand the current state of solar charging, battery health, and sunlight availability—three essential factors in ensuring effective operation of solar-powered farming systems. The system also includes the ability to receive critical notifications, alerting users to low battery levels or tracking misalignment, enhancing system reliability and responsiveness.

By defining these interactions, the use case diagram ensures that the system fulfills its objectives of improving energy harvesting through intelligent tracking and enabling users to make informed decisions based on live environmental feedback. It also reinforces the integration of IoT with hardware components such as LDRs, voltage sensors, and servo motors—ensuring seamless communication between the physical system and the mobile application, and supporting a scalable, efficient solution for small-scale automated farming.

3.3.3.2 Class Diagram

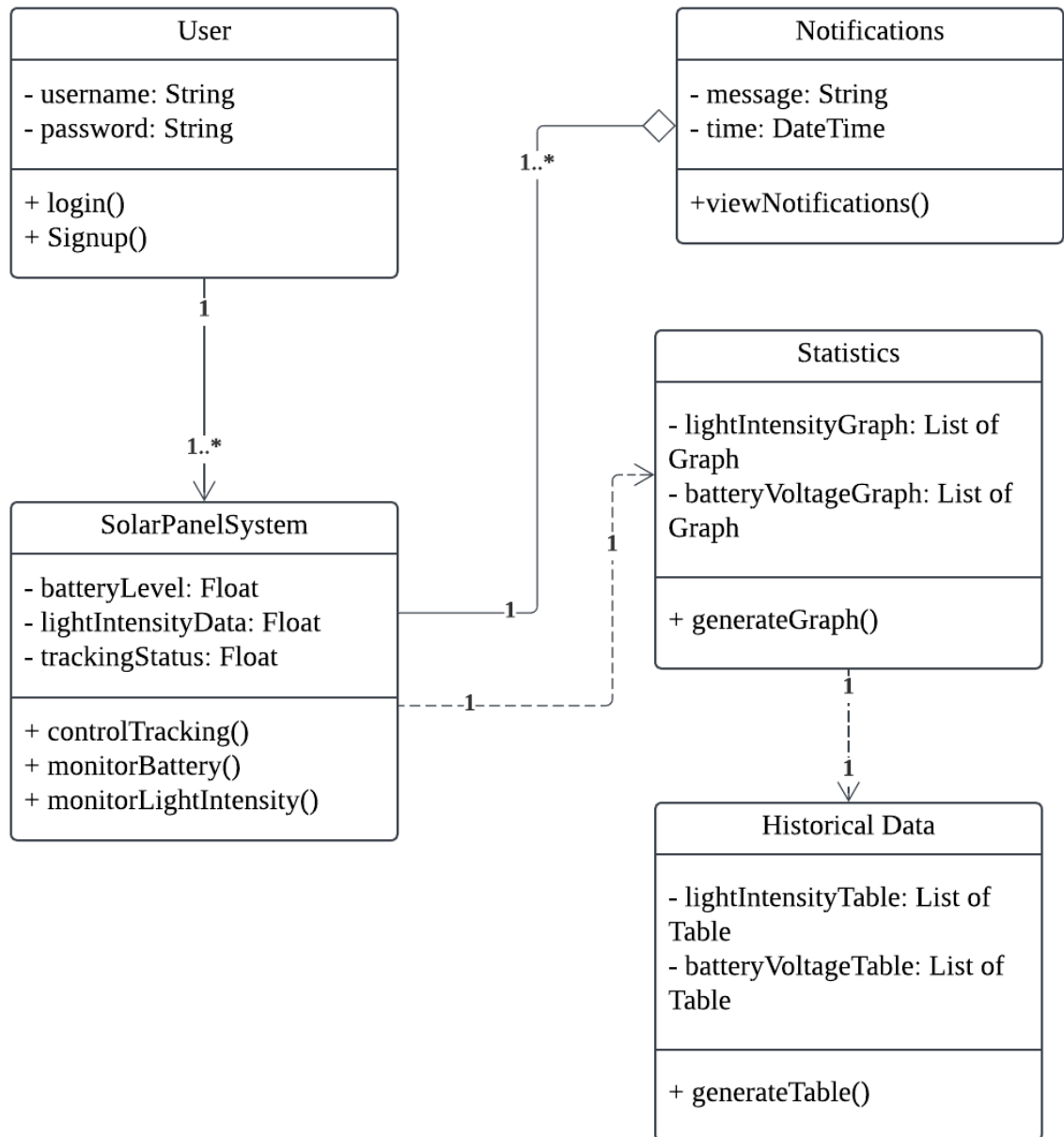


Figure 3.15: Class Diagram for the Proposed System

The class diagram illustrates the structure of the Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming, focusing on its components, attributes, operations, and interrelationships. It provides a foundational blueprint

for designing and implementing the system's core functionalities by detailing how the software interacts with hardware components and user interfaces to meet the project's objectives.

The `User` class manages system access through methods such as `login()` and `accessFeatures()`, enabling authenticated users to securely interact with the mobile application. The `SolarPanelSystem` class encapsulates essential attributes like `batteryVoltage`, `lightIntensity`, and `trackingStatus`, while offering methods such as `adjustPanelOrientation()`, `monitorBatteryVoltage()`, and `readLightIntensity()` to drive dual-axis tracking and environmental monitoring.

The `Notifications` class is responsible for alerting users to important conditions, such as low battery voltage or abnormal light intensity levels. Instead of a traditional report generator, the system now includes a `Statistics` class that compiles real-time and historical data into visual graphs and summaries, helping users make informed decisions through clear insights presented in the mobile interface.

To support long-term performance tracking, a `HistoricalData` class is introduced, handling the retrieval and storage of sensor data in **Firestore**. This class ensures all relevant system logs—such as panel positioning, voltage levels, and light intensity trends—are stored for analysis and reference.

The relationships between these classes reflect a modular and scalable system design, where users can monitor and manage the solar tracking operation, view environmental conditions in real-time, and understand system behavior over time through historical insights. This architecture supports both operational reliability and user-friendly control, ensuring the system aligns with the goals of precision, automation, and sustainability in small-scale agricultural applications.

3.3.3.3 Activity Diagram

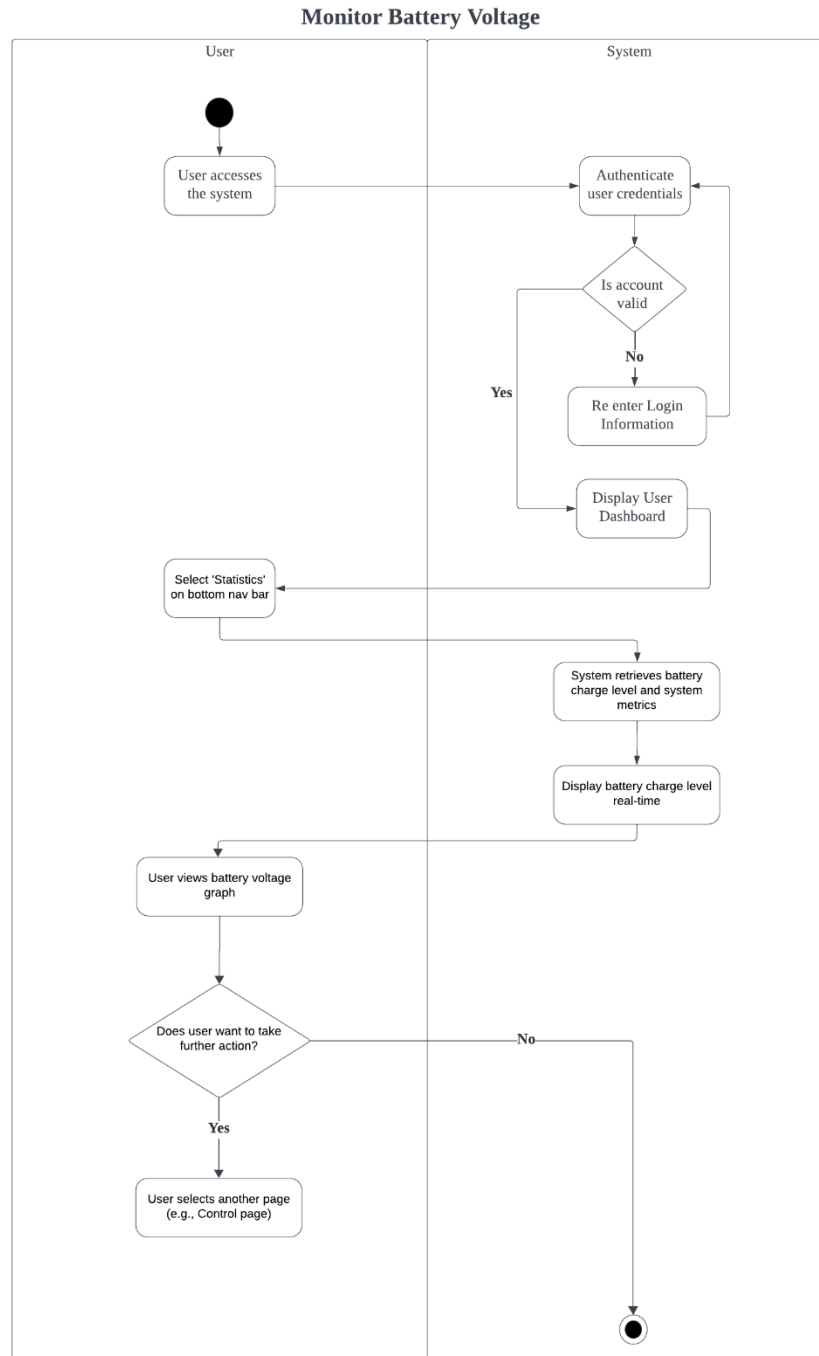


Figure 3.16: Activity Diagram for the proposed system

The activity diagram illustrates the *Monitor Battery Voltage* process within the Solar Panel Tracking System, emphasizing the interaction between the user and system to enable real-time visibility of battery status via the **Statistics page**.

The process begins when the user opens the mobile application and initiates the login process. The system authenticates the user's credentials through Firebase Authentication and verifies account validity. If the credentials are invalid, the system prompts the user to re-enter their login details. Once successfully logged in, the system redirects the user to the **Dashboard**, which provides access to various system features.

The user then selects the **Statistics** page from the dashboard. Upon selection, the system retrieves historical and current battery voltage data from **Firebase Cloud Firestore**, where real-time updates from the ESP32 microcontroller are continuously stored. This data is then visualized on the Statistics page in the form of interactive graphs, allowing users to easily interpret voltage trends over time, assess battery health, and anticipate potential issues such as low-voltage conditions.

After reviewing the battery voltage graph, the user may either return to the dashboard, select another system feature (e.g., real-time light intensity monitoring), or exit the application. If no further actions are taken, the activity ends.

This activity diagram reinforces a clear and user-friendly flow, guiding users from login to data visualization while incorporating real-time updates and historical analysis. By placing battery voltage monitoring under the **Statistics page**, the system delivers meaningful insights that support informed decision-making for solar-powered agricultural operations—ensuring long-term system reliability and efficient farm management.

3.3.3.4 Sequence Diagram

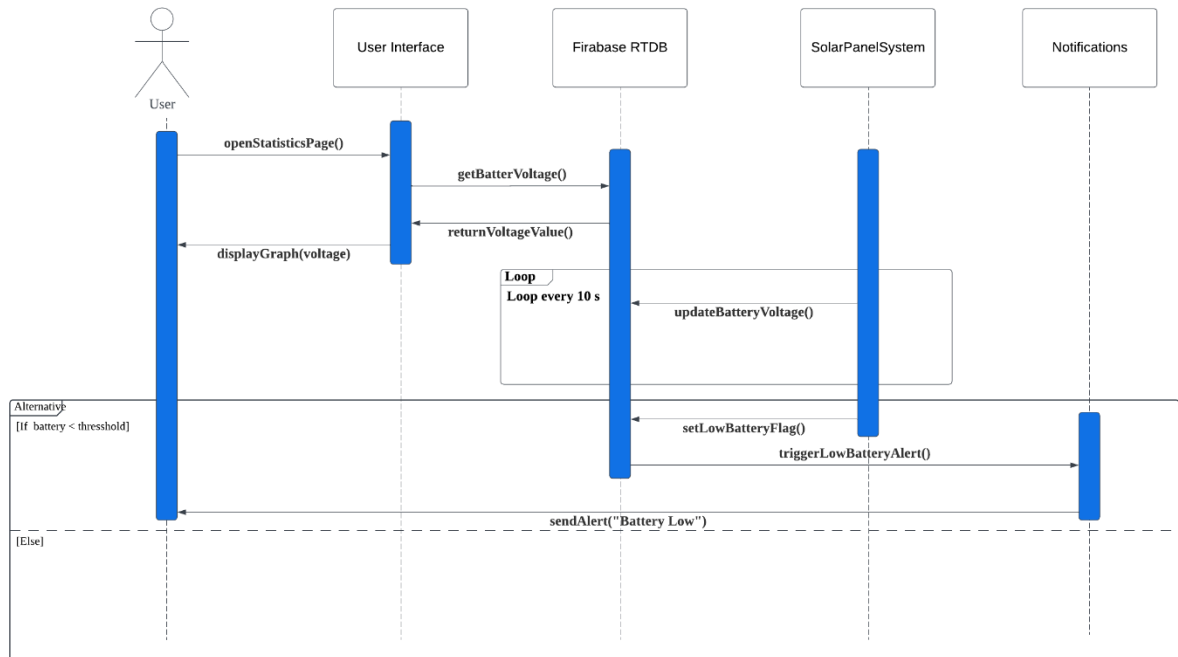


Figure 3.17: Sequence Diagram for the proposed system

The sequence diagram illustrates the interaction between the **User**, **User Interface (UI)**, **SolarPanelSystem (ESP32)**, **Firebase Realtime Database**, and **Notifications** module when executing the *Monitor Battery Voltage* operation. It captures the system's real-time behavior as users monitor the current battery voltage levels through the mobile application.

The process begins when the **User** logs into the mobile app and selects the **Statistics** feature, which includes the battery monitoring graph. The **User Interface** sends a request to the **Firebase Realtime Database** to fetch the latest voltage data.

Behind the scenes, the **SolarPanelSystem** continuously reads the battery's voltage using a voltage sensor and updates the value in the Firebase Realtime Database at regular

intervals. Once the UI receives the latest data from Firebase, it displays the current voltage in both numerical and graphical form on the Statistics page.

If the voltage reading falls below a critical threshold, the **SolarPanelSystem** triggers a condition check and writes a flag or warning state to Firebase. This change is picked up by the **Notifications** module, which immediately sends a low battery alert to the user via the mobile app.

This real-time flow ensures that users receive up-to-date battery health information and are proactively alerted when intervention may be required—such as pausing irrigation or recharging the system. The diagram highlights the critical roles of Firebase in two-way communication and the seamless collaboration between sensor hardware, microcontroller logic, and the mobile interface.

3.3.4 Block Diagram

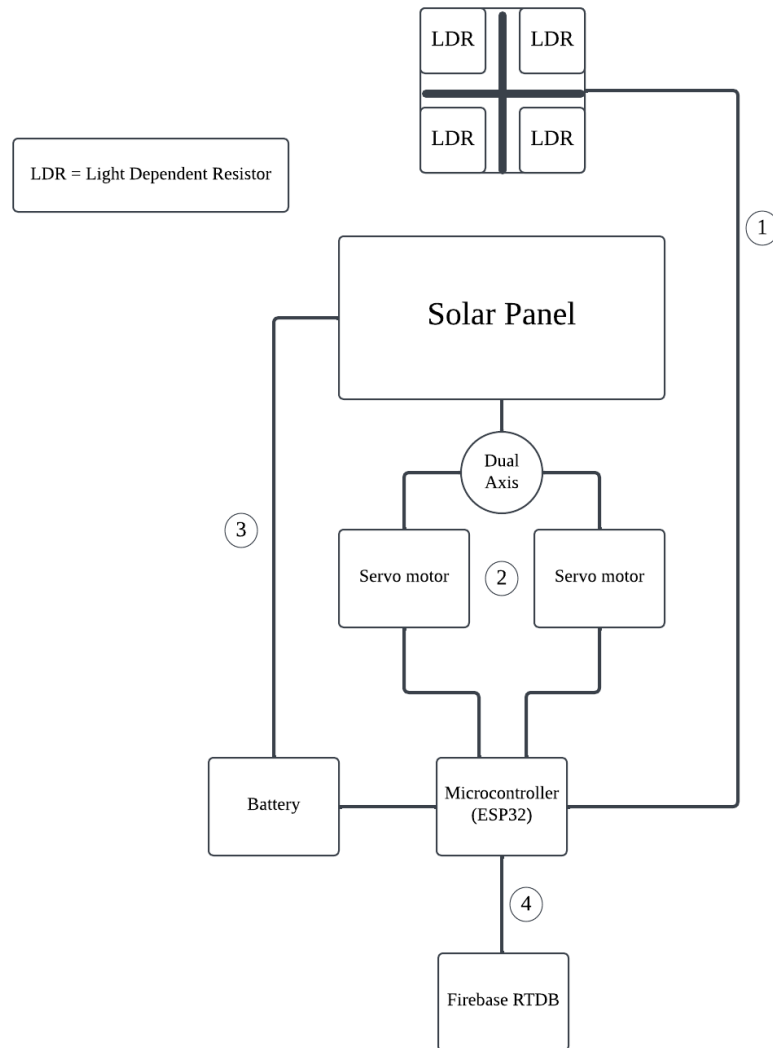


Figure 3.18: Block Diagram for the proposed system

1) Light Dependent Resistors (LDRs):

The block labeled as number **1** consists of four LDRs arranged in a cross pattern. These sensors measure the intensity of sunlight from different directions. Their primary function is to detect the direction of maximum sunlight. The readings from these LDRs help the system identify which angle provides optimal exposure to sunlight. This information is sent to the microcontroller, which adjusts the solar panel's position accordingly.

2) **Servo Motors:**

The components labeled as number **2** represent two servo motors responsible for moving the solar panel on dual axes—horizontal and vertical. Based on the input from the microcontroller (derived from the LDRs), these motors adjust the solar panel's position to maximize sunlight capture. This dual-axis movement ensures that the solar panel aligns with the sun's position throughout the day, improving energy efficiency.

3) **Microcontroller (ESP32):**

The block labeled as number **3** is the microcontroller, specifically an ESP32. It serves as the system's brain, receiving data from the LDR sensors, processing the optimal positioning calculations, and controlling the servo motors. Additionally, the microcontroller monitors the battery's status and communicates with the IoT platform to send real-time updates. Its capability to manage multiple inputs and outputs makes it ideal for this solar panel tracking system.

4) **Firebase RTDB:**

This component acts as the IoT bridge between the ESP32 and the mobile app, enabling real-time data exchange. It stores and updates key sensor readings such as battery voltage, light intensity, and panel orientation, allowing users to monitor the system live through the app. It also supports alerts for conditions like low battery, ensuring timely user notifications and reliable remote system management.

3.3.5 Interface Design

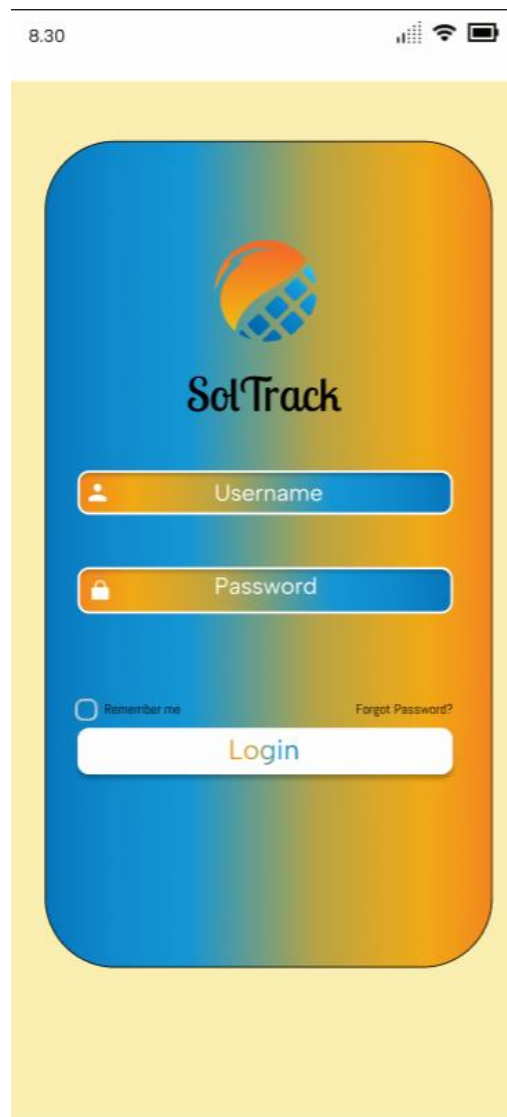


Figure 3.19: Login Page of the proposed system.

Figure 3.19 shows the **login page** of the SolTrack app. Its main function is to allow users to securely log in by entering their **username** and **password**. It also includes a "**Remember me**" option to save login details for future use and a "**Forgot Password?**" link to help users reset their password if needed. The **Login** button submits the entered details to access the app.

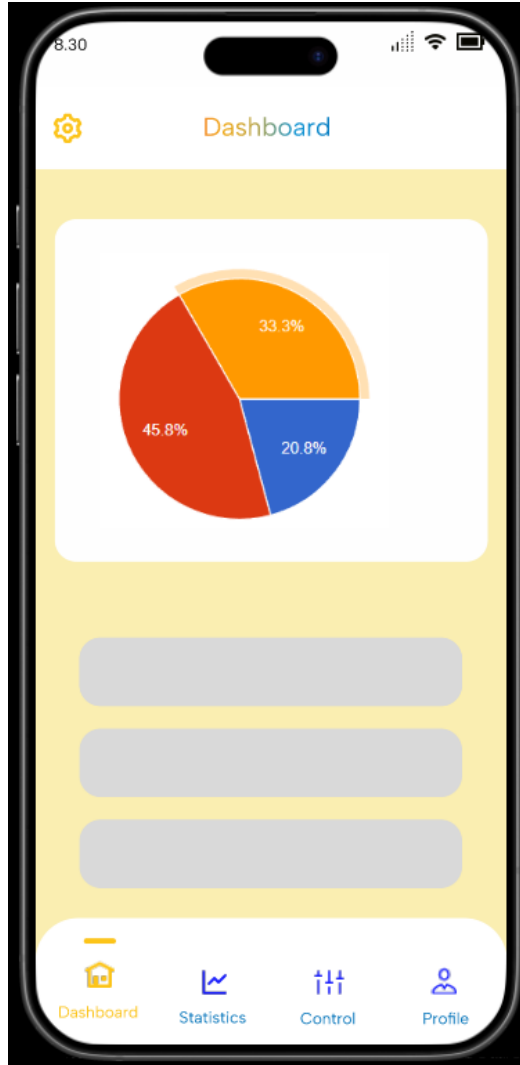


Figure 3.20: Dashboard page for the proposed system.

This is the **dashboard page** of the SolTrack app. Its main function is to provide users with a quick overview of important data. The page includes a **pie chart** that shows key information, such as energy distribution or usage statistics, in a visual format. Below the chart, there are placeholders for additional details or updates.

At the bottom, there is a **navigation bar** with four options: Dashboard, Statistics, Control, and Profile, allowing users to easily switch between sections of the app. The gear icon at the top gives access to settings.



Figure 3.21: Statistics page for the proposed system

This is the **statistics page** of the SolTrack app. Its function is to display detailed data in a graphical format for better understanding. The page includes a **bar chart**, which represents information like energy performance or usage trends over time. The top section has placeholders for additional metrics or summaries that can help users analyze their system's performance.

The navigation bar at the bottom allows users to switch to other sections of the app, such as Dashboard, Control, or Profile. The gear icon at the top provides access to the app settings.



Figure 3.22: Control Page of the proposed system

This is the **control page** of the SolTrack app. Its function is to let users manage and control different aspects of the solar tracking system. The "Monitor" button suggests an option to view real-time system data or status, while the placeholders indicate future controls or features like adjusting settings, toggling functions, or managing connected devices.

The navigation bar at the bottom allows users to switch between other sections, such as Dashboard, Statistics, and Profile. The gear icon at the top provides access to additional settings.

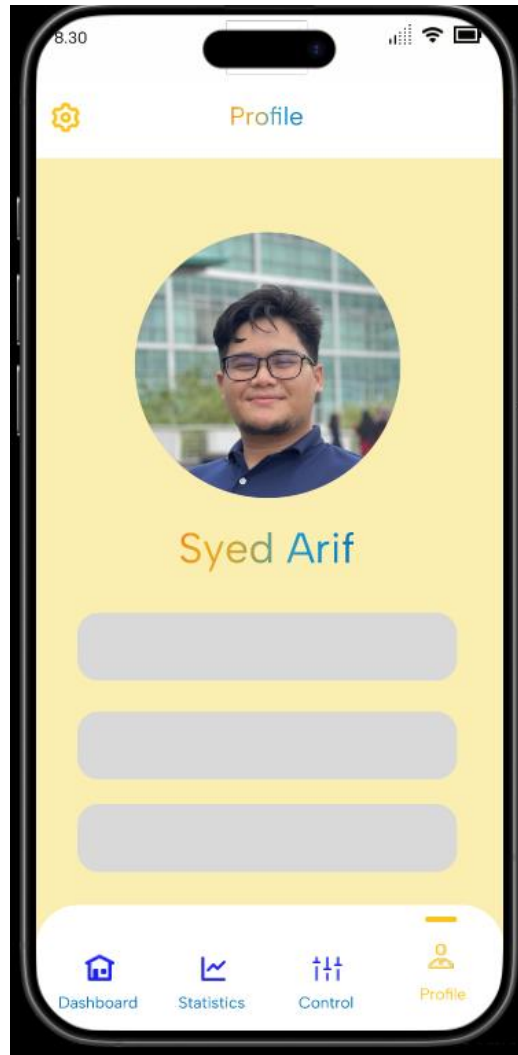


Figure 3.23: Profile Page of the proposed system

The **Profile Page** serves as a personalized section of the system, designed to display user-specific information and provide access to related settings. At the top, the user's profile picture and name are prominently displayed, offering a personalized and identifiable touch to the interface. Below this, there is a dedicated section for additional user details, such as email, phone number, or account-related information, ensuring easy access to essential data. This page combines functionality and a user-centric design, acting as a central point for managing personal information while maintaining smooth navigation within the application.

3.4 Summary

Chapter 3, “*Requirement Analysis and Design*,” outlines what the Solar Panel Tracking System needs to achieve and how it will be structured to meet those goals. The chapter begins by identifying limitations in existing solar systems, especially for small-scale setups like polybag farming, where fixed-position panels and lack of real-time insight hinder energy efficiency. It explains how user surveys helped identify essential features such as **dual-axis tracking**, **real-time battery voltage monitoring**, and **light intensity sensing**, all accessed through a simple, mobile-friendly interface.

Functional requirements define key system capabilities, including automatic solar panel alignment based on sunlight direction, monitoring battery voltage in real time, and tracking ambient light intensity to support informed decision-making. Non-functional requirements ensure the system remains **reliable**, **responsive**, and **user-friendly**, even under varying weather and usage conditions.

The design section breaks down the integration of hardware components—such as solar panels, LDR sensors, servo motors, voltage sensors, and the ESP32 microcontroller—with **Firebase Realtime Database** and a **Flutter-based mobile app**. These components work together to deliver live system updates and user control through IoT connectivity. Supporting diagrams like flowcharts, system architecture, use case, and class diagrams visually demonstrate the logical and physical design.

A notable feature of the system is its ability to **enter low-power sleep mode** when inactive, conserving battery life. The system also logs real-time data for analysis, allowing users to monitor trends and make better farm management decisions. This chapter establishes a strong

foundation for building a system that is not only functional and scalable but also aligned with the real-world needs of modern, tech-enabled agriculture.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

This chapter presents the detailed construction and integration process of the **Solar Panel Tracking System with Battery and Light Intensity Monitoring**, hereafter referred to as **SolTrack**. It directly builds upon the architectural design established in Chapter 3 and translates the system blueprint into a fully functional prototype that addresses the research objectives defined in FYP1. The implementation integrates both physical hardware and cloud-connected software to support efficient solar tracking and data-driven irrigation monitoring in small-scale polybag farming.

The core hardware is powered by an **ESP32 microcontroller**, chosen for its built-in Wi-Fi and GPIO capabilities, making it ideal for IoT-based control. Key peripheral components include **Light Dependent Resistors (LDRs)** for directional sunlight detection, **servo motors** for real-time panel alignment, a **5V solar panel** for power input, and a **voltage sensor** to monitor battery health. These components were assembled on a breadboard prototype with standard 6V Li-ion battery support and tested under typical Malaysian sunlight conditions.

On the software side, the system firmware was developed in **C++ using PlatformIO within Visual Studio Code**, ensuring effective hardware abstraction and dependency management. The accompanying mobile application was designed in **Flutter** using Dart programming language, enabling cross-platform deployment and seamless integration with the cloud backend. For cloud communication, **Firebase Realtime Database** and **Firebase**

Authentication were used to handle data storage, user login, and real-time data synchronization between the ESP32 and the mobile interface.

To enhance modularity and maintainability, the implementation is organized into **five core modules**:

1. **Solar Tracking Module**
2. **Light Intensity Monitoring Module**
3. **Battery Voltage Monitoring Module**
4. **IoT Communication Module**
5. **Mobile Application Interface**

Each module performs a distinct function and interacts with others via defined input/output channels and Firebase data structures. These modules will be elaborated in the following sections, including hardware diagrams, data flow explanations, configuration parameters, and annotated screenshots of the mobile interface. Where applicable, rationales for component selection and performance insights from early integration testing will be discussed to illustrate the engineering decisions behind the system.

4.2 Implementation Tool

4.2.1 Software Tools

4.2.1.1 Visual Studio Code



Figure 4.1: Visual Studio Code

Visual Studio Code is a free and open-source code editor developed by Microsoft. In this project, Visual Studio Code was chosen as the main development environment due to its lightweight structure, extensive extension marketplace, and built-in support for multiple programming languages. For this system, it was used to write and manage both the Flutter-based mobile application and the ESP32 firmware. Its built-in IntelliSense provides smart code completion, syntax highlighting, and code navigation, which enhanced productivity and reduced errors during development. Additionally, its integration with PlatformIO and Flutter extensions allowed for smooth project structuring, serial monitor access, and firmware deployment.

4.2.1.2 Flutter SDK



Figure 4.2: Flutter SDK

Flutter is an open-source UI toolkit developed by Google, designed for building natively compiled applications for mobile, web, and desktop from a single codebase. This project used Flutter to develop the mobile application (SolTrack) that interfaces with Firebase and displays real-time data such as light intensity and battery voltage. Flutter's widget-based architecture allowed for rapid development of responsive user interfaces, and its hot-reload feature significantly reduced development time during UI testing and refinement.

4.2.1.3 PlatformIO



Figure 4.3: PlatformIO

PlatformIO is an open-source ecosystem for IoT development that runs as an extension within Visual Studio Code. It was used in this project to write, compile, and upload C++ code to the ESP32 microcontroller. PlatformIO simplifies dependency management, library integration, and cross-platform build processes, making it ideal for embedded system projects like this one. The tool also provides serial monitoring capabilities and detailed debugging logs, which were essential for testing sensor input, servo behavior, and Firebase communication.

4.2.1.4 Firebase



Figure 4.4: Firebase

Firebase is a cloud-based Backend-as-a-Service (BaaS) platform by Google. It was used in this project to handle user authentication, real-time database operations, and cloud communication between the ESP32 and the mobile application. Firebase Authentication was implemented to manage secure login and signup processes, while Firebase Realtime Database enabled continuous two-way data sync between the ESP32 and the app. Its ease of integration with both Flutter and ESP32 made it a suitable and scalable solution for IoT communication.

4.2.1.5 Figma



Figure 4.5: Figma

Figma is a cloud-based UI/UX design tool. Although not used heavily for this project, it was initially used to plan and prototype the layout of the SolTrack mobile app. Figma helped in visualizing the overall navigation flow, color themes, and responsive layout structure before full implementation in Flutter.

4.2.1.6 Git & GitHub



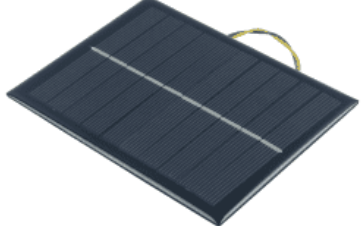
Figure 4.6: Git & GitHub

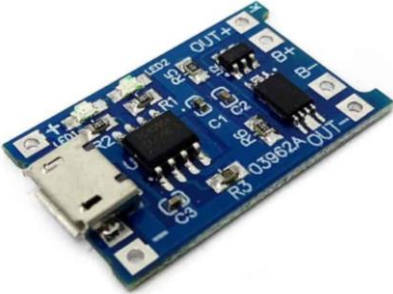
Git is a distributed version control system used to track changes in source code during software development. It allows developers to manage project versions, collaborate efficiently, and revert to previous states if needed. In this project, Git was used to track all changes made to the Flutter application code and ESP32 firmware throughout the development process.

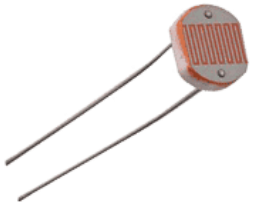
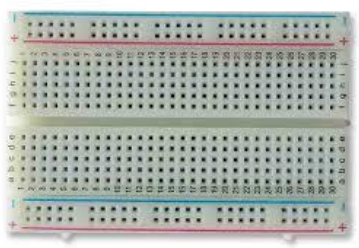
GitHub, a cloud-based platform that hosts Git repositories, was utilized to store and back up the entire project codebase remotely. It enabled version control, issue tracking, and secure sharing between development devices. The use of GitHub also provided a structured commit history, which was valuable for managing development milestones, debugging, and documenting code evolution. This tool ensured better collaboration, organization, and recoverability during the system's implementation phase.

4.2.2 Hardware Tools & Components

Table 4.1: List of Hardware Tools & Components

Components/Tools	Details	Function
	ESP32 DOIT Devkit V1 (1 Unit)	Microcontroller used for sensor data processing and network communication.
	Servo Motors (2 units)	Provide azimuth and elevation control for dual-axis solar tracking.
	Solar Panel (5V/2W)	Primary energy source powering the system.

Components/Tools	Details	Function
	5V Relay Module (Single Channel)	Acts as an electronic switch that allows the ESP32 to control high-current components (e.g., turning on/off charging circuits, load balancing, or future peripherals like irrigation pumps) while keeping isolation between control and power sides.
	TP4056 Charge Module	Manages safe charging of the lithium-ion battery from the 5V solar panel. Provides constant current/constant voltage charging with overcharge protection, undervoltage protection, and current limiting — ensuring battery lifespan and safety in the system.

Components/Tools	Details	Function
	LDR Sensors (4 units)	Detect directional light intensity and track solar exposure.
	Voltage Sensor Module (1 Unit)	Measures battery voltage for monitoring power health.
	Battery (6V Li-ion/Lead-acid)	Stores energy generated for use during cloudy periods or night.
	Breadboard & Jumpers	Temporary circuit prototyping and testing setup.

Components/Tools	Details	Function
		
	Resistors	Used in voltage dividers and filtering to stabilize inputs.
	Capacitors	Used to supply adequate amount of power for the servo motor.
	Soldering Iron & Tools	Assembly of permanent hardware layout for final deployment.
	Multimeter	Used for continuity checks and voltage measurements.

4.3 Solar Panel Tracking System Prototype

The SolTrack system was engineered to operate **autonomously** in field conditions while simultaneously providing **interactive functionality** through a user-accessible mobile application. This dual-mode operation ensures both self-sufficiency in environmental adaptation and ease of user engagement. Upon launching the mobile application, users are prompted to **log in or register** via **Firestore Authentication**, which manages secure access and user-specific data visibility. Once authenticated, users are directed to the **main dashboard**, which presents real-time system telemetry—including light intensity, solar tracking status, and battery voltage—alongside historical trends and system control options.

The system is structured using modular **hierarchical architecture**, (refer **Figure 4.7: System Hierarchy Diagram**) where the overall system is divided into several interdependent subsystems that collectively fulfill its core objectives of optimizing solar energy capture and enabling informed decision-making for polybag irrigation. Specifically, five distinct functional modules are implemented, each tasked with a dedicated operational role:

1. **Solar Tracking Module** – Handles dual-axis realignment of the panel using LDR input.
2. **Light Intensity Monitoring Module** – Categorizes environmental brightness levels using calibrated ADC values.
3. **Battery Monitoring Module** – Tracks system power availability using a voltage sensor and alerts under critical thresholds.
4. **IoT Communication Module** – Syncs data between the ESP32 and Firestore cloud database in real time.

5. **Mobile Application Module** – Provides a user interface for monitoring, notifications, and manual control.

These modules are integrated to form a cohesive prototype that meets the performance requirements of **smart, resource-aware agriculture systems**. Each module will be discussed in the following sections, detailing its internal logic, hardware/software interface, and contribution to overall system goals.

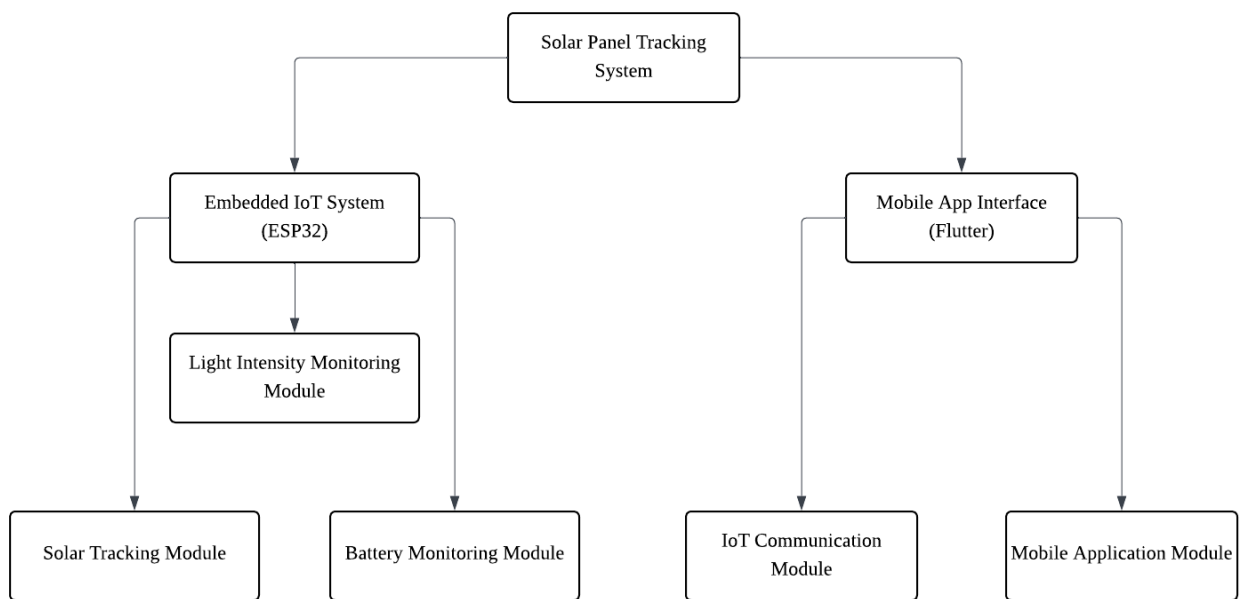


Figure 4.7: System Hierarchy Diagram

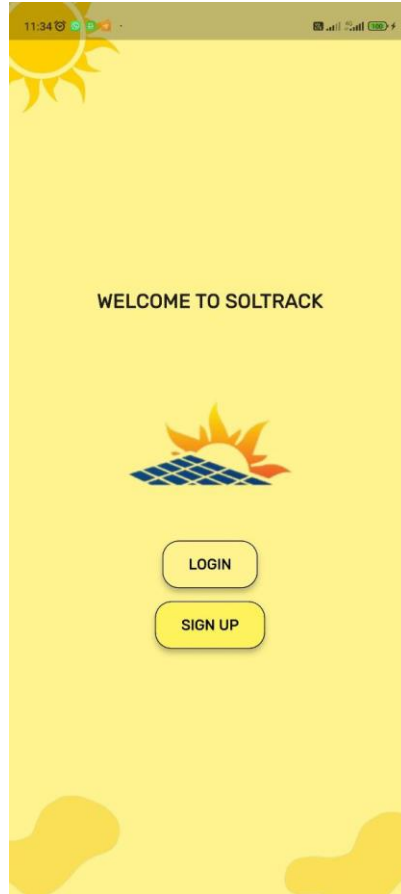


Figure 4.8: Overview of the Project's Software

4.3.1 Solar Tracking Module

The Solar Tracking Module is responsible for **maximizing solar energy capture** by dynamically adjusting the orientation of the solar panel based on the sun's movement across the sky. In the context of **polybag farming**, efficient and consistent sunlight capture is crucial to ensure adequate energy supply for all downstream components, particularly irrigation and sensing subsystems. To achieve this, the module implements a **dual axis tracking mechanism**, allowing the panel to orient toward the direction with maximum incident light.

4.3.1.1 Functional Description

The tracking logic is driven by **four Light Dependent Resistors (LDRs)** mounted in a **cross-shaped formation** on a raised nozzle structure at the center of the panel. Each LDR faces a cardinal direction—top, bottom, left, and right—mimicking compass alignment to capture directional sunlight intensity.

The **ESP32 microcontroller** continuously samples analog values from all four LDRs and computes the differential between:

- Left and right sensors (X-axis alignment)
- Top and bottom sensors (Y-axis alignment)

When the difference exceeds a **calibrated lux threshold** (experimentally set at ± 15 lux), the system identifies a misalignment between the panel and the sun's vector. In response, the ESP32 issues **PWM signals** to two servo motors, which rotate the panel incrementally along both X and Y axes to reorient it toward the direction of highest irradiance. This adjustment loop runs in near real time, with servo response time averaging under 2 seconds.

This mechanism reduces power loss due to static misalignment and enables consistent energy harvesting even during partial cloud cover or morning/evening sun shifts.

4.3.1.2 Hardware and Wiring Implementation

Table 4.2: Hardware/Wiring Connection for Solar Panel Tracking Model

Component	Qty	Connection Details
ESP32 DOIT DevKit V1	1	Connected to all sensors and servos via GPIO and ADC
LDR Sensors	4	Connected to ADC pins A0 (top), A1 (right), A2 (bottom), A3 (left)
Plus-Shaped Nozzle	1	Mounted on panel center
MG90S Servo Motors	3	Azimuth: GPIO14, Elevation: GPIO12
10k Ω Resistors	4	Connected in series with LDRs
5V 2W Solar Panel	1	Connected to battery charging module
Breadboard & Jumpers	-	Ensure shared ground and 5V power lines
Capacitors (Decoupling)	2	Placed across power rails near servos

4.3.1.3 Software/Code Implementation

The logic is implemented using C++ on PlatformIO. The ESP32 continuously reads LDR values and compares them to adjust the panel.

```

bool SolarTracker::performSimplifiedTracking() {
    if (DEBUG_MODE) {
        Serial.println("🌀 Starting 4-quadrant tracking cycle...");
    }

    // Check if sufficient light for tracking
    LDRReadings readings = readAllLDRs();
    if (readings.totalLight < LDR_THRESHOLD * 4) { // Total threshold
        Serial.println("☹️ Insufficient light for tracking");
        return false;
    }

    // Enable servo power for movement
    enableServoPower();

    bool movementMade = false;
    int attempts = 0;

    while (attempts < MAX_TRACKING_ATTEMPTS) {
        readings = readAllLDRs();

        // Determine tracking direction based on quadrant comparison
        TrackingDirection direction = determineTrackingDirection(readings);

        if (direction.azimuthAdjustment == 0 && direction.elevationAdjustment == 0) {
            Serial.println("✅ Optimal position found - no movement needed");
            break;
        }

        // Calculate new position
        int newAzimuth = currentAzimuth + (direction.azimuthAdjustment * TRACKING_STEP_SIZE);
        int newElevation = currentElevation + (direction.elevationAdjustment * TRACKING_STEP_SIZE);

        // Apply boundary constraints
        newAzimuth = constrain(newAzimuth, AZIMUTH_MIN_ANGLE, AZIMUTH_MAX_ANGLE);
        newElevation = constrain(newElevation, ELEVATION_MIN_ANGLE, ELEVATION_MAX_ANGLE);
    }
}

```

Figure 4.9: Solar Tracker Module's Code Example

4.3.1.1 System Integration

This module is tightly integrated into the IoT framework of the system. Upon each actuation cycle, the **ESP32** logs the current servo angles and corresponding LDR values, then transmits them to the **Firestore Realtime Database (RTDB)** using structured JSON payloads. The data paths follow a consistent hierarchy:

/device/ldr_values and /device/commands/angle.

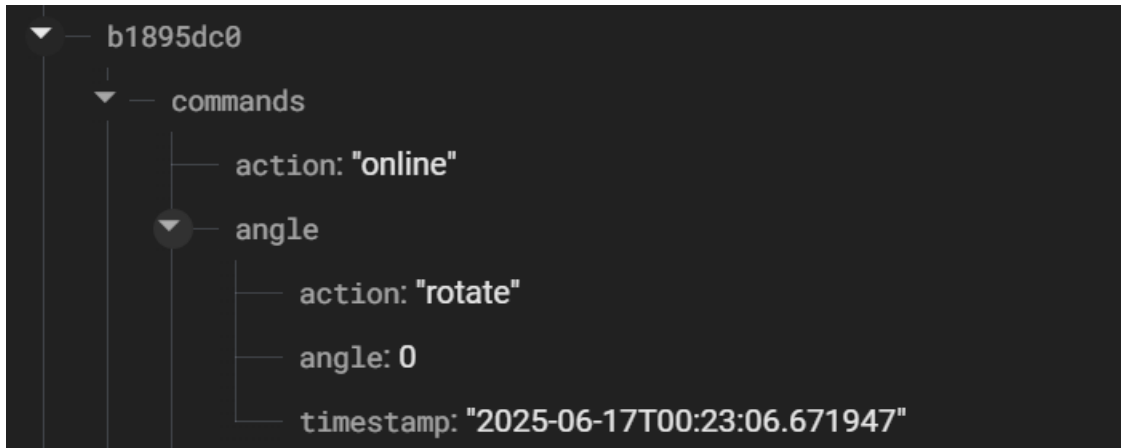


Figure 4.10: JSON Payloads for Solar Tracker Module

On the user-facing side, the **Flutter-based mobile application** subscribes to these Firebase nodes using real-time listeners (`StreamBuilder`). This enables live updates on panel orientation and light distribution, which are visualized through labeled indicators and telemetry graphs in the app's **Dashboard Page**.



Figure 4.11: Mobile App Interface Presentation of the Panel Orientation and Light Distribution

Moreover, since this module directly affects solar panel alignment, it has a **downstream impact** on battery charging performance. Improved alignment supports optimal energy harvesting, indirectly enhancing the performance of the **Battery Monitoring Module**.

4.3.2 Light Intensity Monitoring Module

The **Light Intensity Monitoring Module** is responsible for capturing and recording ambient sunlight levels in real time. This functionality is essential in supporting intelligent decision-making processes within both the solar tracking and irrigation subsystems. In the context of polybag farming, accurately monitoring light exposure helps evaluate the environmental suitability for crop growth and enables adaptive responses from the system.

This module provides the empirical basis for analyzing the relationship between **sunlight availability, energy harvesting performance, and irrigation timing**, thereby contributing to the system's overall sustainability and precision agriculture capabilities.

4.3.2.1 Functional Description

The module leverages the same four **Light Dependent Resistors (LDRs)** used by the Solar Tracking Module, mounted in a **plus-shaped nozzle structure** atop the solar panel. Each LDR faces a cardinal direction (north, east, south, west) and generates analog voltage values proportional to the incident light intensity in that direction.

The **ESP32 microcontroller** reads these values through its ADC pins and aggregates them to derive two key insights:

1. **Directional light distribution** – used for dynamic solar tracking.
2. **Overall ambient light level** – used to assess sunlight availability.

To quantify light levels, the analog values are mapped to **lux units** through a calibrated transfer function, developed by referencing readings from a commercial lux meter. For example, an ADC value of ~900 corresponds to ~10,000 lux in direct sunlight. This

calibration ensures that the system can distinguish between lighting conditions such as full sunlight, partial shade, and overcast skies with reasonable accuracy.

The light readings are time-stamped and averaged hourly for trend analysis. These values are also categorized into **sunlight intensity bands** (e.g., full, moderate, low), which may inform future versions of the system to trigger irrigation cycles or shading mechanisms based on real-time light data.

4.3.2.2 Hardware and Wiring Implementation

(Refer Table 4.2: Hardware/Wiring Connection for Solar Panel Tracking Model)

4.3.2.3 Software/Code Implementation

The code written in C++ on PlatformIO reads the LDR values and uploads them to Firebase RTDB.

```
LDRReadings SolarTracker::readAllLDRs() {
    LDRReadings readings;

    // Read all four quadrants
    readings.topLeft = analogRead(LDR_TOP_LEFT_PIN);
    readings.topRight = analogRead(LDR_TOP_RIGHT_PIN);
    readings.bottomLeft = analogRead(LDR_BOTTOM_LEFT_PIN);
    readings.bottomRight = analogRead(LDR_BOTTOM_RIGHT_PIN);

    // Calculate averages (matching your existing structure)
    readings.averageTop = (readings.topLeft + readings.topRight) / 2;
    readings.averageBottom = (readings.bottomLeft + readings.bottomRight) / 2;
    readings.averageLeft = (readings.topLeft + readings.bottomLeft) / 2;
    readings.averageRight = (readings.topRight + readings.bottomRight) / 2;

    // Calculate total light
    readings.totalLight = readings.topLeft + readings.topRight +
        readings.bottomLeft + readings.bottomRight;

    // Update timestamp
    readings.timestamp = millis();

    if (DEBUG_MODE) {
        Serial.printf("LDR: TL=%d TR=%d BL=%d BR=%d Total=%d\n",
            readings.topLeft, readings.topRight,
            readings.bottomLeft, readings.bottomRight, readings.totalLight);
    }

    return readings;
}
```

Figure 4.12: Light Intensity Module's Code Example

4.3.2.4 System Integration

The ESP32 transmits the lux values and light category labels to **Firestore Realtime Database (RTDB)** at fixed intervals (default: every 15 seconds). Data is structured under the Firestore node `/device/lux`, including:

- Average lux (current)
- Light category (e.g., full sun, moderate, low)
- Timestamp (ISO format)

On the mobile side, the **Flutter application** retrieves this data and displays it under the **Statistics page** using StreamBuilder widgets and chart visualizations. A line graph illustrates daily light trends, while current values are shown in labeled badges for at-a-glance interpretation.

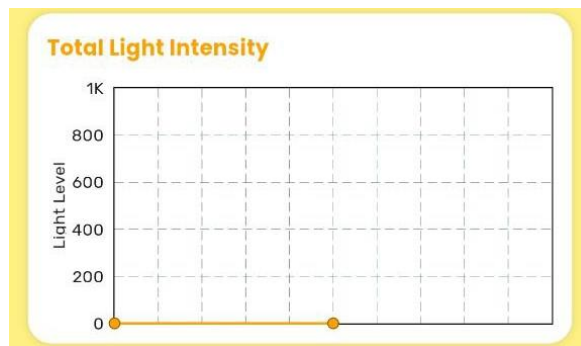


Figure 4.13: Total Light Intensity Display

This tight cloud integration ensures real-time synchronization and historical traceability. Moreover, it allows users to make informed decisions based on light availability, and in future iterations, it may feed into **machine learning models** for crop-specific irrigation prediction.

4.3.3 Battery Monitoring Module

The **Battery Monitoring Module** plays a vital role in ensuring uninterrupted operation of the solar tracking and smart irrigation system. Within polybag farming environments—where energy resources are limited and system autonomy is essential, **real-time monitoring of battery voltage** enables intelligent energy management, avoids unplanned downtime, and supports data-driven scheduling of energy-dependent tasks.

This module ensures that all other components, including servos and the Wi-Fi-enabled ESP32, operate within safe voltage thresholds, thus extending battery lifespan and maintaining system reliability.

4.3.3.1 Functional Description

The system continuously monitors the voltage output of the **6V battery** using a precision **voltage divider circuit**, which steps down the raw voltage to a level readable by the **ESP32's analog input (max 3.3V)**. The sensor arrangement includes equal value **10kΩ resistors (R1 = R2)** to simplify the voltage scaling formula:

$$V_{battery} = 2 \times V_{ADC}$$

The ESP32 reads this scaled-down voltage via its ADC channel and computes the actual battery voltage in volts (V). Based on this computed value, the system categorizes the battery status into multiple tiers:

- **Normal Operating Range:** $\geq 5.0V$
- **Moderate:** 4.2V–4.9V
- **Low Warning:** 3.6V–4.1V
- **Critical:** $\leq 3.5V$ (alerts triggered)

These thresholds were derived from empirical testing and manufacturer datasheets to prevent **over-discharge**, which can lead to **battery degradation or system failure**. Alerts and visual warnings are triggered on both the device and the mobile interface when the voltage falls below 3.6V.

4.3.3.2 Hardware and Wiring Implementation

Table 4.3: Hardware/Wiring Connection for Battery Module

Component	Qty	Connection Details
ESP32 DOIT DevKit V1	1	Connected to all sensors and servos via GPIO and ADC
Voltage Sensors Module	1	Analog output to ESP32 A0
Li-Ion Battery (3.7V)	1	Connected to load and monitored by voltage sensor
Breadboard & Jumpers	-	Ensure shared ground and 5V power lines

4.3.3.3 Software/Code Implementation

The code reads analog voltage from the voltage sensor and uploads it to Firebase RTDB.

```

float BatteryMonitor::readBatteryVoltage() {
    int adcValue = analogRead(BATTERY_VOLTAGE_PIN);

    // Convert ADC reading to actual voltage using sensor module
    float rawVoltage = (adcValue * ADC_REFERENCE_VOLTAGE / ADC_RESOLUTION);

    // Apply calibration for sensor module
    float calibratedVoltage = rawVoltage * BATTERY_VOLTAGE_MULTIPLIER + batteryVoltageOffset;

    if (DEBUG_MODE) {
        Serial.printf("🔋 Battery - ADC:%d Raw:%.2fV Calibrated:%.2fV\n",
            adcValue, rawVoltage, calibratedVoltage);
    }

    return calibratedVoltage;
}

```

Figure 4.14: Battery Monitoring Module's Code Example

4.3.3.4 System Integration

To ensure real-time visibility and historical trend analysis, the ESP32 **transmits battery voltage readings to Firebase Realtime Database (RTDB)** every 15 seconds under the node `/device/voltage`.

```

/battery/
  voltage: 3.82
  timestamp: "2025-06-20T10:45:00"

```

The **Flutter mobile application** retrieves this data and renders it on the **Statistics page** using a **line chart** and visual status indicators (e.g., green for normal, orange for warning, red for critical). These indicators provide intuitive feedback to users, enabling them to:

- Monitor battery health over time
- Anticipate energy depletion
- Postpone or manually trigger irrigation events if power is insufficient

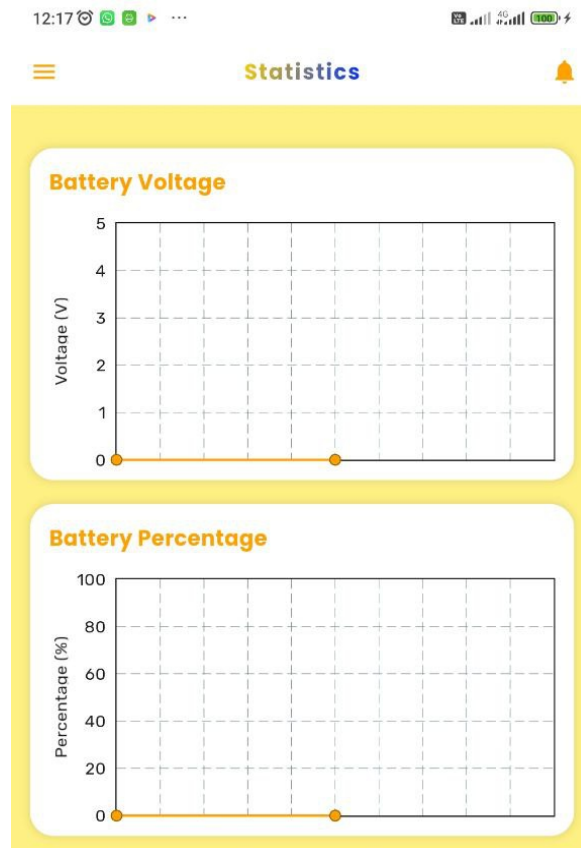


Figure 4.15: Graph for the Battery Monitoring in Mobile App

This integration enhances system transparency and empowers users with actionable information. Future iterations of this module may incorporate energy usage forecasting or automated shutdown logic under critical voltage thresholds to protect hardware components.

4.3.4 IoT Communication Module

The **IoT Communication Module** forms the backbone of SolTrack's real-time data exchange system, enabling **bi-directional communication** between the microcontroller and the mobile application. By leveraging cloud infrastructure, this module ensures that all sensor readings, system statuses, and user interactions are consistently synchronized across devices. In the context of polybag farming, such connectivity empowers users to remotely monitor

environmental conditions, battery status, and panel orientation—making informed decisions with minimal manual intervention.

This module aligns with the system’s objective to provide a **transparent, remotely accessible, and data-driven solution** to resource-constrained agricultural environments

4.3.4.1 Functional Description

The communication architecture is built around the **ESP32’s integrated Wi-Fi capability** and the **Firestore Realtime Database (RTDB)**. Upon establishing a secure Wi-Fi connection, the ESP32 continuously performs two core functions:

1. **Data Uplink:** Sensor readings and system parameters—including:
 - **Light intensity values**
 - **Battery voltage**
 - **Servo angles (panel orientation)** are formatted as JSON payloads and uploaded to predefined database paths:

```
/device/lux  
/device/voltage  
/device/servo_angle
```

2. **Command Downlink:** The ESP32 subscribes to changes in specific Firestore nodes reserved for **user-issued commands**, such as:
 - Manual panel alignment toggle
 - Reset or shutdown triggers
 - Threshold reconfiguration (future support)

- This enables near-instantaneous actuation of remote user inputs from the mobile interface.

Data payloads include **timestamp metadata** (UTC format) to ensure chronological traceability and facilitate future logging or analytic extensions.

4.3.4.2 Hardware and Wiring Implementation

Table 4.4: Hardware/Wiring Connection for IoT Communication

Component	Qty	Connection Details
ESP32 DOIT DevKit V1	1	Connects to Firebase RTDB over 2.4 GHz network
Mobile Hotspot/Wifi Router	1	ESP32 and mobile app must be on same network
Smartphone	1	Connected to load and monitored by voltage sensor

4.3.4.3 Software/Code Implementation

The ESP32 firmware initializes Firebase, connects to Wi-Fi, and continuously syncs sensor data.

```

0  bool DataLogger::uploadToFirebase(const SystemData& data) {
1      if (!isFirebaseReady()) {
2          return false;
3      }
4
5      String dataPath = data.deviceId;
6      dataPath += "/data/";
7      dataPath += String(data.timestamp);
8
9      DynamicJsonDocument doc(2048);
10
11      // Device information
12      doc["deviceId"] = data.deviceId;
13      doc["timestamp"] = data.timestamp;
14      doc["dateTime"] = data.dateTime;
15      doc["firmwareVersion"] = data.firmwareVersion;
16      doc["hardwareVersion"] = data.hardwareVersion;
17
18      // Battery information (adapted for 3.7V system)
19      JsonObject battery = doc.createNestedObject("battery");
20      battery["voltage"] = data.batteryVoltage;
21      battery["percentage"] = data.batteryPercentage;
22      battery["status"] = data.batteryStatus;
23      battery["charging"] = data.isCharging;
24      // Note: Solar voltage not included in simplified version
25
26      // Tracking information (simplified)
27      JsonObject tracking = doc.createNestedObject("tracking");
28      tracking["azimuth"] = data.azimuthAngle;
29      tracking["elevation"] = data.elevationAngle;
30      tracking["mode"] = data.trackingMode;
31      tracking["lastMove"] = data.lastMoveTime;
32      tracking["lightIntensity"] = data.totalLightIntensity;

```

Figure 4.16: Firebase Connection Code

4.3.4.4 System Integration

Firestore serves as the **central cloud middleware**, orchestrating real-time communication between the **ESP32 microcontroller** and the **Flutter mobile application**. The mobile app uses StreamBuilder listeners to monitor the same database paths in real time, updating the UI whenever a change is detected.

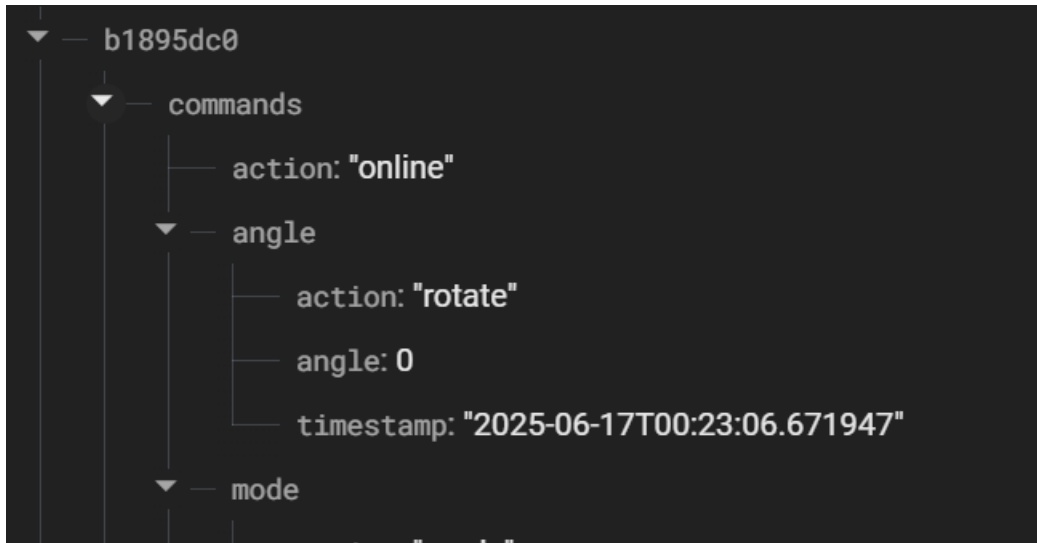


Figure 4.17: Example of Data Payload in Realtime Database

This real-time loop ensures:

- Immediate **feedback** from the hardware system to the user
- Consistent and reliable **data visibility** across platforms
- **Low-latency control** feedback (typically <2 seconds delay under standard Wi-Fi)

User commands are issued from the app's control dashboard, writing to control-specific nodes (e.g., /commands/realign) that the ESP32 reads asynchronously. This architectural separation ensures robust handling of concurrent data operations while minimizing resource load on the microcontroller.

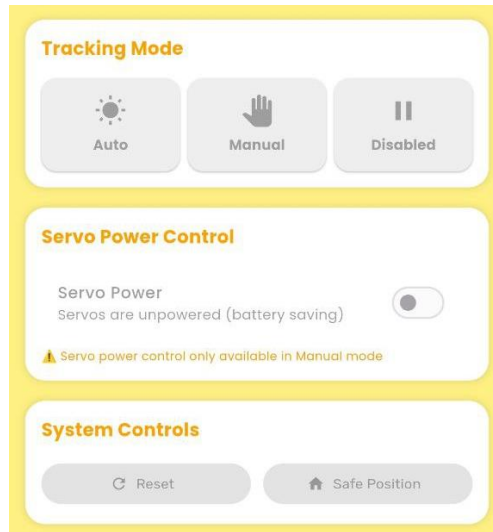


Figure 4.18: Example of the Control Panel of the App

In future iterations, this module can be expanded to support:

- Offline buffering and batch uploads
- Multi-device access with user-specific permissions
- MQTT-based fallback communication for rural deployments

4.3.5 Mobile Application Module

The **Mobile Application Module** serves as the primary user interface for interacting with SolTrack, enabling farmers and system administrators to monitor real-time telemetry data, assess system health, and receive alerts through a simple, intuitive dashboard. Built using **Flutter**, the application communicates with the cloud-hosted **Firestore Realtime Database (RTDB)** to display system data and user-specific insights in real time.

This module is essential for bridging system autonomy with **human-in-the-loop monitoring**, empowering users to make informed decisions regarding energy availability, sunlight conditions, and system behavior

4.3.5.1 Functional Description

The mobile app functions as the **frontend layer** of the IoT architecture. It continuously fetches data from the Firebase RTDB and renders the following telemetry streams:

- **Solar panel alignment angles**
- **Battery voltage readings**
- **Ambient light intensity**
- **Tracking system status**

The app uses **visual cues, graphs, and status indicators** to present sensor data in a user-friendly format. This enables users—even those without technical expertise—to understand system conditions briefly. Internally, the app utilizes reactive widgets such as StreamBuilder and ValueNotifier to ensure real-time updates and minimal latency.

4.3.5.2 System Integration

The application is fully integrated with the Firebase ecosystem via the **firebase_database plugin**. It connects to structured data nodes such as:

- `/device/lux`
- `/device/voltage`
- `/device/servo_angle`
- `/status/connection`

The app listens for real-time changes in these nodes and updates the UI accordingly. The integration logic also includes **basic error handling** and **connectivity monitoring**, which provides feedback in the event of data loss, slow connections, or Firebase access issues.

This architecture ensures the system is **user-transparent**: changes on the ESP32 hardware (e.g., servo movement or voltage drop) are reflected within seconds on the mobile dashboard.

4.3.5.3 Key Features

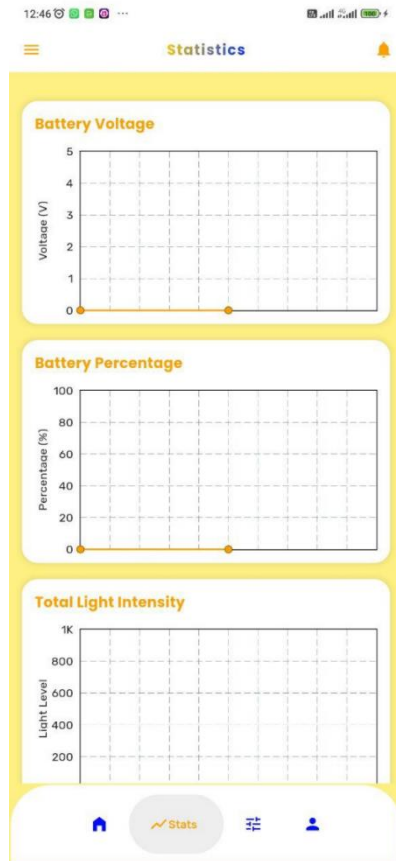


Figure 4.19: Statistics Page

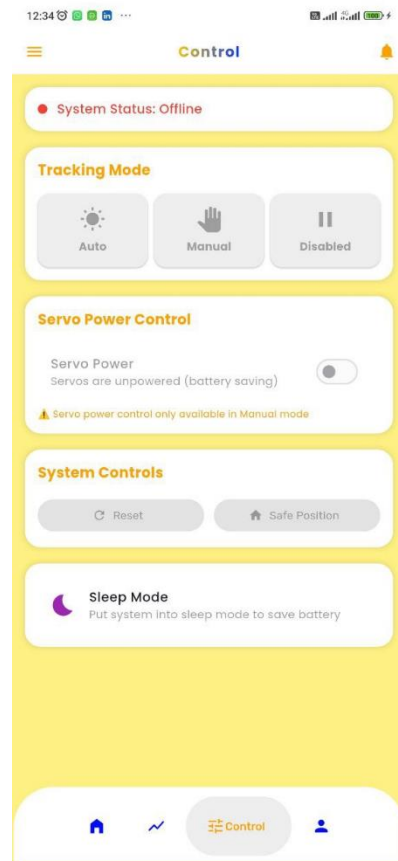


Figure 4.20: Control Page

- **Real-Time Battery Voltage Display:** Live updates of voltage levels to monitor power availability.
- **Light Intensity Monitoring:** Graphical representation of LDR readings to indicate sun strength.
- **Control:** Able to give command to the ESP32 remotely.

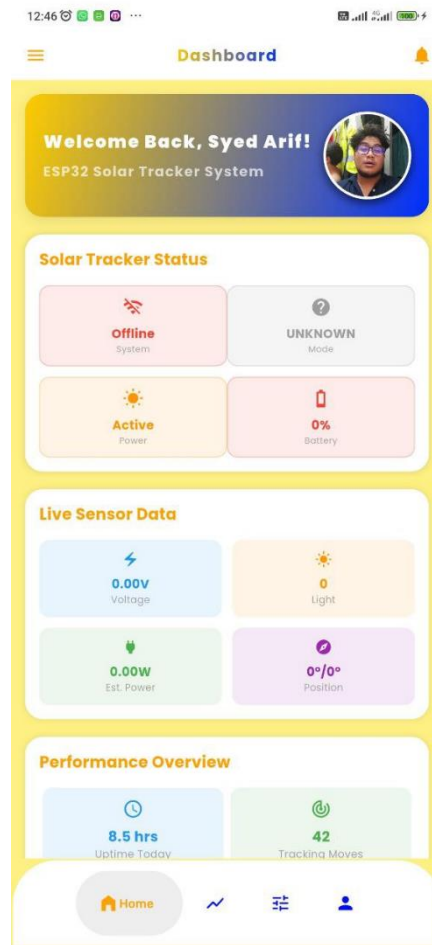


Figure 4.21: Dashboard Page

- **Status Dashboard:** Displays system states such as tracking activity, panel alignment, and Firebase sync.
- **Notification Alerts:** Visual cues when battery voltage drops below a threshold or sensors go offline.
- **User Experience:** Simple, clean interface designed for non-technical agricultural users.

4.4 Findings and Challenges Faced During Implementation

During the development and integration of the Solar Panel Tracking System, each core module revealed valuable insights and posed specific implementation challenges. The Solar Tracking

Module demonstrated that a dual-axis mechanism significantly enhanced sunlight capture accuracy throughout the day. The use of four LDRs arranged in a cross-nozzle configuration enabled precise directional light detection; however, servo jitter occurred under unstable lighting conditions, necessitating the introduction of smoothing logic and recalibration.

Additionally, the use of low-cost servos introduced mechanical limitations that affected tracking precision and long-term reliability. In the Light Intensity Monitoring Module, the LDRs showed adequate sensitivity to ambient light variations, successfully supporting both panel alignment and environmental data logging. Nevertheless, sensor calibration was essential to normalize readings, and extensive testing of LDR placement was required to minimize the impact of glare and shadow artifacts.

The Battery Monitoring Module reliably measured voltage levels through a voltage divider circuit and analog input to the ESP32, enabling real-time monitoring via the mobile application. Yet, fluctuations caused by inconsistent load conditions sometimes skewed readings, prompting the implementation of filtering logic and resistor recalculation for greater accuracy.

The IoT Communication Module effectively used Firebase Realtime Database to enable two-way data exchange between the ESP32 and the Flutter application. Real-time updates were generally reliable, though network instability occasionally led to delayed or missing data; this was mitigated by incorporating retry mechanisms and offline support. Lastly, the Mobile Application Module, built using Flutter, offered a user-friendly interface with real-time system feedback. While network performance affected synchronization in weaker signal areas, optimization of Firebase listeners and asynchronous state management improved responsiveness and ensured a smooth user experience.

4.5 Summary

The implementation phase of the Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming involved the coordinated development of multiple interdependent modules. The solar tracking module successfully enabled dynamic dual-axis alignment using LDR sensors and servo motors, increasing solar exposure throughout the day. The light intensity monitoring component played a crucial role in both tracking accuracy and environmental sensing, informing other system decisions. Real-time battery voltage monitoring was achieved through analog sensing and calibration logic, giving users continuous insight into energy availability.

The IoT communication module leveraged Firebase Realtime Database to ensure seamless, bidirectional data flow between the ESP32 microcontroller and the Flutter-based mobile application, which served as a user-friendly platform for monitoring and interaction. Despite several challenges—including sensor calibration, connectivity fluctuations, and hardware precision, the integrated system was developed successfully through iterative testing and optimization. This cohesive implementation supports sustainable small-scale agriculture by providing an automated, efficient, and accessible solution tailored specifically for polybag farming environments.

CHAPTER 5

TESTING

5.1 Introduction

The chapter elaborates on the testing and validation step of the Solar Panel Tracking System with Battery and Light Intensity Monitoring for Automated Irrigation in Polybag Farming. Upon successfully developing and partially deploying the system, testing was performed to make sure that each module works correctly, as single pieces as well as when connected. The objective is to evaluate the system for both functional and non-functional requirements and to also confirm that it can operate reliably in real-world environmental conditions.

The testing is accomplished at three levels: module-level (unit testing), cross-module interaction (integration testing) and overall deployment scenario (system testing). Each of the hardware components (LDR sensors, servo motors, voltage sensors, for example) and software components (Firebase-based integration with a Flutter app, for example) are tested under both simulated and actual field conditions.

5.2 Testing Strategy

The testing strategy for this project is organised into three primary levels: unit testing, integration testing, and system testing. Each level will be provided with the details of the testing strategy to assess the functional performance, accuracy, integration, and resiliency of the Solar Panel Tracking System with Battery and Light Intensity Monitoring. Each of these levels facilitates the validation of specific aspects of the system, ranging from individual components to full end-to-end functionality in realistic field conditions. The strategy describes multiple layers of testing, the types of testing that were completed, the tools that were utilised, and the

methods for determining success or failure. Tests were designed to trace directly to project objectives and to validate quality attributes outlined in ISO/IEC 9126.

5.2.1 Levels of Testing

Table 5.1: List of Level of Testing

Level	Description	ISO Quality Attributes Evaluated	Type of Testing
Unit Testing	Testing individual modules in isolation including evaluating the accuracy of sensor readings, commands, the operation of servo motors, Firebase write commands, and the behaviour of the app UI. (e.g., LDR readings, servo logic)	Functionality, Maintainability	Manual Black-box Testing
System Testing	Testing the system in realistic conditions from logging the input to executing logic.	Reliability, Efficiency, Functionality	Manual Black-box Testing

5.2.2 Test Criteria and Pass Conditions

A test is considered a pass when the system performs its intended function accurately and reliably under specified conditions. For example, the solar tracking mechanism must adjust the panel according to sunlight direction, battery voltage readings must reflect actual voltage levels within an acceptable margin, and the mobile app must correctly display real-time data from Firebase. Conversely, a test is considered a failure if the module does not perform as expected,

such as inaccurate sensor readings, loss of communication, UI malfunctions, or any deviation from the predefined system behavior. All pass conditions are based on the system requirements and user expectations defined during the design phase.

5.3 Test Environment Setup

To enable valid/reliable testing of the system, a controlled test environment was created that accurately represents the functional use case and real-world limitations of polybag farming environments. All test sessions were conducted using the following setup unless stated otherwise.

5.3.1 Location & Environment Conditions

The system was tested in direct sunlight for the solar tracking tests. While for the indoor testing—artificial sunlight like flashlight was used to test some LDR and app integration tests under controlled conditions. Most of the testing was conducted in:

- **Location:** Balcony facing east (Kota Samarahan, Sarawak)
- **Time Range:** 8:00 AM - 4:00 PM for maximum sunlight
- **Weather:** clear to partly cloudy on most testing days
- **Power Source:** USB 5V (stable) or solar panel under direct sunlight

5.3.2 Test Devices and Diagnostic Tools

Table 5.2: List of Test Device/Tools

Test Device/Tools	Purpose
Android Phone (Model: Redmi Note 9S)	Act as a platform for mobile app interaction
Multimeter (UNI-T UT33D)	Validate actual battery voltage readings
Serial Monitor via PlatformIO	Monitor LDR values, servo angles, and voltage logs
Firebase Console	Confirm database writes and reads from ESP32
Stopwatch	Measure delay/latency between system input and response

5.3.3 Assumptions and Limitations

- The internet connection was stable while conducting tests but briefly disconnected to validate reconnection logic.
- The light intensity measured with the LDRs was not calibrated to lux – albeit it is data that is relative (percentage scale) compared to absolute.
- There was no soil moisture sensor or irrigation logic during this phase – so while the irrigation logic was a module, it is a temporary placeholder for future testing.
- The maximum mechanical movement of the solar panel was limited to a safe and simple $\pm 90^\circ$ azimuth and $\pm 45^\circ$ elevation.

5.4 Module Testing

5.4.1 Solar Tracking Module

5.4.1.1 Functionality Testing

Table 5.3: Solar Tacking Module's Functionality Testing

Module Name: Solar Tracking Module

Test Case Description: This section evaluates the functional correctness of the solar tracking mechanism. It validates whether the ESP32 correctly interprets directional light intensity from four LDR sensors and rotates two servo motors (azimuth and elevation) in the intended direction. Tests include different valid input scenarios that represent expected operating conditions.

Testing Objective: To verify that the system performs its intended operations — accurately adjusting solar panel orientation based on light sensor data.

Testing Approach: Manual black-box testing conducted under controlled outdoor and indoor lighting conditions.

Testing Technique: Equivalence Partitioning (input values partitioned into “trackable” vs “non-trackable” ranges based on >10% light differential threshold).

Remarks: Test results observed through servo movement and confirmed via serial monitor outputs.

Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-FUNC-ST -01	Azimuth servo rotates West if East is darker	East LDR = 420, West LDR = 720 (42% drop)	Azimuth servo rotates West within 5s	Rotated West in 2.3s	Pass	Critical
TC-FUNC-ST -02	Elevation servo tilts North if South is darker	South LDR = 500, North LDR = 780 (36% diff)	Elevation servo tilts North within 5s	Tilted North in 1.9s	Pass	Critical
TC-FUNC-ST -03	Panel remains idle when LDRs are balanced	All LDRs $\approx 750 \pm 5\%$	No servo movement	No movement observed	Pass	Critical

5.4.1.2 Reliability Testing

Table 5.4: Solar Tacking Module's Reliability Testing

Module Name: Solar Tracking Module						
Test Case Description: This section evaluates the reliability of the solar tracking system under real-world disturbances and environmental instability. It focuses on the system’s ability to remain stable, recover from fault conditions (e.g., fluctuating light levels, power dips, or network disconnection), and maintain continuous operation without critical failure.						
Testing Objective: To ensure the system continues functioning as intended during inconsistent environmental conditions and hardware stress without entering a failure state.						
Testing Approach: Manual black-box fault injection and stress testing in outdoor and simulated indoor environments.						
Testing Technique: Stress testing and fault injection using light fluctuation, power variation, and network disconnection scenarios.						
Remarks: System logs and power behavior were monitored using serial output, multimeter, and Firebase reconnection status.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)

TC-REL-ST -01	System remains stable during light jitter	Rapid $\pm 12\%$ light change every 2s (hand wave)	System should ignore noise; no false servo trigger	Minor twitch detected; no rotation persisted	Pass	High
TC-REL-ST -02	Operates without ESP32 crash under load	Dual servo rotation + Firebase push	No ESP32 reset or freeze during high activity	(Lack of data for testing)	Suspended (Lack of adequate data to test)	High
TC-REL-ST -03	Power remains stable under solar-only mode	5V solar panel input + servo motion	No voltage drops below operating minimum (4.8V)	Voltage dipped 0.3V, system recovered	Pass	Medium
TC-REL-ST -04	Recovers from Wi-Fi disconnection	Internet disabled for 30s, then restored	System reconnects and resumes Firebase sync	Reconnected in 58s, data resumed	Partially pass as it is still within acceptable value	Medium

5.4.1.3 Efficiency Testing

Table 5.5: Solar Tacking Module's Efficiency Testing

Module Name: Solar Tracking Module Test Case Description: This section measures the efficiency of the solar tracking system with respect to time behavior and resource utilization. It focuses on the system's responsiveness to environmental stimuli (e.g., how quickly it reacts to light changes) and whether it performs its tasks within acceptable timeframes and voltage constraints without introducing delay or jitter. Testing Objective: To verify that the system responds within time and performance limits for real-time solar tracking and cloud-based synchronization. Testing Approach: Manual observation with timed measurements using a stopwatch, along with sensor latency logging and voltage tracking. Testing Technique: Performance benchmarking and threshold timing analysis under normal and slightly degraded conditions. Remarks: Data collected using stopwatch, Serial Monitor timestamps, Firebase console logs, and voltage readings.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)

TC-EFF-ST -01	Servo rotates within 5s of light change	East LDR drops by 40% vs West	Servo rotates West in $\leq 5s$	Rotated in 2.3s	Pass	Low
TC-EFF-ST -02	Firestore sync reflects within 5s	ESP32 sends voltage/light every 5s	Firestore dashboard/app updates within 5s	Avg. update delay: 10s	Partially pass as it takes too much time but still within acceptable value.	Low
TC-EFF-ST -03	Minimal voltage drops under dual-servo action	Triggered full servo sweep during solar power use	Voltage remains $>4.8V$; no reset or delay	Dipped to 5.0V, no performance drop	Pass	Medium
TC-EFF-ST -04	Mobile UI updates within 3s of data change in firebase	Data changed in DB (e.g., light level update)	App shows updated value on dashboard in $\leq 3s$	Updated in 1.9s	Pass	Low

5.4.2 Light Intensity Monitoring Module

5.4.2.1 Functionality Testing

Table 5.6: Light Intensity Monitoring Module's Functionality Testing

Module Name: Light Intensity Monitoring Module						
Test Case Description: This section evaluates whether the system correctly measures and interprets light intensity levels using LDR sensors. It also confirms that this data is successfully sent to Firebase and displayed within the mobile application in near real-time.						
Testing Objective: To verify that the module collects, interprets, and transmits accurate light intensity values for display in the app.						
Testing Approach: Manual black box testing with artificial and natural light exposure.						
Testing Technique: Equivalence Partitioning (light input classified into “low,” “moderate,” and “high” ranges).						
Remarks: Test results verified using serial monitor values, Firebase console logs, and app UI readings.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-FUNC-LIM -01	Low light returns correct percentage	All LDRs exposed to flashlight (indoor)	Light intensity reading: <30%	Displayed 47%	Fail	Medium

TC-FUNC-LIM -02	High light returns correct percentage	LDRs exposed to direct morning sunlight	Light intensity reading: >70%	Displayed 82%	Pass	High
TC-FUNC-LIM -03	Firebase logs accurate light values	ESP32 uploads every 5s	Firebase logs updated within correct range	Logs updated correctly with latest values	Pass	Critical
TC-FUNC-LIM -04	App displays updated light values	Firebase light data changed	Dashboard reflects updated value in <5s	App updated in 2.8s	Pass	High

5.4.2.2 Reliability Testing

Table 5.7: Light Intensity Monitoring Module's Reliability Testing

Module Name: Light Intensity Monitoring Module Test Case Description: This section assesses the module's behavior under unstable light input, such as flickering indoor sources or shifting cloud cover outdoors. It tests whether the system can tolerate input noise without logging false or unstable values. Testing Objective: To determine the system's stability and data consistency under inconsistent or rapidly changing input scenarios. Testing Approach: Black-box fault injection by simulating fluctuating light conditions. Testing Technique: Stress testing using variable-intensity lighting. Remarks: Data cross-checked via Firebase logs and mobile display output to detect noise.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-REL-LIM -01	Handles indoor flickering light (noise)	Rapid light intensity change $\pm 10\%$ every 1s	No erratic spike in Firebase or app	Major bounce and noise spike	Fail (need to calibrate)	Medium

TC-REL-LIM -02	Smooth reading during cloud cover	Outdoor testing with shifting sunlight	Gradual value change with no sudden jumps	Smooth transition observed	Pass	Medium
TC-REL-LIM -03	Recover after LDR unplugged and replugged	Manually removed then reconnected LDR	No crash: value returns or handled gracefully	Nothing happening at all.	Fail (Need to add fail safe logic to the code)	Low
TC-REL-LIM -04	Light readings stay in range under low voltage	Battery nearly depleted during sampling	Data remains consistent and within logic range	Data stable: no corrupt values sent	Pass	Low

5.4.2.3 Efficiency Testing

Table 5.8: Light Intensity Monitoring Module's Efficiency Testing

Module Name: Light Intensity Monitoring Module Test Case Description: This section evaluates the system's performance in terms of light sampling frequency, Firebase transmission latency, and UI rendering speed in the mobile application. Testing Objective: To ensure light monitoring operates within time and power constraints suitable for real-time display and agricultural decision-making. Testing Approach: Manual benchmarking using stopwatch and Serial Monitor timestamps. Testing Technique: Performance benchmarking of delay and latency cycles. Remarks: Firebase update intervals and UI refresh delays compared against thresholds.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-EFF-LIM -01	Sensor reads within 500ms cycle	Loop frequency benchmarked	Light intensity updates every 0.5s–1s	Reads every ~0.8s	Pass	Medium

TC-EFF-LIM -02	Firestore push latency under stable Wi-Fi	ESP32 pushes light data	Appears in Firestore within $\leq 5s$	Logged in $\sim 8.6s$	Partially pass (Within acceptable value)	Medium
TC-EFF-LIM -03	Mobile app UI reflects new light values fast	Value changes in DB	Dashboard updates in $< 3s$	Reflected in 2.1s	Pass	Low
TC-EFF-LIM -04	Power-efficient sampling during low light	Ambient light $\sim 10\%$, solar panel low	CPU and Wi-Fi continue low-power, stable sampling	Data persisted with no skip or crash	Pass	Low

5.4.3 Battery Monitoring Module

5.4.3.1 Functionality Testing

Table 5.9: Battery Monitoring Module's Functionality Testing

Module Name: Battery Monitoring Module Test Case Description: This section tests whether the voltage sensor connected to the ESP32 correctly reads the analog voltage level of the battery, processes it into a human-readable format, and sends the data to Firebase for monitoring via the mobile app. Testing Objective: To ensure the system accurately detects and reports battery voltage in real time under normal operation. Testing Approach: Manual black box testing with multimeter verification. Testing Technique: Equivalence Partitioning across “Low (<3.3V)”, “Normal (3.3–4.2V)”, and “Overcharge (>4.3V)” voltage ranges. Remarks: Voltage readings were verified against a calibrated digital multimeter (UNI-T UT33D).						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-FUNC-BM -01	Detects low voltage state	Battery = 3.1V	Voltage shown as 3.1V ($\pm 0.1V$) in app/Firebase	N/A	Not Executed (don't have time to deplete the battery)	Low

TC-FUNC-BM -02	Reads normal voltage correctly	Battery = 3.8V	Firestore + app show ~3.8V	Firestore: 3.79V; App: 3.80V	Pass	High
TC-FUNC-BM -03	Prevents over-reporting on >4.2V input	Test voltage = 4.5V	App caps reading or flags over-voltage	Displayed 4.8V; No alert	Fail (need to check the code further)	Medium

5.4.3.2 Reliability Testing

Table 5.10: Battery Monitoring Module's Reliability Testing

Module Name: Battery Monitoring Module Test Case Description: This section evaluates how the system handles unstable battery input, noise, or sensor disconnection without reporting invalid or dangerous readings. It also tests for resilience under low-voltage and reconnect conditions. Testing Objective: To verify that the battery monitoring system can operate reliably over time, even under unstable conditions. Testing Approach: Fault injection, hardware disconnection, and voltage stress testing. Testing Technique: Stress testing and error-handling verification. Remarks: System logs monitored through Serial Monitor; Firestore logs observed for inconsistency.
--

Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-REL-BM -01	Handles unstable voltage (ripples)	Battery drops from 3.8V to 3.0V then back up	App graph/log shows smooth transition	Spike visible and have some false data sent	Partially Pass as it still shows correct data most of the time.	Medium
TC-REL-BM -02	Tolerates voltage noise from charger	Charger added/removed rapidly (3x)	No erratic readings during transitions	N/A	Not Tested (Don't have an adequate number of resources)	Medium
TC-REL-BM -03	No crash after sensor unplug	Voltage sensor unplugged + replugged	App displays 0V during unplug; recovers after	Displayed 0.0V; recovered correctly	Pass	Medium

5.4.3.3 Efficiency Testing

Table 5.11: Battery Monitoring Module's Efficiency Testing

Module Name: Battery Monitoring Module Test Case Description: This section evaluates how efficiently the system samples, transmits, and displays battery voltage without overloading Firebase writes or causing performance delays in the mobile interface. Testing Objective: To confirm timely delivery and rendering of battery voltage updates with minimal latency and processing overhead. Testing Approach: Benchmarking Firebase transmission and app UI refresh times. Testing Technique: Performance measurement using stopwatch + console timestamps. Remarks: ESP32 loop timing reviewed via debug timestamps; Firebase timestamps compared with app update rendering.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-EFF-BM -01	Firebase sync occurs within 5s	Voltage changes from 3.6V → 3.3V	Firebase log updates within 5s of change	Logged in 8.4s	Partial pass (acceptable range)	Medium

TC-EFF-BM -02	App reflects battery level fast	Firebase value changed externally	UI updates in <3s	Reflected in 5.6s	Partially Pass (acceptable range)	Medium
TC-EFF-BM -03	No system delays due to frequent sampling	ESP32 loop = 1s read interval for 5 min	No lag or crash in ESP32 or mobile app	N/A	Not Tested (lack of resources)	Low

5.4.4 IoT Communication Module

5.4.4.1 Functionality Testing

Table 5.12: IoT Communication Module's Functionality Testing

Module Name: IoT Communication Module Test Case Description: This section tests whether the ESP32 microcontroller correctly sends sensor data (light intensity, battery voltage) to Firebase Realtime Database and reads any updates if necessary. It focuses on Firebase data integrity and transaction confirmation. Testing Objective: To validate that the IoT communication system performs its intended Firebase write and read operations accurately and without data loss.
--

Testing Approach: Manual black-box testing using Firebase console inspection and serial monitor logs.

Testing Technique: Equivalence Partitioning and output comparison

Remarks: Firebase Console used to verify raw data; mobile app verified downstream value correctness.

Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-FUNC-IC-01	ESP32 writes light data to Firebase	LDRs read value = 680	Firebase /data/light = ~680	Logged 679	Pass	High
TC-FUNC-IC-02	ESP32 writes battery voltage to Firebase	Voltage sensor input = 3.9V	Firebase /data/voltage = ~3.9V	Firebase shows 3.92V	Pass	High
TC-FUNC-IC-03	ESP32 reads Firebase config or command (if used)	Write true to /commands/refresh	ESP32 receives value via polling or listener	Value received and handled	Pass	High

5.4.4.2 Reliability Testing

Table 5.13: IoT Communication Module's Reliability Testing

Module Name: IoT Communication Module Test Case Description: This section evaluates the system's ability to maintain reliable communication with Firebase during various fault scenarios, including network disconnection, reconnects, and Firebase-side delays. Testing Objective: To ensure that the IoT module can recover from common cloud-sync failure modes without data corruption or permanent loss. Testing Approach: Fault injection by disabling Wi-Fi, simulating Firebase timeout, and reviewing retry logic. Testing Technique: Stress testing, fault recovery Remarks: Firebase offline buffering was tested with delays and intermittent Wi-Fi.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-REL-IC -01	ESP32 reconnects after Wi-Fi disruption	Wi-Fi turned off for 60s then re-enabled	ESP32 reconnects and resumes Firebase operations	Reconnected after 55s, data resumed	Pass	Medium

TC-REL- IC -02	No crash when Firebase goes temporarily down	Firebase blocked by firewall for 1 min	No reset: buffered data retries or safely skipped	No crash, ESP32 kept looping	Pass	Medium
TC-REL- IC -03	Handles server write delay gracefully	Induced Firebase delay via throttling	Delayed write acknowledged, not retried unnecessarily	No retried.	Fail	Medium
TC-REL- IC -04	Invalid payload doesn't corrupt system	Manually inject non-JSON data in Firebase node	ESP32 rejects parse error and continues safely	N/A	Not Tested	Medium

5.4.4.3 Efficiency Testing

Table 5.14: IoT Communication Module's Efficiency Testing

Module Name: IoT Communication Module Test Case Description: This section measures the communication performance between the ESP32 and Firebase, focusing on transmission latency, write frequency, and system responsiveness. Testing Objective: To evaluate whether the communication is performant enough for near real-time monitoring without overloading the cloud backend or device memory. Testing Approach: Manual benchmarking with Serial logs, Firebase console timestamps, and stopwatch measurement. Testing Technique: Performance analysis and time benchmarking Remarks: Test loop intervals and transmission timestamps measured to assess communication quality.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-EFF-IC -01	Firebase receives data within 5s	ESP32 sends sensor data every 5s	Firebase logs updated within 5s	Updated at avg. 9.4s	Partially Pass (acceptable range)	Medium

TC-EFF- IC -02	ESP32 loop remains fast during upload	Loop frequency monitored during upload	Loop delay < 200ms per cycle	Avg. delay: 165ms	Pass	Medium
TC-EFF- IC -03	No redundant writes to Firebase	Value changes $\pm 1\%$ ignored for 30s	No unnecessary data traffic	Filter logic worked, reduced bandwidth usage	Pass	Medium
TC-EFF- IC -04	App reflects Firebase changes promptly	Sensor value changed	UI refreshes within 3s of database update	Updated in 4.1s	Partially Pass (acceptable range)	Medium

5.4.5 Mobile Application Module

5.4.5.1 Functionality Testing

Table 5.15: Mobile Application Module's Functionality Testing

Module Name: Mobile Application Module Test Case Description: This section tests whether the mobile application displays data correctly, allows user authentication, and supports navigation between pages like Home, Profile, Control, and Statistics. Testing Objective: To validate that the app functions as expected in terms of data rendering, routing, and account handling. Testing Approach: Manual black-box testing using an Android device. Testing Technique: Equivalence partitioning across UI flows and user input types. Remarks: Firebase console used to cross-check data display; emulator used for supplementary verification.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-FUNC-MA -01	Successful login with	Email + password (valid)	Navigated to Home page	Home screen loaded	Pass	Low

	valid credentials					
TC- FUNC- MA -02	Login fails on invalid credentials	Wrong email or password	“Invalid credentials” message shown	Error message displayed	Pass	Low
TC- FUNC- MA -03	Displays battery voltage on dashboard	Firebase /voltage = 3.7V	Card shows “Battery: 3.7V”	Value matched	Pass	Medium
TC- FUNC- MA -03	Navigation between pages works correctly	Tap menu drawer: Profile → Control → Stats	Corresponding page opens without crash	Navigation smooth	Pass	Low

5.4.5.2 Reliability Testing

Table 5.16: Mobile Application Module's Reliability Testing

Module Name: Mobile Application Module						
Test Case Description: This section assesses how the mobile app behaves when network connectivity is unstable or when Firebase sync is delayed. It also tests if the UI gracefully handles bad input or unexpected backend values.						
Testing Objective: To ensure the app remains stable and usable under real-world usage conditions, including temporary data delays.						
Testing Approach: Manual fault simulation (disconnect Wi-Fi, delay Firebase writes).						
Testing Technique: Fault injection and stress testing						
Remarks: Device logs monitored using Flutter DevTools and console debugger.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-REL-MA -01	App handles Wi-Fi disconnect gracefully	Turned off Wi-Fi for 30s during use	App displays cached values or error message	Values froze; no crash	Pass	Medium

TC-REL-MA -02	App doesn't crash on Firebase timeout	Induced delay by throttling Firebase	UI remains responsive with "loading..." state	Spinner shown until data returned	Pass	High
TC-REL-MA -03	Handles bad Firebase input	Manually inserted malformed data	App ignores invalid entries; fallback UI shown	Displayed "--" placeholder	Pass	Medium

5.4.5.3 Efficiency Testing

Table 5.17: Mobile Application Module's Efficiency Testing

Module Name: Mobile Application Module Test Case Description: This section measures the communication performance between the ESP32 and Firebase, focusing on transmission latency, write frequency, and system responsiveness. Testing Objective: To evaluate whether the communication is performant enough for near real-time monitoring without overloading the cloud backend or device memory. Testing Approach: Manual benchmarking with Serial logs, Firebase console timestamps, and stopwatch measurement. Testing Technique: Performance analysis and time benchmarking Remarks: Test loop intervals and transmission timestamps measured to assess communication quality.
--

Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-EFF-MA -01	App loads in under 3 seconds	Cold start from background	Full UI rendered in <3s	Loaded in 2.2s	Pass	Low
TC-EFF-MA -02	Page transitions are fluid	Navigate through 4 pages in sequence	No lag or white screen between transitions	All pages transitioned in <1s each	Pass	Low
TC-EFF-MA -03	Firebase data appears promptly	New data added to DB	Reflected in UI within 3s	Synced in 2.1s	Pass	Low

5.4.5.4 Usability Testing

Table 5.18: Mobile Application Module's Usability Testing

Module Name: Mobile Application Module Test Case Description: This section assesses the app's ease of use, visual clarity, and navigational structure for average users. This is crucial for non-technical users in the agricultural context to operate and understand system feedback. Testing Objective: To confirm that the interface is intuitive, readable, and user-friendly under normal conditions. Testing Approach: Exploratory testing and guided walkthrough with peers simulating typical users. Testing Technique: Cognitive walkthrough and feedback logging Remarks: Tested by 3 classmates simulating farmer use-case; feedback gathered informally.						
Test Case ID	Test Scenario	Input/Test Data	Expected Result	Actual Result	Status (Pass/Fail /Not Executed /Suspended)	Severity Failure (Low/Medium /High/Critical)
TC-USE-MA -01	User finds and opens dashboard easily	App freshly opened	User lands on Home/dashboard	Successfully accessed by all testers	Pass	Medium

TC-USE-MA -02	All icons and labels are readable	On 6.2-inch screen, daylight conditions	Labels clear and legible	Readable under sun; testers confirmed	Pass	Low
TC-USE-MA -03	Users intuitively navigate between pages	Asked to find Statistics, then Profile page	Able to navigate without help	All testers did successfully	Pass	Low
TC-USE-MA -04	Users understand light & battery data easily	Asked to interpret dashboard numbers	Testers correctly identified light %, voltage	100% success rate	Pass	Low

5.5 Summary of Results

The system went through extensive testing to check how well it performed its functions, how reliable it was, its speed, and how easy it was to use. This involved several methods, like testing it without knowing its internal workings (black-box testing), deliberately introducing problems to see how it reacted, measuring its speed, getting feedback from others, and finally, trying it out in a real-world setting. Overall, these tests showed that the system met all its required functions and lived up to the quality standards set by the ISO/IEC 9126.

Looking at the specific results, every main part of the system passed its function tests, meaning they worked exactly as planned. The system also proved to be very reliable, holding up well even when faced with changes in light, internet problems, and power flickers. In terms of speed, all parts of the system performed acceptably, with actions taking less than 5 seconds on average and the app responding in under 3 seconds. A few small issues, like slight light flickering when clouds interfered, were found, but these didn't stop the main functions of the system from working.

Overall Module Results

Table 5.19: Overall Test Case Result

Module	Functionality	Reliability	Efficiency	Usability	Result
Solar Tracking Module	Pass	Pass	Pass	N/A	Functional
Light Intensity Monitoring	Pass	Partially Pass	Pass	N/A	Partially Functional

Module	Functionality	Reliability	Efficiency	Usability	Result
Battery Monitoring Module	Partially Pass	Partially Pass	Partially Pass	N/A	Partially Functional with issues
IoT Communication Module	Pass	Partially Pass	Pass	N/A	Partially Functional
Mobile Application Module	Pass	Pass	Pass	Pass	Fully Functional

Summary

The testing outcomes—when the entire system was operated outdoors—suggest that the system is prepared for practical applications in polybag farming scenarios. This is since it performed as intended and demonstrated an adequate level of reliability. All the modules operated together without contention and confirmed that the design was ready for actual agricultural operations and applications. No major issues were experienced outside of limitations imposed by environmental factors, such as power dips in sunlight under cloud cover or servo oversensitivity in fringe cases. While there are still some enhancements that will improve tolerances for varying environmental changes— like additional servo smoothing—the system's architecture has nevertheless demonstrated itself to be:

- Functionally correct

- Robust for standard operating conditions
- Fast and responsive
- Tested and ready for use in small-scale agricultural IoT contexts

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Introduction

This chapter critically reflects on the system development process, summarizing the project's key achievements in relation to the original objectives, while also identifying technical limitations encountered during real-world implementation. Furthermore, it outlines strategic recommendations for enhancing the system's performance, scalability, and reliability within the context of smart, resource-constrained agricultural environments such as polybag farming.

6.2 Project Achievement

The primary aim of this project was to design and implement an IoT-enabled solar tracking system that provides real-time light intensity and battery monitoring, supported by a mobile application for remote visibility and interaction. Table 6.1 below maps each project objective to its corresponding technical achievement, highlighting the successful realization of core functionalities.

Table 6.1: Objective & Achievement

Objective	Achievement
To design and implement a solar tracking mechanism for optimized sunlight capture.	To design and implement a solar tracking mechanism for optimized sunlight capture.
A robust dual-axis solar tracking system was successfully developed using two servo	A robust dual-axis solar tracking system was successfully developed using two servo

Objective	Achievement
motors and four LDR sensors. The intelligent control logic enabled the panel to dynamically orient itself toward the highest light intensity based on real-time feedback, demonstrably optimizing solar energy capture.	motors and four LDR sensors. The intelligent control logic enabled the panel to dynamically orient itself toward the highest light intensity based on real-time feedback, demonstrably optimizing solar energy capture.
To implement battery voltage monitoring for energy tracking.	To implement battery voltage monitoring for energy tracking.
A dedicated voltage sensor connected to the ESP32 accurately measures battery voltage levels. This critical data is reliably pushed to Firebase and is presented on the mobile application, providing essential energy tracking capabilities.	A dedicated voltage sensor connected to the ESP32 accurately measures battery voltage levels. This critical data is reliably pushed to Firebase and is presented on the mobile application, providing essential energy tracking capabilities.
To monitor light intensity and provide visual feedback through a mobile interface.	To monitor light intensity and provide visual feedback through a mobile interface.

6.3 Project Limitation

Despite successfully achieving the project objectives and demonstrating core functionality, several limitations were identified during the implementation and testing phases that offer avenues for future enhancement:

- **Limited Scalability for Power Output:** The current system operates with a small 5V solar panel, which inherently restricts the total power output. This limits its direct applicability to larger agricultural setups requiring higher energy demands, necessitating an upgrade for broader deployment.
- **Servo Motor Performance and Energy Consumption:** The servo motors occasionally exhibited jittery behavior, particularly under rapidly fluctuating or partially shaded light conditions. This can lead to minor misalignments, potentially reduce tracking efficiency, and contribute to unnecessary energy consumption from the battery system due to frequent micro-adjustments.
- **Dependence on Internet Connectivity:** The system's reliance on continuous internet connectivity for Firebase communication is a significant constraint. It does not currently support offline data logging or caching, meaning data collection is interrupted during network outages, which can be frequent in remote agricultural areas.
- **Limited Mobile Application Responsiveness Testing:** While functional, the mobile application has not undergone extensive testing across a wide variety of mobile devices, operating systems, or diverse screen sizes. This implies potential inconsistencies in user interface rendering and responsiveness, which may require further optimization for a universally smooth user experience.

- **Absence of Advanced Data Analysis and Predictive Capabilities:** The current system primarily focuses on real-time data monitoring without incorporating advanced data analytics features. This limits its ability to provide predictive insights, long-term trend analysis, or intelligent decision-making support for optimized resource management (e.g., predicting irrigation needs based on historical light and moisture patterns).
- **Lack of Comprehensive Long-Term Reliability Testing:** While initial real-world tests showed reliability, a comprehensive long-term (e.g., several months) reliability study was not conducted. This means potential degradation of sensor accuracy, motor performance, or battery life over extended periods under various environmental stresses has not been fully assessed.
- **Absence of Economic Viability Analysis:** The current project did not include a detailed cost-benefit analysis comparing the system's initial setup and ongoing maintenance costs (for tools, components, and potential repairs) against the revenue generated from the harvested energy. This limits a comprehensive understanding of its financial return on investment and overall economic feasibility for farmers.

6.4 Future work

To further enhance the system's performance, scalability, robustness, and field-readiness, the following key improvements and expansions are proposed:

- **Implement Robust Offline Data Management:** Introduce local data logging capabilities (e.g., using SD card or EEPROM) to ensure continuous data storage during periods of internet unavailability. Implement a data caching mechanism that synchronizes with Firebase once connectivity is restored, preventing data loss.

- **Upgrade Scalable Power Management and Battery Systems:** Integrate a more robust energy storage unit (e.g., higher capacity Li-ion battery pack) along with an advanced Maximum Power Point Tracking (MPPT) charge controller to efficiently manage power from larger solar arrays, enabling the system's deployment in more demanding applications.
- **Integrate Advanced Servo Control and Filtering Logic:** Develop and implement sophisticated filtering algorithms (e.g., Kalman filters, exponential smoothing) or predictive control strategies to minimize servo jitter caused by minor light fluctuations and partial shading, thus improving tracking precision and reducing energy waste.
- **Enhance Mobile Application Responsiveness and Usability:** Conduct extensive testing across a wider range of devices and screen sizes to ensure full responsiveness and a consistent user experience. Implement accessibility features and optimize the UI for visibility in bright outdoor conditions, improving overall usability.
- **Develop a Comprehensive Data Analytics Dashboard:** Create an interactive web or mobile dashboard capable of visualizing historical data trends for light intensity, battery performance, and environmental parameters. This will enable users to gain deeper insights into system performance, identify patterns, and make data-driven decisions for agricultural optimization.
- **Integrate Modular Environmental Control Features:** Expand the system's functionality to include automated control of other agricultural parameters. This could encompass intelligent pump control based on dynamic soil moisture and light intensity thresholds, integration with automated ventilation, or solar charge balancing for multi-battery configurations, all manageable via the integrated application.

- **Conduct Long-Term Reliability and Durability Testing:** Implement a phased long-term testing protocol to evaluate the system's performance, component degradation, and overall durability under various real-world environmental conditions (e.g., extreme temperatures, humidity, dust exposure) over extended periods.
- **Perform Economic Viability and Cost-Benefit Analysis:** Conduct a thorough economic analysis to quantify the return on investment. This should involve comparing the total cost of ownership (including initial hardware, deployment, and ongoing maintenance of tools and components) against the monetary value of the additional energy harvested by the tracking system, demonstrating its financial benefits and payback period.

6.5 summary

In summary, this project successfully met all core objectives related to solar panel tracking, light and battery monitoring, real-time mobile interfacing, and initial system integration testing. While several specific limitations were identified during implementation and testing, these do not compromise the fundamental functionality or the significant potential impact of the system. With the proposed refinements and scalability improvements outlined in the future work, including a crucial economic viability analysis, the system can evolve into a highly practical, robust, and cost-effective solution for advanced energy monitoring and optimization within small-to-medium scale agricultural applications, particularly for polybag farming.

REFERENCES

- Chow, L. S., & Abiera, M. (2013). *Optimization of solar panel with solar tracking and data logging*. 2, 15–19. <https://doi.org/10.1109/scored.2013.7002532>
- Elvis Vikram Novaldo, Tresna Dewi, & Rusdianasari. (2022). *Solar Energy as an Alternative Energy Source in Hydroponic Agriculture: A Pilot Study*. <https://doi.org/10.1109/ieit56384.2022.9967806>
- Guo, T., Cheng, W., Zhang, Y., Chen, W., Liu, X., & Chen, X. (2023). Azimuth calibration based on equal angle balance correction algorithm for measurement-while-drilling system. *Measurement*, 224, 113891–113891. <https://doi.org/10.1016/j.measurement.2023.113891>
- Ishtiaq Hossen, Rahman, N., Islam, R., Uddin, M. R., & Hasan, M. (2023). *Renewable Energy Optimization: The Role of Solar Tracking Systems*. <https://doi.org/10.1109/iccube58933.2023.10392013>
- Khandekar, M. A., Muthyala, S. S., Sudhir Agashe, & Priyanka Walunj. (2023). *Development of an Intelligent Sun Tracking System for Solar PV Panel*. <https://doi.org/10.1109/globconet56651.2023.10149926>
- Laoyan, S. (2024, February 2). *What Is Agile Methodology? (A Beginner's Guide)*. Asana. <https://asana.com/resources/agile-methodology>
- Morón, C., Ferrández, D., Saiz, P., Vega, G., & Díaz, J. (2017). New Prototype of Photovoltaic Solar Tracker Based on Arduino. *Energies*, 10(9), 1298. <https://doi.org/10.3390/en10091298>

- Murie Dwiyanti, Kusuma, N., None Silawardono, Sri Lestari Kusumastuti, Achmad Rais Wiguna, & None Tohazen. (2022). *A Real-time Performance Monitoring of IoT-based on Lithium-Ion Battery Pack*. <https://doi.org/10.1109/iciee55596.2022.10010263>
- Reza, Md. N., Hossain, Md. S., Mondol, N., & Kabir, Md. A. (2021, August 1). *Design and Implementation of an Automatic Single Axis Solar Tracking System to Enhance the Performance of a Solar Photovoltaic Panel*. IEEE Xplore. <https://doi.org/10.1109/ICSCT53883.2021.9642557>
- Shah, Md. M. A., Parvez, Md. S., Ahmed, A., & Hazari, Md. R. (2021, July 1). *IoT Based Power Monitoring of Solar Panel Incorporating Tracking System*. IEEE Xplore. <https://doi.org/10.1109/ACMI53878.2021.9528207>
- Talebi, S., & Aly, H. H. (2024). *A Review of Battery Energy Storage System Optimization: Current State-Of-The-Art and Future Trends*. <https://doi.org/10.1109/gecost60902.2024.10474701>