



Faculty of Computer Science and Information Technology

SMART CLASSROOM MANAGEMENT SYSTEM IN EDUCATIONAL INSTITUTION

Siti Nur Farhana Binti Hamdan

**Bachelor of Computer Science with Honours
(Network Computing)**

2024

SMART CLASSROOM MANAGEMENT SYSTEM IN EDUCATIONAL INSTITUTION

SITI NUR FARHANA BINTI HAMDAN

This project is submitted in partial fulfilment of the requirements for the
degree of Bachelor of Computer Science with Honours
(Network Computing)

Faculty of Computer Science and Information Technology
UNIVERSITY MALAYSIA SARAWAK

2024

SMART CLASSROOM MANAGEMENT SYSTEM IN EDUCATIONAL INSTITUTION

SITI NUR FARHANA BINTI HAMDAN

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dengan Kepujian
(Pengkomputeran Rangkaian)

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2024

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE SMART CLASSROOM MANAGEMENT SYSTEM IN EDUCATIONAL INSTITUTION.

ACADEMIC SESSION: 2024/2025

SITI NUR FARHANA BINTI HAMDAN

(CAPITAL LETTERS)

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

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

☐

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

☐

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

☐

/

UNRESTRICTED



(AUTHOR'S SIGNATURE)

Permanent Address

NO. 22 Jalan Cengal 5/3,
Taman Malawati Jaya,
45000 Kuala Selangor, Selangor Darul Ehsan.

Date: 21/7/2025

Validated by


SELEVIAWATI TARMIZI
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

(SUPERVISOR'S SIGNATURE)

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

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

DECLARATION OF ORIGINALITY

I hereby declare that this research together with all its content is none other than that of my own work, with consideration of the exception of research-based information and relative materials that were adapted and extracted from other resources, which have evidently been quoted or stated respectively

Signed,



.....

SITI NUR FARHANA BINTI HAMDAN

Faculty of Computer Science and Information Technology

University Malaysia Sarawak.

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Madam Seleviawati Binti Tarmizi, for her invaluable guidance, patience and support throughout my Final Year Project journey. Her advice and encouragement have been instrumental in completing this work. I am also sincerely thankful to my parents for their endless love, prayers and unwavering support, which gave me the strength to keep going. A heartfelt thank you goes to my friends, who stood by me through every challenge and provided encouragement and motivation when I needed it most. Additionally, I would like to extend my appreciation to myself for the dedication, perseverance and hard work I put into overcoming obstacles and achieving this milestone. Completing this project has been a valuable experience and I am truly grateful to everyone who contributed to this journey.

Abstract

The Smart Classroom Management System in Educational Institutions automation tackles the critical challenge of energy inefficiency in classrooms, where light and fan are frequently left on when the rooms are unoccupied. This report presents a comprehensive overview of the design, implementation and testing of a system that utilize Internet of Things (IoT) technology and Arduino-based sensors to monitor classroom occupancy and automatically control electricity energy usage. Resulting the system that successfully automates the control of classroom appliances, improving efficiency and contributing to energy management. During the implementation phase, the system underwent rigorous testing to assess its functionality, including usability and performance evaluations, ensuring alignment with the initial design objectives. The significance of this project lies in its potential to serve as a practical example for promoting energy-efficient practices in schools, fostering awareness among students and staff about responsible electrical energy use.

Abstrak

The Smart Classroom Management System in Educational Institutions dapat menangani masalah kritikal pembaziran tenaga elektrik di dalam bilik darjah, di mana lampu dan kipas sering dibiarkan terpasang apabila bilik darjah tidak digunakan. Penyelidikan ini memberikan gambaran menyeluruh tentang reka bentuk, pelaksanaan, dan sistem yang memanfaatkan teknologi *Internet of Things (IoT)* dan sensor berasaskan Arduino untuk memantau kehadiran di dalam bilik darjah dan secara automatik mengawal penggunaan tenaga elektrik. Hasilnya, sistem ini berjaya mengautomatikkan kawalan lampu dan kipas di dalam bilik darjah, meningkatkan kecekapan dan menyumbang kepada pengurusan tenaga yang baik. Semasa fasa pelaksanaan, sistem ini menjalani ujian untuk menilai fungsinya, termasuk kebolegunaan dan prestasi untuk memastikan kesemua objektif telah dicapai. Kepentingan projek ini terletak pada potensinya untuk berfungsi sebagai contoh praktikal dalam mempromosikan amalan penjimatan tenaga elektrik di sekolah, serta meningkatkan kesedaran di kalangan pelajar dan kakitangan mengenai penggunaan tenaga elektrik dengan bertanggungjawab.

Table of Contents

Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Scope	3
1.4 Aims and Objectives.....	4
1.5 Brief Methodology	4
1.5.1 Planning.....	5
1.5.2 Iteration Phase	5
1.5.3 Launch.....	7
1.6 Significance of the Project	7
1.7 Project Schedule.....	7
1.8 Expected Outcome	8
1.9 Conclusion.....	8
Chapter 2: Literature Review	9
2.1 Introduction	9
2.2 Review of similar existing system	9
2.2.1 Lutron Vive Wireless Lighting Control System.....	9
2.2.2 Honeywell Smart Energy Management System.....	11
2.2.3 Siemens Desigo CC Energy Management System.....	12
2.2.4 Comparison with the Proposed Solution.....	13
2.3 Review of tools and technologies.....	14
2.3.1 Arduino Nano 33 IoT	14
2.3.2 Passive Infrared Sensors (PIR).....	16
2.3.3 DHT11–Temperature and Humidity Sensor.....	17
2.3.4 Real-Time Clock (RTC) Module.....	18
2.3.5 LEDs and Resistor.....	18
2.3.6 Connecting Wires and Breadboards	19
2.3.7 MG90S Micro Servo Motor	20
2.3.8 LDR (Light Dependent Resistor)	20
2.3.9 Arduino IoT Cloud	21
2.4 Conclusion.....	22
Chapter 3: Requirement Analysis and Design.....	23
3.1 Introduction	23
3.2 Agile Methodology.....	23

3.2.1 Plan.....	24
3.2.2 System Architecture Design	25
3.2.3 Product Backlog	26
3.2.4 Use Cases and Scenarios for Smart Classroom System.....	27
3.2.5 Use Case Description	28
3.2.6 Activity Diagram	34
3.2.7 Sequence Diagram.....	37
3.3 Iteration Phase	39
3.4 Launch.....	42
3.4.1 UI Design	42
3.5 Conclusion.....	43
Chapter 4: Implementation and Testing	45
4.1 Introduction	45
4.2 Tools and Platforms Used.....	45
4.2.1 Arduino IDE	46
4.2.2 Arduino IoT Cloud	47
4.3 Hardware Modules	51
4.4 Cloud Dashboard Controls	56
4.5 Implementation Process	57
4.5.1 Motion Detection Logic	58
4.5.2 Environmental Condition Response.....	60
4.5.3 Time-Based Automation.....	61
4.5.4 Cloud Connectivity and Manual Override Functionality	62
4.6 Conclusion.....	63
Chapter 5: System Testing.....	64
5.1 Introduction	64
5.2 Testing and Validation	64
5.2.1 Functional Testing	64
5.2.2 Timing Accuracy Test.....	65
5.3 Test Case.....	66
5.3.1 Test Case for Detect Student Entering Module	67
5.3.2 Test Case for Detect Classroom Empty Module	68
5.3.3 Test Case for Check Time with RTC Module	68
5.3.4 Test Case for Turn on Light and fan Module	69
5.3.5 Test Case for Show System Status on LED	70
5.3.6 Test Case for System Activity Logging and Manual Control	70

5.3.7 Test Case for Control Lights Based on Ambient Light Module.....	71
5.4 User Testing.....	73
5.5 Conclusion.....	80
Chapter 6: Conclusion and Future Work	81
6.1. Introduction	81
6.2 Achievement.....	81
6.3 Contribution	82
6.4 Limitation	83
6.5 Future Work.....	84
6.6 Conclusion.....	86
References	87
Appendix	89

List of tables

Table 2.1. Comparison of all similar systems with the proposed project.....	13
Table 3.1. Detect Student Entering.....	28
Table 3.2. Detect Classroom Empty.....	29
Table 3.3. Check Time with RTC.....	30
Table 3.4. Turn on Light and fan.....	31
Table 3.5. Show System Status on LED.....	31
Table 3.6. System Activity Logging and Manual Control.....	32
Table 3.7. Control Lights Based on Ambient Light	33
Table 3.7. Iteration 1: Motion Detection to Control Light and fan.	40
Table 3.8. Iteration 2: Automatic Turn-Off After Class Hours.....	40
Table 3.9. Iteration 3: Status Display via LED.....	41
Table 4.1. Libraries Installed in Arduino IDE.....	46
Table 4.2. Variables setup in Arduino Cloud.....	49
Table 4.3. Widgets Used for Dashboard.....	51
Table 4.4. List of the Components	52
Table 4.5. Pin Connection Overview.....	53
Table 4.6. Two Main Environmental Sensors.	55
Table 5.2. Detect Student Entering Module.	67
Table 5.3. Detect Classroom Empty Module.	68
Table 5.4. Check Time with RTC Module.....	68
Table 5.5. Turn on Light and fan Module.....	69
Table 5.6. Show System Status on LED.....	70
Table 5.7. System Activity Logging and Manual Control.....	70
Table 5.8. Control Lights Based on Ambient Light Module.....	71
Table 6.1. Objectives Achievement.....	81
Table 6.2. Contribution Overview.....	83
Table 6.3. Key Limitations.....	84
Table 6.4. Future Improvement.....	85
Table 6.5. Priority of Future Work.....	86

List of figures

Figure 1.1. Agile Methodology.	5
Figure 1.2. Gantt Chart of Project Schedule.	8
Figure 2.1. Lutron Vive system.	10
Figure 2.2. Honeywell Smart Power for Buildings.	11
Figure 2.3. Desigo CC Energy Management System.	12
Figure 2.4. Arduino Nano.	14
Figure 2.5. Arduino Uno.	15
Figure 2.6. PIR Sensors.	16
Figure 2.7. PIR Detection.	16
Figure 2.8. DHT11	17
Figure 2.9. Real-Time Clock (RTC) Module.	18
Figure 2.10. LEDs and Resistor.	18
Figure 2.11. Connecting Wires and Breadboards.	19
Figure 2.12. MG90S Micro Servo Motor.	20
Figure 2.13. Light Dependant Resistor.	20
Figure 2.14. Arduino Cloud.	21
Figure 3.1. Agile Methodology.	24
Figure 3.2. System Architecture Design.	26
Figure 3.3. Use Case Diagram	27
Figure 3.4. Activity Diagram.	36
Figure 3.5. Sequence Diagram.	38
Figure 3.6. Flow Chart of Iteration Phase.	39
Figure 3.7. Web View Chart of Dashboard Arduino Cloud	42
Figure 3.8. Mobile View Chart of Dashboard Arduino Cloud	43
Figure 4.1. Board and Port Configuration via Arduino IDE.	46
Figure 4.2. Serial Monitor on Arduino IDE.	47
Figure 4.3. Devices on Arduino Cloud.	48
Figure 4.4. Setup Tab on Arduino Cloud.	49
Figure 4.5. Sketch Tab on Arduino Cloud.	50
Figure 4.6. Implementation on Web View of Dashboard Arduino Cloud.	56
Figure 4.8. PIR Sensor with Component Label.	58
Figure 4.10. Serial Monitor Output when Motion is Detected.	59
Figure 4.11. Arduino Cloud Dashboard when Motion is Detected.	60
Figure 4.13. RTC Configure Working Hours.	61
Figure 4.14. Arduino Cloud Dashboard when Motion is Detected.	62

Figure 4.15. Manual Control on Arduino Cloud Dashboard.....	63
---	----

Chapter 1: Introduction

1.1 Introduction

The classroom defines as a place in a school or college where teachers instruct groups of students (Wolff et al., 2020). Arduino is an open-source electronics platform that is easy to use, combining both hardware and software. It can be control by writing instructions for its microcontroller using the Arduino programming language and the Arduino Software (IDE) (Kondaveeti et al., 2021). The concept of smart automation has been applied in so many ways such as Smart Buildings, Smart Health or Smart Agriculture.

Today, managing energy is a big challenge around the world. With rising energy demands, it's more important than ever to use power in a smart and sustainable way. Poor management of electrical efficiency often leads to high costs, wasted resources and a larger carbon footprint, contributing to environmental problems like climate change (Yang et al., 2023). This highlights the need for solutions that can reduce waste and improve the usage of electricity in daily life.

Educational institutions, particularly in classrooms, face challenges in efficiently managing energy (Daly et al., 2022). As Malaysia's population grows, the education sector is expanding quickly. Schools, teachers and both public and private universities now have more students and staff. This growth means they are using significant amounts of energy for activities inside the classroom, whether for learning or for engaging, enjoyable events held in the classroom (Nasrudin et al., 2021). Light and fan are frequently left on even when classrooms are empty, leading to wasted energy and higher costs. By improving energy management, schools and universities can reduce unnecessary energy use and lower expenses.

Implementing smart systems to automatically control light and fan when rooms are unoccupied can help these institutions become more sustainable and set a positive example for students and the community. The Smart Classroom Management System offers a practical

solution to reduce energy waste in classrooms. By using IoT (Internet of Things) technology and Arduino based sensors, this system detects classroom occupancy and automatically manages light and fan. This approach optimizes energy usage and creates a comfortable and responsive environment.

Furthermore, the Smart Classroom Management System also enables remote monitoring and control through web and mobile applications, utilizing Arduino Cloud platforms and the IoT Remote application., enabling staff to manage classrooms in real time. This provides an effective way for institutions to track electricity usage and respond quickly to any issues, ultimately helping schools and universities reduce costs and support sustainability goals.

The paper is divided in following sections. In Chapter 1, the Background or Introduction of this project are briefly discussed in this section. After that in Chapter 2, the similar system related to the smart classroom are reviewed. Section 3, the chosen methodology of this project is presented in detail. In Chapter 4, the implementation of the Smart Classroom system is described, including the tools used, hardware setup and cloud integration. Chapter 5 presents the testing process, covering functional tests, timing accuracy and real-world user validation through detailed test cases. Chapter 6 concludes the report by summarizing the achievements, discussing limitations and proposing future improvements for the system.

1.2 Problem Statement

Educational Institutions are currently facing significant challenges related to energy management in classrooms. One major issue is energy waste, as many classrooms leave light and fan on when unoccupied. This inefficient usage leads to substantial energy waste, increasing operational costs and contributing to a larger carbon footprint, which causes bad environmental and climate change (Daly et al., 2022).

Another critical challenge is the lack of automation in energy management practices (Mustaffa & Kudus, 2022). Despite advancements in technology, most educational institutions do not utilize automated systems for managing energy consumption. This results in missed opportunities to optimize resource usage and implement sustainable practices within classroom environments. Without these systems, energy management remains a manual process, which can be both time-consuming and ineffective.

Finally, current energy management practices often lack real-time monitoring and control capabilities. Without access to data on energy usage and classroom conditions, staff cannot effectively manage energy consumption or respond promptly to issues. This insufficient monitoring leads to ongoing inefficiencies and increased costs. A centralized networked system that gathers and shares energy data could simplify monitoring and enhance control over energy consumption.

1.3 Scope

The Smart Classroom Management System is designed primarily for educational institutions, particularly classroom environments that aim to optimize energy usage. This system can also be adapted for office or organizational settings that wish to reduce electricity costs by automating the control of fans and lights based on occupancy status.

This project will use an Arduino and IoT (Internet of Things) technologies to build an automated system for managing classroom resources, specifically focusing on controlling fans and lights. Motion sensors will detect the presence of individuals and based on occupancy data, the system will control fans and lights, turning them on or off as needed.

1.4 Aims and Objectives

The aim of this project is to develop an intelligent, sensor-based system that automates the control of classroom light and fan based on classroom occupancy to optimize energy usage in educational institutions.

The following objectives that will achieve by the end of this project:

- i. To develop a wireless system that uses sensors for monitoring the classroom conditions like occupancy status.
- ii. To deploy low-cost sensors and IoT (Internet of Things) technology and send real-time updates of electrical appliances in the classroom.
- iii. To create a detailed installation guide specifically for educational institutions, ensuring easy deployment of the system in classrooms.
- iv. To thoroughly test the performance of the prototype and ensure its functionality meets the project's objectives.

1.5 Brief Methodology

The Agile methodology is used in this project to develop the Smart Classroom Management System efficiently by breaking down the system into a smaller iterative phase. This approach allows continuous evaluation and ensures that the functionalities can be tested and refined before the entire system is completed (Guerrero-Ulloa et al., 2023). Figure 1.1 illustrates the Agile methodology, which consists of planning, iterative development cycles and final deployment. Each iteration involves designing, developing, testing, deploying and reviewing, ensuring flexibility and continuous improvement throughout the project lifecycle.

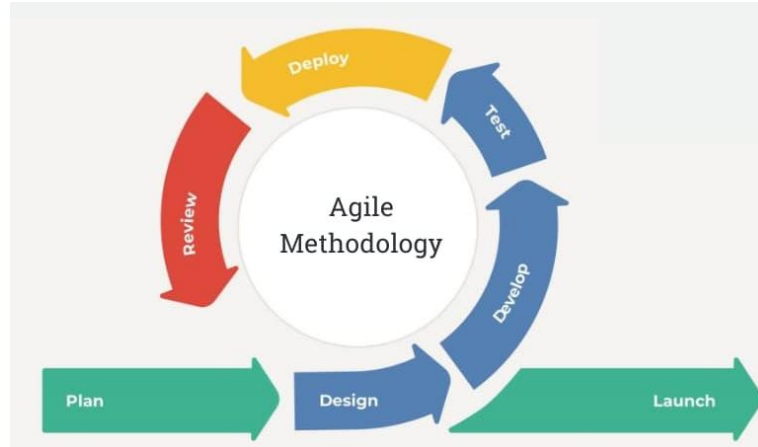


Figure 1.1. Agile Methodology.

1.5.1 Planning

The planning phase defines the objectives and scope of the Smart Classroom Management System. Essential components, including PIR sensor, DHT11 temperature sensor, light-dependent resistor (LDR), relay module, RTC module and Arduino Nano 33 IoT, are identified. Additionally, a backlog of tasks is created, prioritizing features such as occupancy-based automation, temperature-triggered fan control, ambient light-based LED control and system status indicators.

1.5.2 Iteration Phase

The project is developed in multiple iterations, where each cycle delivers functional components of the system. This phase ensures that key features are implemented incrementally, allowing for testing, feedback and adjustments before moving to the next iteration.

1. Design

In each iteration, the system design is refined, focusing on integrating all sensors for automated control. The architecture is planned to ensure seamless communication between components and cloud integration for monitoring.

2. Developing

The development phase involves coding and integrating hardware components. For example, one iteration may focus on motion detection using the PIR sensor, while others implement DHT11-based temperature control for the fan, involve RTC-based scheduling for automatic power management and LDR-based lighting adjustment to turn off LEDs when sufficient ambient light is detected. System status logging and cloud variable syncing are also incorporated.

3. Testing

Each feature undergoes testing to verify its functionality and compatibility with other components. This includes testing temperature thresholds for fan activation, LDR light sensitivity for LED control, motion detection accuracy and RTC synchronization for scheduling.

4. Deployment

Once tested, the components are deployed. Initial iterations deploy basic functionality, such as sensor-based control, while later iterations introduce refinements like cloud data logging and enhanced user interactions with the system.

5. Review

After deployment, feedback from testing and observations is analyzed. Adjustments are made based on performance, ensuring that the system meets efficiency and reliability.

1.5.3 Launch

Upon completion of all iterations, the Smart Classroom Management System is finalized. The system ensures automated energy management, providing an efficient solution for smart classroom automation environments. A user manual is prepared to guide future users in operating and maintaining the system.

1.6 Significance of the Project

The following list highlights the significance of the project:

- i. The system demonstrates automated control of fan and light by using motion detection, temperature sensing and ambient light measurement potentially contributing to more efficient energy usage by reducing unnecessary operation of electrical appliances.
- ii. This project shows how IoT (Internet of Things) technology can be used in schools to help manage resources more efficiently and encourage eco-friendly practices.
- iii. Students and staff will gain awareness of energy-efficient practices, encouraging responsible resource use within classroom settings.

1.7 Project Schedule

Figure 1.2 shows the expected project schedule for Final Year Project in Gantt chart form. The estimated duration of the project is 350 days, which is start from 07/10/2024 until 01/07/2025.

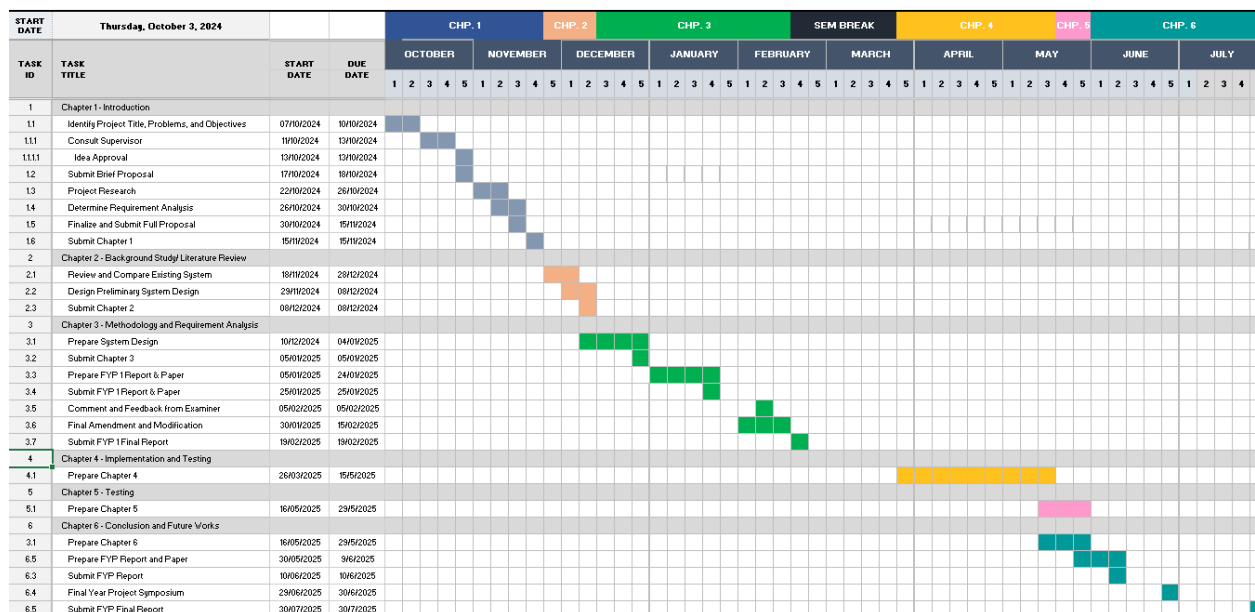


Figure 1.2. Gantt Chart of Project Schedule.

1.8 Expected Outcome

An expected outcome for this project is a fully working prototype of the Smart Classroom Management System, capable of automatically controlling light and fan based on real-time occupancy, temperature and ambient light levels. This prototype will serve as a demonstration of the system's effectiveness in system's ability to manage the electricity based on the occupancy status in the classroom.

1.9 Conclusion

In conclusion, this chapter introduces the Smart Classroom Management System, which aims to address the energy management challenges faced by educational institutions. Also, highlighting the lack of real time monitoring systems and limited usage of IoT in this era. By implementing IoT technology and Arduino based sensors, the system will automate the control of light and fan, reduce energy waste and lower operational costs.

Chapter 2: Literature Review

2.1 Introduction

This chapter explores existing work and resources relevant to the project. Section 2.2 reviews similar systems currently in use, highlighting their features, benefits and limitations. This analysis provides insight into the existing system and identifies areas that require improvement. Section 2.3 examines the tools and technologies selected for this project, explaining their functionalities and justifying their suitability for the proposed solution.

2.2 Review of similar existing system

Several smart systems have been developed to enhance energy efficiency and automate environmental control in classrooms. These systems vary in scope, cost and functionality. Below is a review of three existing systems relevant to classroom management, focusing on their features, limitations and how they compare them to the proposed solution.

2.2.1 Lutron Vive Wireless Lighting Control System

The Lutron Vive system shown in Figure 2.1 provides an innovative wireless solution for lighting control, specifically designed to improve energy efficiency in educational and commercial environments. This system integrates sensors that detect occupancy and natural light, automatically adjusting lighting levels to match the needs of space, which helps minimize energy waste (Lutron Electronics Co., n.d.).

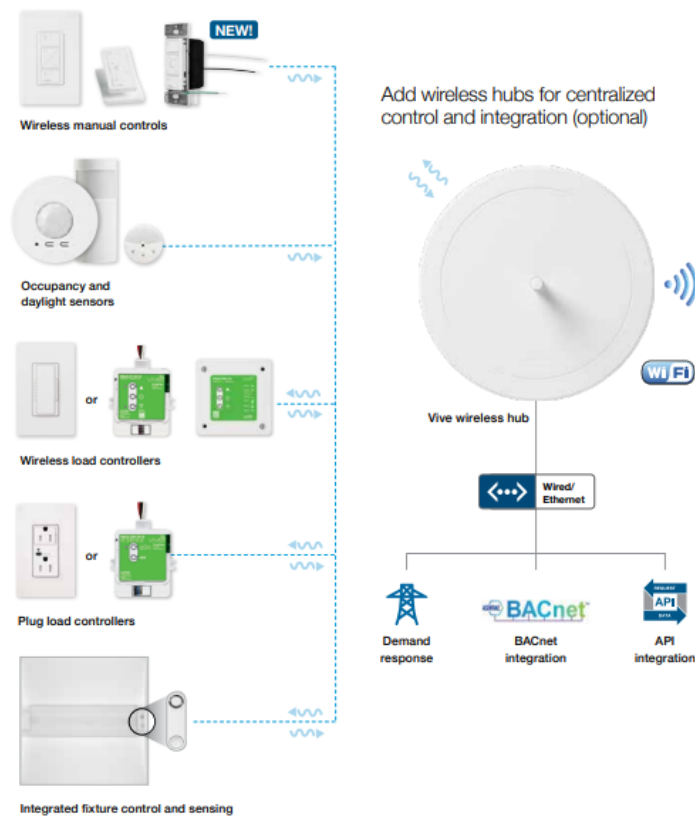


Figure 2.1. Lutron Vive system.

One specific example of Vive in action is at Franklin & Marshall College, where it was implemented to enhance energy efficiency across multiple building environments. The system's ease of installation and scalability allowed for seamless integration across the campus, where lighting could be adjusted remotely and efficiently (Lutron Electronics Co., n.d.). While Vive is highly effective for lighting automation, it does not extend to other systems such as HVAC or fans, limiting its scope to more comprehensive building automation setups. Additionally, the cost of the system may be prohibitive for smaller installations or organizations with tighter budgets.

2.2.2 Honeywell Smart Energy Management System

The Honeywell Smart Energy Management System (SEMS) provides an advanced solution for managing energy use in large buildings as shown in Figure 2.2. It integrates controls for lighting, HVAC and fans, utilizing occupancy sensors to dynamically adjust energy consumption based on real-time usage patterns. This results in enhanced energy efficiency, particularly in large environments such as campuses and office buildings (Honeywell, 2020).



Figure 2.2. Honeywell Smart Power for Buildings.

Similarly, the proposed Smart Classroom Management System aims to optimize energy use but in a more focused and simplified manner, addressing the specific needs of classrooms by controlling essential elements like lighting and fans. While Honeywell's SEMS is designed for larger infrastructures, the Smart Classroom Management System is streamlined for cost-effectiveness and practicality in smaller spaces. Compared to Honeywell SEMS Yefi et al. (2024), emphasize that integration with efficient APIs is key for ensuring seamless communication between devices, a feature essential for scaling smart systems to different environments.

2.2.3 Siemens Desigo CC Energy Management System

The Siemens Desigo CC Energy Management System is a comprehensive solution for managing building energy use. Based on Figure 2.3 it integrates various systems, including heating, cooling and lighting and collects data through sensors and controllers. The system monitors parameters like temperature, energy consumption and solar radiation, which are displayed on a user-friendly interface for real-time management.

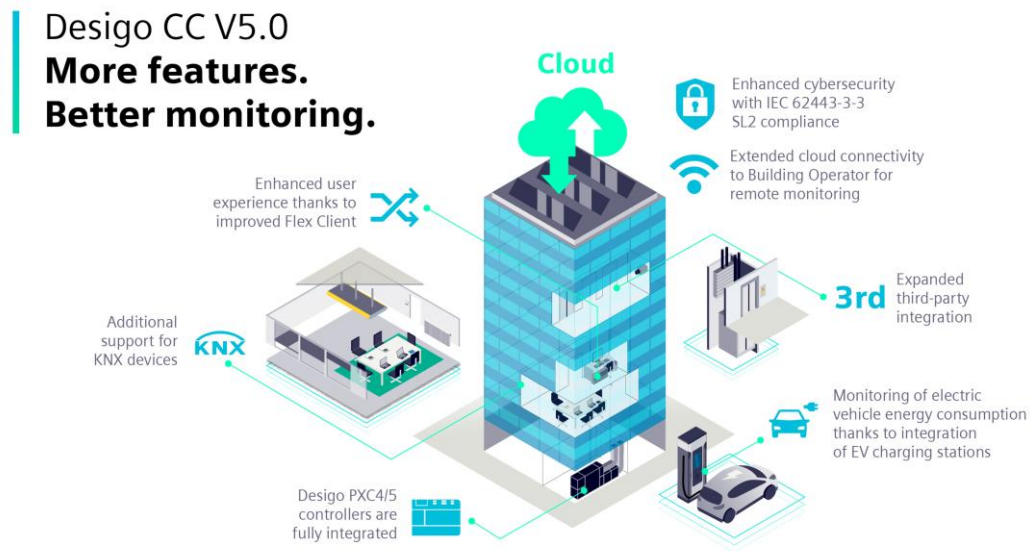


Figure 2.3. Desigo CC Energy Management System.

For example, in the office building in Tomislavgrad, Desigo CC manages energy sources like a heat pump and photovoltaic power plant. It tracks key data such as air temperature, thermal energy and electrical power consumption, helping to optimize energy efficiency (Medjugorac et al., 2020). While this system is highly effective for large buildings, its complexity and broad scope may not be necessary for smaller environments like classrooms, which require more basic energy management features.

2.2.4 Comparison with the Proposed Solution

As shown in Table 2.1, the proposed Smart Classroom Management System stands out by offering a simpler, more cost-effective solution tailored specifically for classrooms, in contrast to the more complex and expensive systems like Lutron Vive, Honeywell SEMS and Siemens Desigo CC.

Table 2.1. Comparison of all similar systems with the proposed project.

Feature	Lutron Vive	Honeywell SEMS	Siemens Desigo CC	Smart Classroom System
Scope	Lighting	Lighting, HVAC, Fans	Lighting, HVAC, Heating, Cooling	Lighting, Fans
Primary Focus	Lighting efficiency	Energy efficiency in large buildings	Energy management in large buildings	Energy efficiency in classrooms
Complexity	Simple	Complex	Very Complex	Simple and focused
Installation Ease	Easy	Moderate	Moderate	Very Easy
Cost	Moderate to High	High	High	Low

Target Environment	Commercial, Educational	Large buildings, Campuses	Large infrastructures	Classrooms
Limitations	No HVAC or fans	Overly complex for small spaces	Too complex for classrooms	Focused on classroom needs
Key Strength	Easy to install, scalable	Manages multiple systems	Advanced data integration	Tailored for classrooms, cost-effective

2.3 Review of tools and technologies

To implement a smart classroom management system, various tools and technologies must be considered, including hardware components and software platforms. In this section, the key tools and technologies used in the development of the proposed system are being reviewed, focusing on those that contribute to its functionality, simplicity and cost-effectiveness.

2.3.1 Arduino Nano 33 IoT

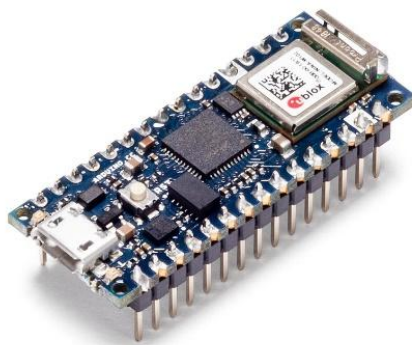


Figure 2.4. Arduino Nano.

As shown in Figure 2.4 the Arduino Nano with integrated Wi-Fi which manages all inputs and outputs. This microcontroller is compact, making it ideal for installations where space is limited, such as in a classroom prototype. Its built-in Wi-Fi module is particularly beneficial for IoT applications, enabling remote data collection and monitoring. Based on Figure 2.5 the Arduino Uno is also known for its low power consumption, cost-efficiency and ease of programming through the Arduino IDE.



Figure 2.5. Arduino Uno.

Compared to Arduino Uno, which lacks native wireless communication, Nano with Wi-Fi integration eliminates the need for additional modules like the ESP8266, reducing complexity and enhancing performance. This makes it highly suitable for building a connected, energy-efficient classroom system.

2.3.2 Passive Infrared Sensors (PIR)



Figure 2.6. PIR Sensors.

PIR sensors as shown in Figure 2.6 are essential for detecting motion, making them a key component of the system's energy saving functionality. PIR sensors are affordable, easy to implement and consume minimal power, making them a practical choice for classrooms where cost and energy efficiency are priorities.

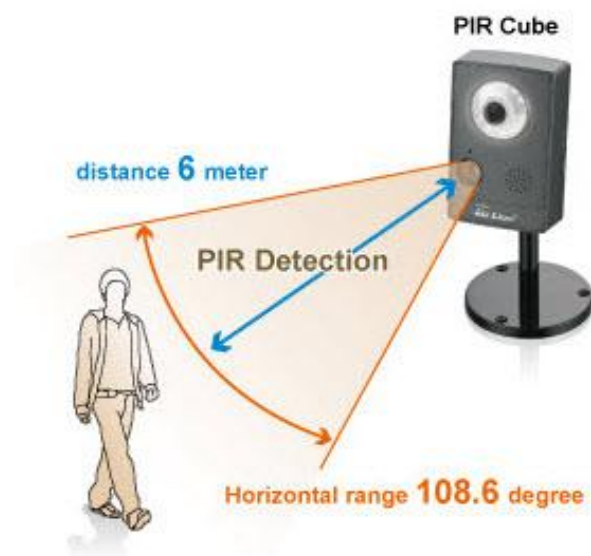


Figure 2.7. PIR Detection.

These sensors detect infrared radiation emitted by human bodies, triggering actions like turning light and fan on or off when occupancy is detected as shown in Figure 2.7. To ensure complete coverage, multiple PIR sensors can be strategically placed in different parts of the classroom. This arrangement addresses the potential for blind spots, ensuring accurate motion detection even when students are seated or stationary for long periods.

2.3.3 DHT11–Temperature and Humidity Sensor

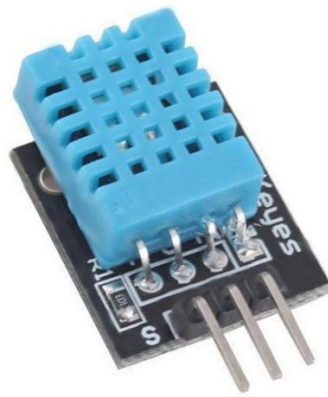


Figure 2.8. DHT11

Figure 2.8 shows the DHT11 sensor is used in this project to measure the temperature and humidity inside the classroom. The main function is to help the system decide when to turn the fan on or off, depending on how hot the room is. For example, if the temperature goes beyond a certain level, the fan will turn on automatically to cool the space. This adds to the smart and energy-saving nature of the system. The DHT11 was selected because it is affordable, easy to use and works well with the Arduino Nano 33 IoT. It only needs a few simple connections, making it convenient to set up without adding too much complexity to the circuit.

2.3.4 Real-Time Clock (RTC) Module



Figure 2.9. Real-Time Clock (RTC) Module.

The RTC module in Figure 2.9 ensures that the system can track time accurately, enabling time-based automation. For example, the system can automatically shut down appliances after operational hours. This feature adds an extra layer of energy efficiency and convenience, especially for classrooms with predictable schedules. RTC modules are particularly useful for scenarios where internet connection might be unreliable, as they provide timekeeping independent of online sources.

2.3.5 LEDs and Resistor

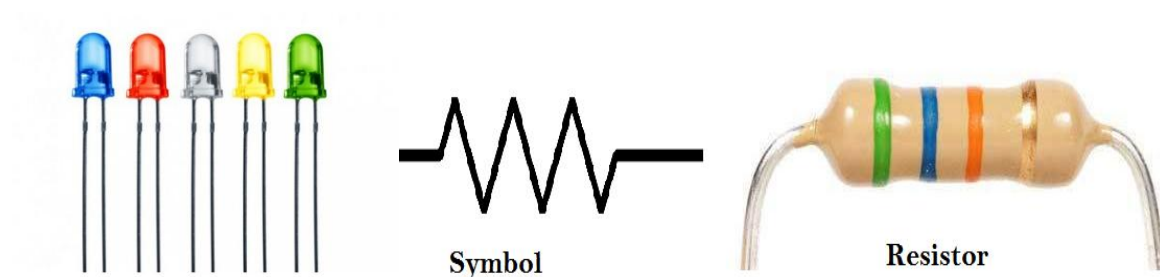


Figure 2.10. LEDs and Resistor.

Figure 2.10 shows Light-emitting diode (LEDs) and Resistors. LED provides visual indicators for system status such as motion detection. For example, an LED could light up when motion is detected. Paired with resistors to control current flow, these indicators make the system more user-friendly and assist in debugging during development.

2.3.6 Connecting Wires and Breadboards

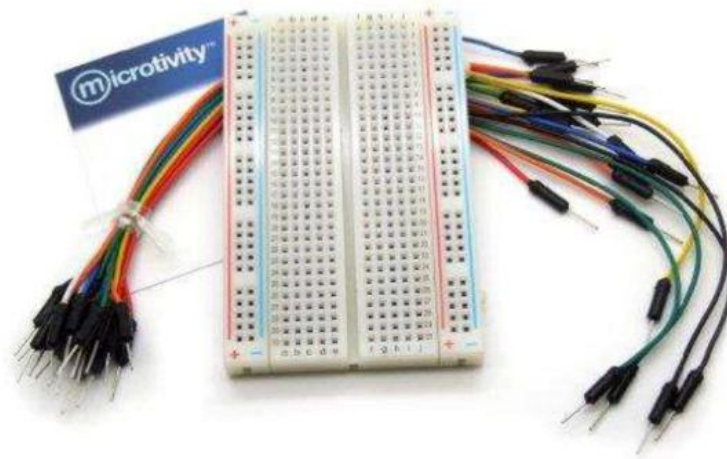


Figure 2.11. Connecting Wires and Breadboards.

Figures 2.11 show the connecting wires and breadboard. These components are fundamental for prototyping and testing the system. They allow for flexible connections between the Arduino Nano and other modules, enabling easy modifications during the development phase. Once the system design is finalized, the connections can be soldered for a more permanent setup.

2.3.7 MG90S Micro Servo Motor



Figure 2.12. MG90S Micro Servo Motor.

Figure 2.12 shows that MG90S micro servo motor was used to simulate the function of a fan. This component was chosen for its compact size, fast response and durability due to its metal gears. The servo motor rotates to a set angle based on motion detection and time conditions, effectively representing the on and off directional behavior of a real fan in the smart classroom system. Its integration helps demonstrate how automation can control basic appliances based on sensor input and predefined rules.

2.3.8 LDR (Light Dependent Resistor)

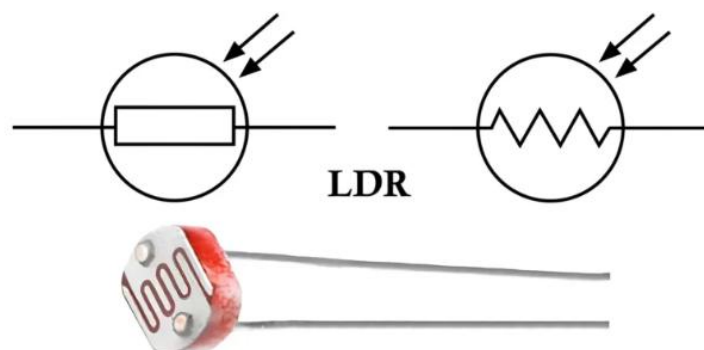


Figure 2.13. Light Dependant Resistor.

The Light Dependent Resistor (LDR) that shown in Figure 2.13 is used to adjust the brightness of the classroom lights based on the surrounding light level, but only when motion is detected by the PIR sensor. Instead of simply turning the lights on or off, the system uses the LDR to decide how bright the lights should be. For instance, if someone enters the room and its already bright outside, the system will keep the lights dim or at a lower brightness. If the room is dark, the lights will automatically become brighter. This smart adjustment helps save energy while still maintaining a comfortable environment. The LDR is a suitable choice for this because it's affordable, easy to use with Arduino and effectively responds to changes in ambient light.

2.3.9 Arduino IoT Cloud



Figure 2.14. Arduino Cloud

Arduino Cloud, as shown in Figure 2.14, was chosen for this project due to its simplicity and reliability in managing IoT devices. The web interface, provided through the Arduino Cloud dashboard, offers an easy-to-use platform for monitoring and controlling the system. Mobile access is facilitated through the IoT Remote app, allowing convenient control and monitoring on the go.

The integration of these tools saves time during setup and ensures smooth operation. Features such as real-time updates and customizable widgets make Arduino Cloud an ideal choice for the smart classroom system, as it enables seamless connectivity and efficient management.

2.4 Conclusion

In this chapter, we explored the key factors that guide the design of the proposed smart classroom management system. By looking at similar systems like Lutron Vive, Honeywell SEMS and Siemens Desigo CC, we identified that most existing solutions are either too complex or costly for smaller environments like classrooms.

The tools and technologies selected for this project include the Arduino Nano 33 IoT with built-in Wi-Fi, PIR sensors, DHT11 temperature and humidity sensor, LDR, MG90S servo motor, real-time clock (RTC) module, LEDs, resistors and essential prototyping components like breadboards and connecting wires. These components were chosen for their affordability, ease of integration and suitability for real-time classroom automation. The selected tools support the main goal of creating a responsive and energy-efficient smart classroom system, capable of adjusting lighting and fan operations based on environmental and occupancy data.

Chapter 3: Requirement Analysis and Design

3.1 Introduction

In this chapter, the methodology to conduct this research will be discussed further in detail on how this project or system develops. There are many types of methodologies used to conduct research or develop software and systems. Each methodology offers a structured way to approach a project, depending on the goals and needs. Research by Guerrero-Ulloa et al. (2023), in software development, some methodologies, like Waterfall, follow a clear step-by-step process, while others, like Agile, focus on flexibility and quick changes based on feedback. Some approaches, like Spiral and Rapid Application Development (RAD), emphasize improving the product over time with frequent updates. The right methodology depends on the specific project and what needs to be achieved.

The chosen methodology is Agile because it offers flexibility and allows for continuous progress. It helps break the project into smaller, manageable tasks, making it easier to track progress and stay on schedule. Al-Saqqa et al. (2020) stated that Agile also allows for quick adjustments if things change or unexpected challenges come up. Its iterative process ensures that feedback can be used to improve the project step by step. This makes Agile a great choice for projects where changes and improvements are expected along the way. Overall, the Agile methodology is well-suited for projects that require quick implementation and the ability to respond to changing requirements.

3.2 Agile Methodology

Agile methodology as shown in Figure 3.1 is an approach that focuses on delivering small, incremental improvements over time. It works by breaking the project into manageable cycles or sprints, with each one producing a part of the final product. This approach allows for constant evaluation and adjustments, ensuring the project meets changing requirements. Agile

emphasizes flexibility, adaptability and continuous progress, making it a great choice for projects that need to be refined and improved.

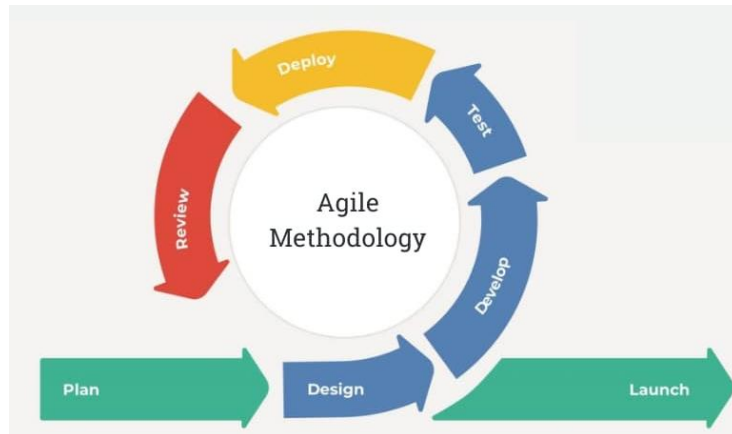


Figure 3.1. Agile Methodology.

3.2.1 Plan

The planning phase lays the foundation for the project. The overall goal of the Smart Classroom Management System is to automate energy usage by controlling light and fan based on occupancy, temperature and lighting conditions. The system will emphasis on energy efficiency and user convenience. Hardware components include a PIR sensor for motion detection, an RTC module for time-based control, a DHT11 sensor for temperature monitoring and an LDR for ambient light sensing. An Arduino Nano 33 IoT board with built-in Wi-Fi is used for cloud connectivity via the Arduino IoT Cloud.

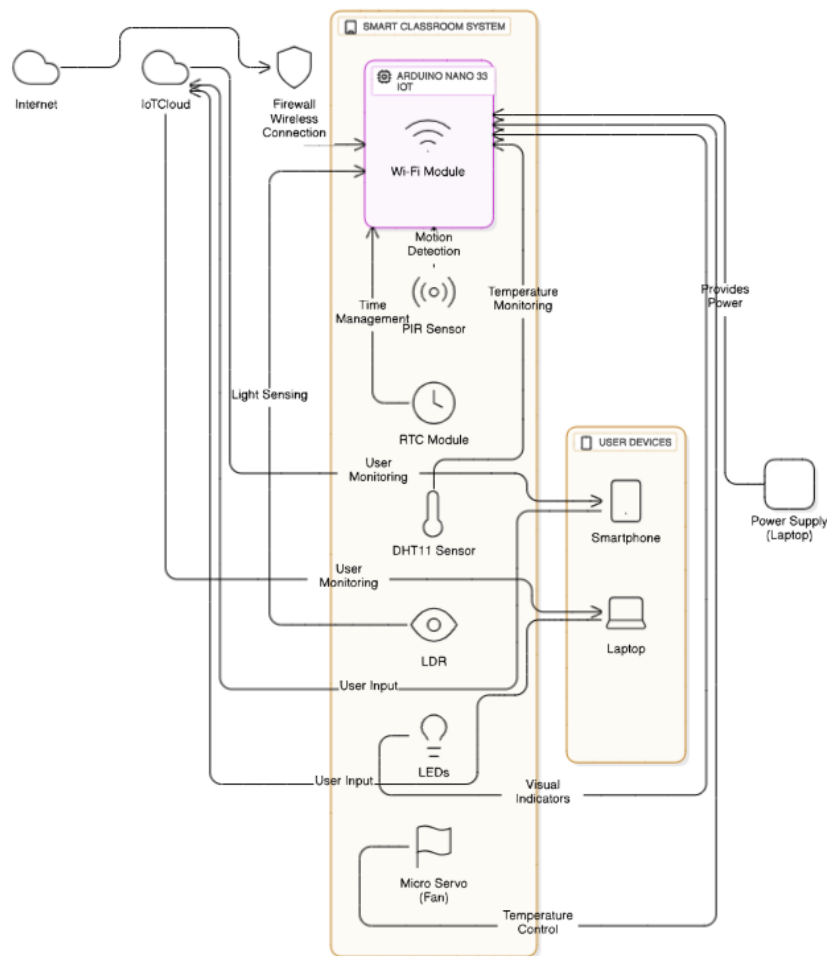
Software development tools include the Arduino IDE and the required libraries for all sensors. The product backlog includes user stories such as integrating PIR for motion detection, implementing temperature-based fan control using the DHT11 sensor, light automation using the LDR and auto power-off scheduling via the RTC. The sprint backlog breaks these stories into smaller, manageable development tasks.

3.2.2 System Architecture Design

Figure 3.2 shows the Smart Classroom system setup, where the Arduino Nano 33 IoT connects various components like the PIR sensor, RTC module, DHT11 temperature sensor, LDR and LEDs. The functions of each component are listed below.

- The PIR sensor detects human motion.
- The RTC module supports time-based management.
- The DHT11 sensor helps monitor temperature to automate the fan.
- The LDR adjusts lighting based on ambient brightness.
- LEDs function as system status indicators.

Data and control signals are transmitted to the Arduino IoT Cloud via Wi-Fi, enabling real-time monitoring and manual control from a smartphone or Laptop. This system enhances classroom energy efficiency and remote management.



3.2.3 Product Backlog

The product backlog is a prioritized list of features, user stories, enhancements and bug fixes required for the Smart Classroom Management System. It serves as a key element of the Agile methodology, guiding the development process. The backlog is adaptable, evolving over time based on feedback, changing requirements and progress made during each sprint.

It is essential as it provides clear guidelines, helping to organize and prioritize tasks. This ensures that development remains focused on the project's goals and user needs, breaking down the overall system into manageable tasks that deliver incremental value throughout the iterations phase.

3.2.4 Use Cases and Scenarios for Smart Classroom System

There are a few scenarios that shown in the Use Case diagram in the Figure 3.3 below:

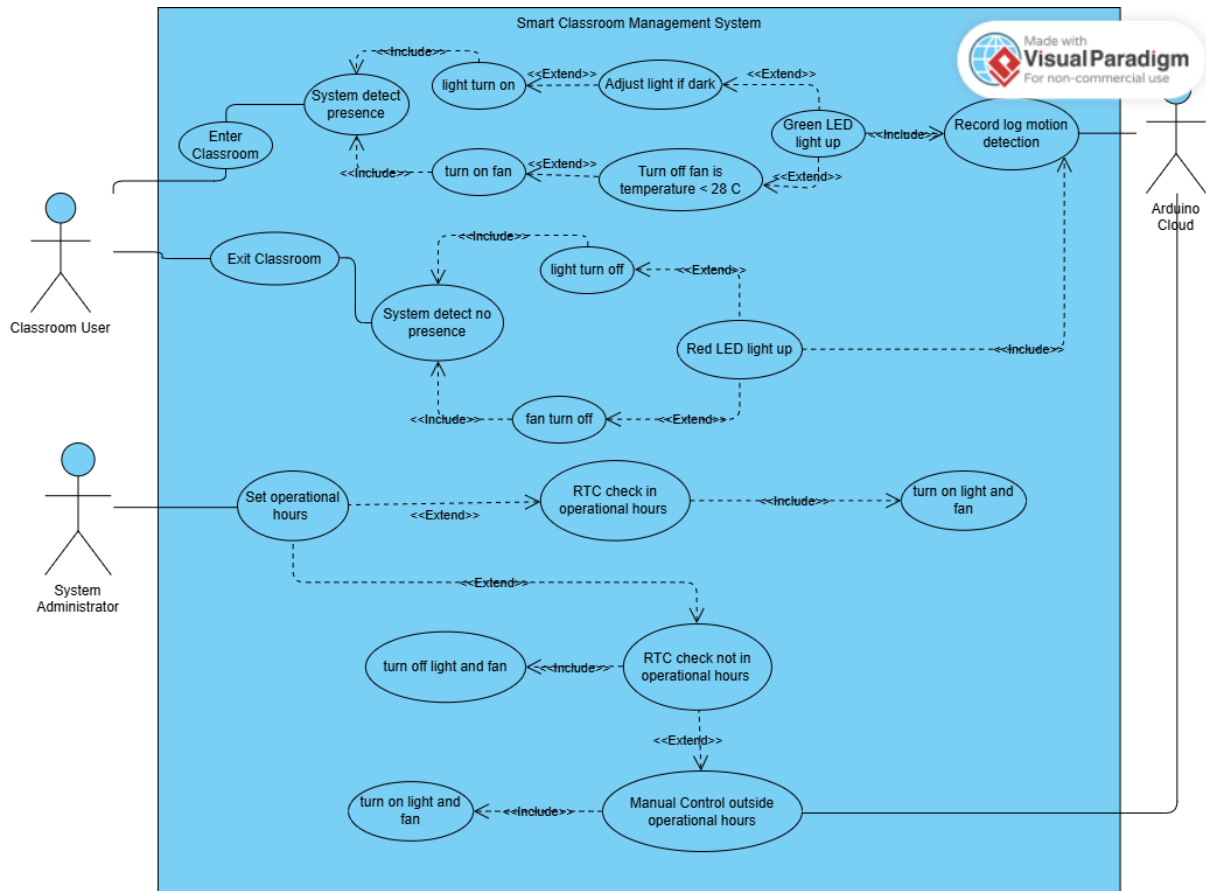


Figure 3.3. Use Case Diagram

The first scenario states that the system should detect classroom user presence using a PIR motion sensor. When a user enters the classroom, the system will automatically turn on the lights. For fan control, the system considers both motion and temperature. Using the DHT11 temperature sensor, if the room is detected to be hot and motion is present, the fan will be turned on to improve comfort. However, if the room is already cool, the fan will remain off even with motion detected. If no motion is detected, the fan will always stay off regardless of temperature, ensuring efficient energy usage.

Additionally, the system integrates a Light Dependent Resistor (LDR) to assess ambient lighting levels. If motion is detected and the room is dim, the lights will be turned on. However, if sufficient natural light is present, the system will keep the lights off even when motion is detected, reducing unnecessary electricity usage.

The System Administrator will define the classroom's operational hours, and the Real-Time Clock (RTC) module will fetch the current time to enforce this schedule. Once operational hours end, the system will automatically shut down all connected devices to conserve energy. However, manual override control is also available outside operational hours, allowing authorized personnel to activate lights or fans if needed for cleaning, events or unusual cases.

In the third scenario, LED indicators provide real-time feedback of the system status. A green LED indicates motion detection and system activity, while a red LED signals inactivity. All activity logs including motion detection, temperature, ambient light level, device status and manual overrides are recorded in real-time to the Arduino IoT Cloud, offering reliable monitoring and data accessibility for future analysis and maintenance.

3.2.5 Use Case Description

Table 3.1 shows how the system detects a student entering the classroom using a PIR motion sensor. Based on the motion and room temperature, it automatically turns on the lights and activates the fan only when necessary.

Table 3.1. Detect Student Entering

Detect Student Entering	
Actor	Classroom User

Brief Description	The system detects a user entering the classroom using a PIR motion sensor. It automatically turns on the lights and activates the fan if the temperature is high.
Pre-condition	The system is powered on, and all sensors are functional.
Post-condition	The lights and fan are turned on based on motion and temperature.
Main Flow	1. A classroom user enters the classroom. 2. The PIR motion sensor detects presence. 3. The system evaluates temperature using the DHT11 sensor. 4. If hot, the fan is turned on. 5. Lights are turned on.
Alternate Flow	If the lights or fan are already on, the system takes no further action.
Exception Flow	None

Table 3.2 illustrates the scenario where the classroom becomes empty. If no motion is detected for a specified duration, the system conserves energy by turning off both the lights and the fan. If the sensors fail to detect the room as empty, the light and fan remain on, and no alternate actions are defined.

Table 3.2. Detect Classroom Empty

Detect Classroom Empty	
Actor	Classroom User
Brief Description	The system detects that the classroom is empty using the PIR motion sensor and turns off the lights and fan after a specified period.
Pre-condition	Lights and fan are on, and sensors are functional.

Post-condition	Lights and fan are turned off.
Main Flow	1. The system monitors the classroom for motion. 2. If no motion is detected for a set duration, it determines the classroom is empty. 3. The lights and fan are turned off.
Alternate Flow	None
Exception Flow	If the sensor fails to detect absence, the devices remain on.

Table 3.3 shows the use case scenario when the system needs to cross check time with Real Time Clock (RTC) to get the real time and determine whether it is within operational hours or not as RTC module will fetch the current time and will give a signal to the system.

Table 3.3. Check Time with RTC

Check Time with RTC	
Actor	System Administrator
Brief Description	The system checks the current time using the RTC module to verify whether it's within the operational hours defined by the administrator.
Pre-condition	RTC module and system are operational.
Post-condition	The system identifies if the current time is within operational hours.
Main Flow	1. The administrator sets operational hours. 2. The RTC module retrieves the current time. 3. The system compares it with the defined hours.
Alternate Flow	If the RTC fails, the system uses a pre-configured fallback time.
Exception Flow	None

Table 3.4 describes the system's behavior for controlling light and fan during operational hours. Using the Real Time Clock (RTC) module to fetch real-time data, the system determines whether it is within operational hours. If so, the light and fan are turned on. If they are already on, no additional action is taken.

Table 3.4. Turn on Light and fan

Turn on Light and fan	
Actor	System Administrator
Brief Description	The system turns on the light and fan if it is determined to be within operational hours.
Pre-condition	The system has identified that it is within operational hours.
Post-condition	Light and fan are turned on.
Main Flow	1. The system verifies the time is within operational hours. 2. The light and fan are turned on.
Alternate Flow	If the devices are already on, no further action is taken.
Exception Flow	None

Table 3.5 explains how the system uses LEDs to display its status, such as motion detection. The LED must be functional and the system operational for this feature to work. The system monitors its status and updates the LED accordingly. If no changes occur, the LED will remain as they were.

Table 3.5. Show System Status on LED

Show System Status on LED	
Actor	System Administrator

Brief Description	Displays system status using LEDs to indicate motion detection.
Pre-condition	LEDs are functional and the system is operational.
Post-condition	System status is displayed on LEDs.
Main Flow	1. The system determines the status. 2. The system updates LEDs to reflect the status.
Alternate Flow	If no status change is detected, the LEDs retain their previous state.
Exception Flow	None

Table 3.6 shows how the system keeps track of its activity using the Arduino IoT Cloud. Whenever motion is detected, or temperature and light levels change, the system records these events along with the exact time they happened. Instead of only logging system errors, it now also notes specific cases like motion during non-operational hours. This helps the administrator understand when and why the system responded. If needed, the administrator can also manually take control of the devices through the cloud dashboard, especially useful if the classroom needs to be used outside normal hours.

Table 3.6. System Activity Logging and Manual Control

System Activity Logging and Manual Control	
Actor	System Administrator
Brief Description	The system logs motion, temperature, light and manual override events to the Arduino IoT Cloud. It allows remote manual control of devices. The administrator can also manually control the system when needed.
Pre-condition	The motion sensor, RTC and logging functionalities are active and connected to the Arduino Cloud.

Post-condition	All events are recorded and viewable on the Arduino Cloud dashboard. Manual overrides are also logged when performed.
Main Flow	1. Sensors detect motion, temperature and ambient light. 2. System logs status and timestamp. 3. Logs are sent to Arduino Cloud. 4. Admin can override device status manually.
Alternate Flow	If no motion is detected, the system logs the status as "No motion detected" along with a timestamp and continues in its previous state.
Exception Flow	The administrator manually overrides system control, and this action is logged along with time and reason.

Table 3.7 describes how the system decides whether to turn on the lights based on how bright the room is. When someone enters and motion is detected, the LDR sensor checks the natural light level. If the room is dark or dim, the system will turn on the lights. But if there's already enough daylight, the lights stay off. This not only saves energy but also makes sure the classroom is always well-lit only when needed. It's a simple way to make the lighting smarter and more responsive to real conditions.

Table 3.7: Control Lights Based on Ambient Light

Control Lights Based on Ambient Light	
Actor	Classroom User
Brief Description	The system uses the LDR sensor to detect ambient light levels and decides whether to turn on lights when motion is present.
Pre-condition	PIR and LDR sensors are operational.

Post-condition	Lights are turned on only if motion is detected and the room is dim.
Main Flow	1. Motion is detected via PIR. 2. LDR sensor checks ambient light level. 3. If dim, lights are turned on.
Alternate Flow	If ambient light is sufficient, lights stay off.
Exception Flow	None

3.2.6 Activity Diagram

Figure 3.4 presents the activity diagram of the smart classroom management system, which begins operating only when someone enters the classroom. As soon as the PIR motion sensor detects movement, the system becomes active and begins checking other conditions. Motion detection is the key trigger for all further actions. If no one is present, the system stays idle. It does not turn on the fan, lights, or respond to environmental readings, regardless of how hot or dark the room is. This helps reduce energy waste and keeps the system efficient.

Once motion is detected, the system checks the current time using the RTC module to determine whether it is within the scheduled operational hours set by the administrator. If the current time is outside of these hours, the system shuts down all devices and ignores any sensor readings.

However, if the time is valid, the system continues by evaluating the room's environmental conditions. The DHT11 sensor reads the temperature and if the room is too hot, the fan turns on. If the room is too cold, the fan turns off automatically. The LDR sensor also measures the ambient light level. If the classroom is too dim, the lights are switched on. If there is already enough natural light, the lights stay off. All these actions only take place because

motion was detected in the first place. Without that initial detection, none of the devices will activate.

To keep users informed, the system uses two LEDs as status indicators. A green LED lights up when motion is detected, while a red LED indicates no presence. All activity, including sensor readings, device responses and system decisions, are recorded and sent to the Arduino IoT Cloud. Each log includes a timestamp and an explanation, such as motion detected after hours, or the fan turned off due to low temperature.

This logging allows the system administrator to monitor usage patterns, identify anomalies and maintain full control of the classroom environment. If needed, the administrator can manually override the system through the cloud platform, turning devices on or off for special use cases outside of normal hours. Altogether, this setup makes the smart classroom responsive, energy-aware and easy to manage remotely.

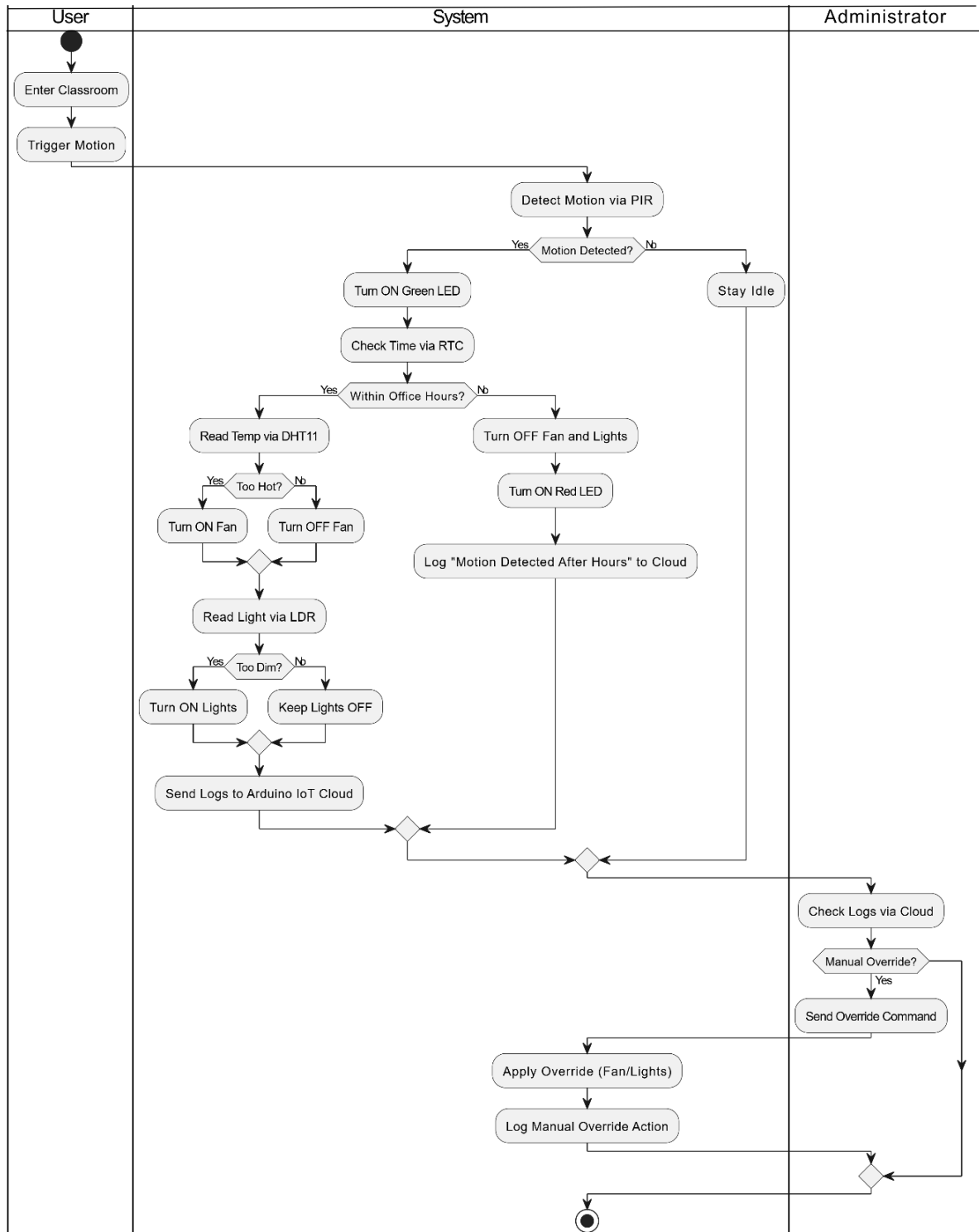


Figure 3.4. Activity Diagram.

3.2.7 Sequence Diagram

Figure 3.5 illustrates the sequence diagram of the smart classroom system, showing how various components work together when a user enters the room. The process begins when the PIR motion sensor detects movement and notifies the system. Once motion is confirmed, the system checks the current time using the RTC module to ensure that the action takes place within operational hours. If the time is valid, the system proceeds to evaluate the classroom environment.

The DHT11 sensor measures the temperature and if the room is too hot, the fan is turned on. If it becomes too cold, the fan is turned off. At the same time, the LDR sensor checks the ambient light level. If the room is dim, the lights are turned on, but if there is enough natural light, they remain off. These environmental responses are only triggered if motion is detected. If no one is present, the system will not activate the fan or lights regardless of the conditions.

The system also updates the LED indicators, with the green LED lighting up when motion is present and the red LED showing when the room is empty. All system activity, including motion events, temperature readings, ambient light levels and device status changes, are logged in real time and sent to the Arduino IoT Cloud. This allows the system administrator to monitor all classroom activity remotely, review system decisions and manually control devices when needed. For example, to enable classroom use outside of operational hours.

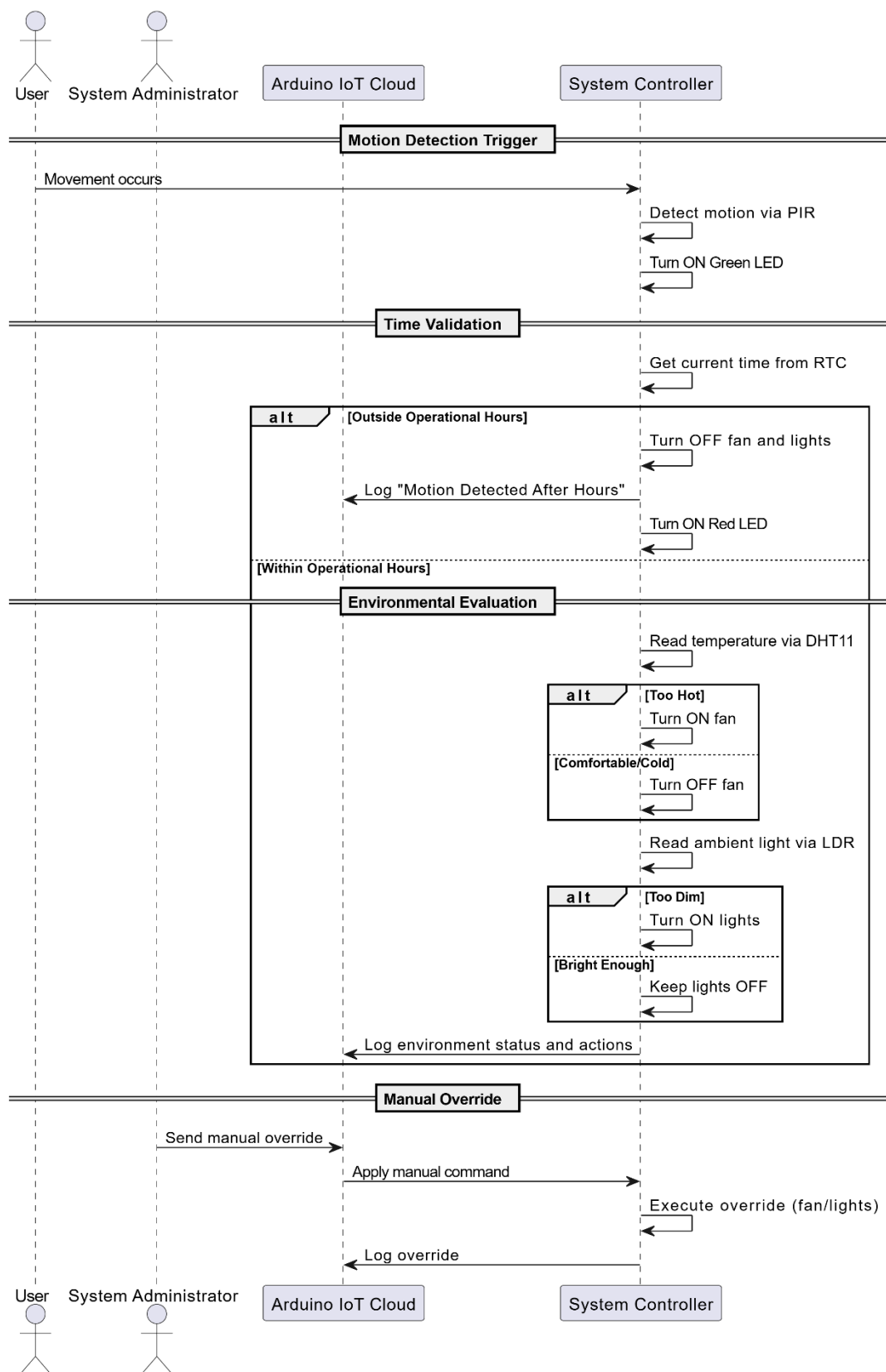


Figure 3.5. Sequence Diagram.

3.3 Iteration Phase

Figure 3.5 shows the Flow Chart of Iteration Phase to describe where the system works on each part step by step. It starts with design, build, test and make improvements based on what we learn from testing. By repeating this process, the system gradually improves until it works well and meets all user needs.

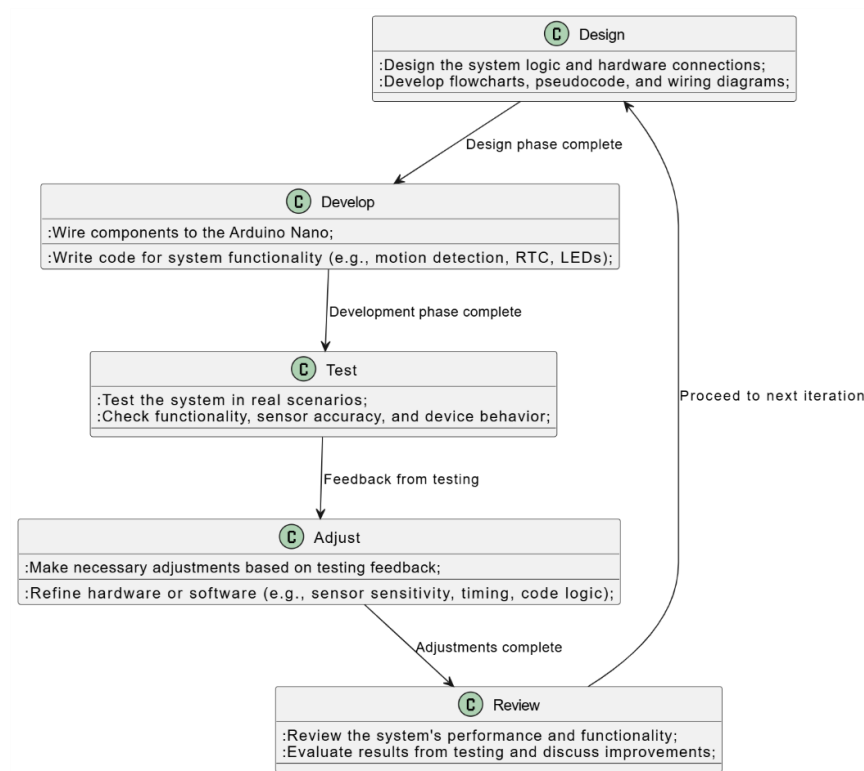


Figure 3.6. Flow Chart of Iteration Phase.

Table below show the detailed breakdown of the iteration phases for each part of the project. Table 3.7 shows the Iteration for phase 1 which is motion detection to control light and fans. Table 3.8 shows the Iteration for phase 2 which is to develop a system that will automatically turn off after operational hours. Table 3.9 shows the Iteration for phase 3 to develop the Status display via LED based on motion and occupancy status. These tables outline the steps of design, development, testing, adjustments and review for each feature. They provide

a clear overview of how each iteration progresses to ensure the system is built effectively and meets all requirements.

Table 3.7. Iteration 1: Motion Detection to Control Light and fan.

Phase	Details
Design	Define how the system should respond when motion is detected, integrating both temperature and ambient light considerations. Pin connections from the Arduino Nano to the PIR sensor, DHT11 sensor, LDR, LEDs and relay modules are carefully mapped for accurate implementation and debugging.
Develop	Connect the PIR sensor, relay modules, DHT11 and LDR to the Arduino Nano. Write code that triggers lights and fan only when motion is detected. If motion is present, DHT11 checks temperature to decide on fan activation, while the LDR determines lighting needs.
Test	Test the system in a real environment to verify that motion detection correctly activates light and fan depending on current temperature and light conditions.
Develop	Adjust based on feedback, such as refining sensor thresholds.
Review	Confirm that the system responds appropriately to motion, temperature and lighting, ensuring it avoids unnecessary energy use when no one is present.

Table 3.8. Iteration 2: Automatic Turn-Off After Class Hours.

Phase	Details
Design	Plan the logic for automatically disabling the system after hours using the RTC module. Define operational hours during which the system is allowed to function.

Develop	Write code to read the current time from the RTC and control whether the system can activate based on motion. Outside of operating hours, motion should not trigger any device.
Test	Test the system's ability to ignore sensor input after operational hours and verify proper shutdown of devices when the schedule ends.
Develop	Fine-tune the operational time range and improve handling of edge cases based on testing results.
Review	Evaluate the time-based control system's reliability and ensure it prevents unnecessary energy consumption during off-hours.

Table 3.9. Iteration 3: Status Display via LED.

Phase	Details
Design	Design the system status LED display logic, including when to light up LEDs based on motion and occupancy status.
Develop	Wire LEDs to Arduino Nano and write code to control the LED display according to the motion and occupancy status.
Test	Test the LED display to ensure it accurately reflects the motion and occupancy status.
Develop	Adjust LED logic based on testing feedback for improved clarity or visibility.
Review	Review the effectiveness of the LED display in conveying motion and occupancy status and discuss improvements if necessary.

3.4 Launch

The launch phase for this project focused on deploying a small prototype of the Smart Classroom Management System in a controlled environment, rather than a full-scale classroom. This phase aims to test the prototype's functionality in real-world conditions and identify any issues that might require adjustments before full implementation.

The prototype, which featured automated control of light and fan based on PIR motion detection, temperature monitoring with the DHT11 sensor and ambient light evaluation using an LDR. During the launch, the system's performance was closely monitored to evaluate sensor accuracy, response times and overall user interaction, providing crucial insights for further refinement.

3.4.1 UI Design

Figure 3.6 shows the expected dashboard for web view that will be implemented in this project which will utilize the drag and drop widgets in the Arduino Cloud dashboard.

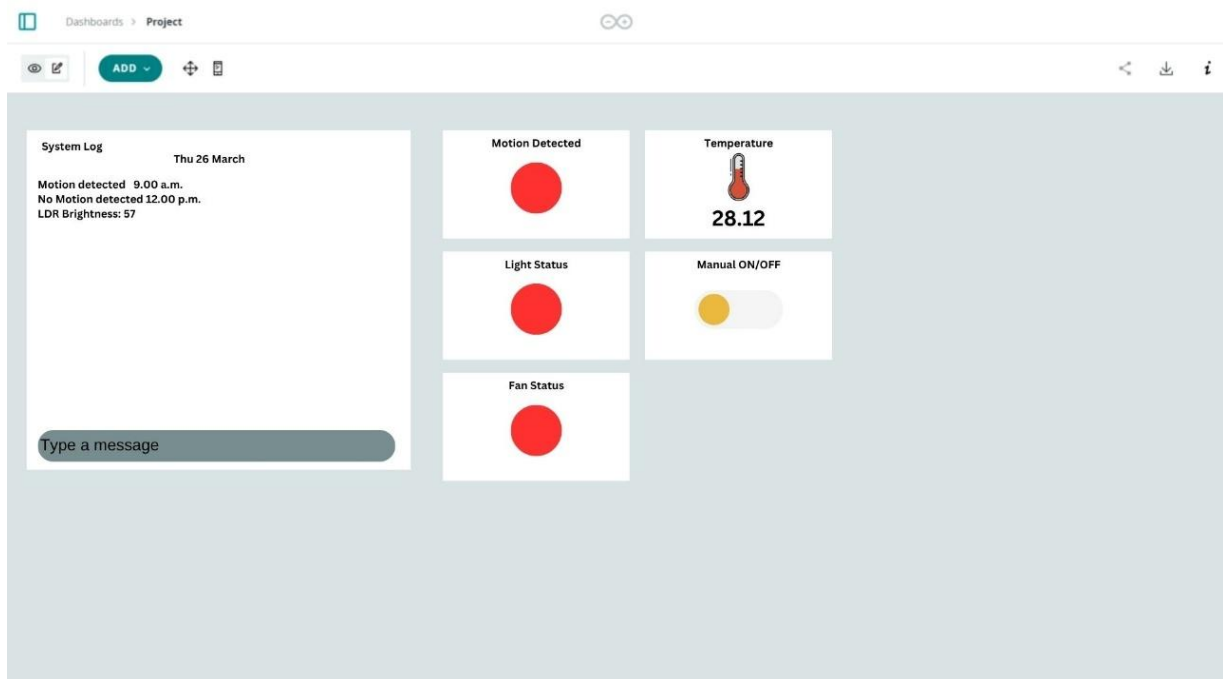


Figure 3.7. Web View Chart of Dashboard Arduino Cloud

Figure 3.7 shows the same dashboard in mobile view, providing a compact and responsive layout optimized for smaller screens. This version ensures that users can easily access the system's features and functionalities on mobile devices.

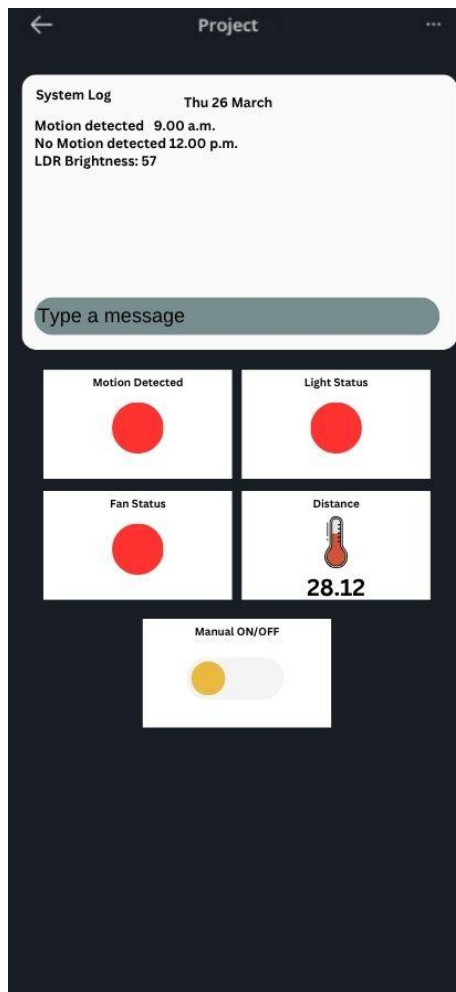


Figure 3.8. Mobile View Chart of Dashboard Arduino Cloud

3.5 Conclusion

To summarize this chapter the methodology for developing the Smart Classroom Management System is discussed, with Agile being chosen for its flexibility and iterative approach. The chapter outlines the process of breaking down the project into smaller,

manageable tasks through sprints, starting with planning and a prioritized product backlog. Use cases for the system are presented, including motion detection for controlling light and fan, automatic shutdown after class hours and motion and occupancy status indication via LEDs.

The iteration phase follows a cycle of design, development, testing, adjustments and review to ensure continuous progress and system improvement. By breaking the project into manageable phases, the development process ensured continuous testing and improvement, ultimately leading to a functional prototype that can be refined further for real-world implementation. This approach allowed for flexibility and adaptability, ensuring that the system met both technical requirements and user needs.

Chapter 4: Implementation and Testing

4.1 Introduction

This chapter describes the implementation and testing of the Smart Classroom system. The setup was built using Arduino Nano and focused on automating classroom utilities based on motion, temperature and ambient light. A Passive Infrared (PIR) sensor served as the primary trigger to detect human presence. Once motion was detected, the DHT11 sensor measured the room temperature to control the fan, while a Light Dependent Resistor (LDR) adjusted the brightness of the lighting based on current ambient light levels.

A Real-Time Clock (RTC) module ensured that the system only functioned during predefined working hours, preventing unnecessary power usage outside of scheduled times. In addition, integration with the Arduino IoT Cloud allowed real-time monitoring, status updates and manual override functionality through an online dashboard.

The test covered various scenarios to confirm that the system responded accurately to motion detection, lighting conditions and temperature changes. The RTC scheduling was validated, and the cloud connectivity was tested for reliable data synchronization and remote control. Any issues identified during testing were resolved to improve stability and efficiency. Overall, the Smart Classroom system successfully automated environmental control in a responsive and energy-efficient manner.

4.2 Tools and Platforms Used

This section explains in detail the tools and platforms used to build and test the system, including how each was configured and applied during development.

4.2.1 Arduino IDE

The Arduino IDE was used during the early development stage of the project. It served as the main platform for writing, compiling and uploading the code to the Arduino Nano 33 IoT board. Based on the Figure 4.1, the first step involved configuring the correct board and port by going to the “Tools” menu, selecting “Board” and choosing “Arduino Nano 33 IoT.” Then, under “Port,” the correct COM port connected to the board was selected to enable successful communication.

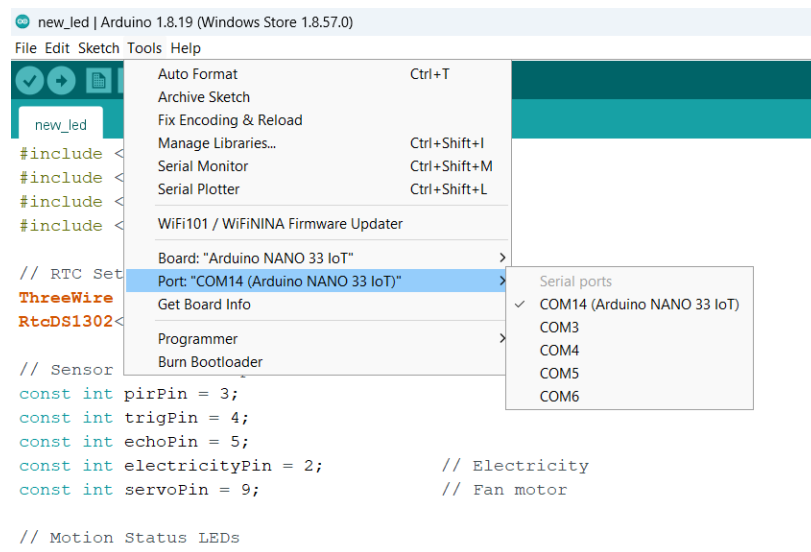


Figure 4.1. Board and Port Configuration via Arduino IDE.

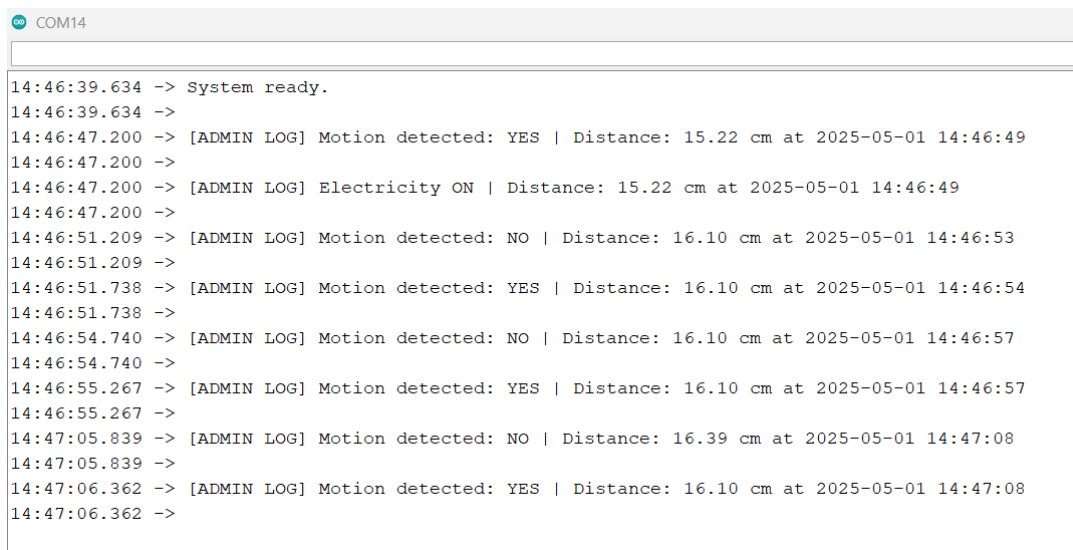
Based on Table 4.1, after setting up the Arduino Nano 33 IoT, the necessary libraries were installed to support the components used in the project.

Table 4.1. Libraries Installed in Arduino IDE.

Library	Usage	Install From
RtcDS1302.h	For DS1302 RTC module	Rtc by Makuna
ThreeWire.h	Communication interface for RTC	Comes with RTC library
Servo.h	Servo motor control	Built-in

DHT.h	For reading DHT11 temperature & humidity sensor	DHT sensor library by Adafruit
Adafruit_Sensor.h	Required dependency for DHT	Adafruit Unified Sensor

Each component, including the DHT11 sensor, LDR and RTC module, was individually tested using the Arduino IDE to ensure proper functionality. For example, the RTC module was set with the current time to enable accurate scheduling of the system. As shown in Figure 4.2, the Arduino IDE's Serial Monitor displayed helpful debugging messages during development, allowing for easier identification and correction of errors. This step-by-step testing approach ensured that all sensors and modules operated as expected before full system integration.



```

COM14
14:46:39.634 -> System ready.
14:46:39.634 ->
14:46:47.200 -> [ADMIN LOG] Motion detected: YES | Distance: 15.22 cm at 2025-05-01 14:46:49
14:46:47.200 ->
14:46:47.200 -> [ADMIN LOG] Electricity ON | Distance: 15.22 cm at 2025-05-01 14:46:49
14:46:47.200 ->
14:46:51.209 -> [ADMIN LOG] Motion detected: NO | Distance: 16.10 cm at 2025-05-01 14:46:53
14:46:51.209 ->
14:46:51.738 -> [ADMIN LOG] Motion detected: YES | Distance: 16.10 cm at 2025-05-01 14:46:54
14:46:51.738 ->
14:46:54.740 -> [ADMIN LOG] Motion detected: NO | Distance: 16.10 cm at 2025-05-01 14:46:57
14:46:54.740 ->
14:46:55.267 -> [ADMIN LOG] Motion detected: YES | Distance: 16.10 cm at 2025-05-01 14:46:57
14:46:55.267 ->
14:47:05.839 -> [ADMIN LOG] Motion detected: NO | Distance: 16.39 cm at 2025-05-01 14:47:08
14:47:05.839 ->
14:47:06.362 -> [ADMIN LOG] Motion detected: YES | Distance: 16.10 cm at 2025-05-01 14:47:08
14:47:06.362 ->

```

Figure 4.2. Serial Monitor on Arduino IDE.

4.2.2 Arduino IoT Cloud

After the basic system was functioning locally, the next step was integrating it with the Arduino IoT Cloud. This platform allows remote control and monitoring of the system in real

time. The process began by logging into the Arduino Cloud and creating an account. Once logged in, a new device was added. This redirected to a separate page prompting the installation of the Arduino Cloud Agent plugin, which is required to detect and communicate with the Arduino Nano 33 IoT board. After successful installation, the board was automatically detected. Based on Figure 4.3, the status of the device history can be monitored via Devices Tab in Arduino Cloud.

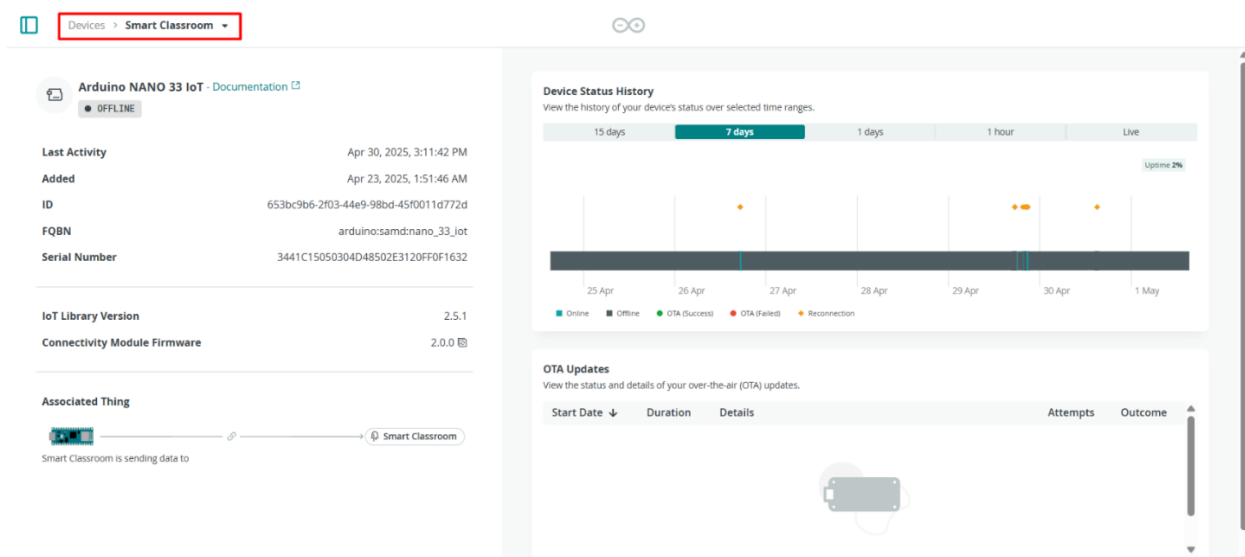


Figure 4.3. Devices on Arduino Cloud

Under the “Things” section of the Arduino IoT Cloud platform, the system configuration continued with the setup of essential cloud components. As shown in the Figure 4.4, several cloud variables were defined and appropriately mapped to the hardware functionalities within the “Setup” tab.

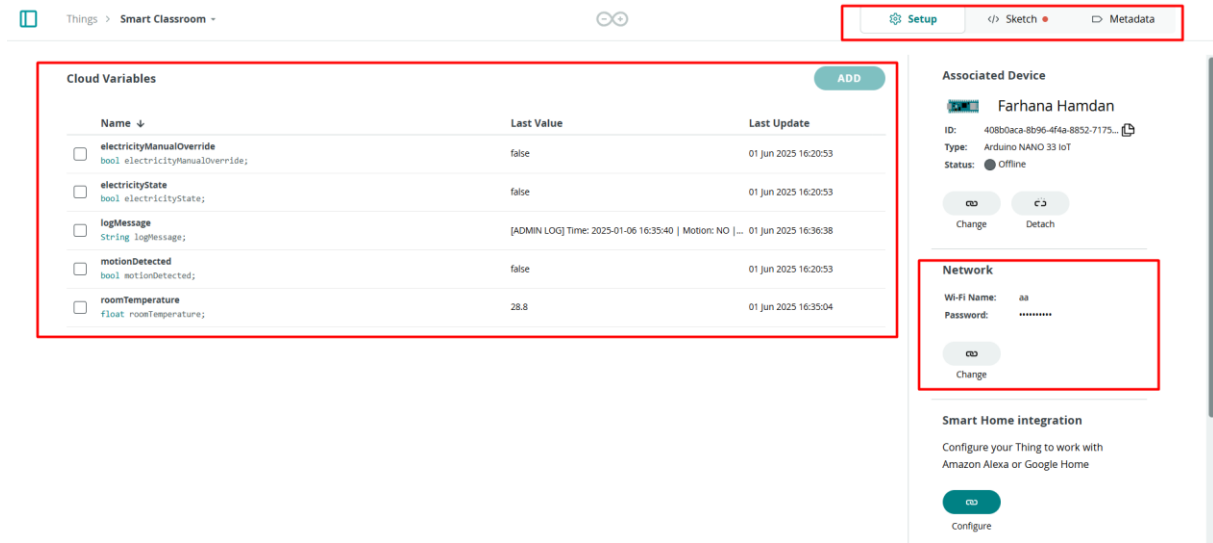


Figure 4.4. Setup Tab on Arduino Cloud

The details of these variables, including their data types and access permissions, are summarized in Table 4.2. These variables served as the interface between the device and the cloud, enabling real-time monitoring and control. The Wi-Fi credentials, including the network SSID and password, were then configured to allow the Arduino Nano 33 IoT to connect to the internet and communicate with the cloud services.

Table 4.2. Variables setup in Arduino Cloud.

Name	Data Type	Last Value	Last Update
electricityManualOverride	bool electricityManualOverride;	false	01 Jun 2025 16:20:53
electricityState	bool electricityState;	false	01 Jun 2025 16:20:53

logMessage	String logMessage;	[ADMIN LOG] Time: 2025-01-06 16:35:40 Motion: NO Temp: 28.8°C Electricity OFF Reason: No Motion Brightness: 0	01 Jun 2025 16:36:38
motionDetected	bool motionDetected;	false	01 Jun 2025 16:20:53
roomTemperature	float roomTemperature;	28.8	01 Jun 2025 16:35:04

Following this, the required libraries were installed using the library manager, accessible through the book shaped icon in the “Sketch” tab. Once the necessary libraries were added, the main program code was uploaded to the board, finalizing the integration with the Arduino IoT Cloud as shown in Figure 4.5.

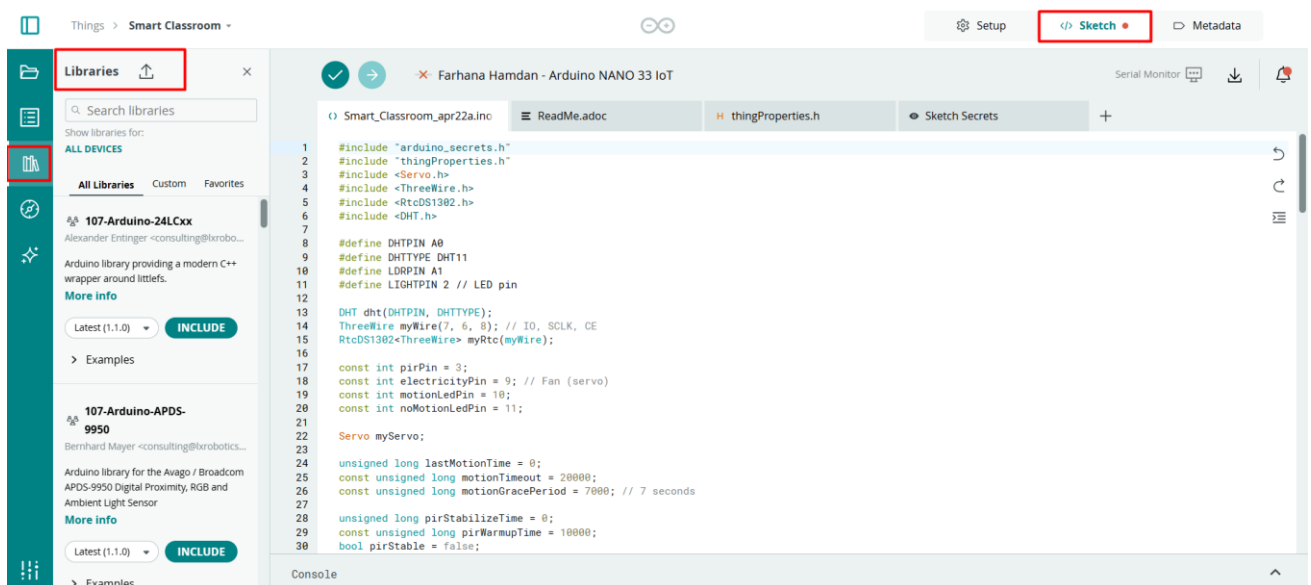


Figure 4.5. Sketch Tab on Arduino Cloud

Once the cloud variables were created and configured, the Dashboard tab was used to design a custom interface that allows users to monitor and control the smart classroom system in real time, as shown in Table 4.3, which lists the widgets used and their respective functions.

Table 4.3. Widgets Used for Dashboard.

Type	Bind to Variable	Label	Function
Switch	electricityManualOverride	Manual Control	Allows manual ON/OFF control of fan and LED
LED	motionDetected	Motion Detected	Lights up green when motion is detected
LED	electricityState	Fan & LED	Shows whether the fan and light are currently active
Text	logMessage	System Log	Displays system messages like motion, time and decisions
Value	roomTemperature	Room Temperature	Displays current room temperature in °C
Chart	roomTemperature	Room Temperature Chart	Plots temperature changes over time for monitoring trends

4.3 Hardware Modules

The smart classroom system uses common hardware modules such as the PIR sensor for motion detection, an ultrasonic sensor for distance measurement and an RTC module to

keep accurate time for scheduling. A micro servo motor acts as a fan, while LEDs provide visual indicators, all controlled by the Arduino Nano 33 IoT board. Table 4.4 shows the list of the components used in this project along with their functions.

Table 4.4. List of the Components

No.	Component	Description
1	DHT11 Sensor	Measures room temperature and humidity for controlling the fan (servo motor).
2	LDR (Photoresistor)	Detects ambient light levels to adjust LED brightness or control lighting.
3	RTC Module (DS1302)	Keeps track of real-time clock for scheduling system operation hours.
4	Arduino Nano 33 IoT	Microcontroller is used to process sensor data and connect to the Arduino IoT Cloud.
5	Resistors (10k Ω)	Used with LEDs to limit current and protect electronic components.
6	LED (Yellow)	Simulates the classroom lamp; brightness controlled based on motion and light.
7	LEDs (Red & Green)	Green LED lights up when motion is detected, Red LED indicates no motion.
8	Micro Servo MG90S	Simulates a fan; rotates based on temperature and motion detection.

The pin connection overview summarizes how each hardware component is connected to the Arduino Nano 33 IoT. As shown in Table 4.5, each sensor, LED and component is assigned a specific digital or analog pin on the microcontroller, ensuring proper communication and functionality within the smart classroom system.

Table 4.5. Pin Connection Overview.

Connected Component	Arduino Pin	Component Pin / Wire	Purpose
DHT11 Sensor	A0	DATA	Reads room temperature to control fan logic.
	5V	VCC	Powers the DHT11 sensor.
	GND	GND	Ground connection for the DHT11 sensor.
LDR + 10k Ω Resistor	A1	Middle of LDR– 10k Ω divider	Reads ambient light level for LED brightness control.
	5V	LDR side	Connected to one leg of LDR.
	GND	Resistor side	Resistor connected to GND (bottom of voltage divider).
Yellow LED (Lamp)	D2	Anode	Adjusts brightness automatically based on LDR.
	GND	Cathode (via 10k Ω resistor)	Ground through current-limiting resistor.
RTC Module (DS1302)	D6	CLK	Clock signal for RTC sync.
	D7	DAT	Data I/O for RTC communication.

	D8	RST	Reset signal to initiate communication.
	3.3V	VCC	Powers the RTC module.
	GND	GND	Ground connection for RTC module.
PIR Sensor	D3	OUT	Detects motion; HIGH when triggered.
	5V	VCC	Powers the PIR sensor.
	GND	GND	Ground connection for PIR sensor.
Micro Servo (Fan)	D9	Orange (Signal or PWM)	Controls fan ON/OFF via rotation.
	5V	Red (VCC)	Powers the servo motor.
	GND	Brown (GND)	Ground connection for the servo.
Green LED (Motion Detected)	D10	Anode	Turns ON when motion is detected.
	GND	Cathode (via 10k Ω resistor)	Ground with current-limiting resistor.
Red LED (No Motion)	D11	Anode	Turns ON when no motion is detected.
	GND	Cathode (via 10k Ω resistor)	Ground with current-limiting resistor.

The Smart Classroom system functions intelligently based on synchronized sensor inputs, responding with automated controls only when motion is detected. A PIR sensor on pin

D3 acts as the trigger, detecting movement in the room. Without motion, the system remains inactive to conserve energy. The system uses a DS1302 RTC module, connected to D6 (CLK), D7 (DAT) and D8 (RST), to keep track of real-time clock. It limits operations to working hours (8 AM to 6 PM) unless manual override is used.

Once motion is detected during working hours, the system synchronizes two main environmental sensors as listed in Table 4.6:

Table 4.6. Two Main Environmental Sensors.

No.	Sensors	Function
1.	DHT11 temperature and humidity sensor	A0 measures room temperature. If the room is too hot (above 28°C), the MG90S micro servo on D9 rotates to simulate fan activation.
2.	LDR (light-dependent resistor)	A1 monitor's ambient light. If the room is dark, the system powers the yellow LED on D2 to simulate lighting. The brightness level is dynamically adjusted using PWM based on LDR readings.

All LEDs and sensors are protected with 10k Ω resistors to ensure safety and prolong hardware lifespan. The system is powered using a regulated 3.3V supply, compatible with various sources like USB ports, adapters, or portable power banks. This enables reliable performance for all components, including sensors, LEDs and the servo motor.

4.4 Cloud Dashboard Controls

To enhance usability and allow remote monitoring, a custom dashboard was designed using the Arduino IoT Cloud platform. After successfully creating and configuring the cloud variables, the dashboard interface was developed to visually represent system activity and provide manual control options. The dashboard includes widgets such as switches, LEDs, value displays text fields and chart all of which are bound to the respective variables in the system.

This interface allows users to monitor motion status, check the electricity state and view detailed log messages generated by the system in real time. It also includes a manual override switch, enabling users to control classroom equipment manually, regardless of sensor conditions. The dashboard is designed to be intuitive and user-friendly, ensuring easy access for users even without technical experience, as illustrated in Figure 4.6 and Figure 4.7.

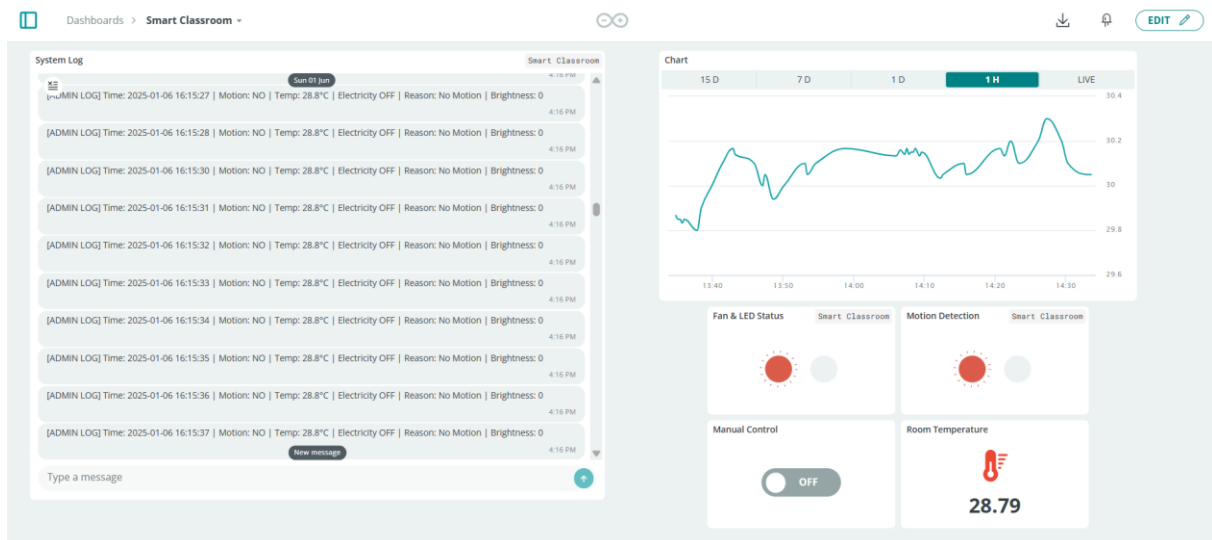


Figure 4.6. Implementation on Web View of Dashboard Arduino Cloud.

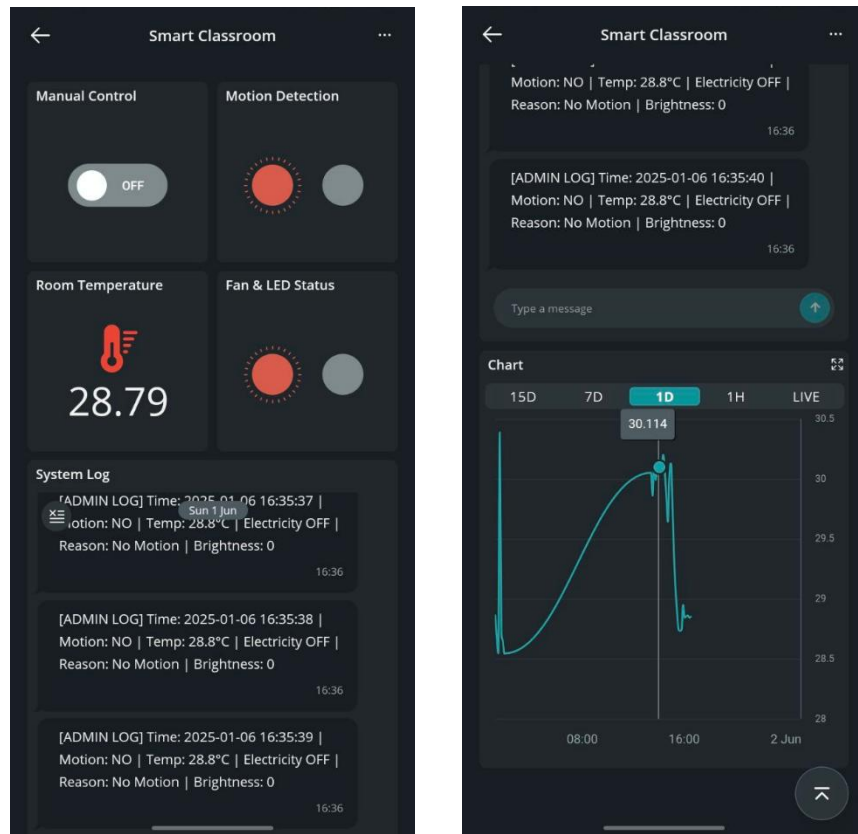


Figure 4.7. Implementation on Mobile View of Dashboard Arduino Cloud.

4.5 Implementation Process

This section outlines the development and implementation process of the smart classroom system. The system was designed by integrating motion detection, temperature and light sensing, time-based automation and cloud connectivity. Its main goal is to ensure that devices like light and fan operate only when necessary, improving energy efficiency. The development involved both hardware configuration and software programming, starting with offline testing and later connected to the Arduino IoT Cloud for real-time remote monitoring and control.

4.5.1 Motion Detection Logic

The motion detection part of the system used a PIR (Passive Infrared) sensor to detect human presence in the classroom. When the PIR sensor detected any movement, the system automatically turned on both the fan and the LED lamp. This created a smart environment where energy was only used when someone was present.

To make the motion detection more accurate, the PIR sensor's settings were manually adjusted using the small orange knobs located on the side of the sensor module as shown in Figure 4.8. One of these knobs controls the delay time and the other controls the detection range or sensitivity.

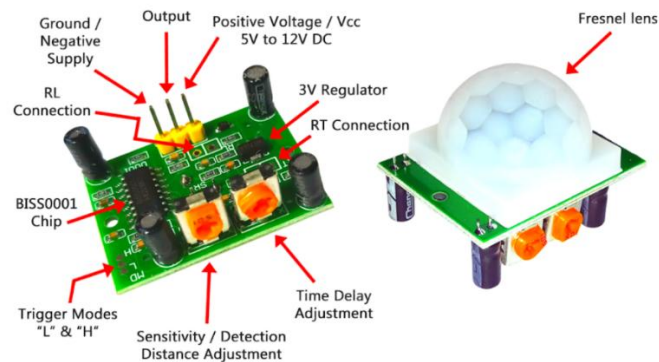


Figure 4.8. PIR Sensor with Component Label.

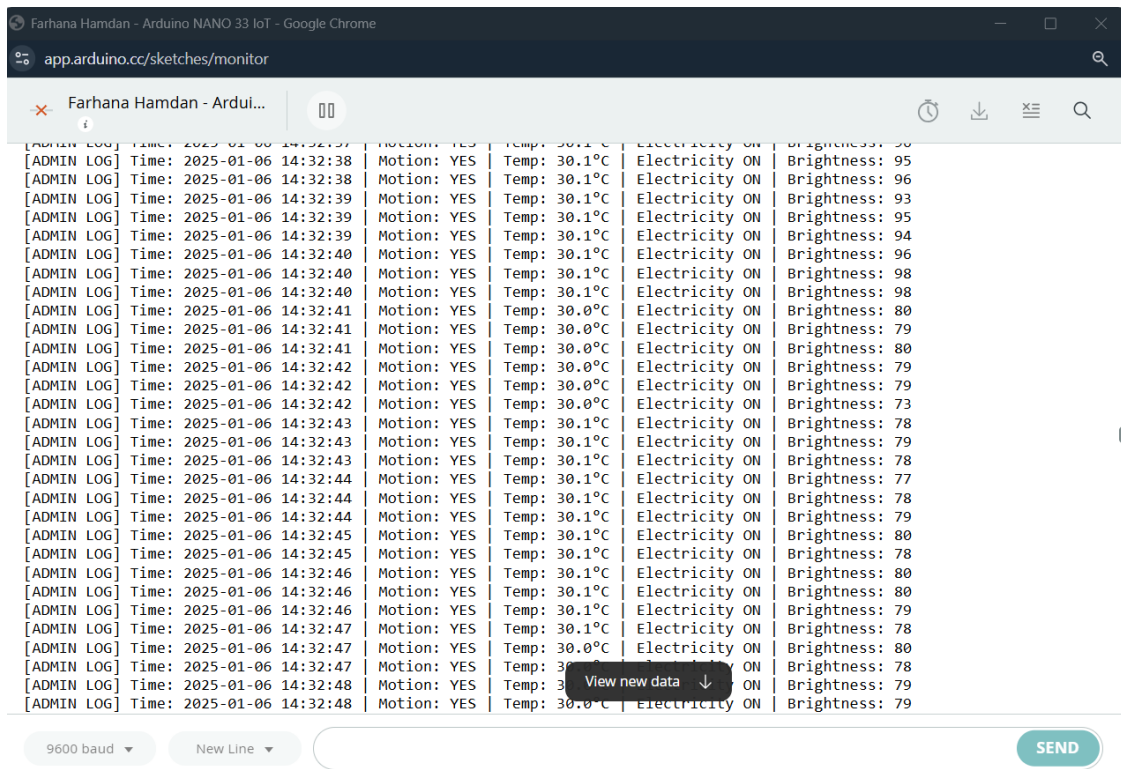
By carefully adjusting these two, we made sure the system wouldn't trigger too easily or stay on for too long unnecessarily. This tuning helped especially in real classroom settings, where slight movement from students should be enough to keep the devices on, but also turn them off when the room is empty for a while.

As shown in the code snippet in Figure 4.9, after a 7 second grace period has passed since the last detected motion, the system automatically turns off the fan and lamp to conserve electricity.

```
unsigned long lastMotionTime = 0;
const unsigned long motionGracePeriod = 7000; // 7 seconds
```

Figure 4.9. Serial Monitor Output when Motion is Detected.

Figure 4.10 displays the Serial Monitor output when motion is detected, while Figure 4.11 shows the Arduino IoT Cloud dashboard indicating motion detection, with the fan and lamp status highlighted in green.



Time	Motion	Temp	Electricity	Brightness
2025-01-06 14:32:37	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 95
2025-01-06 14:32:38	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 96
2025-01-06 14:32:39	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 93
2025-01-06 14:32:39	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 95
2025-01-06 14:32:39	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 94
2025-01-06 14:32:40	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 96
2025-01-06 14:32:40	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 98
2025-01-06 14:32:40	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 98
2025-01-06 14:32:41	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 80
2025-01-06 14:32:41	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 79
2025-01-06 14:32:41	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 80
2025-01-06 14:32:42	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 79
2025-01-06 14:32:42	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 79
2025-01-06 14:32:42	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 73
2025-01-06 14:32:43	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 78
2025-01-06 14:32:43	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 79
2025-01-06 14:32:43	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 78
2025-01-06 14:32:44	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 77
2025-01-06 14:32:44	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 78
2025-01-06 14:32:44	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 79
2025-01-06 14:32:45	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 80
2025-01-06 14:32:45	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 78
2025-01-06 14:32:46	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 80
2025-01-06 14:32:46	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 80
2025-01-06 14:32:46	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 79
2025-01-06 14:32:47	Motion: YES	Temp: 30.1°C	Electricity ON	Brightness: 78
2025-01-06 14:32:47	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 80
2025-01-06 14:32:47	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 78
2025-01-06 14:32:48	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 79
2025-01-06 14:32:48	Motion: YES	Temp: 30.0°C	Electricity ON	Brightness: 79

Figure 4.10. Serial Monitor Output when Motion is Detected.

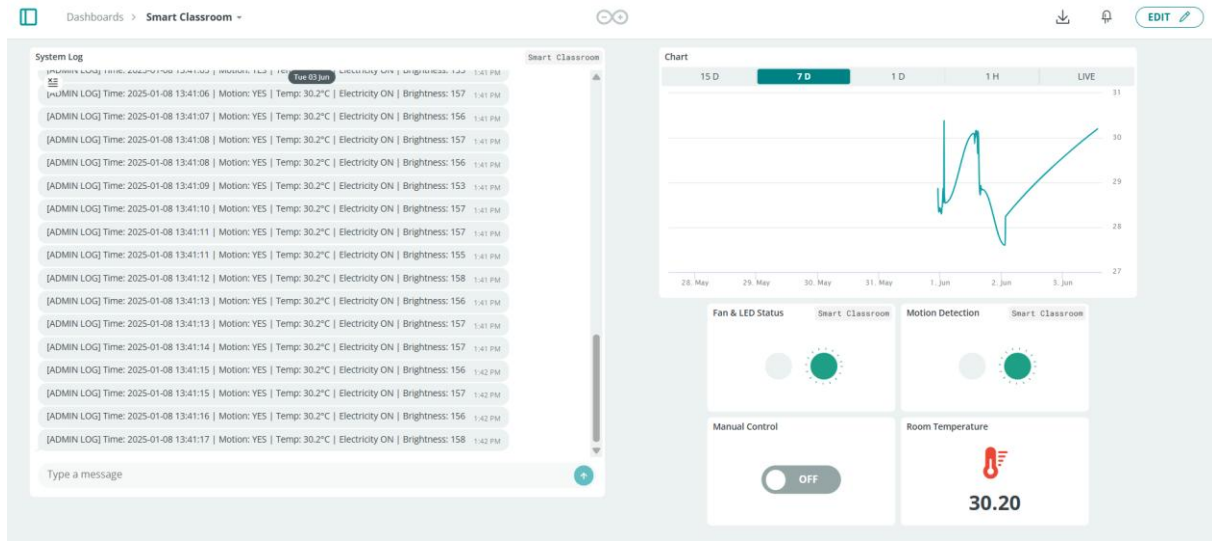


Figure 4.11. Arduino Cloud Dashboard when Motion is Detected.

4.5.2 Environmental Condition Response

The smart classroom system is designed to respond dynamically to environmental changes, ensuring optimal comfort and energy efficiency. The DHT11 temperature sensor and Light Dependent Resistor (LDR) are used in this project to simulate environmental awareness, allowing the system to respond to heat and light levels accordingly. However, these sensors only trigger actions when motion is detected in the room.

The LDR checks the ambient brightness. If the light level is low, especially during cloudy evening, the system automatically turns on the lamp and adjusts its brightness using Pulse Width Modulation (PWM). This ensures just enough lighting is provided.

At the same time, the DHT11 sensor continuously monitors the room temperature. If the temperature rises above a defined threshold which in this case is 28C, the system activates the fan by rotating a servo motor to simulate turning it on. This logic helps keep the room cool and comfortable while avoiding unnecessary fan usage in cooler conditions. By combining real-time motion, temperature and lighting data, the system creates a responsive and adaptive classroom environment.

4.5.3 Time-Based Automation

To follow classroom schedules and avoid unnecessary energy usage outside of school hours, a Real-Time Clock (RTC) module was included in the setup. This module keeps track of the current time even when the Arduino is restarted or loses power. At the start of the project, we set the current time manually using a separate code. After that, the RTC module kept counting time accurately on its own.

With the help of this RTC, we programmed the system to operate only during working hours, for example, between 8:00 AM and 6:00 PM as shown in the code screenshot in Figure 4.13. If the system detected motion after 6:00 PM, it would ignore the signal and turn off the devices automatically. This made the system even more energy efficient by double checking both the motion status and the current time before activating any devices.

```
bool isWorkingHours = now.Hour() >= 8 && now.Hour() < 18;  
String statusMessage;
```

Figure 4.13. RTC Configure Working Hours.

Figure 4.14 shows the dashboard when motion is detected. Although the motion status is highlighted in green, indicating motion was detected, the fan and LED statuses are shown in red, signifying that both the fan and lamp remain off. The log will display reason of this system behaviour which is outside working hours.

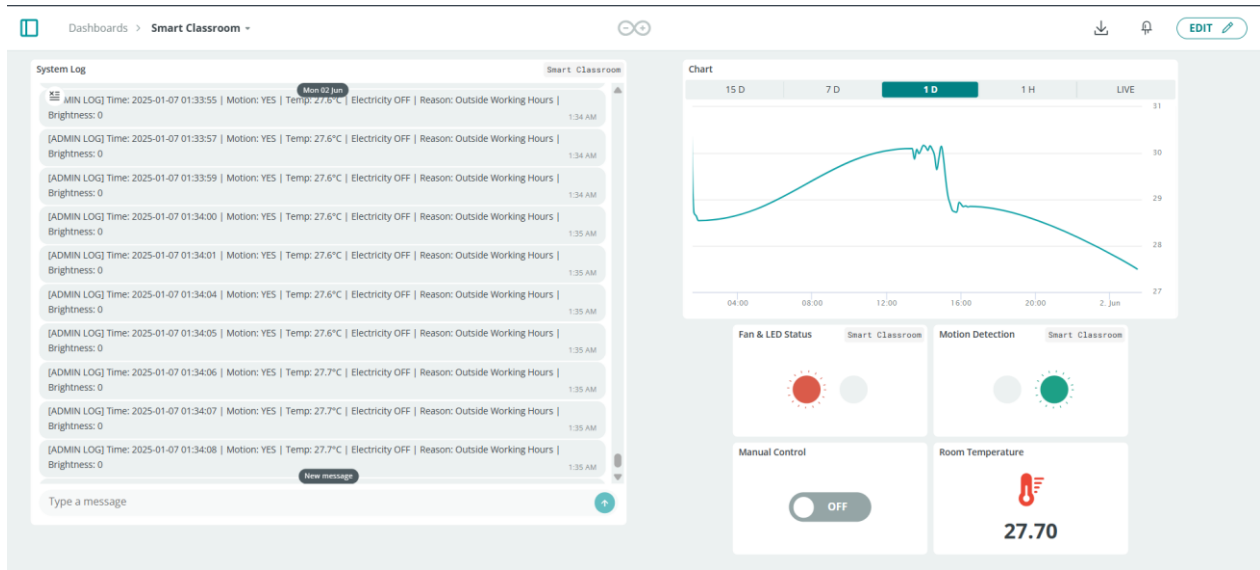


Figure 4.14. Arduino Cloud Dashboard when Motion is Detected.

4.5.4 Cloud Connectivity and Manual Override Functionality

Once the hardware and logic were working well in offline mode, the system was connected to the Arduino IoT Cloud for real-time monitoring and control. Using the Arduino Nano 33 IoT, which has built-in Wi-Fi, we created a cloud dashboard where key variables such as motion detection status, room temperature and LED status could be viewed remotely.

We configured these variables in the Arduino IoT Cloud with different access types which some were read-only while others allowed control such as turning the system on or off remotely. This setup allowed users or administrators to check the classroom's condition from their smartphones or computers.

Cloud integration made the system smarter and more flexible, giving remote access to view sensor readings, getting motion logs and manually override controls if needed. In particular, the manual override feature was added to allow users to activate the fan and lamp outside of regular working hours, ensuring that the classroom could still be used comfortably during special sessions, meetings, or extended study periods. These controls can be accessed

through the Arduino IoT Cloud dashboard, as shown in Figure 4.15. The system log will show the reasons of the system behaviour because of manual override.

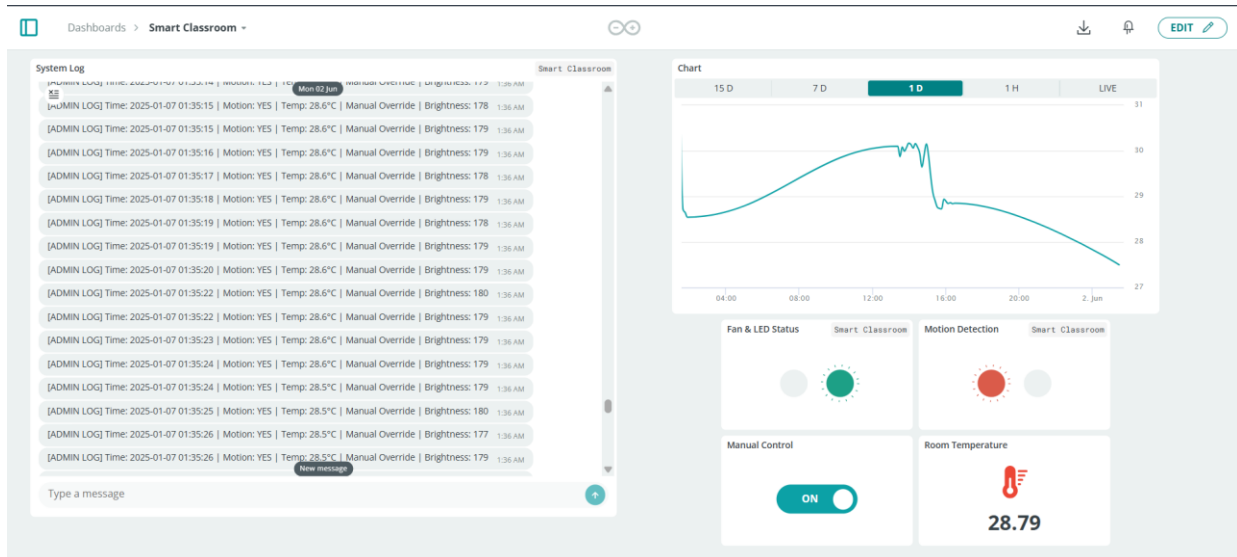


Figure 4.15. Manual Control on Arduino Cloud Dashboard.

4.6 Conclusion

In this chapter, the implementation of the smart classroom system was explained in detail. The system was developed using Arduino IDE and Arduino IoT Cloud, with various modules enabling remote control of electrical appliances. The motion detection feature was used to automate light and fan, while an automatic turn-off function ensured energy savings after operational hours. Additionally, an LED status display provided real-time system updates.

Chapter 5: System Testing

5.1 Introduction

System testing is a crucial phase in the development of any project, as it ensures that all components function together as intended. In this phase, the smart classroom system underwent a series of tests to verify the functionality, accuracy and reliability of both the hardware and software. The testing focused on assessing the performance of sensors, actuators and the cloud connection, ensuring that each element of the system was responsive and integrated correctly. Additionally, the system's behavior during various scenarios was evaluated to confirm that it operated as expected.

5.2 Testing and Validation

After assembling the smart classroom system, a series of tests were carried out to verify motion detection, sensor accuracy and time-based automation. Each component was tested on its own, then evaluated together to ensure the system functioned smoothly as a whole.

5.2.1 Functional Testing

To begin testing, each sensor was checked for proper functionality. The PIR sensor was tested by walking in front of it at different angles and distances to confirm it successfully detected motion. When motion was detected, both the LED light and fan via servo motor activated correctly, confirming the logic was working.

The DHT11 temperature sensor and the Light Dependent Resistor (LDR) were also tested to simulate environmental response. When motion was detected, the LDR adjusted the brightness of the lamp based on room lighting and the DHT11 triggered the fan if the temperature exceeded 28°C. These behaviors confirmed that the system could respond accurately to both motion and environmental conditions.

5.2.2 Timing Accuracy Test

The Real-Time Clock (RTC) module was tested by setting the current time and comparing it to a reference clock. During testing, the system correctly activated and deactivated devices based on whether the current time was within working hours. For example, if the time was after 6:00 PM, the devices remained off even if motion was detected, which confirmed the RTC-based logic worked as planned.

As shown in Table 5.1, a comparison of System Log output shows how the system behaves during operational hours versus outside of them.

Table 5.1. System Log Overview

Scenario	Conditions	Example Log Output
Manual Override ON	<code>electricityManualOverride == true</code>	[ADMIN LOG] Time: 2025-06-01 14:32:10 Motion: YES Temp: 30.2°C Manual Override Brightness: 120
Automatic: Motion Detected, Working Hours, Temp > 28°C	<code>electricityManualOverride == false AND motionDetected == true AND hour >= 8 && hour < 18 AND temp > 28.0</code>	[ADMIN LOG] Time: 2025-06-01 10:45:23 Motion: YES Temp: 29.5°C Electricity ON Brightness: 145
Automatic: Motion Detected, Working Hours, Temp ≤ 28°C	same as above but Temp ≤ 28	[ADMIN LOG] Time: 2025-06-01 11:15:48 Motion: YES Temp: 26.4°C Electricity OFF Room Cool Enough Brightness: 160

Automatic: No Motion (Working Hours)	electricityManualOverride == false AND motionDetected == false AND hour >= 8 && hour < 18	[ADMIN LOG] Time: 2025-06-01 13:18:05 Motion: NO Temp: 30.1°C Electricity OFF Reason: No Motion Brightness: 0
Automatic: Outside Working Hours	electricityManualOverride == false AND `hour < 8	[ADMIN LOG] Time: 2025-06-01 06:12:17 Motion: NO Temp: 27.7°C Electricity OFF Reason: Outside Working Hours Brightness: 0

5.3 Test Case

The system was tested across multiple modules from Table 5.2 to Table 5.8, each addressing a specific functionality. Table 5.2 (Detect Student Entering Module) and Table 5.3 (Detect Classroom Empty Module) verified the effectiveness of motion detection and the automatic control of light and fan. Table 5.4 (Check Time with RTC Module) and Table 5.5 (Turn on Light and fan Module) confirmed the system's ability to operate based on defined working hours. Table 5.6 (Show System Status on LED) ensured accurate visual indicators, while Table 5.7 (System Activity Logging and Manual Control) validated comprehensive event logging to the Arduino Cloud and support for manual override. Lastly, Table 5.8 (Control Lights Based on Ambient Light Module) tested the system's ability to adjust lighting intelligently based on ambient light levels, enhancing energy efficiency.

5.3.1 Test Case for Detect Student Entering Module

Table 5.2. Detect Student Entering Module.

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
1	Detect Student Entering	1. System is powered on and sensors (PIR, DHT11, LDR, RTC) are functional. 2. User enters the classroom. 3. PIR detects motion. 4. RTC verifies it is within operational hours or override is enabled. 5. DHT11 checks temperature. 6. LDR checks ambient light. 7. If hot, turn on fan. 8. If dark, turn on LED.	Fan and LED turn on if conditions are met. LEDs indicate device activation.	Fan and LED turn on if conditions are met. LEDs indicate device activation.	Pass
		User enters again while lights and fan are already ON.	System takes no further action.	System takes no further action.	Pass

5.3.2 Test Case for Detect Classroom Empty Module

Table 5.3. Detect Classroom Empty Module.

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
2	Detect Classroom Empty	1. Light and fan are ON. 2. System monitors for motion. 3. No motion detected for set duration.	Lights and fan are turned OFF.	Lights and fan are turned OFF.	Pass
		Sensor fails to detect absence.	Lights and fan remain ON.	Lights and fan remain ON.	Pass

5.3.3 Test Case for Check Time with RTC Module

Table 5.4. Check Time with RTC Module.

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
3	Check Time with RTC	1. RTC module is operational. 2. System fetches current time. 3. Compares with working hours.	System correctly identifies time status.	System correctly identifies time status.	Pass

		RTC fails to fetch time.	System defaults to preconfigured fallback time.	System defaults to preconfigured fallback time.	Pass
--	--	--------------------------	---	---	------

5.3.4 Test Case for Turn on Light and fan Module

Table 5.5. Turn on Light and fan Module.

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
4	Turn on Light and fan	1. Time verified within operational hours. 2. Motion is detected. 3. Temp and light levels are evaluated.	Devices turn ON if hot or dark.	Devices turn ON if hot or dark.	Pass
		Devices are already ON.	No further action taken.	No further action taken.	Pass

5.3.5 Test Case for Show System Status on LED

Table 5.6. Show System Status on LED.

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
5	Show System Status on LED	1. System and LEDs functional. 2. Status changes (e.g., motion detected) 3. LEDs update accordingly.	LED status reflects changes.	LED status reflects changes.	Pass
		No system status change.	LEDs retain previous state.	LEDs retain previous state.	Pass

5.3.6 Test Case for System Activity Logging and Manual Control

Table 5.7. System Activity Logging and Manual Control.

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
6	System Activity Logging and Manual Control	1. Motion, temp and light are detected. 2. RTC checks time.	Logs are visible on Arduino Cloud.	Logs are visible on Arduino Cloud.	Pass

		3. Logs sent with reason and timestamp. 4. Admin checks logs.			
		No motion detected for long duration.	"No motion" log with timestamp.	"No motion" log with timestamp.	Pass
		Admin manually overrides outside hours.	Override action is logged.	Override action is logged.	Pass

5.3.7 Test Case for Control Lights Based on Ambient Light Module

Table 5.8. Control Lights Based on Ambient Light Module

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
7	Control Lights Based on Ambient Light	1. Motion is detected. 2. RTC checks if time is within working hours. 3. LDR checks brightness.	LED turns ON in darkness; OFF in brightness.	LED turns ON in darkness; OFF in brightness.	Pass

		4. If dark, LED turns ON. 5. If bright, LED stays OFF.			
		Ambient light is sufficient.	LED remains OFF.	LED remains OFF.	Pass
		Admin manually turns ON LED.	LED turns ON regardless of LDR.	LED turns ON regardless of LDR.	Pass

5.4 User Testing

To evaluate the effectiveness and user satisfaction of the Smart Classroom system, a user testing session was conducted involving 25 participants. A simple Yes-or-No questionnaire format was used to make it easier for users to respond clearly and efficiently. This approach allowed quick feedback collection while covering all core functionalities of the system. To make the feedback more meaningful, anyone who answered “No” was asked to give a short reason for their response.

This approach helped gather clear, honest feedback without overwhelming the participants. The responses were mostly positive, providing valuable insight into how well the system performed in a real-use environment. The results showed that the system met all of its intended objectives.

As shown in Figure 5.1, all 25 participants agreed that the system successfully detected their presence in the classroom. This result reflects the reliability of the motion detection feature during the testing session. However, two users initially noted a slight delay. One mentioned the sensor did not pick them up until they moved closer, while another said it responded after a second or two when walking slowly. While the system still managed to detect everyone, this feedback suggests that a minor improvement in sensor sensitivity or placement could help it react more promptly in all situations.

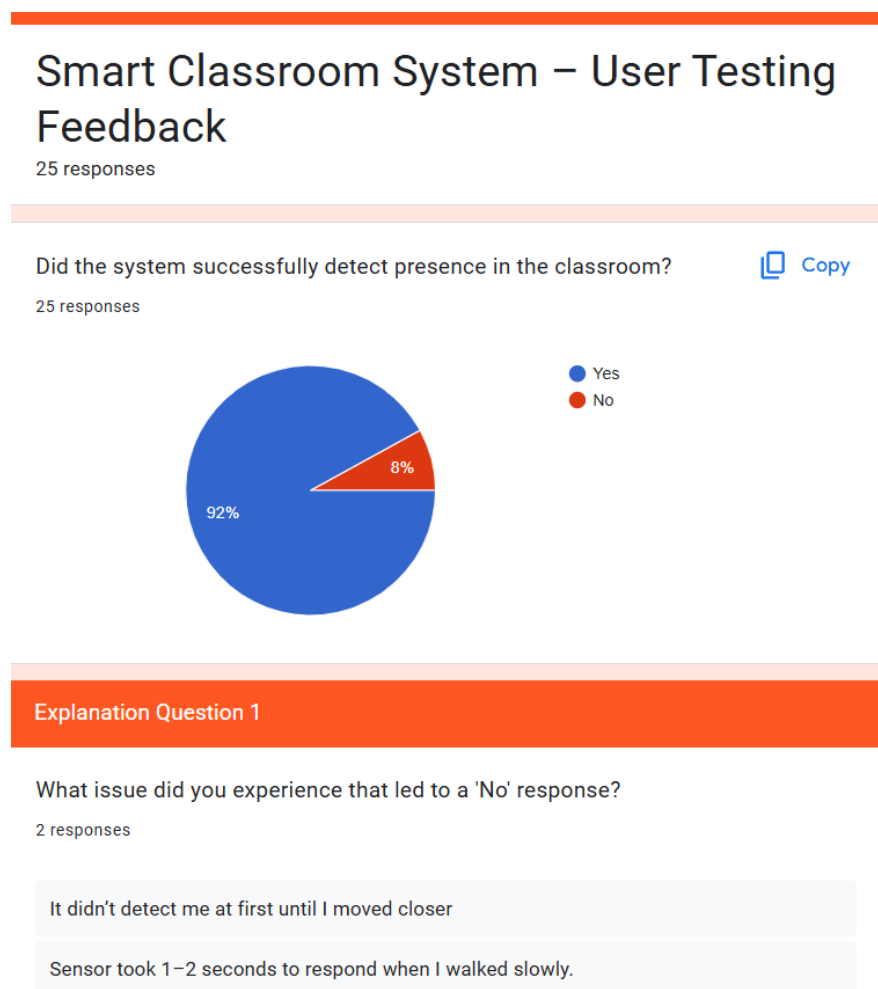


Figure 5.1. User Feedback on Presence Detected.

As shown in Figure 5.2, all 25 participants responded “Yes” when asked whether the fan activated automatically when the temperature went above 28°C. No one reported any problems for this feature. This shows that the temperature-based automation worked exactly as intended, with the DHT11 sensor and fan responding accurately to changes in the room’s heat level. The result confirms that the system can maintain comfort while also supporting energy efficiency.

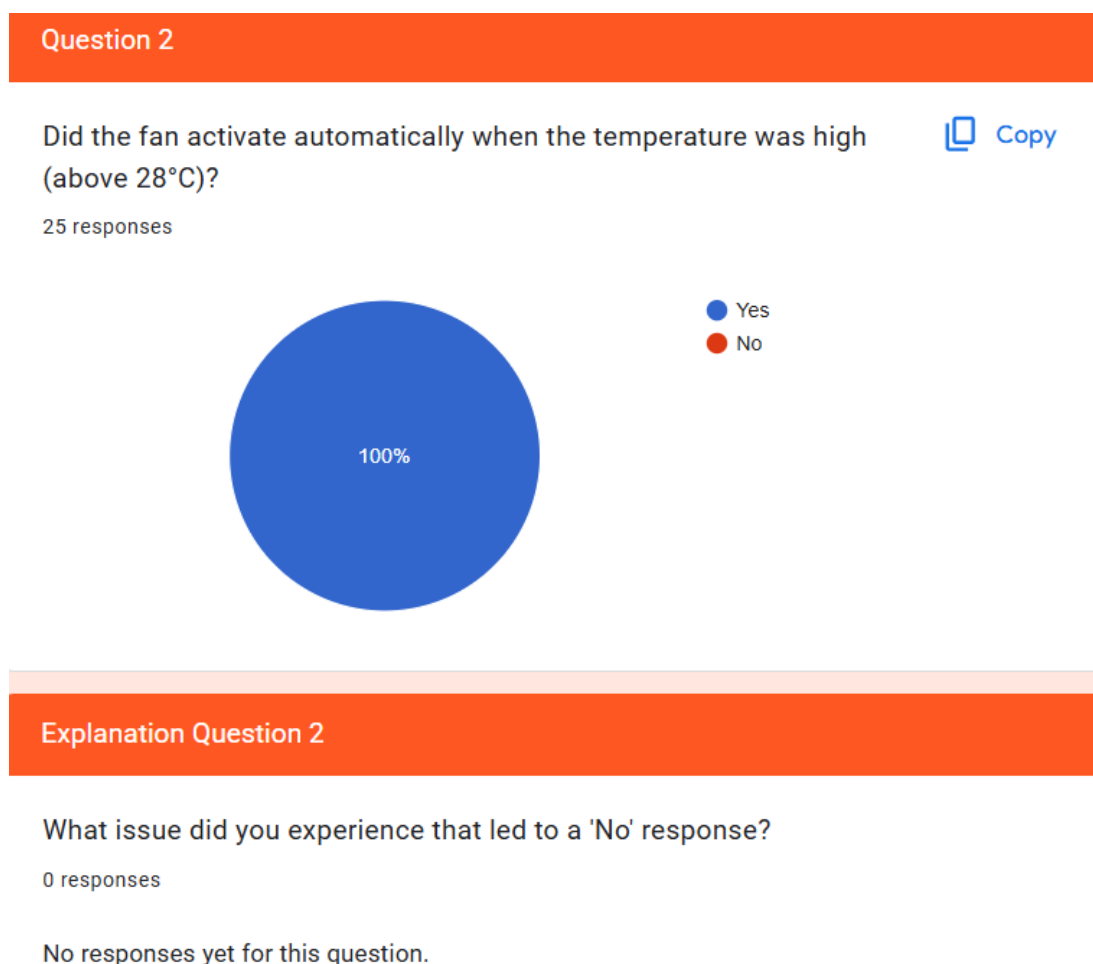


Figure 5.2. User Feedback on Temperature Controlled.

As shown in Figure 5.3, 24 out of 25 participants said the lighting responded appropriately based on how bright or dark the room was. One person, however, answered "No" and explained that although the system worked, they personally preferred the room to be brighter. This feedback doesn't indicate a fault in the system but shows that preferences for brightness can vary from person to person. It could be helpful in the future to give users the option to adjust the lighting intensity to suit their comfort.

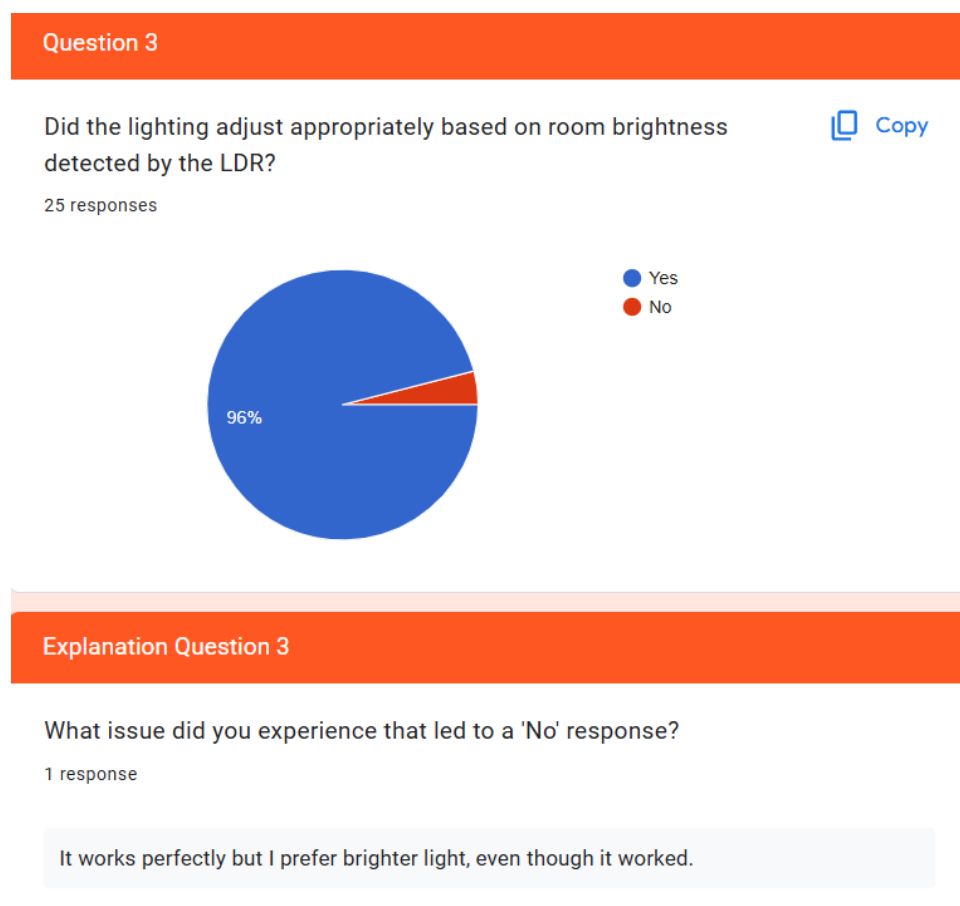


Figure 5.3. User Feedback on Ambient Light.

As shown in Figure 5.4, 24 participants confirmed that the manual control through the Arduino IoT Cloud worked smoothly. One user mentioned that they experienced a slight delay when turning the devices on or off using the dashboard. Although the feature still worked, this

feedback suggests there may be a small lag in response time, especially depending on internet connection or cloud syncing. Overall, the manual override feature was effective and did not create confusion for most users.

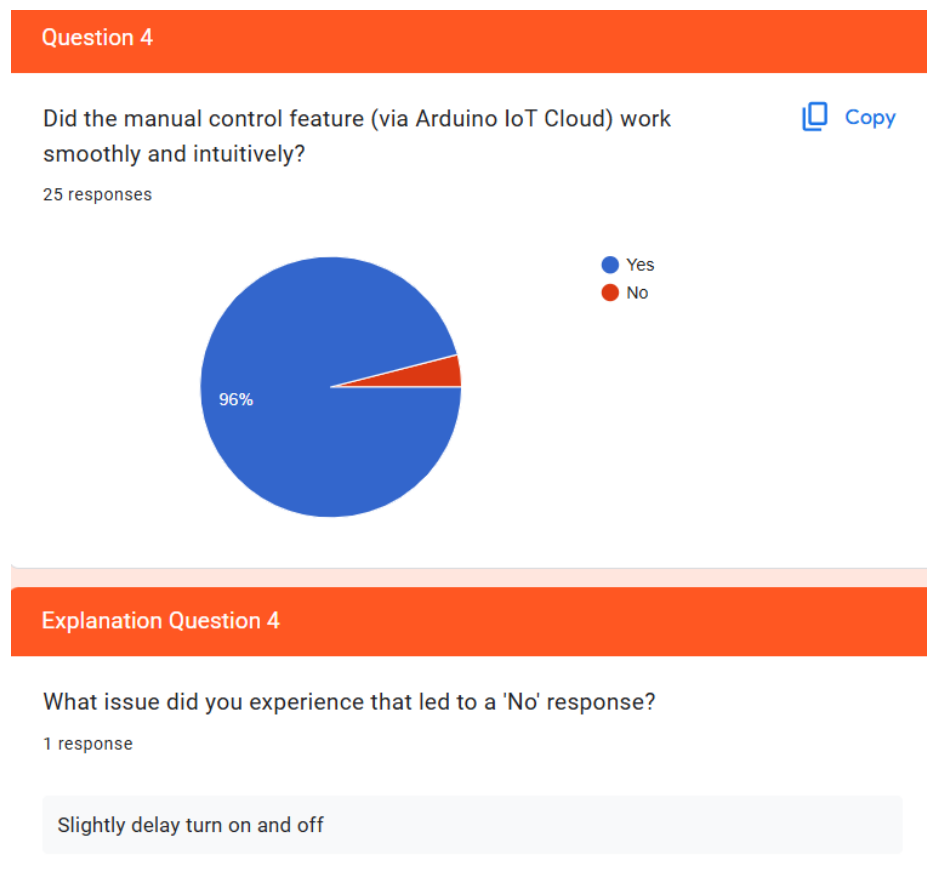


Figure 5.4. User Feedback on Manual Control.

As shown in Figure 5.5, most users found the Arduino dashboard easy to use. Only one participant answered "No" and said they were unsure at first which switch controlled the fan. This response indicates that the interface could benefit from clearer labels or a short guide to help new users navigate it more confidently. Despite that, the general feedback shows that users were able to understand and control the system without major difficulty.

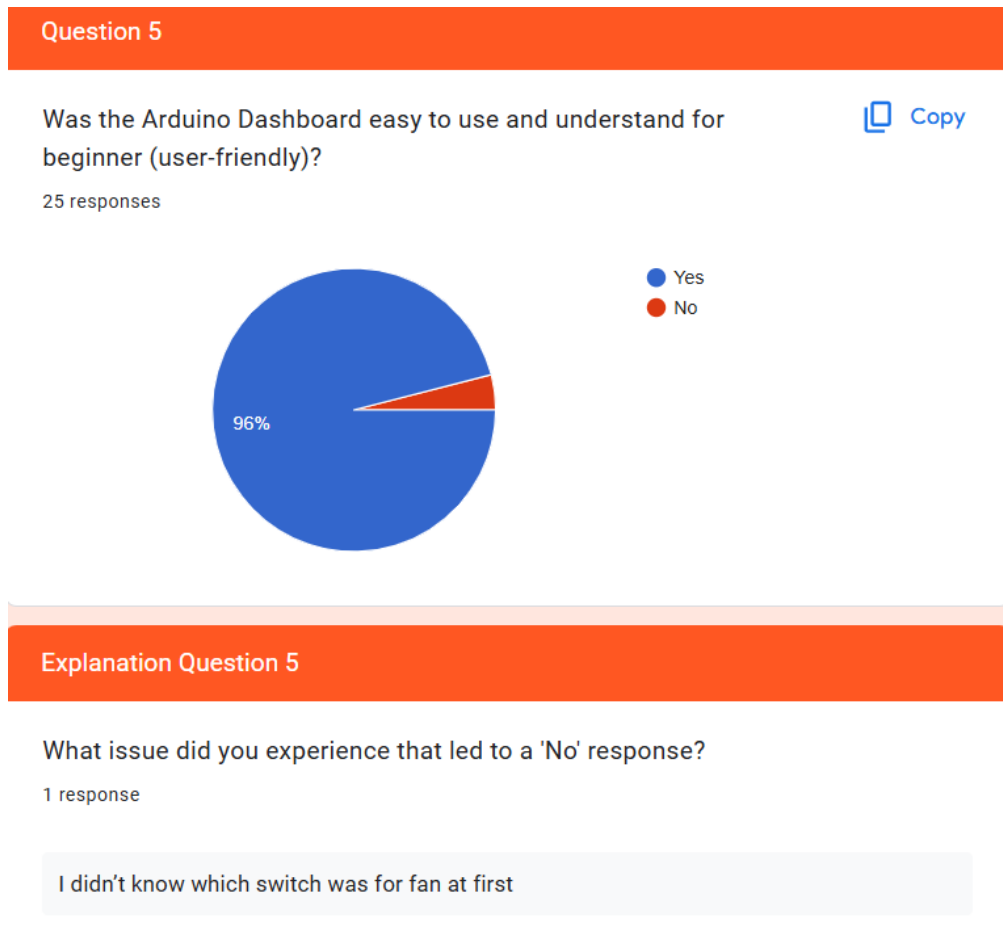


Figure 5.5. User Feedback User Friendly GUI.

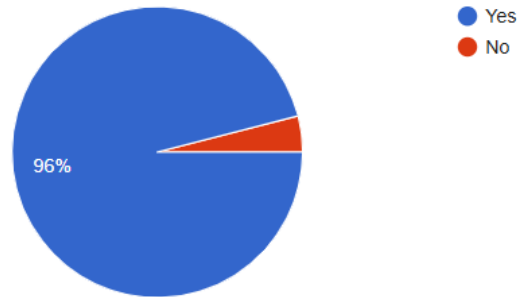
As shown in Figure 5.6, 24 out of 25 participants said they were able to monitor or control the system through the Arduino IoT Cloud without any problems. One user mentioned being slightly confused by the widget layout until someone explained it. This feedback highlights the importance of widget labelling or guidance for first-time users. Even so, the majority found the platform accessible and easy to use.

Question 6

Could you monitor or control the system via the Arduino IoT Cloud without issues?

 Copy

25 responses



Explanation Question 6

What issue did you experience that led to a 'No' response?

1 response

The widget confused me until someone explained it

Figure 5.6. User Feedback on Monitor and Control System.

As shown in Figure 5.7, all participants said they would recommend this Smart Classroom system for actual classroom use to save energy. This unanimous response shows strong confidence in the system's performance and practicality. It also reflects that users believe the system meets real needs in a classroom setting, not just during testing.

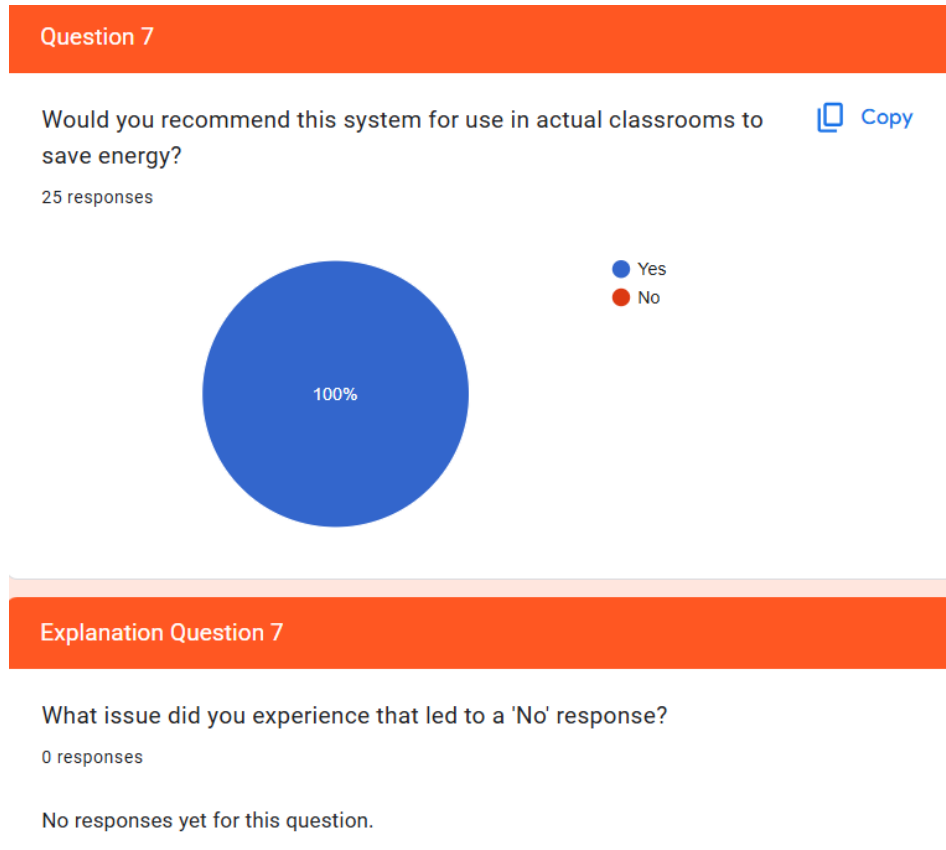


Figure 5.7. User Feedback on Recommendation.

5.5 Conclusion

Overall, the feedback from users was very positive. Most participants found the system easy to use and were satisfied with how it responded to changes in the environment. The features, including motion detection, temperature-based fan control, and lighting adjustment, worked as expected and were well-received. Even the few comments that pointed out minor delays or confusion were constructive and did not affect the overall functionality. These results suggest that the Smart Classroom system is not only practical but also suitable for daily use, especially in efforts to improve energy efficiency in real classroom settings.

Chapter 6: Conclusion and Future Work

6.1. Introduction

This chapter presents the final overview of the project, summarizing the work completed in the development and testing of the smart classroom system. This section discusses the outcomes of the project, and the lessons learned during the implementation process and the areas where the system met its objectives. Furthermore, it explores future work and enhancements that could be made to improve the system's functionality, scalability and user experience in a real-world educational setting.

6.2 Achievement

Table 6.1 below shows the main objectives of the project and what has been successfully achieved:

Table 6.1. Objectives Achievement.

Objectives	Achievements
To develop a wireless system that uses sensors for monitoring the classroom conditions like occupancy status.	A functional prototype was successfully developed using motion and temperature sensors. The system accurately detects student presence and automatically controls devices like light and fan.
To deploy low-cost sensors and IoT (Internet of Things) technology and send real-time updates of electrical appliances in the classroom.	Affordable components, including the PIR sensor, DHT11 sensor, LDR and Arduino Nano IoT board, were used. Real-time status updates were successfully transmitted to the Arduino IoT Cloud for remote monitoring. Although a slight delay was observed during manual control, the system still met the intended

	functionality and responded consistently within acceptable time limits.
To create a detailed installation guide specifically for educational institutions, ensuring easy deployment of the system in classrooms.	A complete step-by-step installation guide, including wiring diagrams and annotated code, was created to help schools or institutions implement the system with minimal technical expertise.
To thoroughly test the performance of the prototype and ensure its functionality meets the project's objectives.	The system was tested under various scenarios to check all functions. Features like motion detection, temperature and light control, time checks and manual control were all confirmed to work properly.

6.3 Contribution

As listed in Table 6.2, this project makes several valuable contributions in both technical and educational areas. The system provides a practical solution for automating energy usage in classrooms by integrating sensor-based monitoring with IoT technology. It helps reduce electricity waste by turning off devices when the classroom is empty and activating them when students are present. This contributes toward environmental sustainability and cost saving in schools.

Additionally, the system improves the way classrooms are managed by introducing real time monitoring through the Arduino Cloud. School administrators can check the status of the room from anywhere, which is especially useful for ensuring energy efficient practices are being followed. The dashboard design was kept simple so that even users with minimal technical experience could operate it. While one or two users found the interface slightly

confusing at first, overall feedback showed that most people could understand and use the system without issues.

The educational contribution lies in the fact that the project demonstrates simple and low-cost components can be used to solve real-world problems. The setup process and code are well documented, making it easier for others, especially students or educators to replicate or expand the project.

Table 6.2. Contribution Overview.

Area	Contribution
Technical	Integration of PIR, DHT11, LDR, RTC, servo motor and Arduino Cloud for automation.
Environmental	Promote energy saving in educational sectors.
Educational	Offers a hands-on learning experience with IoT using simple, low-cost components.
Accessibility	Designed for easy use and installation in schools
Monitoring	Cloud logging allows remote status checking and historical tracking

6.4 Limitation

Despite the positive outcomes, there are still a few limitations that should be acknowledged, as shown in Table 6.3. One of the main issues is the PIR sensor's limited motion detection. It responds best to noticeable movement, which means if a person enters slowly or remains still for too long, the system might assume the room is empty and switch off the lights and fan. A few users during testing mentioned slight delays in detection when walking in slowly or standing still near the edges of the room. This can be uncomfortable for users.

Secondly, the system relies heavily on a stable internet connection. If the Wi-Fi signal is lost, the cloud data will not update, and administrators will not be able to monitor the classroom remotely. This also affects logging and cloud-based control features.

Lastly, the system currently supports only one classroom. For wider use across a school or institution, the system needs to be scaled to handle multiple rooms, which requires more advanced data handling and dashboard support.

Table 6.3. Key Limitations.

Limitation	Explanation
Limited Motion Detection	The system might not always detect subtle movements, which can cause it to mistakenly assume the room is empty.
Internet Dependency	If the Wi-Fi connection is unstable or lost, real-time cloud updates and remote monitoring won't function properly.
Single Classroom Focus	The current setup is designed for one classroom and would need major changes to support multiple rooms or larger deployments.

6.5 Future Work

There are several ways this project can be improved in the future to make it more efficient, flexible and scalable, which are listed in Table 6.4. One major upgrade would be integrating AI or machine learning to analyze usage patterns and optimize energy settings automatically. For instance, the system could learn which hours classrooms are usually occupied and adjust activation times based on past data.

Another important development would be to create a mobile application or a web dashboard for easier control. Instead of depending only on Arduino Cloud, users could receive push notifications, control devices remotely, or see visual graphs of system activity.

Some users found the PIR detection slightly delayed, especially when entering slowly or standing off to the side. This could be improved by repositioning the sensor or installing more than one PIR module to increase coverage and responsiveness.

The system should also support multi-class management, where data from several rooms can be collected and controlled centrally. This will be useful for large schools or colleges that want to adopt the solution across multiple classrooms.

To improve usability, the system could also support offline data storage, allowing it to log data locally when the internet is not available and sync later. Voice control or gesture detection using additional sensors like a microphone or camera can also be explored to make the system more user-friendly. The integration of these features can be planned and executed based on their priority levels, as illustrated in Table 6.5.

Table 6.4. Future Improvement.

Future Work	Details
AI-Based Automation	Learn and adapt to real usage patterns
Mobile App and Web Dashboard	User-friendly control and data visualization
Improved PIR Coverage	Use multiple PIR sensors or better positioning to detect motion more accurately
Multi-Classroom Expansion	Monitor and control several rooms together
Offline Logging	Save data during internet outages and sync later
Enhanced Detection	Add gesture or voice control for accessibility

Table 6.5. Priority of Future Work.

Future Task	Priority Level
AI-Based Automation	██████████ (High)
Multi-Classroom Expansion	██████████ (High)
Improved PIR Coverage	██████████ (Medium)
Mobile App and Web Dashboard	██████████ (Medium)
Offline Logging	██████ (Low)
Gesture or Voice Control	██████ (Low)

6.6 Conclusion

This project successfully designed and tested a smart classroom system that uses IoT and sensors to monitor classroom conditions and control electrical appliances. The system aims to promote energy saving and reduce manual effort by automating fan and light control. The project achieved its goals by using low-cost components, building a real-time cloud monitoring system and validating it through functional testing. While some limitations exist, the system has strong potential for future expansion and real-world implementation in educational environments.

References

- Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11).
- Daly, D., Roth, J., Kokogiannakis, G., McDowell, C., Tibbs, M., & Cooper, P. (2022). Energy consumption in Australian primary schools: Influences and metrics. *Energy and Buildings*, 277, 112549.
- Guerrero-Ulloa, G., Rodríguez-Domínguez, C., & Hornos, M. J. (2023). Agile methodologies applied to the development of Internet of Things (IoT)-based systems: A review. *Sensors*, 23(2), 790.
- Honeywell. (2020). *Honeywell launches first autonomous building sustainability solution to fight rising global energy consumption*. Retrieved from <https://www.honeywell.com/us/en/press/2020/02/honeywell-launches-first-autonomous-building-sustainability-solution-to-fight-rising-global-energy-consumption>
- Kondaveeti, H.K. *et al.* (2021) A systematic literature review on prototyping with Arduino: Applications, challenges, advantages and limitations, *Computer Science Review*, 40, p. 100364. doi: 10.1016/j.cosrev.2021.100364.
- Lutron Electronics Co., Inc. (n.d.). *Franklin & Marshall College case study*. Retrieved December 7, 2024, from https://assets.lutron.com/a/documents/3684466franklinandmarshallcasestudy_english.pdf.

- Medjugorac, M., Susac, S., Maric, P., & Filipovic, T. (2020, January). Importance of Energy Management in Buildings. In *Proceedings of the 31st DAAAM International Symposium* (pp. 0891-0898).
- Mustaffa, N.K. and Kudus, S.A. (2022) ‘Challenges and way forward towards best practices of Energy Efficient Building in Malaysia’, *Energy*, 259, p. 124839. doi: 10.1016/j.energy.2022.124839.
- Nasrudin, M.W. *et al.* (2021) Smart classroom for electricity-saving with integrated IOT (INTERNET OF THINGS) system’, *Journal of Physics: Conference Series*, 2107(1), p. 012019. doi:10.1088/1742-6596/2107/1/012019.
- Wolff, C.E., Jarodzka, H. and Boshuizen, H.P. (2020) ‘Classroom management scripts: A theoretical model contrasting expert and novice teachers’ knowledge and awareness of classroom events’, *Educational Psychology Review*, 33(1), pp. 131–148. doi:10.1007/s10648-020-09542-0.
- Yang, M., Chen, L., Wang, J., Msigwa, G., Osman, A. I., Fawzy, S., ... & Yap, P. S. (2023). Circular economy strategies for combating climate change and other environmental issues. *Environmental Chemistry Letters*, 21(1), 55-80.
- Yefi, P., Menon, R., & Eicker, U. (2024, April). Evaluation of APIs for Data Exchange with Building Management Systems. In *Proceedings of the ACM/IEEE 6th International Workshop on Software Engineering Research & Practices for the Internet of Things* (pp. 1-6).

Appendix

7/7/25, 2:16 PM

Smart Classroom System – User Testing Feedback

Smart Classroom System – User Testing Feedback

Dear participant,

My name is Farhana Hamdan (78538). Thank you for participating in the testing of Smart Classroom System. Please answer the following yes/no questions based on your experience. Your feedback will help to improve the system's performance and usability. Thank you!

** Indicates required question*

1. Did the system successfully detect presence in the classroom? *

Mark only one oval.

- ☐ Yes *Skip to question 3*
☐ No *Skip to question 2*

Explanation Question 1

2. What issue did you experience that led to a 'No' response? *

Question 2

3. Did the fan activate automatically when the temperature was high (above 28°C)? *

Mark only one oval.

- ☐ Yes *Skip to question 5*
☐ No *Skip to question 4*

Explanation Question 2

4. What issue did you experience that led to a 'No' response? *

Question 3

5. Did the lighting adjust appropriately based on room brightness detected by the LDR? *

Mark only one oval.

☐ Yes *Skip to question 7*

☐ No *Skip to question 6*

Explanation Question 3

6. What issue did you experience that led to a 'No' response? *

Question 4

7. Did the manual control feature (via Arduino IoT Cloud) work smoothly and intuitively? *

Mark only one oval.

- ☐ Yes Skip to question 9
- ☐ No Skip to question 8

Explanation Question 4

8. What issue did you experience that led to a 'No' response? *

Question 5

9. Was the Arduino Dashboard easy to use and understand for beginner (user-friendly)? *

Mark only one oval.

- ☐ Yes Skip to question 11
- ☐ No Skip to question 10

Explanation Question 5

10. What issue did you experience that led to a 'No' response? *

Question 6

11. Could you monitor or control the system via the Arduino IoT Cloud without issues? *

Mark only one oval.

- ☐ Yes Skip to question 13
- ☐ No Skip to question 12

Explanation Question 6

12. What issue did you experience that led to a 'No' response? *

Question 7

13. Would you recommend this system for use in actual classrooms to save energy? *

Mark only one oval.

- ☐ Yes
- ☐ No Skip to question 14

Explanation Question 7

7/7/25, 2:16 PM

Smart Classroom System – User Testing Feedback

14. What issue did you experience that led to a 'No' response? *

This content is neither created nor endorsed by Google.

Google Forms