UNIVERSITI POLY-TECH MALAYSIA

UPTM IOT SMART CLASSROOM BOOKING SYSTEM (SCBS)

NURUL AIN SHAZWANI BINTI AHMAD ZAINI

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLICATION DEVELOPMENT

UNIVERSITI POLY-TECH MALAYSIA
Faculty of Computing & Multimedia

UPTM IOT SMART BOOKING CLASSROOM SYSTEM (SBCS)

NURUL AIN SHAZWANI BINTI AHMAD ZAINI
AM2405015967

FYP4132

AUGUST 2025

Declaration of Originality

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc.) has been acknowledged within the main report to an item in the References or Bibliography lists.

I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Copyright Acknowledgement

Signed:  Date:

Office Stamp

Abstract

Classroom management in institutions of higher learning has been usually based on manual scheduling and physical exchange of keys, which causes administrative bottlenecks, overlapping of reservation and a wastage of energy due to utilities that are left on in empty rooms. To help overcome these inefficiencies, the following project suggests the UPTM IoT Smart Classroom Booking System (SCBS) which is a software and hardware solution that will mechanize the process of booking rooms and controlling the environment. The system was created based on an Agile-Prototyping approach, where a responsive Laravel-based web application was built and an Internet of Things (IoT) hardware network was built using ESP32 microcontrollers.

The web dashboard offers a centralized Role-Based Access Control (RBAC) system through which students and lecturers will access real-time room availability, bookings requests, and produce secure access tokens in the form of QR codes. An ESP32-CAM module, which serves as a digital gatekeeper at the hardware level, scans the QR codes generated at the classroom entrance and activates a servo-motor smart lock when a server verifies it. Additionally, Passive Infrared (PIR) sensors and electromagnetic relay modules are installed to check the occupancy of people so that after 15 minutes of inactivity the power to the lighting and air conditioning of the room is turned off.

A series of systematic system testing ensured that the web platform is efficient in avoiding scheduling conflicts, whereas the IoT components are able to avoid the necessity of manual collection of keys and proactively decrease unnecessary electrical usage. Although with some slight limitations as to network dependence and voltage stability during actuation, the SCBS is a very precise, safe and energy-saving prototype. In the end, this project proves that a combination of modern web platforms and low-cost IoT devices can be used to greatly improve administrative functionality and institutional sustainability, with a scalable basis of a "Smart Campus" ecosystem at UPTM.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Alhamdulillah, first and foremost, I would like to express my deepest gratitude to Allah S.W.T for granting me the strength, health, and perseverance to successfully complete this Final Year Project, the Smart Classroom Booking System (SCBS).

This project would not have been possible without the continuous support and guidance of many individuals. I would like to extend my sincerest appreciation to my esteemed project supervisor, Norfazlina binti Johar, for their invaluable advice, constructive feedback, and endless patience throughout the development of this project. Their expertise and encouragement were instrumental in helping me navigate the technical challenges of integrating the Laravel web application with the IoT hardware.

I also wish to express my gratitude to the Faculty of Computing & Multimedia at Universiti Poly-Tech Malaysia (UPTM) for providing the educational foundation, resources, and platform necessary to undertake this research. A special note of thanks goes to the Dean and the Academic Affairs (AAS) department for their willingness to participate as the primary client during the system evaluation phase. Their insightful feedback greatly shaped the outcome of the SCBS.

Furthermore, I am deeply thankful to the lecturers, staff, and the 50+ UPTM students who generously volunteered their time to participate in the User Acceptance Testing (UAT). Your practical feedback and enthusiastic interaction with the prototype provided the critical data needed to validate this system.

Last but certainly not least, I want to express my profound gratitude to my beloved parents and family members. Thank you for your unwavering moral support, understanding, and countless sacrifices throughout my academic journey. To my friends and classmates, thank you for the shared late nights, the brainstorming sessions, and the constant encouragement.

Thank you all.

1 INTRODUCTION

1.1 Introduction

With the cost of operation and the issue of sustainability on the increase, there is mounting pressure on the universities to maintain efficiency in managing classrooms and other physical resources. Some of the problems that reduce productivity and lead to costs include schedule conflicts, empty classrooms, administrative time-wastes, and energy waste caused by lights and air conditioners left on in unused rooms (Noor, Atlam, Almars, Noor, and Malki, 2023; Mohd Wafi Nasrudin et al., 2021).

The modern Internet of Things (IoT) technology, including automatic controls, occupancy sensors, and real-time dashboards, can potentially address these problems. A reservation system can be integrated with IoT to reduce the operating costs, enhance room occupancy, reduce the workload of employees, advance responsibility and visibility, and provide information to guide space design in the future (Abassi, Ahmad, and Iqbal, 2021).

In this project, a Smart Classroom Booking System (SCBS) is proposed to University Poly-Tech Malaysia (UPTM). The SCBS will also contain online reservation, automatic energy control, access control, real-time availability, and administrative analytics that will make classroom administration more efficient, sustainable, and easy to use.

1.2 Project Background

The suggested Smart Classroom Booking System (SCBS) of UPTM is a web-based application with a flexible approach to mobile users, in which the browser access is the central point of focus. The front end will be built with HTML5, CSS3 and JavaScript to develop a user-friendly interface that is compatible with diverse screen sizes. The back end will be developed based on the Laravel PHP framework and a MySQL database to provide occupancy monitoring with the use of the IoT and a user authentication process to ensure data protection. The combined security systems in Laravel like CSRF protection and role-based access control will also contribute to the maintenance of system integrity and the prevention of unwanted system access.

The SCBS will enable real-time booking, automated energy control, and comprehensive administrative dashboards to combine the state-of-the-art front and a stable Laravel back end and IoT-based automation. Ultimately, this solution aims to facilitate a smarter, more environmentally-friendly, and efficient campus life through reducing the number of people who have a manual task, enhancing transparency, decreasing the amount of energy waste, and ensuring safe access to classrooms.

1.3 Problem Statement

A problem statement is a concise overview of an existing problem that highlights the existing state of affairs, the absence of a solution or a weakness, and the importance of finding a solution (Alavi and Leidner, 2018). By explaining who, what, where, when, why and how the problem affects operations, it becomes possible to set project goals (5W1H). Classroom management is important to University Poly-Tech Malaysia (UPTM) so that it can have smooth academic operations and maximize on the use of resources. However, the current manual reservation system, lack of immediate availability, automated occupancy verifications and unauthorized use of classrooms have contributed to scheduling, energy wastage, management inefficiencies, and increased operational expenses (Rahman et al., 2021). A smart, unified system is required to improve sustainability, transparency, and efficiency in general.

1.3.1 Inefficient Manual Booking Process

At present UPTM manages classroom booking with a request form that is written by hand or communicating directly with the administrative staff. This is a manual process, which is likely to cause human error, thus there are frequent cases of double-booking and booking conflicts. More so, it brings about a lot of delay in approval and transparency and requires the user to be physically present during office hours and puts pressure on the administrative staff. These are the archaic approaches that do not have the digital infrastructure required to handle resources in an efficient and modern manner (Garcia & Lee, 2022).

1.3.2 Large Energy Wastage in Empty Classrooms

One of the direct and expensive results of the existing system is the mass wastage of electrical power. In the absence of any automatic connection between the booking status of a classroom, its current occupancy and its environmental systems, lights, fans, and air conditioning systems are operated in the same way, hours long, in totally vacant classrooms. Such lack of connectedness between the agenda and the physical reality implies that energy is used without any advantage. Studies show that schools that do not use smart facility management software can squander up to 30 per cent of their power in unused areas (Rahman et al., 2021), causing unnecessary high operational expenses and unnecessary expansion of the environmental footprint of UPTM.

1.3.3 Unavailability of Information about the Real Time Classroom

One of the current challenges that UPTM faces is the lack of the latest data on the availability of the classroom. The schedules are now displayed on paper in front of the classrooms. But since other classes might change their session to online classes without being notified in the notice board, this process will often lead to confusion. Both the lecturers and students are left in the dark about.

The fact that a classroom is in use, thus resulting in time wastage and inefficient resource utilization. According to Alavi and Leidner (2018), access to up-to-date and accurate information is important in making well-informed judgements in an academic environment. Without a centralized digital system that allows monitoring and updating classroom status, UPTM is still grappling with effective utilization of its learning spaces (particularly during the peak of academic days).

1.3.4 The lack of automated occupancy monitoring

Another problem that restricts UPTM is the absence of automated equipment to monitor classroom occupancy. Now it is difficult to maximize classroom management and save on energy since there is no technical method used to check whether a room is in use or not. The researchers state that Internet of Things (IoT)-based occupancy sensors can effectively detect the utilization of a space and adjust energy consumption, including air conditioning and lighting, based on their use (Sharma and Kumar, 2020). Without this kind of devices, the rooms can still be air-conditioned and lighted even when not in use, which leads to unnecessary wastage of energy. Such an imbalance highlights the importance with which UPTM should consider sustainable practices and the necessity to implement the Internet-of-Things-based monitoring to facilitate the efficient and sustainable management of resources.

1.3.5 Unauthorized Classroom Usage and Accountability of Lack of Access

Another issue that UPTM manages is unauthorized classroom use. In some cases, the classes are occupied by students or groups without legal permission as it is not possible to book and keep track of it. This also complicates the control of usage by administrators in addition to disrupting academic schedules. Henriquez (2020) argues that inefficiencies in access control operations and resource misuse may occur in institutions of higher learning because they have yet to establish formalized access control procedures. There is the need to have in place a secure and permission-based reservation system to provide accountability and proper use of the university facilities.

1.4 Project Objectives

The objective of this project is to establish a Smart Classroom Booking system (SCBS) in the University Poly- Tech Malaysia (UPTM) to enhance the sustainability, efficiency, and transparency of the classroom management. To achieve this goal, the following goals are laid out in the project:

1.4.1 To develop a centralized online booking platform.

UPTM is dealing with classroom bookings manually at the current time as a result, it is often inefficient, sluggish, and results in multiple bookings. It entails establishment of a secure, web based system through which lecturers and students will be able to see availability, make reservations and get instant confirmations on any device. This will computerize and simplified the whole work process by removing all forms of creating a duplicate booking by automatically checking the conflicts, shortening approval time, and reducing the administrative load of manual scheduling by a considerable margin.

1.4.2 To implement IoT-based automated energy control

To address the issue of Significant Energy Wastage (1.3.2). It involves the combination of IoT sensors, which are PIR motion sensors, to the electrical system of the building through microcontrollers and relays. The system will automatically switch-off lights, fans, and air conditioners in the classrooms after no occupancy is detected and back to the pre-set conditions when a booking is in progress. This automation will save power and cost of operation directly.

1.4.3 To Offer Updates on Availability in Real Time

The unavailability of real time classrooms has led to confusion and underutilisation of the facilities. To attain this, a centralized and up-to-date portal displaying classroom availability on real time should be availed. Real-time information can be offered to lecturers and students in order to make decisions in terms of class scheduling, group discussions, and other academic activities in a more effective way. The real-time updates will also enable administrators to monitor usage patterns as well as utilise classroom space at peak periods.

1.4.4 To Construct a Dashboard for Administrative Monitoring

The administrative personnel now find it hard to monitor reservations, track the use of classrooms, and determine the illegal use of classrooms. Therefore, the aim of the project is to develop a comprehensive monitoring dashboard to the administrators. This dashboard will display the statistics about the energy consumption, booking trends, and hotel occupancy which will be effective in resources management. The system enhances transparency, accountability and informed decision-making in the classroom allocation and facility planning processes because the system provides actionable data to administrators.

1.4.5 To establish a QR code-based access control system

Besides derailing scheduled operations, illegal classroom practices cause administrative inefficiencies. To offset this, the SCBS will use safe access control measures, using digital approval systems, smart lock or QR code check-ins via email upon reservation. This is to ensure that there are only authorized persons who are allowed to access classrooms. This objective reduces conflicts, promotes responsible usage and also ensures compliance with set academic schedules besides safeguarding university resources.

1.5 Scope and Target User

In order to ensure that the Smart Classroom Booking System (SCBS) meets the needs of the University Poly-Tech Malaysia (UPTM), the scope of the project will provide the activities that will be undertaken, who will use the system, and what will be included. It clarifies the boundaries of the project and prevents unnecessary extensions which exceed the objectives of the project.

1.5.1 Project Scope

All activities necessary to provide the fully functional SCBS are included in the project scope. This entails the overall process of the system planning, designing, development, testing and deployment. This mission will be achieved by initially defining and reporting certain project objectives and user needs by interacting with all the essential stakeholders such as administrators, lecturers and students. The key deliverables are the functional web application itself, source code, a tested database, the combination of APIs, system documentation and a final project report. The most prominent ones include frontend and back-end development, database design, API development, and IoT integration, and its milestones will be in line with the FYP 1 (requirements and analysis, about 14 weeks) and FYP 2 (implementation and testing, about 14 weeks) timeline.

Technical implementation will entail building of a responsive frontend using HTML, CSS, and JavaScript so that it can operate perfectly both in a desktop and a mobile platform. LarvalPHP framework will be used to create the backend which will be coupled with MySQL database to manage data effectively. RESTful APIs that will be custom built will support communication between the front and the backend and will be thoroughly tested with such tools as Postman.

Moreover, the project is to be provided with the implementation of a QR code-based access control system, and IoT-based occupancy sensors that can enhance the automation of energy management. It is also necessary to mention that the project will be oriented on the development of the software and prototype-level IoT integration as a proof of concept instead of assuming any major physical infrastructure improvements or massive hardware implementation throughout the entire campus.

1.5.2 Product Scope

The SCBS is a web-based software that works with both desktops and smartphones at the same time due to its responsive design. It improves classroom management because lecturers and students have an opportunity to make a reservation and see the real-time room availability using any device. The system involves IoT devices, such as ESP32 and ESP32-CAM boards, PIR motion and light sensors to detect the presence of occupants in the classroom and control the air conditioning and light, therefore, minimizing energy wastage. Each confirmed reservation generates a QR code which can be scanned by users at the entrance of the classroom by an ESP32-CAM or QR reader to ensure safe entry and ensure that only authorized individuals enter the room. Accessing a centralized dashboard also allows administrators to track reservations, track the use of classes, and analyze information about energy consumption. Even though the platform is initially designed to serve the purpose of classroom management, it is scalable and might be further expanded in the future to serve more University buildings.

1.5.3 Target User

The SCBS targets three major user groups of all UPTM faculties. Undergraduate and graduate students are going to use the project to book classrooms to study, complete their group projects and extracurricular activities. The increased efficiency in academic staff and lecturers scheduling lectures, tutorials, and academic activities will ensure that scheduling issues with student reservations are minimised. In a bid to enhance allocation of resources, administrators and facilities managers particularly those in the ICT and Facilities Department will verify the use of energy, reservations, monitor the occupancy of classrooms, secure access, and analyse the statistics of room occupancy. The SCBS ensures smooth, efficient, and sustained classroom management system to the benefit of the entire UPTM community by taking into consideration the needs of these users.

1.6 Overview of This Report

This study comprises three comprehensive chapters that ensure the entire development lifecycle of the Smart Classroom Booking System (SCBS). This study consists of three detailed chapters that comprise the entire development lifecycle of the Smart Classroom Booking System (SCBS).

Chapter 1: Introduction This chapter gives the background of the project comprising of the background context of the challenges in classroom management in UPTM, the problem statement, the project objectives and scope, and intended users.

Chapter 2: Methodology, System Design and Implementation gives an elaborate report of how the project was implemented. It starts with literature review of pertinent literature and existing systems to derive the theoretical background and gaps in research. Thereafter, the chapter identifies the development methodology, tools, procedures and functional as well as the non-functional system requirements. Lastly, it outlines the system design and implementation including the architectural approach and the process of development.

Chapter 3: Results, Testing and Conclusion will include the results of the project, beginning with the analysis of the results and a system evaluation. It encompasses in-depth testing procedures and evaluation of the offered solution concerning the initial aims, and it ends with the project results and the suggestions on the further work.

2 LITERATURE REVIEW

2.1 Introduction

This chapter offers a critical analysis of the body of research and the state of technology that are pertinent to the development of Smart Classroom Booking System (SCBS). Its major topics are the development of smart classroom systems, the strong and weak aspects of the existing online booking services, the automation capability of IoT and occupancy data, and the importance of responsive web design to make the system user-friendly. This analysis will find out effective solutions, highlight certain market gaps, including the absence of connection between booking and energy automation, and justify the technology decisions of the SCBS by analysing past studies and existing systems. In order to address the specific issues at UPTM, this foundation will ensure that the project is founded on established expertise and provide innovative integration.

2.2 Investigation

Research on the relevant areas of issues, appropriate solutions, and supporting technologies to manage the challenges of classroom management and energy efficiency in tertiary institutions is very important. This paper will discuss four key areas, which include smart classroom systems, online booking systems in education, IoT and occupancy monitoring, and responsive web and mobile integration. The subjects directly relate to the design of the Smart Classroom Booking System (SCBS) of University Poly-Tech Malaysia (UPTM), which could provide a guide in the implementation of the project and information on best practices.

2.2.1 Smart Classroom Systems

The change in paradigm of smart classroom systems is a break of tradition when it comes to traditional classroom and its advancement into technological-enhanced classrooms that make operations and pedagogy more effective. As Sharma and Kumar (2020) note, such systems are software and hardware integrated to create data-driven and interactive environments. In addition to the fact that it only increases the involvement of the students, there are also significant operational rewards. Indicatively, Lee and Choi (2019) demonstrated that administrative overheads could potentially reduce by up to 25% when educational institutions automated their scheduling and resource tracking processes by using smart classroom technologies.

However, there is a huge literature gap. Most of the studies on smart classrooms such as the works by Garcia and Lee (2022) treat facility management (Lighting, HVAC, access control) as a separate area of research and focus mostly on instructional technologies such as interactive whiteboards and student response systems. The SCBS project will bridge this gap by making facilities management one of the key components of the smart classroom ecosystem. The SCBS embraces a comprehensive "smart classroom" concept that incorporates both learning and facility optimization a synergy that is sometimes disregarded by including automated energy control and access management straight into the booking procedure.

2.2.2 Online Booking Systems in Education

In educational institutions, online booking platforms are now a common method for managing shared resources. Strong capabilities for preventing double booking, controlling user rights, and offering calendar integrations are provided by platforms such as Skedda and Classroombookings (Patel, 2021). Their extensive use highlights how ineffective manual, paper-based solutions are.

These commercial systems have a major drawback despite their advantages, they are largely scheduling-centric. Sustainability measures must be incorporated into the next generation of instructional tools, according to Mohammed et al. (2020). Existing systems can notify user when a room is reserved, but they are unable to ascertain whether the room is truly occupied or regulate the atmosphere in response to that information. As a result, there is a gap between the online schedule and the actual classroom environment.

The SCBS deals with this restriction immediately. It is a platform for occupancy-driven resource management in addition to being a booking system. The SCBS can be configured to automatically release a reservation if no occupancy is detected 15 minutes after the start time (addressing no-shows), in contrast to a system like Skedda that would hold a reservation for the entire booked period. More importantly, it can signal IoT devices to turn off lights and air conditioning, meeting the energy conservation goals recommended by Mohammed et al. (2020).

2.2.3 IoT and occupancy monitoring

Between the digital reservation system and the actual classroom setting, the Internet of Things (IoT) offers the vital connection. IoT sensors are a tried-and-true and reasonably priced way to detect occupancy in real time, especially Passive Infrared (PIR) motion sensors. PIR-based systems might identify human presence in classroom settings with over 95% accuracy, according to research by Chen & Ahn (2021).

It is commonly known that this data can be used to save energy. According to a groundbreaking study by Rahman et al. (2021), occupancy-based automation could help educational institutions cut their HVAC and lighting energy use by 20–30%. According to their research, low attendance, early dismissals, or cancellations leave classrooms empty for a sizable amount of their allotted time.

The innovation of the SCBS lies in its directional data flow. It employs occupancy data to inform the booking system itself in addition to IoT sensors for automation. The system can instantly change a reserved room's status to "available" so that other users can reserve it if sensors show that the room is not there. Our project's primary differentiator is this dynamic approach to resource allocation, which has been confirmed by Rahman et al.'s (2021) energy savings findings.

2.2.4 Responsive Web and Mobile Integration

The possibility of accessing systems should be convenient and universal regarding contemporary education. In contemporary times, the strategy of Responsive Web Design (RWD) is best practice. The approach enables a single web page to adapt its design to a variety of screen sizes, such as desktop, tablet and smartphone (Marcotte, 2011). This is not an option but a necessity of a system like the SCBS. Although the administrators may prefer the bigger interface of a desktop dashboard, instructors and students have no choice but to use their mobile phone to check the availability and make reservations in the process.

Such user interfaces are easier to build with frameworks that include HTML5, CSS3, and JavaScript, including Tailwind CSS or Bootstrap. Deciding to use a web-based strategy instead of a native mobile application is calculated. It reduces barriers to adoption by ensuring that the system is cross-platform compatible, and users will not be required to download and update an application (Smith, 2022). The SCBS will use these technologies to ensure that the experience of all device users is smooth to attain high utilisation rates among its target consumers at UPTM.

2.3 Related Works

The development of this project will rely on four available classroom booking systems that can offer some valuable information regarding the features, usability, and limitations. The references such as Classroombookings, Skedda, Smart Classroom Management System (SCMS) as well as IoT-Based Energy Conservation Smart Classroom System will be used to determine the key functions as well as pointing at the areas that can be improved, especially when it comes to the use of IoT to make classrooms real-time monitoring and automation.

2.3.1 Classroombookings

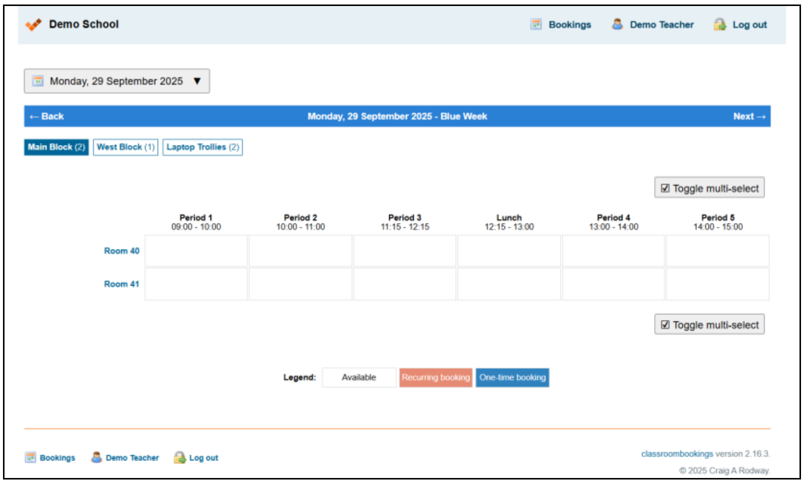


Figure 2.1: Classroombookings Schedule (Rodway, 2025)

Classroombookings is a basic structure of online scheduling at the educational place. As a self-hosted, open-source web application, it is a feasible alternative to organizations that strive to departure the use of traditional, paper-based reservation systems to digital management. The platform design is focused on easy room booking and has a clear organizational structure.

Figure 2.2: Classroombookings Filled Schedule (Rodway, 2025)

The system operates using a centralized booking system that uses a simple to use weekly or daily schedule display. The users have the opportunity to select rooms and use open time slots to open a New Booking form, which requires manual recording of event details including title, description and length. In this interface, a multi-level user role system is provided with special interfaces of Administrators, Teachers and Students to ensure accessibility of different users by different levels.

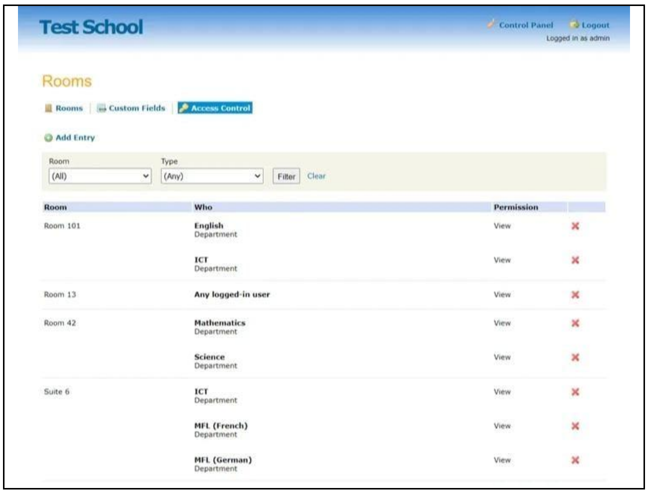


Figure 2.3: Classroombookings Rooms List (Rodway, 2025)

Key features include automatic conflict prevention and shading of booked time slots, possibilities to make a booking over the semester in advance, and automatic email notifications of booking confirmations and notifications. The administrative interface includes the basic reporting functionality in the form of CSV exports, allowing the easy analysis of the patterns of room utilization. Such qualities make it particularly suitable to those organizations that require simple digital scheduling and do not have complex requirements.

Still, Classroombookings has significant lapses in modern smart classroom management. The system is fully operated in the digital realm where room status is defined solely by schedule entries rather than the physical state. It lacks any IoT integration functionality, so occupancy tracking in real-time and automated control of energy are impossible. Also, the system cannot automatically enforce booking policies, such as the cancellation of no-show reservations, so it requires constant human administrator intervention. In the case of the SCBS, Classroombookings provides a stable platform upon which fundamental booking features can be utilized and the significance of integrating physical automation aspects discloses in an excellent manner.

2.3.2 Skedda

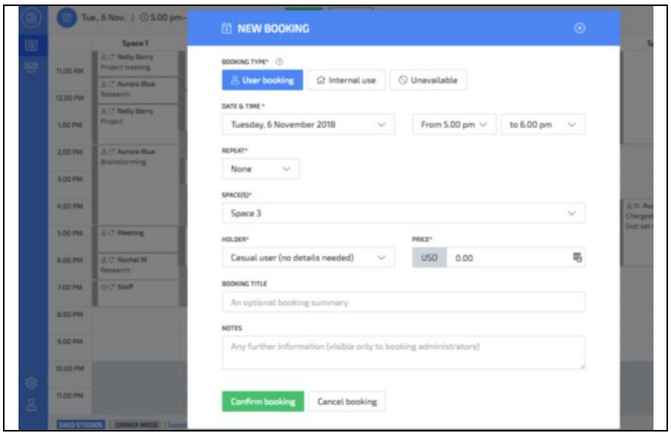


Figure 2.4: Skedda Booking System (Skedda, 2025)

Skedda is a commercial and cloud-based booking and management solution used to book and manage venues, including classes, coworking space and conference rooms. It offers a user-friendly interface and a wide range of administration capabilities, and is the standard of the industry in terms of sophisticated space management solutions. Even though it was designed in a range of settings, it is also used in educational institutions. The platform is more focused on visual interactions and easy to use workflows and light training.

The central booking system of the place focuses on an entertaining experience with floor plan flexibility and easy Book Now option. The responsive design will provide users with the ability to explore available spaces and make reservations in a few clicks, as it ensures the smooth functioning of both desktop and mobile devices. The interface of the venue settings enables the administrator to establish advanced booking restrictions such as restrictions applying to the maximum number of booking periods, advance booking restrictions and buffer time between appointments. This is not the only personalisation choice of the platform.

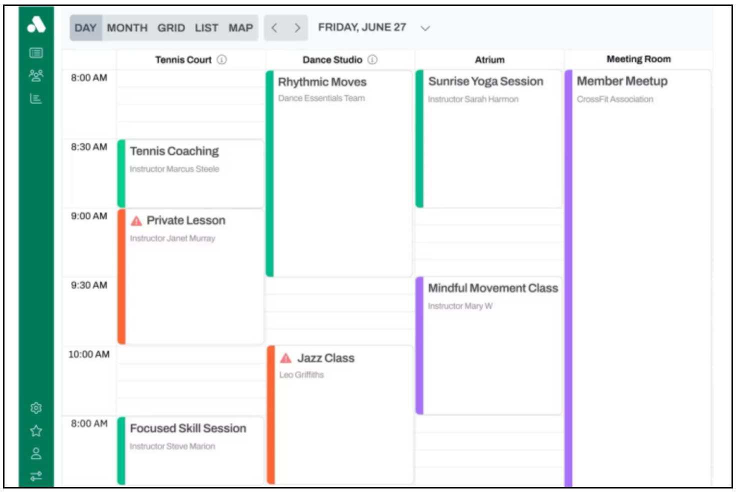


Figure 2.5: Skedda Synchronization (Skedda, 2025)

A good example of its integration is the fact that Skedda integrates well with major calendar apps like Gmail Calendar and Outlook. By using an elaborate dashboard, which allows real-time monitoring and control of all space booking, this feature will ensure users have updated calendars on all platforms as well as offer administrators with overall control. The platform is especially attractive to organisations with low training and adoption requirements as it focuses on user experience.

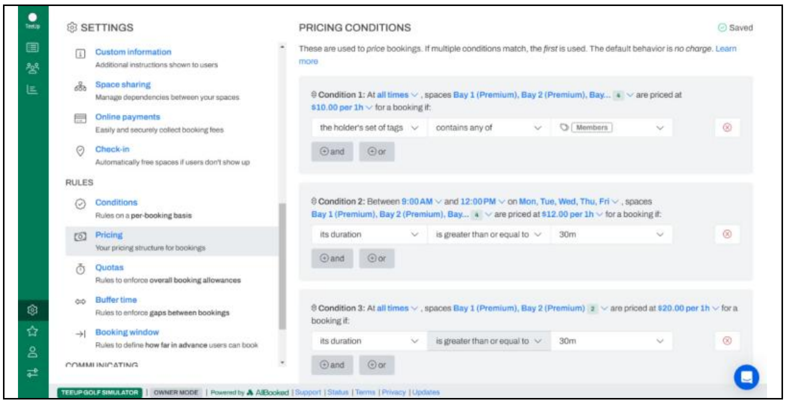


Figure 2.6: Skedda Advance Features (Skedda, 2025)

Even with these features, Skedda has serious drawbacks in the implementation of smart classrooms. The platform is a closed ecosystem in terms of automation features, as it does not have an in-built capability to integrate the IoT devices and support the two-way exchange of data. This will not allow occupancy control in real time and automatic control of environmental systems like lighting and air conditioning. Also, the advanced features pricing model as a subscription can be a challenge to the budgets of some institutions. In the case of SCBS project, Skedda sets high bar on the design of user interface and administrative capabilities and indicates clearly that there is a necessity to provide built-in physical resource management functionality.

2.3.3 Smart Classroom Management System (SCMS)

Figure 2.7: Smart Classroom Management System (Ali, 2022)

The Smart Classroom Management System is an educational way of bringing together digital management and physical access control as a way of balancing out between classroom security and scheduling system. Designed as a research prototype, SCMS illustrates the viable implementation of linking the digital permissions and usage of physical spaces via technological advancement.

Figure 2.8: SCMS System QR Generation (Ali, 2022)

The most significant innovation offered by the system is its unified workflow that puts an online booking into real access. Once the reservation is confirmed, the user will be allowed to open a Generate QR Code option that generates unique, temporary authentication code. There are special readers scanning the codes on the entrances of the classrooms, which usually have ESP32-CAM modules and compare the permissions with the primary booking database on the fly. Not only does this keep the physical access under control but it also changes the room status between Booked and In Use allowing much better occupancy information as compared to fixed scheduling systems.

One of the biggest advantages of this is that the access control system has the in-built functionality of automated attendance tracking. Any QR code scan will serve as a proven attendance record, eliminating the need to use a separate rollcall and providing an accurate account of usage. The system is effective in addressing the issue of the unauthorized use of the classrooms by only allowing those with valid and current reservations, thereby protecting any scheduled academic activity.

Nevertheless, there is a significant weakness of SCMS in its biased application of smart classroom concepts. Although it is able to combine access control with the booking management, the system is totally ignoring the element of energy management. It does not have integration with environmental control systems such as lighting and air conditioning such that they can proceed running independent of the room utilization. This is an important opportunity that has been wasted in energy saving in learning institutions. In the case of SCBS project, SCMS confirms the feasibility of implementing the physical access control with the digital scheduling as well as emphasizing the necessity to include the energy management elements.

2.3.4 IoT-Based Energy Conservation Smart Classroom System

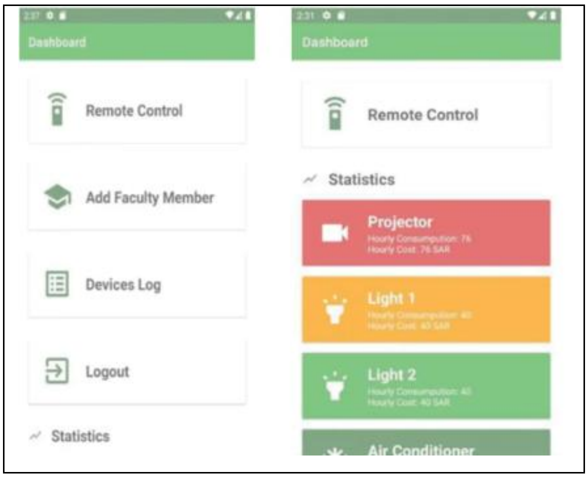


Figure 2.9: IoT-Based Energy Conservation Smart Classroom System (H. Noor, 2023)

The IoT-based Smart Classroom System of Energy Conservation assumes the hardware-based approach to managing resources, focusing on the energy efficiency through automated monitoring and control systems. This study-based method demonstrates that the IoT technologies have significant potential to reduce the operation cost and environmental impact within the educational settings.

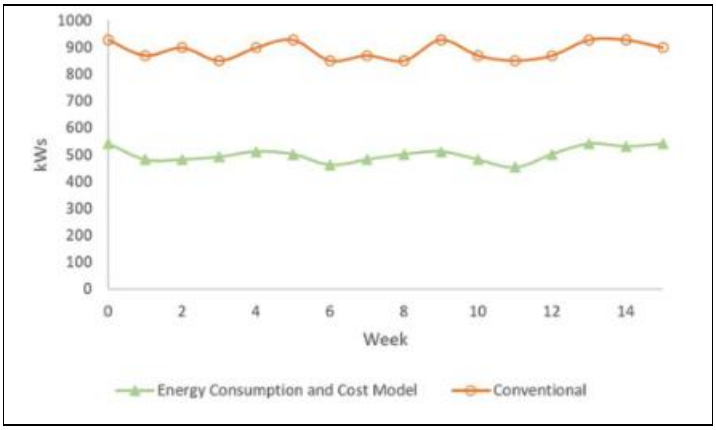


Figure 2.10: IoT-Based Energy Conservation Analysis (H. Noor, 2023)

The system operates on sensor-based automation structure that continuously monitors the occupancy of classrooms using Passive Infrared (PIR) motion sensors. When no movement is detected over a preset period, typically 10-15 minutes, relay switches are operated by a central microcontroller to switch off lighting and air conditioning and other electrical devices. This reactive approach is very effective in minimizing waste of energy in empty spaces as studies indicate that there may be a 20-30 percent reduction in energy consumption.

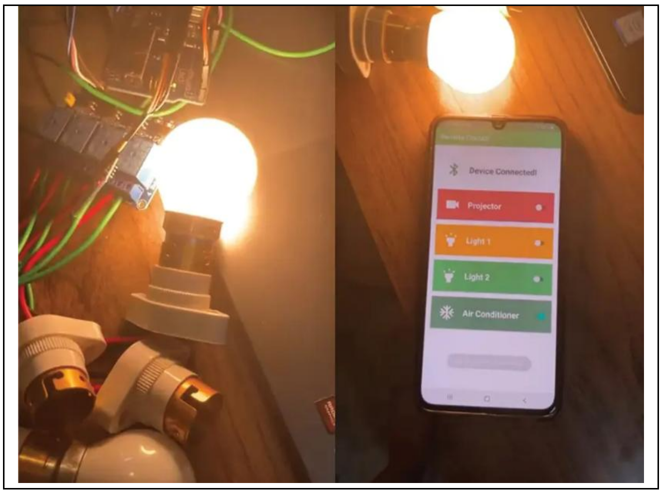


Figure 2.11: IoT-Based Energy Conservation Smart Controller (H. Noor, 2023**)**

To the motion detecting systems, more advanced systems have additional sensors like Light Dependent Resistors (LDRs) to automatically switch artificial lighting depending on Natural light supply, temperature and humidity sensors to enhance climate control. The system demonstrates excellent hardware integration that will efficiently link sensor input to actuator output to evolve an environmental management responsive solution. This is a technical framework that provides a solid foundation of automated resources-saving within the educational environment.

The major disadvantage of the system is that it does not have operational disconnection with institutional scheduling systems. In the absence of integration with the booking platforms, it cannot recognize planned and unscheduled activities and prepare classrooms before the planned activities take place. This limitation presents a barrier to a great user experience because it can not ensure rooms are ready to be comfortable in time before classes begin. Furthermore, the system puts much focus on presence rather than permission, a factor that makes it ineffective in addressing issues of unauthorized occupancy of space. In the case of the SCBS project, this system provides the necessary technical control of the IoT-based automation and clearly demonstrates the need to combine these features with detailed scheduling and authorization systems.

2.4 Comparison

All of the four related systems that had been previously analysed are excellent at some aspect but fail at providing a holistic smart classroom management system. The detailed analysis of all the works that were considered in Section 2.3 contains the clear identification and discussion of the aspects that are analysed in Table 2.1 below: basic online booking, conflict prevention, and IoT connection. To summarise these findings, the table gives a graphical account of how the SCBS integrates and expands these current solutions. The combination of these points makes our project look particularly special in terms of its value proposition, as it compares the features of existing systems with the proposed feature set of the SCBS.

Table 2.1: Feature Comparison of Existing Systems and the UPTM SCBS

Features	Classroombookings	Skedda	SCMS	IoT Energy System	UPTM SCBS
Core Online Booking	Yes	Yes	Yes	No	Yes

Real-Time Availability	No (Static Schedule)	No (Static Schedule)	Limited (via QR Scan)	No	Yes (Dynamic, Sensor-Driven)
Conflict Prevention	Yes	Yes	Yes	Not Applicable	Yes
User Role Management	Yes	Yes	Yes	No	Yes
Access Control / Authorization	No	No	QR Code Authentication	No	QR Code Authentication
IoT Integration	No	No	Limited (for Access Only)	Yes (Core Feature)	Yes (Core Feature)
Real-Time Occupancy Monitoring	No	No	Indirect (via Access)	Yes (PIR Sensors)	Yes (PIR Sensors)
Automated Energy Control	No	No	No	Yes (Lights/AC/Projectors)	Yes (Lights/AC/Projectors)
Admin Dashboard & Analytics	Basic Reports	Advanced Dashboard	Basic	Basic (Energy Data)	Comprehensive Dashboard
Cost Model	Open-Source	Subscription	Academic	Academic	Custom Development

2.5 Discussion

The research gap is supported by the comparative analysis in Table 2.1 and gives a vivid roadmap on how the UPTM SCBS should be developed. The discussion elaborates on the design of the proposed system by modifying and combining certain features of the corresponding works, and justifications are made on each of the decisions to develop a unified solution.

Upon the main online booking system, the SCBS emulates the basic functionalities of Classroombookings and Skedda. Classroombookings adopts the role-based user management system (Administrator, Lecturer, Student) because this is the structure that is essential to ensuring the academic hierarchy and permissions. Skedda, the system copies the visual interactive booking interface and synchronization to provide a user-friendly interface that needs minimum training. These fixed features offer a stable and user-friendly basis of the reservation process.

The SCBS security mechanism is a version of the Smart Classroom Management System (SCMS), which uses the authentication based on QR codes. This mechanism was chosen as it is an affordable solution, which establishes a direct and verifiable connection between a digital booking and physical access. Waterline Reservation System generates the QR code based on a confirmed reservation in the booking system, which makes it compatible. This will ensure that a physical access to a classroom by unauthorized persons is blocked since only the users having a legitimate digital reservation will be permitted to enter.

To ensure automated energy saving, the SCBS incorporates the IoT architecture of the IoT-Based Energy Conservation System. These comprise real time occupancy sensing using PIR motion sensors and light and air conditioning control using relay modules. This change is essential in the realisation of sustainability goals and minimisation of operational expenses. The other components are completely compatible with the IoT system. It will run on physical occupancy independent of the means through which access was obtained hence wastage of energy will not occur.

The important integration of the UPTM SCBS is that of dynamic feedback between the modules. It is the essential feature of integration that is lacking in all the reviewed systems. The access control system (through the generation of QR codes) is informed by the booking schedule, and the booking system is informed by the IoT sensors (through the real-time occupancy information). An example would be to have a classroom that has a booked booking is not occupied, the sensor data will automatically release said booking. Such synergy enables the SCBS to maximize the use of space as well as energy usage according to actual use rather than planned use.

As such, the UPTM SCBS would be a required amalgamation of the divergent technologies into a single platform. The justification of its development is due to its incorporation of efficient scheduling (regulated based on commercial systems), the secured access (regulated based on SCMS), and the automated sustainability (regulated based on the IoT system). This is a combination, and it is managed with the help of an all-inclusive admin dashboard which will satisfy a full-fledged demand of a modern university such as UPTM which is, at the moment, not satisfied with by any existing product.

2.6 Conclusion

The chapter has also provided a literature review and critical analysis of the technological environment applicable to the Smart Classroom Booking System (SCBS). The research conducted created a background in terms of delving into the central themes of smart classroom systems, online booking applications, IoT monitoring, and responsive web design, establishing a considerable gap between scheduling programs and physical facility operations.

Four related works Classroombookings, Skedda, SCMS, and an IoT-Based Energy Conservation System presented in the detailed analysis supplied definite evidence of a particular market gap. Each system proved to be effective in specific tasks but still the lack of integrated features that could be found in the other systems. This limitation and how the proposed UPTM SCBS has implemented all the important features based on strategic adaptation and integration was visually synthesized in the comparison table.

As was determined in the discussion, the SCBS is a justified and needed initiative in its synergistic approach. System embodies the best practices used in the current systems blending the booking features of the commercial systems, access control features of the academic prototypes and energy automation of the IoT systems. This integration forms a feedback process that is dynamic in that digital scheduling and physical monitoring respond to each other to streamline resource usage. The UPTM SCBS thus stands to be able to transcend the shortcomings of the past solutions and offer an all-round, effective and environmentally friendly classroom management solution.

This chapter has managed to provide both theoretical and practical background of the system development. The methodology, system design, and implementation will be elaborated in the following chapters in the basis of the foundations made in this literature review.

3 METHODOLOGY

3.1 Introduction

This chapter explains how the implementation of the Smart Classroom Booking System (SCBS) and attaining its goals will be done systematically. To handle the complexity, meet changing demands, and deliver a quality product, it is necessary to choose a proper development process (Ainul, 2023). The hybrid approach to be used in this project will integrate the concepts of Agile and Prototyping. Agile offers a model of the controlled sprints of incremental development, through which the system can progress based on the ongoing feedback. This framework applies prototyping paradigm to develop the early and functional versions of the system in such a way that the stakeholders would see and engage with critical components prior to complete development. This combined solution is particularly quite appropriate to the SCBS, which implies multiple innovative features, including the integration of IoT and access based on QRs that can benefit through early validation.

3.2 Agile-Prototyping Hybrid Methodology

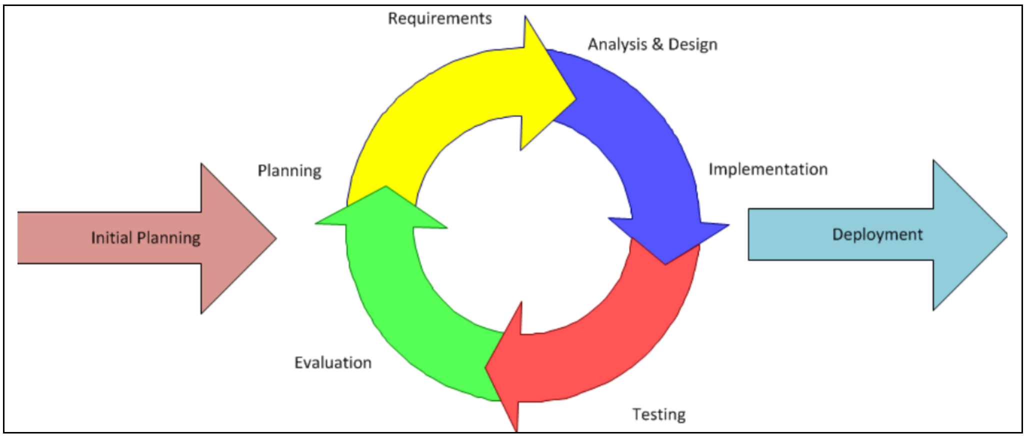


Figure 3.1: Agile-Prototyping Hybrid Methodology Diagram (Inflectra Corporation, 2024)

Agile methodology is selected as the main approach due to the stability of the ads and the ability to do iterative systems it is suitable to have a system with several interconnected elements such as dashboard analysis, web reservations and IoT automation. Agile does not require the system to be created in a fixed and linear manner but rather in smaller manageable bits known as sprints. This helps it to achieve a steady improvement, evaluation, and improvement that is driven by the continual input of interested parties.

The Prototyping model is incorporated into the Agile system in order to address the need to develop innovative features that are not present in the requirements when every requirement is not evident. Prototyping is the development of simple working versions of the system at the start of the development process. In the case of the SCBS, this will involve development of operation models of the booking system, incorporation of IoT and design user interfaces during the first phase. These concepts in turn are presented to users to be tested and reviewed, which directly affects the subsequent Agile sprints, to make sure that the final system is highly aligned to the expectations and requirements of the user.

3.3 Phases in Agile-Prototyping Hybrid Methodology

SCBS development process will consist of six stages which will be iterative but sequential. Each stage will possess specific responsibilities and deliverables, which will ensure steady progress and specific milestones and combine continuous feedback through prototyping.

3.3.1 Phase 1: Planning

The SCBS planning will begin with a dedicated study on identifying the particular areas of the operation at UPTM that have caused the inefficiencies. This will include systematic collection of information among the students, administrators and lecturers in different faculties in order to learn about technical limitations and academic time schedule needs. Based on these interactions, specific problem statements will be captured like 23 day delay in the manual booking authorisation and the failure to monitor classroom utilisation in real-time.. A conceptual diagram of the QR code access flow and a simple wireframe of the booking form will be among the first low-fidelity prototypes that are drawn concurrently. Before any formal design work starts, stakeholders will be able to verify that the suggested solution meets their expectations thanks to these preliminary images. With the selection of Laravel and MySQL for the backend, HTML, CSS, and JavaScript for frontend development, and ESP32 microcontrollers with PIR sensors for the Internet of Things components, the technology stack will be finalised. This selection will ensure compatibility and effectiveness for both the web- based and physical automation aspects of the system.

3.3.2 Phase 2: Design

The design phase will transform the approved concepts from the planning phase into detailed, interactive blueprints. Using Figma, a high-fidelity, clickable prototype of the complete user interface will be created. Key user journeys, including a student looking for a classroom that is open, a lecturer making a reservation, and an administrator seeing the occupancy dashboard, will be simulated by this prototype. Before coding starts, stakeholders will engage with this prototype to offer input on the design, positioning of buttons, and information flow. Technically, this stage will also entail defining tables for users, reservations, classrooms, and iot_sensor_logs as well as developing a thorough Entity-Relationship Diagram (ERD) for the MySQL database. In order to guarantee a unified technical design for the development sprints, a system architecture diagram will also be created to outline how the ESP32-based sensors, the frontend, and the Laravel backend will connect via RESTful APIs.

3.3.3 Phase 3: Development

Development will utilize Agile sprints, and at the conclusion of every sprint, a functional, testable prototype of a specific module will be created. A Minimal Viable Product (MVP) for the primary online booking system will be developed in the initial sprint and set up on a local XAMPP server for initial user testing. The admin dashboard will be created in the third sprint, QR code creation will be incorporated in the second, and IoT connectivity will be the primary emphasis of the final sprint. For the ESP32 microcontrollers to read data from PIR motion sensors and control relay modules, firmware needs to be developed and uploaded. The Laravel backend will create a particular API endpoint (like/Api/occupancy) to receive this sensor data and start automatic processes, like turning off appliances in an empty room.

3.3.4 Phase 4: Testing

The testing will be carried out together with the development once each major module or sprint is complete. Every working prototype will be subjected to an intense testing process. Functional accuracy of booking module will involve the testing of conflict prevention algorithm by trying to book two rooms at the same time. Integration testing: This will ensure that the system is accurate in updating the classroom status and the data received by the ESP32 sensors and sent to the Laravel backend. Lastly, there will be a sample user base (students and lecturers) that will participate in User Acceptance Testing (UAT) whereby they will be required to do certain tasks that will involve the use of the QR code and the process of reservation. Their feedback and success rates will be recorded to determine and address any usability problems or weaknesses before the actual launch.

3.3.5 Phase 5: Deployment

A successful test will see the system deployed to the live environment. The local XAMPP server is going to be substituted by the live web hosting service with the dedicated domain of the Laravel application and MySQL database. Once the firmware is completed the ESP32 devices will be literally installed in pilot classrooms. They will be configured to access the API endpoints of the live system and be connected to the campus Wi-Fi. The safe mode of the relay modules to be interconnected with the electrical circuits of the classroom will be the most important step in ensuring that the automated control of the lights and air conditioners are effective, and safe to be used daily.

3.3.6 Phase 6: Review & Feedback

It will begin with a formal post-deployment reporting and tracking period. To determine the key performance measures, including the number of automated booking releases on the no-shows, as well as the reduction in energy consumption of the pilot classrooms, the data about the use of the system during the first month of operation will be reviewed. Structured feedback forms will be distributed to the first users so that they can be given a chance to provide qualitative information about their experience. To ensure that the SCBS matures in tandem with actual application and sustainability in the long run, the information and input obtained will be synthesized into a report that shall directly reflect on the betterment plan and scope on subsequent system upgrading.

3.4 Requirement

In this section, the intended requirements of the Smart Classroom Booking System (SCBS) will be outlined, and they will be based on the responses to the questionnaire, which will be held with the stakeholders of UPTM. The requirements will contain the needs and preferences of students, lecturers, and the administrators with respect to the booking processes and real-time availability of classrooms, the energy management as well as access control.

3.4.1 Data Gathering Techniques

At least 50 UPTM respondents (including lecturers and students of various faculties) will be employed to gather requirements using a systematic online questionnaire. The questionnaire will be divided into five major sections which are, non-functional expectations, functional preferences, problem identification, objective alignment, and demographic data. The strategy will enable the quantitative analysis of the interests of the users to the new system and current booking problems. It is expected that this information would indicate that the users support digital transformation and automated features and reveal severe issues with the manual booking processes, waste of energy in unoccupied classes, and unauthorized use of rooms.

Besides that, some selected classrooms will be observed directly all over the campus to complement the use of questionnaires and document actual use patterns and inefficiencies in operations. The principal topics of these observations will be peak times, occupancy rates in classrooms, energy consumption patterns, and instances of violation of room use. The number of lights and air conditioners left on in empty classrooms and the time required in real time to undertake the manual booking process at the administrative counters are some of the real-world scenarios that are not best represented through questionnaires.

3.4.2 Functional Requirement

The SCBS will also substitute the existing manual paper-based system with high online reservation functionality in accordance with the likely outcomes of the questionnaire. To reduce the long approval processes that come with the initial research, the system will allow users to make, amend and cancel classroom bookings online without necessarily visiting administrative counters. In order to solve the scheduling issues, real-time updates on the status of the classroom will be initiated which will show availability, booked status, and current occupancy. According to the analysis of user needs, automated email confirmations and monitoring the booking history will be provided. The system will combine the check-in method based on QR code to mitigate the issue of walk-in usage and automated booking cancellation upon 15 minutes of non-arrival to ensure the system can solve the issue of walk-in usage. To promote the issue of energy wastage, the implementation of IoT will automatically regulate the air conditioning and the lights, based on the occupancy detection.

3.4.3 Non-Function Requirement

The system will be required to have performance requirements of loading in 5 seconds of all booking operations as per the user expectations of responsive systems. The security will be ensured with the help of the mandatory login verification with the use of UPTM credentials assuring the privacy of the bookings as the institutional standards presuppose. There will still be availability of the system at any time to respond to booking requests when business is not on. We should have a totally responsive interface that is compatible with both desktop and mobile operating systems. At the same time, reliability standards will ensure that the system remains operational at the time of high academic pressure and the issue of error in the booking transactions will be addressed resiliently.

3.4.4 System Requirement

Technically, it will be implemented using HTML, CSS, and JavaScript as frontend technologies to be used, which will be cross-device compatible. Laravel PHP framework and MySQL database management will be used as the backend that will manage booking data, user accounts and system logs. ESP32 microcontrollers with PIR motion sensors used to detect the presence of people and relay modules to turn electrical tools on and off will be part of IoT components. The tools that will be used in development will be Figma to prototype an interface, Postman to test the API, and XAMPP to develop the local environment. It will need web hosting that will support PHP 8.0+ and MySQL 8.0+ to be deployed, alongside the ability to support concurrent user access in times of peak bookings.

3.5 Analysis

These subsections provide the analytical models to determine the structure and behaviour of the Smart Classroom Booking System (SCBS). The analysis is to be modelled in terms of predicting the result of the data collection, the core functionality of the system, considering the Use Case Model that helps identify the primary interactions between the actors and the system and a Flowchart, defining the logical sequence of the automated energy control process. These models will be used as a guideline of how the system should be developed and all the functional requirements that should be taken care of. The rationale of the IoT automation shall be apparent.

3.5.1 Data Gathering Analysis

3.5.1.1 Observation Analysis

Figure 3.2: Classrooms Observation (Ain,2025**)**

Various major operational inefficiencies were found through direct observations in diverse classes of UPTM, which advocates the need of the SCBS. Although booked, nearly 35 per cent of the scheduled classes were reported to be empty during the most productive academic periods between 10:00 AM and 6:00 PM and this fact implies that resources are not fully utilised. Particularly, the occurrence of energy waste was observed both in the evening and upon breaks, when 60 percent of the observed classes had air conditioning and lights on unoccupied areas. The manual booking process had major administrative bottlenecks, and a mean time of waiting was 15 to 20-minute stays at administration desks when registration is high. Furthermore, an illegal usage was observed in approximately a quarter of the classroom checks in which the students occupied the premises without bookings to use them non-academically and often collided with the official users. The results confirm the essence of improper use of spaces, energy and accessibility that is lacking and that the SCBS aims to address.

3.5.1.2 Questionnaire Analysis

There was a thorough questionnaire that was sent to the UPTM community and it received 50 responses. All 21 questions are analyzed below to confirm the problem statements and outline system requirements.

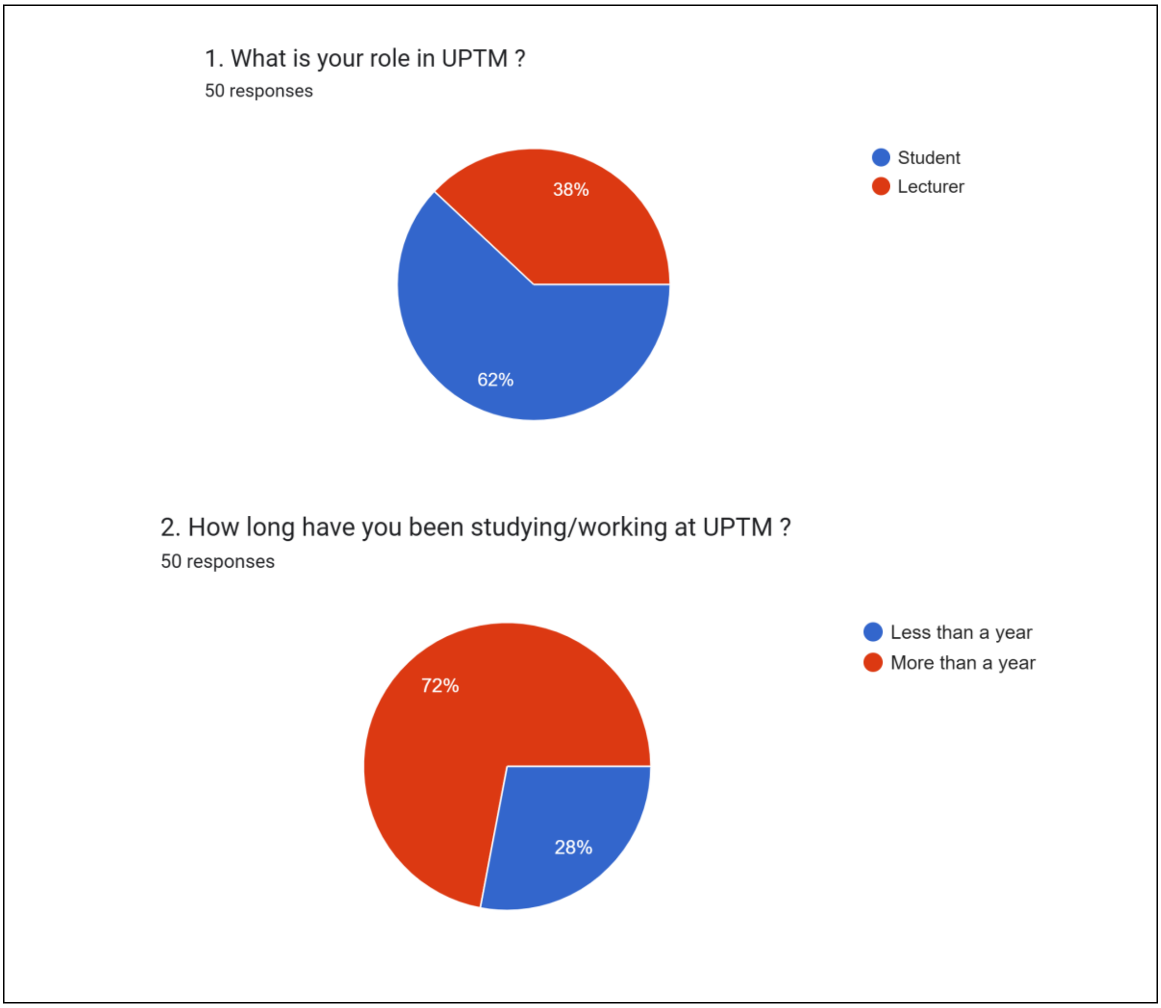
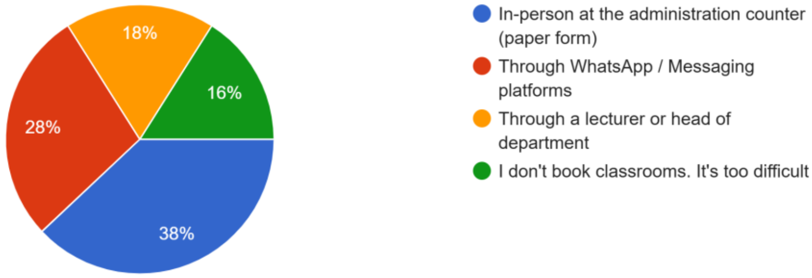


Figure 3.3: Demographic Analysis (Ain,2025)

The questionnaire received the responses of 50 members of the UPTM community including 62 percent students (Q1) and 38 percent lecturers, 72 percent of which stayed at UPTM over a year (Q2) and 28 percent shorter than a year. Such a demographic distribution offers a solid quantitative support of the problem statements and a concise mandate of the proposed Smart Classroom Booking System (SCBS). The results indicate that there is a high level of agreement on the inefficiencies of the established process and overwhelming support of the main features of the system, and most of the respondents were of considerable institutional experience to offer valuable information.

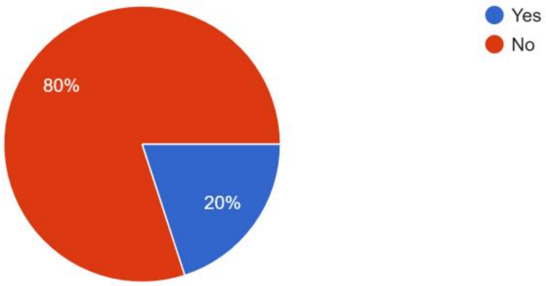
3. What is your primary method for booking a classroom at UPTM ?

50 responses



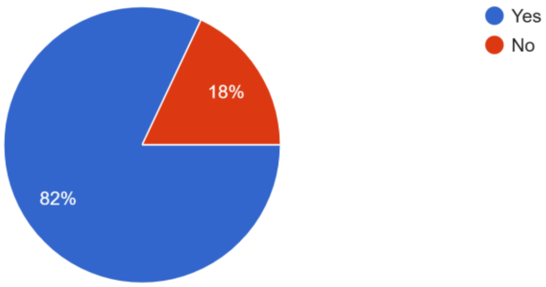
4. Is the classroom always available whenever you want to book it ?

50 responses



5. Does your classroom booking usually take a long time to be approved ?

50 responses



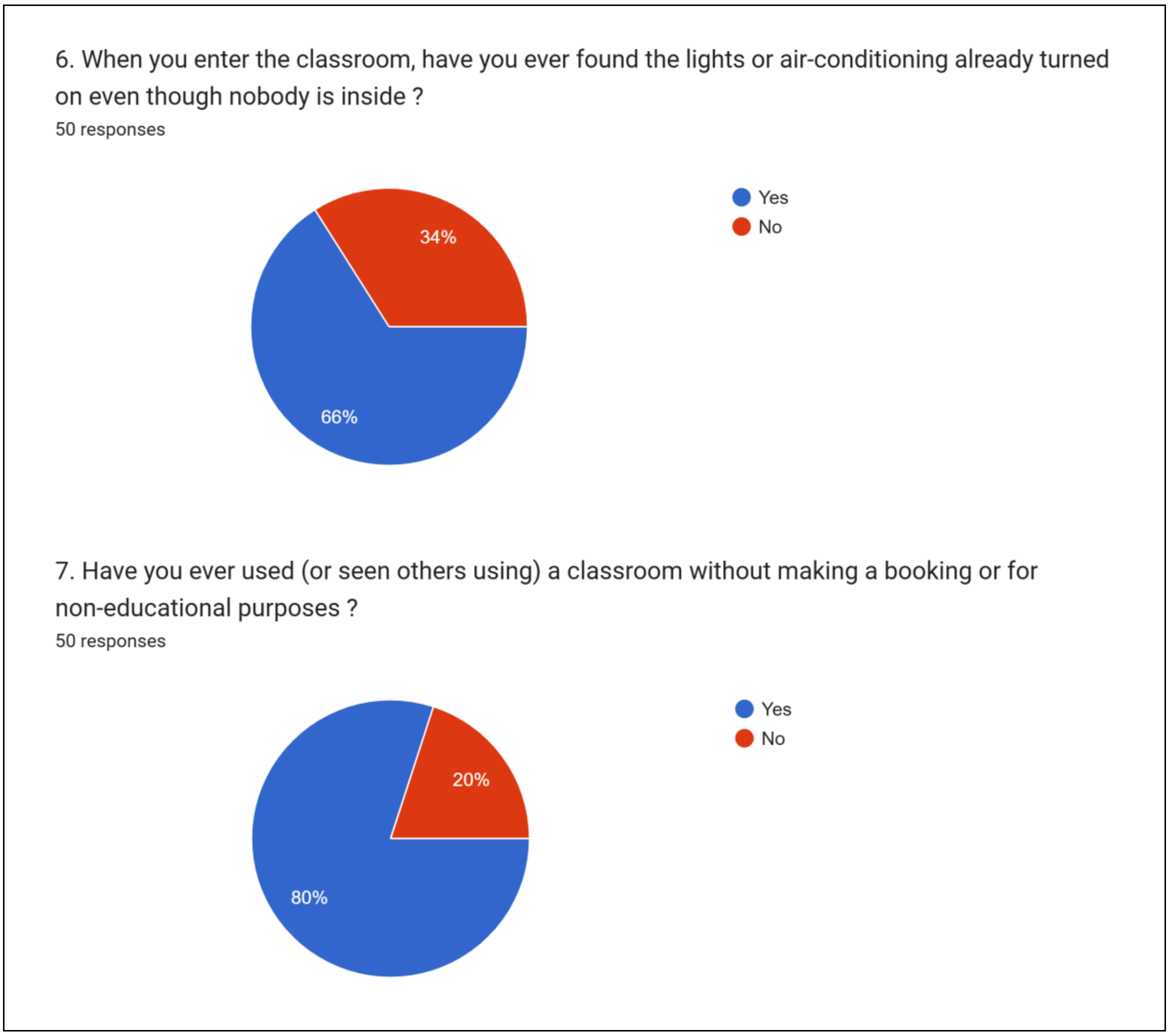
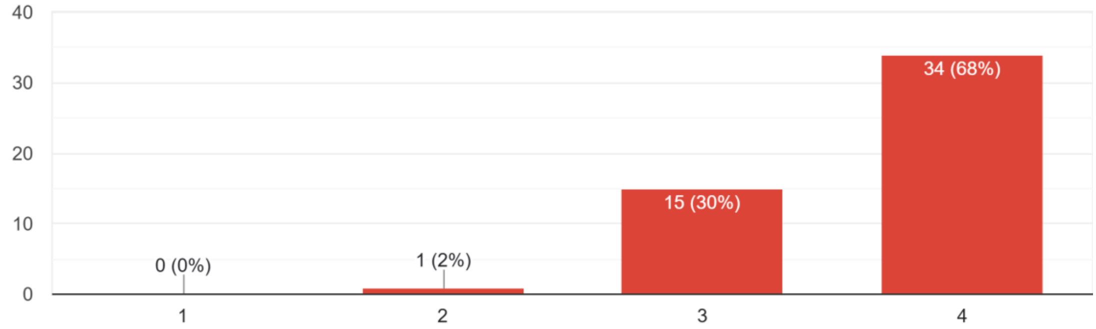


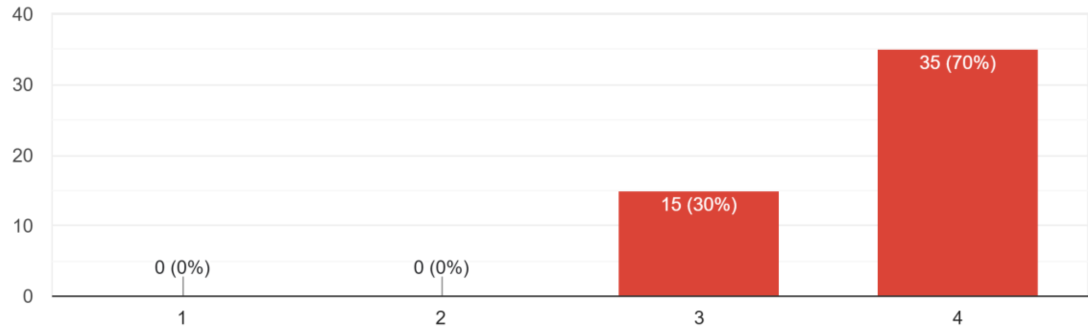
Figure 3.4: Current Process Analysis (Ain,2025)

The existing classroom management system analysis shows that the system is highly inefficient in various aspects. The booking system is so fragmented, as half of the respondents use different methods: 38% of them turn to the administration desk and fill in the paperwork, 28% WhatsApp or other messaging applications, and 18% lecturers and department heads (Q3). Importantly, 16% do not engage in the booking process all, because of the perceived complexity of it (Q3), which also is directly related to the high rate of unauthorized classroom use to non-educational purposes (80%) (Q7). The lack of availability of resources is one more significant challenge, and 80% of respondents stated that they do not always have access to the classroom when it is necessary (Q4). The administrative inefficiencies are also demonstrated by 82 percent of users who report to have had a long booking approval period (Q5). Moreover, 66% of the respondents have attested that they have witnessed classrooms in which the lights and air-conditioning are on even though they are not in use (Q6).

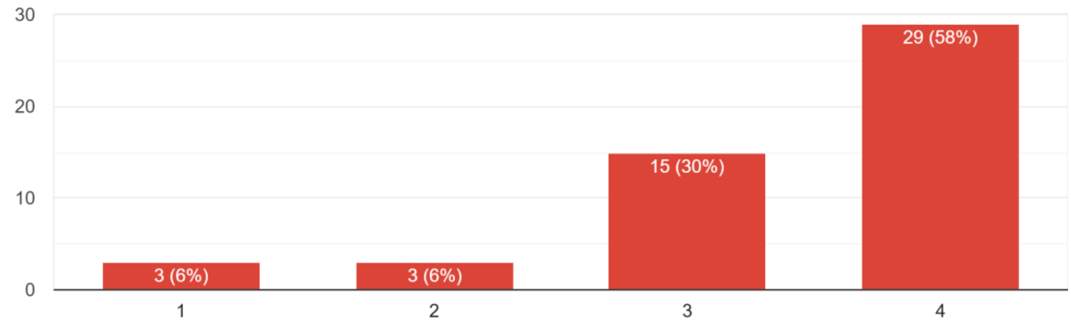
8. Do you agree if the classroom booking method is changed to an online system ?
50 responses



9. Do you agree if the system shows classroom availability based on real-time updates (in used, available, booked)?
50 responses



10. Do you agree if the system automatically turns lights and air-conditioning on/off based on motion detection whether someone is in the classroom or not ?
50 responses



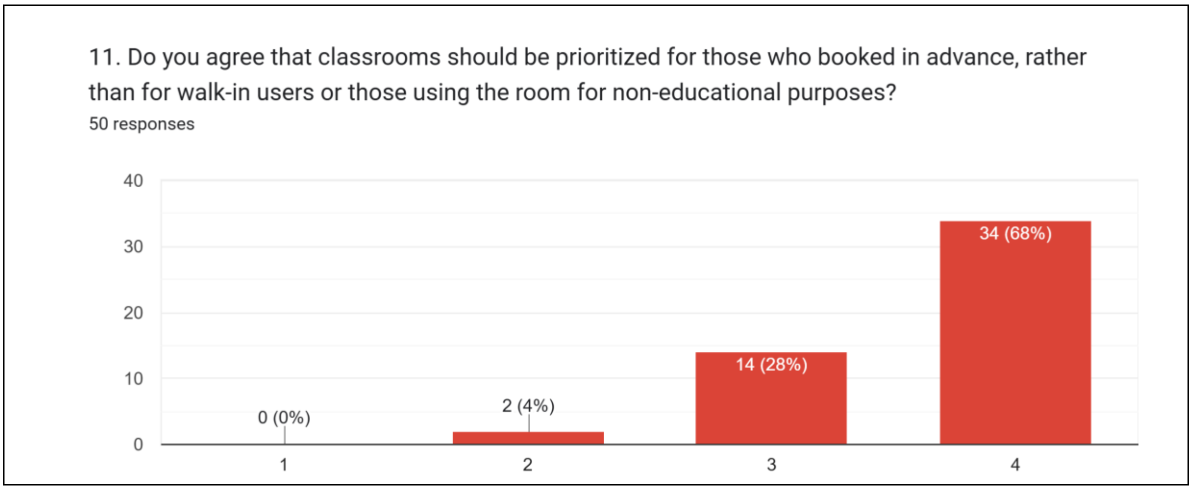
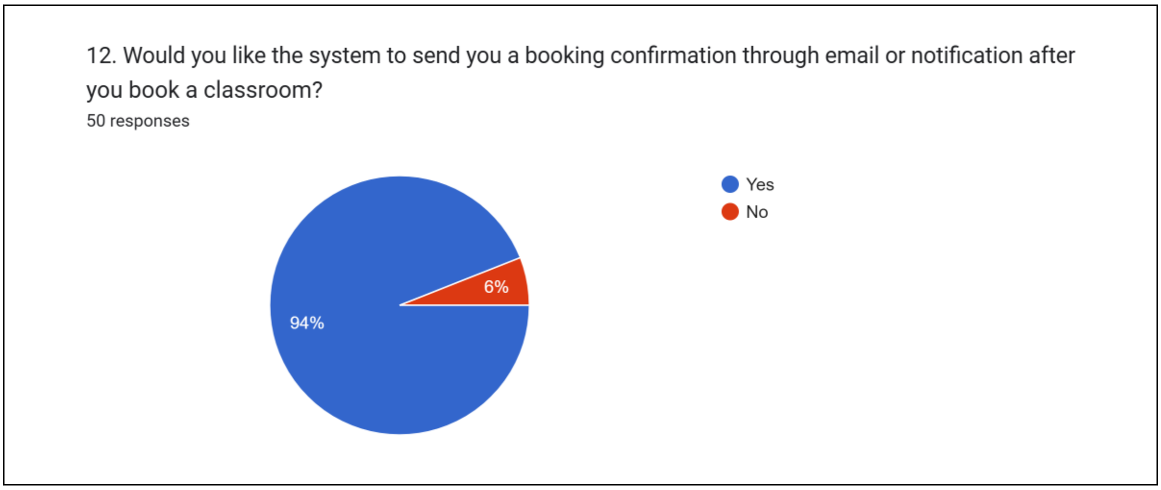


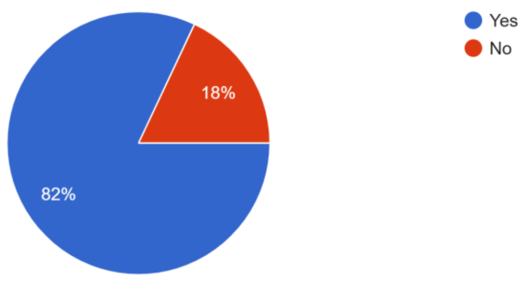
Figure 3.5: System Objectives Validation Analysis (Ain,2025)

The data demonstrates resounding endorsement for the core objectives of the proposed SCBS. An overwhelming majority of respondents (68% strongly agree, 30% agree) support transitioning from the current manual process to an online booking system (Q8), with only 4% expressing disagreement with this change. There is also near-unanimous consensus regarding the implementation of real-time availability updates (70% strongly agree, 30% agree, Q9) and automated energy control utilizing motion detection technology (58% strongly agree, 30% agree, Q10). Furthermore, 68% of respondents strongly agree that classroom access priority should be allocated to users with advance bookings over walk-in users or those utilizing rooms for non- educational purposes (Q11), thereby reinforcing the necessity for a systematic approach to authorized access control.



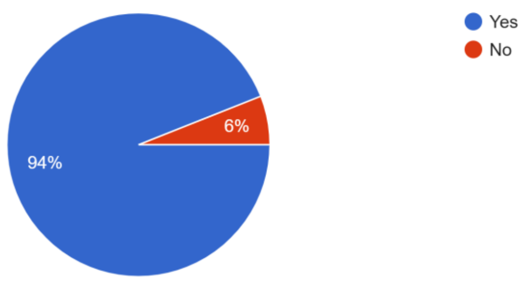
13. Should the system automatically cancel your booking if you do not check in within 15 minutes after the scheduled time?

50 responses



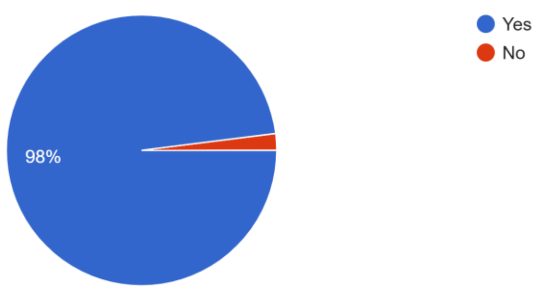
14. Would you like the system to allow you to modify or cancel your booking online (without going to the counter)?

50 responses



15. Should the system show a live timetable of which classrooms are currently occupied, available, or reserved?

50 responses



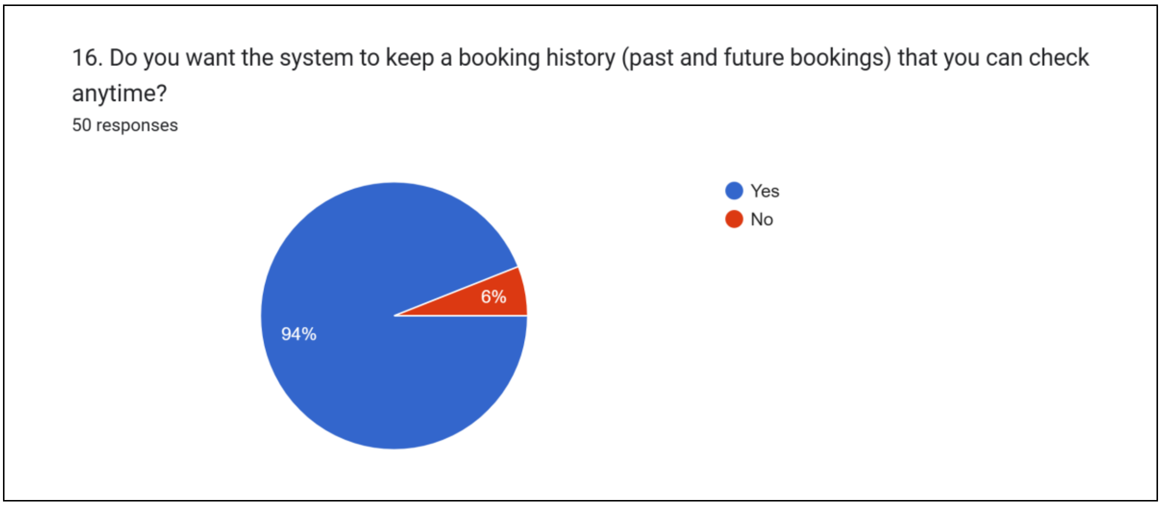
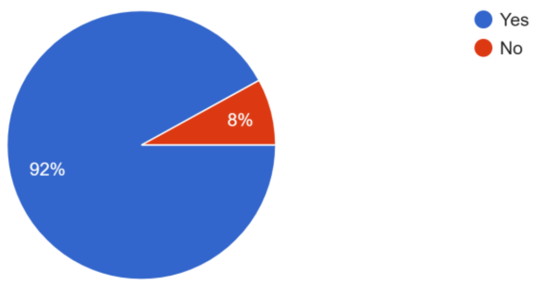


Figure 3.5.1.2.4: Functional Requirements Analysis (Ain,2025)

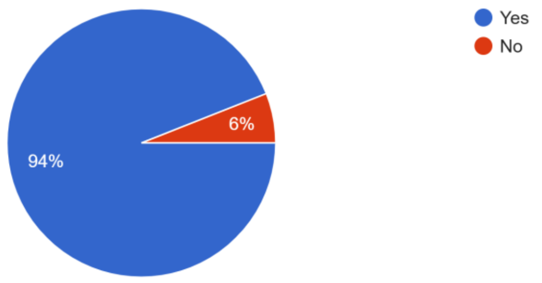
The questionnaire answers give straight guidance to the functional capabilities of the SCBS. The customers are highly demanding certain functionalities that will make the booking process more convenient and efficient at the system level. The statistics show that the number of people that needed automated booking confirmation via email or the notification system (94%, Q12) is overwhelming, and it would allow users to be sure of their booking instantly. Online booking, which allows for changing and cancelling the booking without a visit to the physical counters, had an equivalent level of support (94%, Q14), which is a major source of pain in the existing manual system.

Visualization of classroom status in real-time through live timetable interfaces was noted as a necessity, and 98% of the participants agreed with this option (Q15). Comprehensive booking record management of the completed and future booking was also highly underlined (94%, Q16) in order to allow the user to track the trend of their classroom usage. In addition, a significant percentage (82%) supported the option of developing automatic cancellations processes because of the no-shows which are made within the period of 15 minutes of booked times (Q13), and such a feature would improve the use of classes and decrease waste of resources.

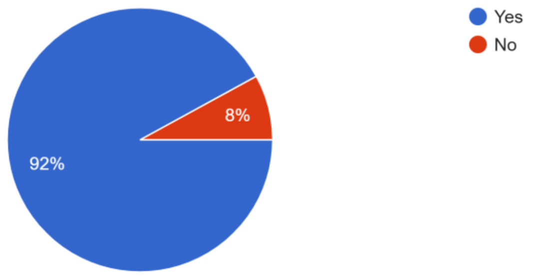
17. Should the booking system load quickly (less than 5 seconds when checking or booking a classroom)?
50 responses



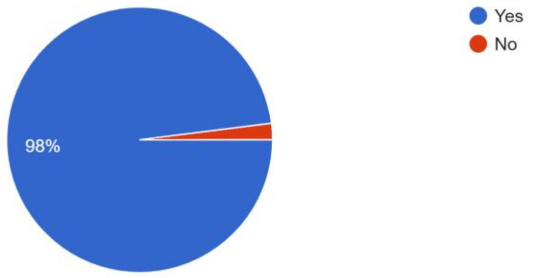
18. Do you think the system should require a secure login using your student/staff ID before booking a classroom?
50 responses



19. Should the booking system be available 24/7 (not limited to office hours)?
50 responses



20. Should the system be mobile-friendly and easy to use on both computers and smartphones?
50 responses



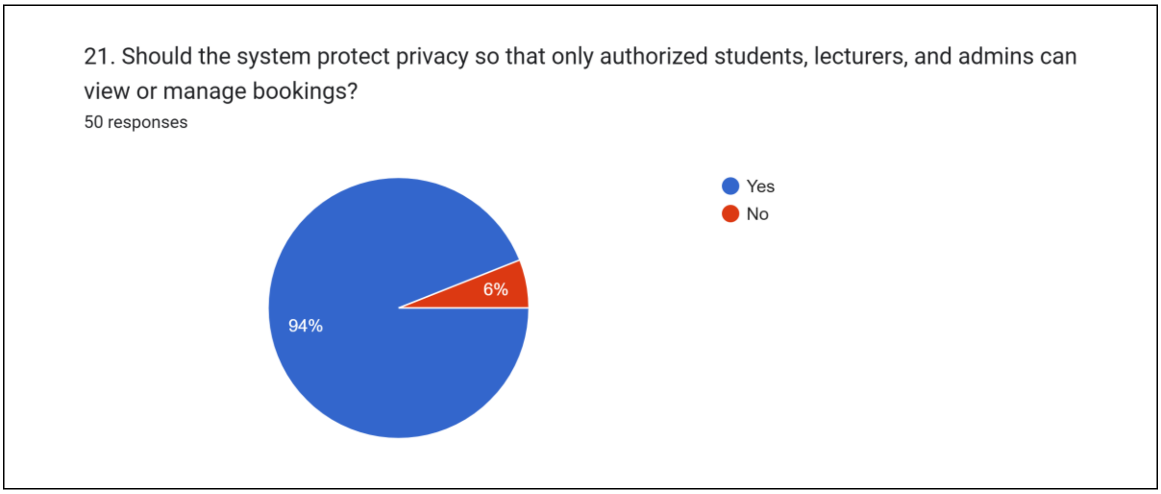


Figure 3.6: Non-Functional Requirements Analysis (Ain,2025)

The questionnaire results clearly describe the quality features and performance requirements of the SCBS. The respondents constantly emphasized the importance of system performance with 92 percent demanding rapid responsiveness in the system with the stipulation that all booking and checking operations take a loading time of less than five seconds (Q17). The issue of security developed to be crucial, as 94 percent of users required secure authentication procedures that relied on official identification credentials of students and staff members (Q18).

Accessibility requirements were well defined with 92 out of the 100 respondents expecting the systems to be accessible even during the non-business hours (Q19). The items which customers praised most were cross-platform compatibility, as 98% expressed a need in a responsive design that would be flawless on desktop computers and mobile devices only (Q20). 94% of the respondents also highlighted the importance of ensuring the protection of privacy by limiting access to the data to authorised people only (Q21).

To sum up, the response of the questionnaire is analysed to provide an effective and user-friendly framework on the development of the SCBS and explicitly justify the operational validity challenges that were experienced. The data confirms all issues associated with manual processes, availability of resources, loss of energy, and unauthorised access as well as clearly defines the functional requirements that need to be fulfilled to implement the system. The readiness of the community toward the digital change in classroom management is indicated by the overall consensus concerning the functional and non- functional requirements that also constitutes a solid foundation of the system design and development.

3.5.2 Use Case Model

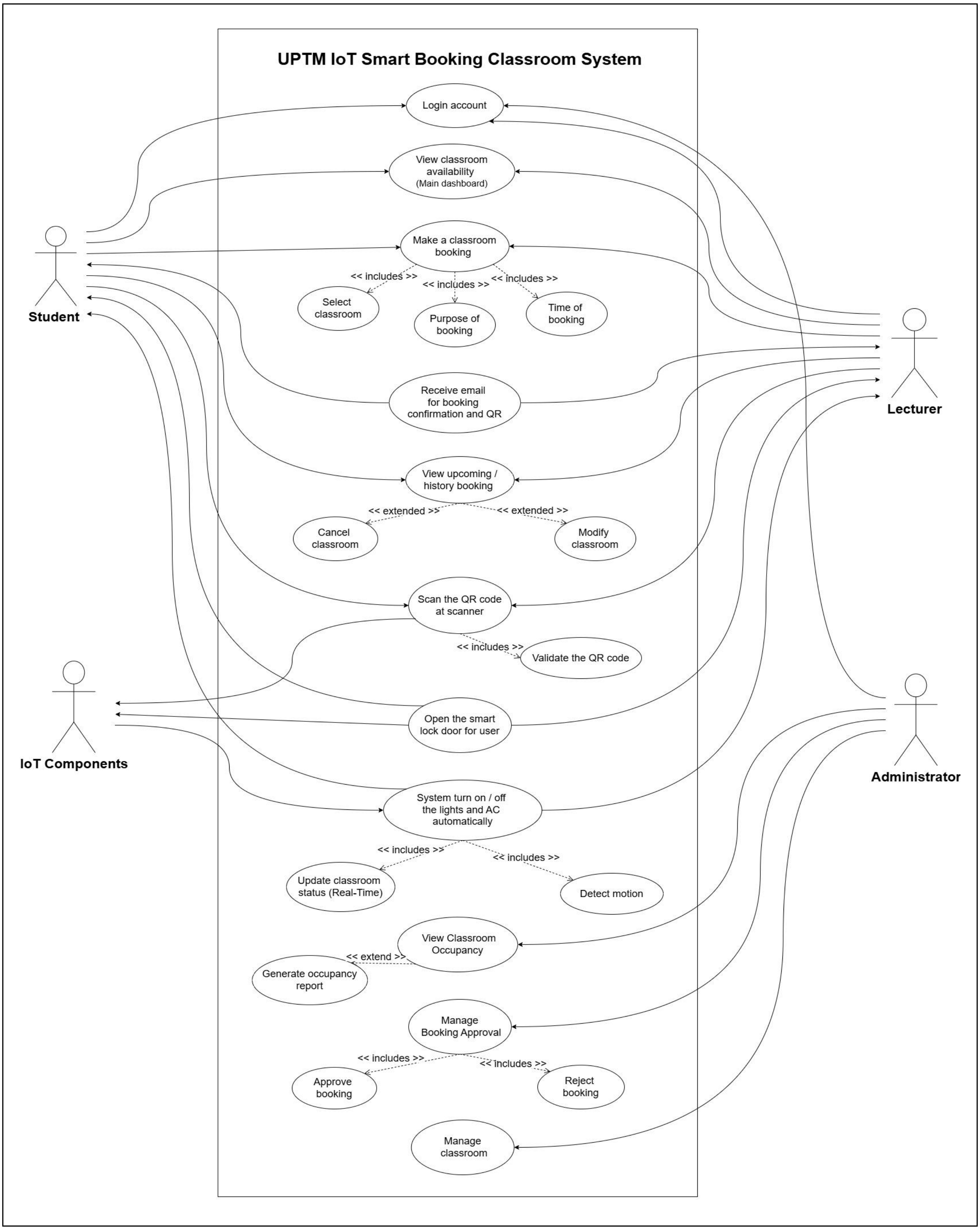


Figure 3.7: Full Use Cases of SCBS (Ain,2025)

The SCBS serves as the boundary for all the functionalities described. Within this system, both the Student and the Lecturer act as users with identical booking privileges. They start their process by logging in and using the main dashboard to see the availability of classrooms in real time. Through a special Booking Dashboard, they can then reserve a classroom by choosing a suitable classroom, state the time and the purpose of booking. The system will instantly create a unique QR code and emails a booking confirmation as soon as a reservation is successfully made. Student and lecturer have complete control over their bookings. In addition to viewing their booking history, consumers can use this interface to change the specifics of a future reservation or cancel it completely.

To access the classroom, the student and lecturer will have to scan the given QR code at the smart door lock first. The system will validate the QR code to match the one in the system once it matched the door will be open. Inside the classroom, IoT components will detect motion to automatically turn the lights and air conditioning on or off for efficiency and update the classroom status in real-time.

The IoT Components serve as the automated, physical layer of the system, executing the decisions made in the digital platform. This subsystem scans and validates the QR codes presented by users, and upon authorization, commands the smart lock door to open. It also employs motion capture in order to provide smart functionality to control the classroom setting by switching lights on and off or activating AC, which would facilitate energy conservation. More importantly, these elements constantly refresh the classroom state on a real-time basis, pushing live information into the system. This will guarantee the users, and the Administrator always have accurate and up to date availability information and the live occupancy monitoring information respectively.

The administrator also uses the Login account feature when starting a session. Once they are authenticated, they can do everything with the system. They will be able to see the occupancy of classrooms in real-time, so as to monitor the utilization in each room. Based on it, they are in a position to develop an occupancy report to enable comprehensive planning and analysis. They can accept or decline booking request, as far as they can manage booking. The basic architecture of the system is also under their complete control using Manage Classroom that also has the Add, Delete and the Edit classroom facilities to ensure a valid and up-to-date list of resources.

3.5.3 Flowchart

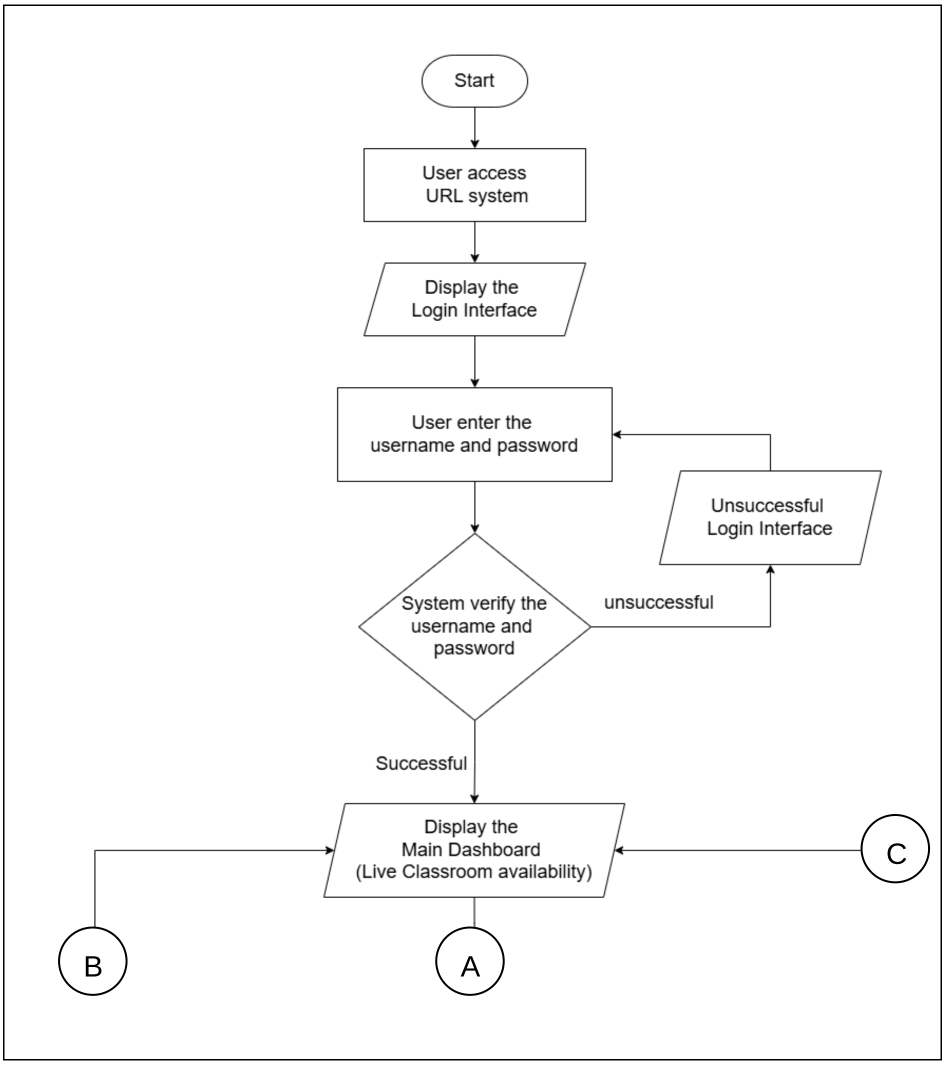


Figure 3.8: Student and Lecturer Flowchart Part 1 (Ain,2025)

Identification is the point of departure among students and lecturers. The system is accessed by a URL and users are presented with a login screen where they are required to enter the right credentials (username and password). This information is verified by the system, which in case it is incorrect an error message is shown, and users are required to renter the credentials. Upon successful login, the user is redirected to the Main Dashboard that will instantly display live availability of classrooms.

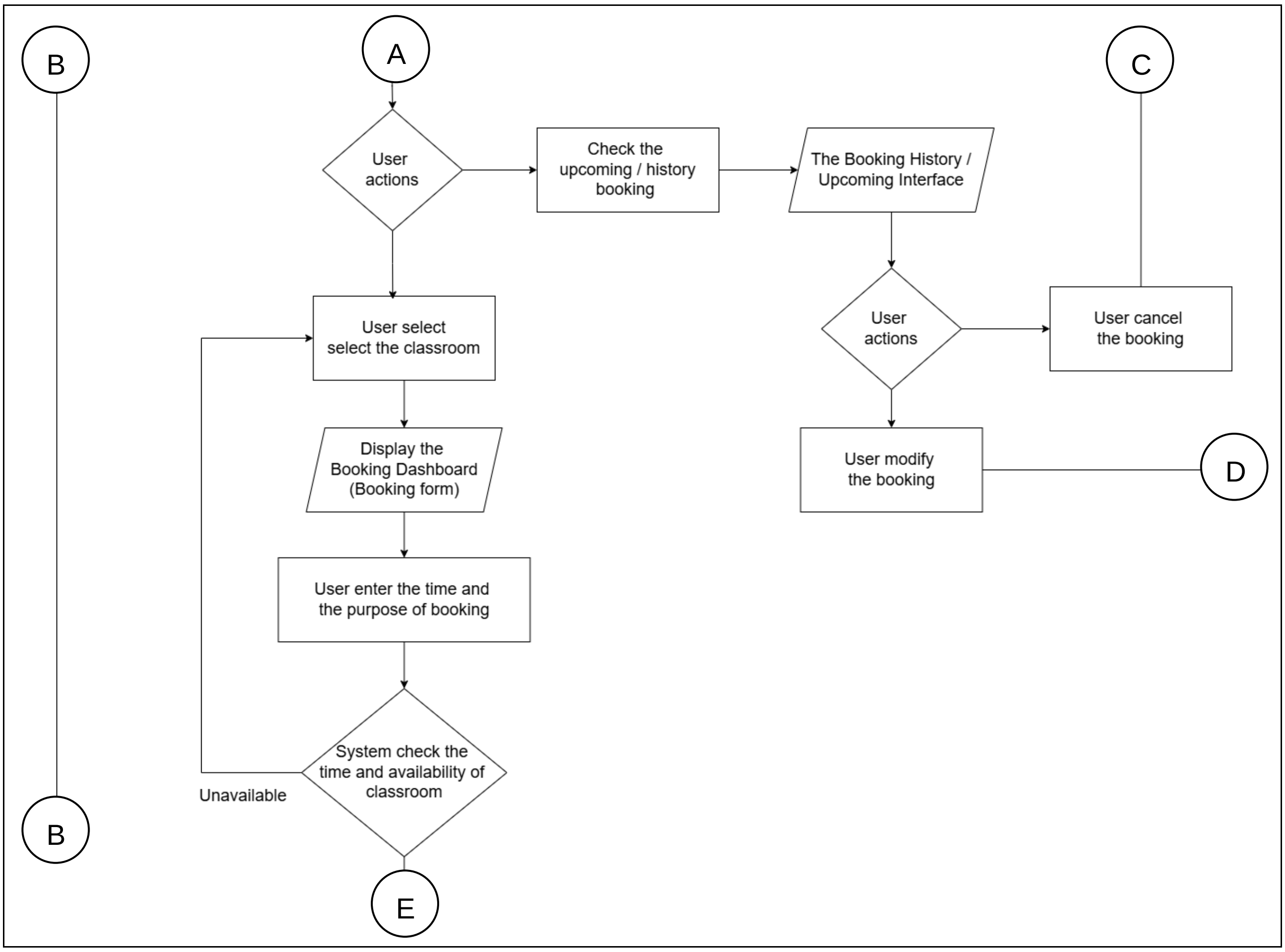


Figure 3.9: Student and Lecturer Flowchart Part 2 (Ain,2025)

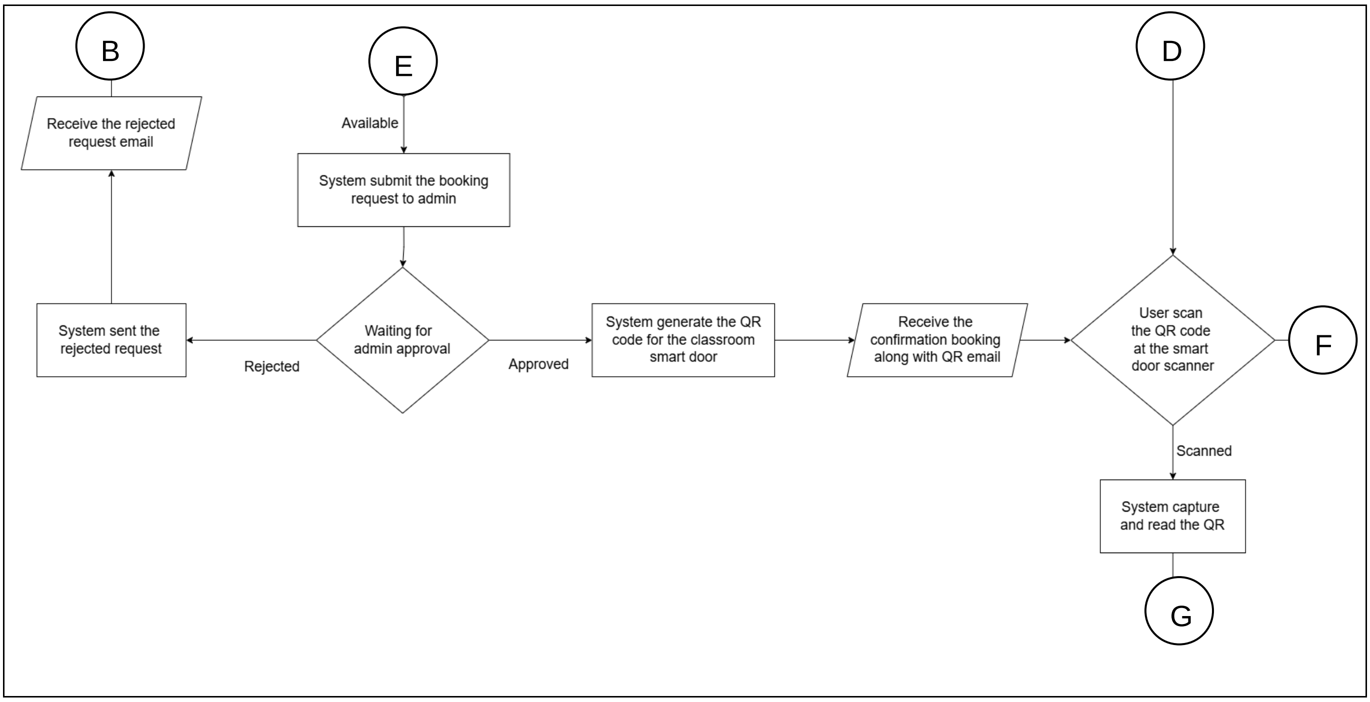


Figure 3.10: Student and Lecturer Flowchart Part 3 (Ain,2025)

The user is left with two main paths that they can take, starting off the main dashboard. The first one will allow them to get a hold of their booking history and subsequent bookings. The option provided to them to choose personal booking schedule discloses their personal booking schedule where they can see their upcoming bookings and in case of need, make changes or cancellations. In case of cancellation of the booking, the users will be notified, and the status of the classroom will be displayed as available in the main dashboard.

The second more complex process is the process of reserving a classroom. By selecting this choice, you will be redirected to the Booking Dashboard, which is a form where the user will be asked to select their preferred time slot and reason why they wanted to book. The system then performs a necessary validation with the database to check whether the classroom chosen is free at that time. In case unavailable, then the user is told, and he or she has to re-select once again. In case it is available, the system sends the request of the booking to an administrator where it is approved, and it becomes in a pending state.

The user then waits to get the decision of the administrator. In the case where the request is rejected by the admin, the user receives an email stating they have been rejected and they will go back to the Main Dashboard to request another new booking. Once the approval is made by the admin, the system generates a unique QR code of the specific booking session and sends a confirmation containing the QR code to the user as an email.

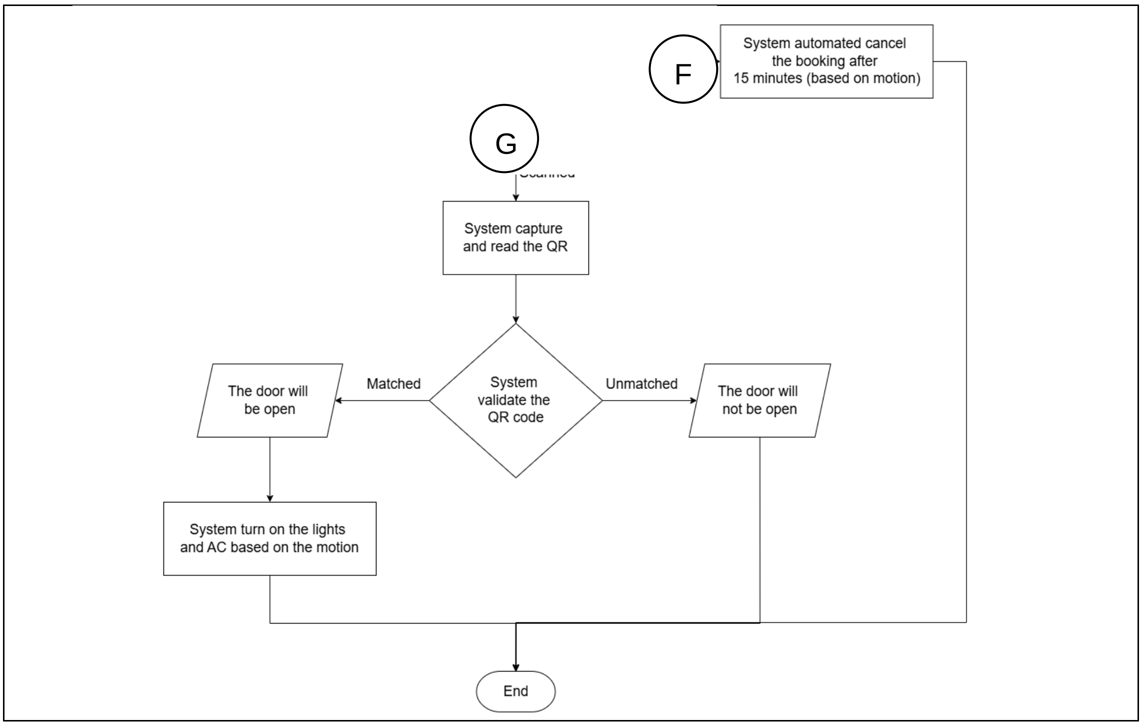


Figure 3.11: Student and Lecturer Flowchart Part 4 (Ain,2025)

The final phase is access to corporeal classrooms. A smart door scanner is used to scan a QR code by the user when he enters the classroom. The system reads the code and deciphers it and also says whether the code is in use or not. The door will never open, and the attempt to enter will be terminated in the event of a faulty code or another event an attempt to enter a code expires after fifteen minutes. The smart lock activates and unlocks the door, provided that the code is correct. Upon entering, a person is recognized by motion sensors and activates the air conditioning and lighting. To enhance efficiency in energy, a timer is continuously running. The device has automatic shutting of the lights and air conditioner to terminate the booking cycle in the case of no activity after fifteen minutes.

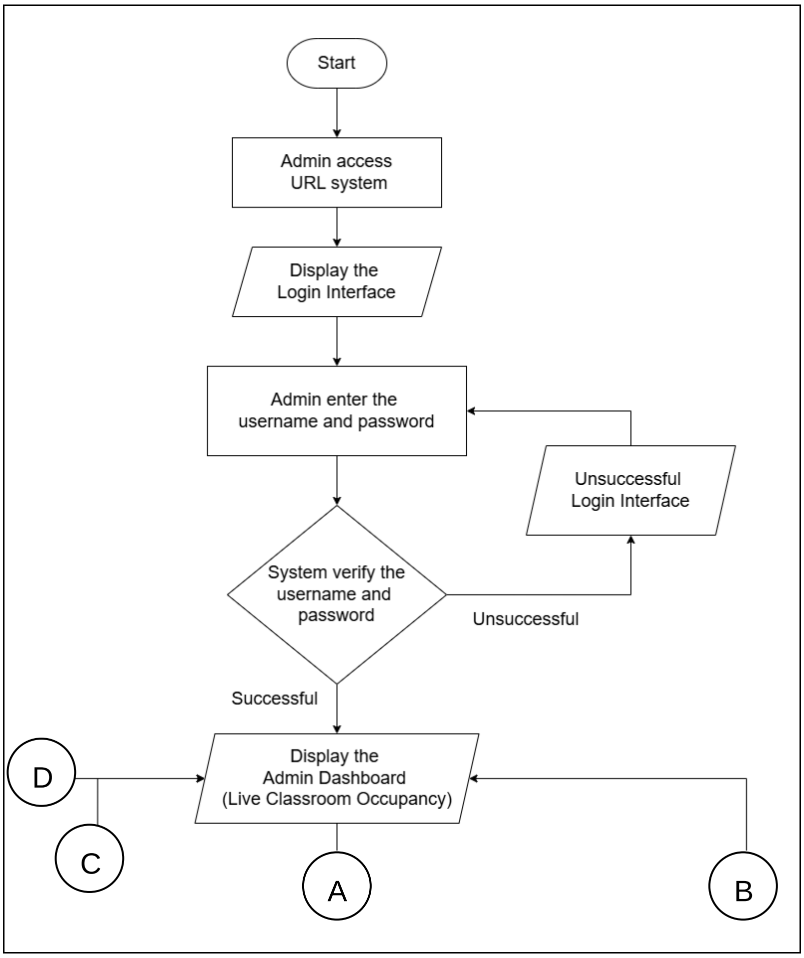


Figure 3.12: Administrator Flowchart Part 1 (Ain,2025)

The administration process starts with a secure log-in process. The administrator type in the URL of the system and gives his or her username and password. These credentials are verified by the system and in case they fail, access is denied. Upon successful logging into the site, one is given access to the Admin Dashboard that provides a comprehensive view of occupied classrooms in real-time.

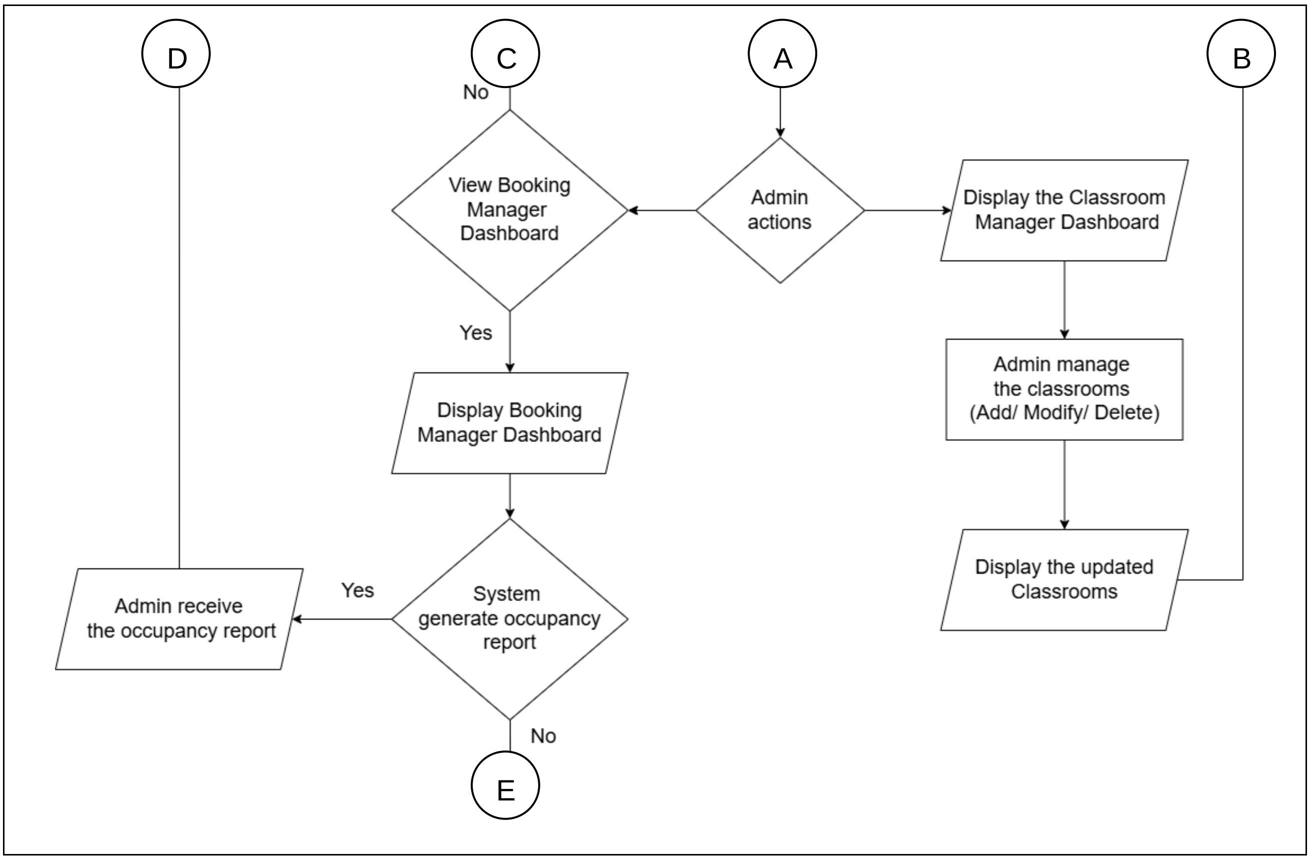


Figure 3.13: Administrator Flowchart Part 2 (Ain,2025)

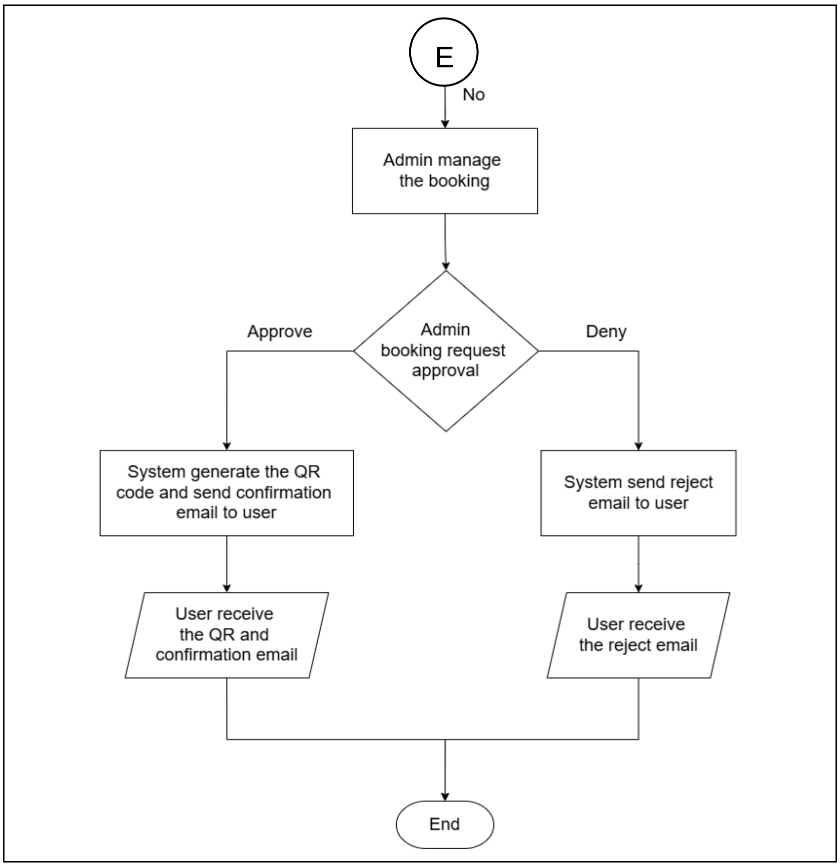


Figure 3.14: Administrator Flowchart Part 3 (Ain,2025)

There are a number of key management functions of this dashboard that the administrator has. The main activity of the administrator is to choose the Booking Manager Dashboard. They indicate booking inquiries immediately after selecting this option. The system has the capability of generating occupancy report before starting the approval process. This helps the administrator to collect analytical information that is relevant to the booking scenario. The administrator can choose to generate his/her report, or to avoid this step and work directly with the reservations.

The administrator reviews all the requests and decides whether they should be approved or not. Provided that the request is rejected by the admin, the system will automatically send an email of rejection to the user bringing this procedure to an end. When the request is approved by the system administrator, two things automatically take place. It generates a QR code to enter the door and sends an email with the QR code to the user, and that is what finishes that administrative activity.

The final position is classroom management. This leads to the Classroom Manager Dashboard which allows the admin to add new classrooms to the system or modify the existing ones such as capacity or equipment or eliminate the classrooms. The system displays the updated classroom after any changes have been made and ensures that the changes have been stored and one goes back to the central to do other things. This structured procedure will help the administrator to track the usage of the system effectively, manage or do the reservation process, and maintain accurate classroom records.

3.6 Conclusion

The chapter has detailed the comprehensive approach to be used in developing the Smart Classroom Booking System (SCBS) at UPTM. The hybrid Agile-Prototyping approach was chosen as it is the most appropriate in terms of early stakeholder validation using functional prototypes and offers the required flexibility during the iterative development. Being executed in six phases, Planning, Design, Development, Testing, Deployment, and Review and Feedback, the six-phase execution strategy provides an open set of guidelines on project implementation because of particular outputs set at each stage.

Requirements analysis has enabled clear functionality, non-functional, and system requirements based on observational studies and structured questionnaires that reflect address the issues of inefficiencies in manual booking, energy waste, and unauthorized classroom use. The analysis models, including Use Case Model and Flowchart, are graphical illustrations of system capabilities and automatic processes which will be the foundation of the system design. Altogether, this strategy is necessary to ensure a flexible but systematic way of developing a robust user-oriented solution, which will accommodate the specific needs of UPTM to manage the classroom environment efficiently, being energy-saving, and controlling safe access entry.

4 DESIGN

4.1 Introduction

The chapter is a proposal of the design of the UPTM IoT Smart Classroom Booking System (SCBS). It defines the interface designs of the system, schema of the database, flow of data and the framework of the functioning of the system. As the project is on planning stage and on verge of implementation, the figures and descriptions given here are a blueprint on development. They show how the system should work once installed so that all the requirements that were identified above are met.

4.2 Interface Design

The interface is easy to use and has a similar, uniform experience to students and lecturers, which have no differences in their functional requirements. It is made to be used both on the PC and mobile. The system has responsive pages that address all the booking lifecycle stages such as booking history to login. Each of the figures below can be interpreted as one of such interfaces, and its purpose and design motivation can be explained.

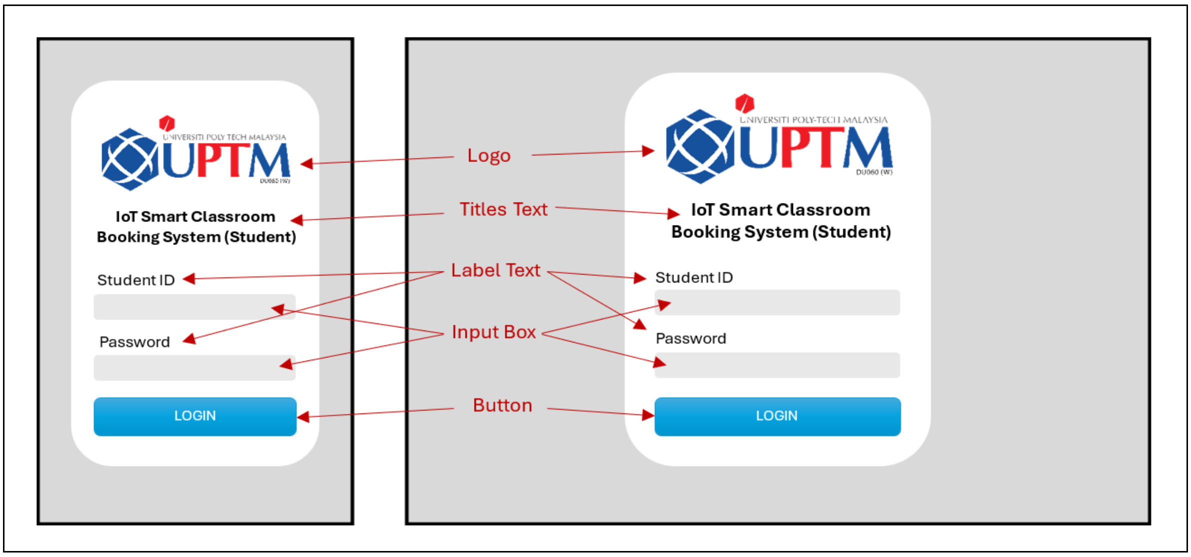


Figure 4.1: Student Login Interface Design (Ain,2026)

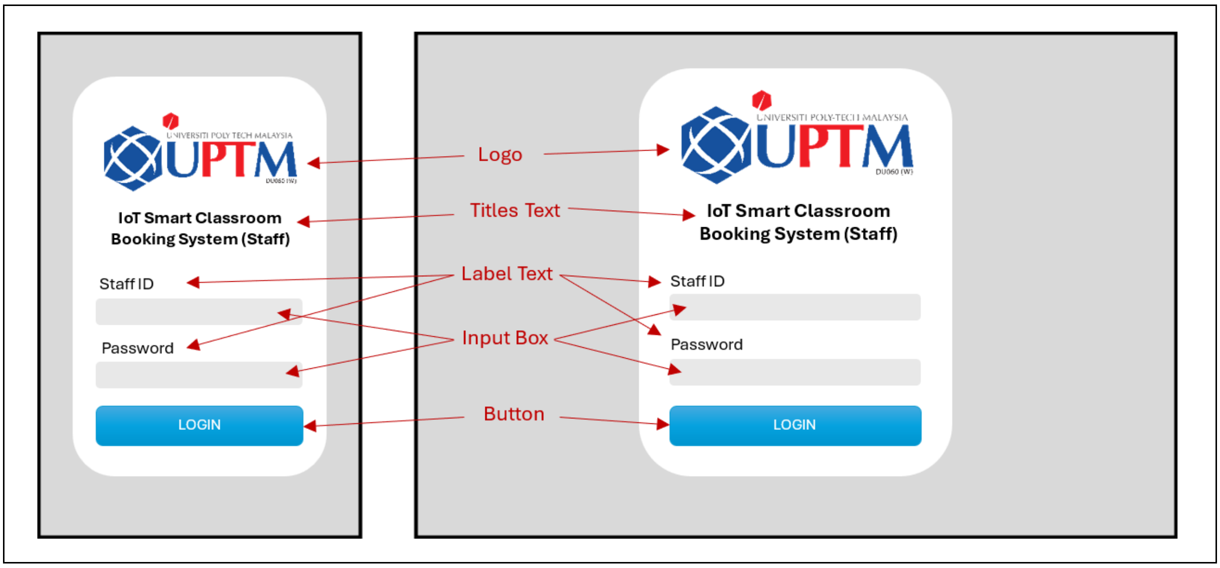


Figure 4.2: Staff Login Interface Design (Ain,2026

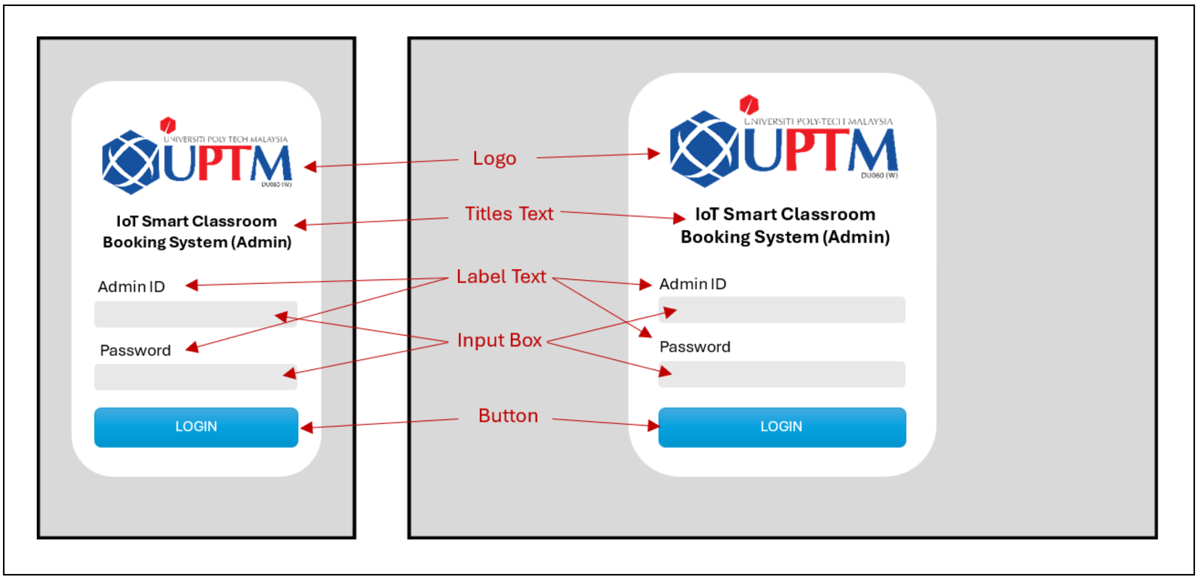


Figure 4.3: Admin Login Interface Design (Ain,2026)

The main point of access to the SCBS is the login page, which is meant to offer secure, role-based access to all the users. The system does support three different roles thus the differentiation between the students, lecturers(staff) and administrators is intended to be differentiated in the login process. Students and lecturers have equal traits, and administrators are provided with higher privileges and are redirected to a special dashboard. Role-based authentication will also ensure that only the functions that are with respect to the duties of a particular user are granted to the user.

The login page user interface should be basic but effective. The system logo and title come first in the page to introduce instant branding and context. Under this, two large input boxes are provided. The first is the user ID that is stored into the database as a unique identifier. The password is typed in the second input box, and this is masked to ensure security. Both fields are used through input validation to avoid cases where the fields are left blank, and the credentials are different to what is maintained in the database. These inputs are important since they decide on the dashboard where the user will be redirected once they are authenticated successfully. Lastly there is the main button the "Login" button that sends the credentials to be checked.

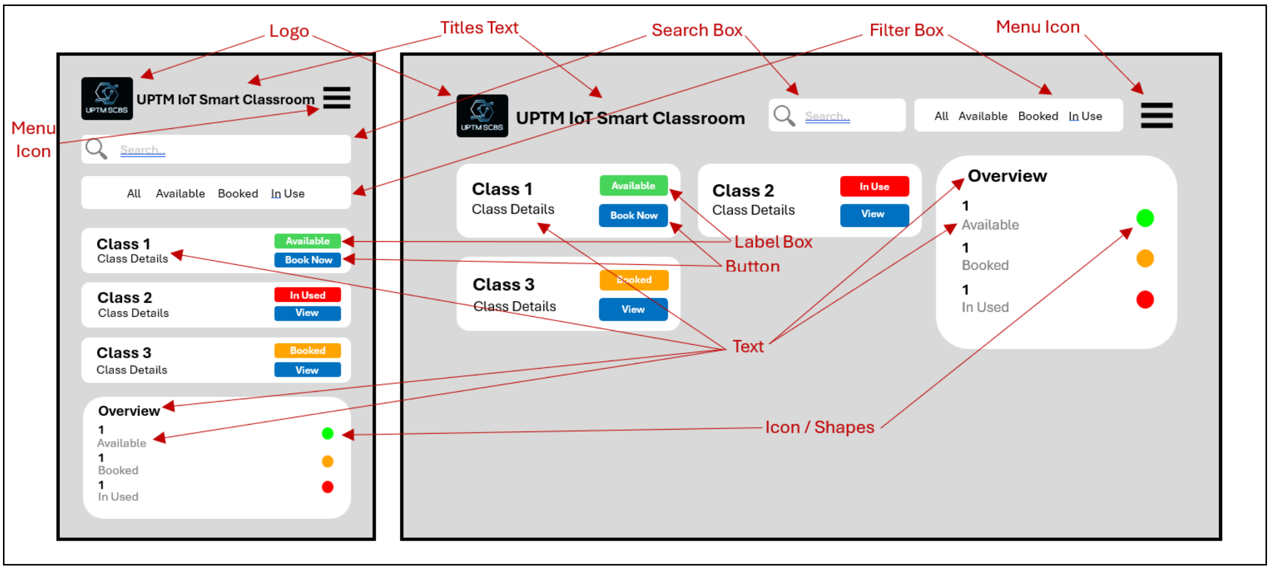


Figure 4.4: Main Dashboard Interface Design (Ain,2026)

Upon successful logging in, the primary dashboard becomes the central point of student and lecturers. This interface has an overall view of the availability of the classroom in real-time, combining records of booking with the data of the IoT sensors. The dashboard is divided into several essential parts. The main element is the grid of classrooms availability or list which shows all the classrooms and their status, Available (green), Booked (yellow), and In Use (red). These statuses are actively updated depending on planned bookings and dynamically updated on occupancy data through IoT sensors. This display takes a short time to be seen by the users and determine which rooms can be used immediately or booking in advance. The interface has filtering possibilities to classify the classrooms based on the capacity, facilities or location so that the user can locate the appropriate rooms quickly. This dashboard satisfies the need of real-time availability reported in the problem statement, which is associated with the challenge of using outdated paper schedules and making informed decisions regarding academic activities.

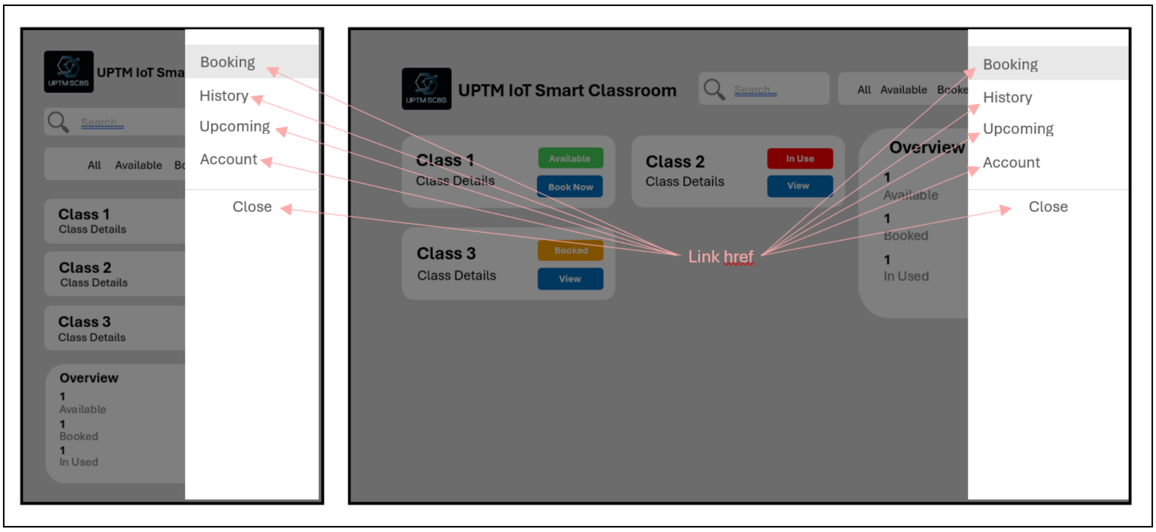


Figure 4.5: Menu section Interface Design (Ain,2026)

The menu section provides intuitive navigation throughout the SCBS application. Designed for consistency across all pages, the menu appears as a hamburger icon for both desktop and mobile devices to preserve screen space while maintaining access to all functions. The menu is organized into logical sections based on user needs and system functions. Primary navigation options include "Booking" for returning to the main availability view and initiating a new reservation, "History" for accessing past booking records, "Upcoming" for viewing and managing existing reservations and "Account" for managing profile settings. Each menu item is represented by text label, improving usability through visual recognition. The active page is highlighted to provide orientation within the system. This menu structure ensures that users can easily navigate between system features without confusion, supporting the usability goals identified in the non-functional requirements.

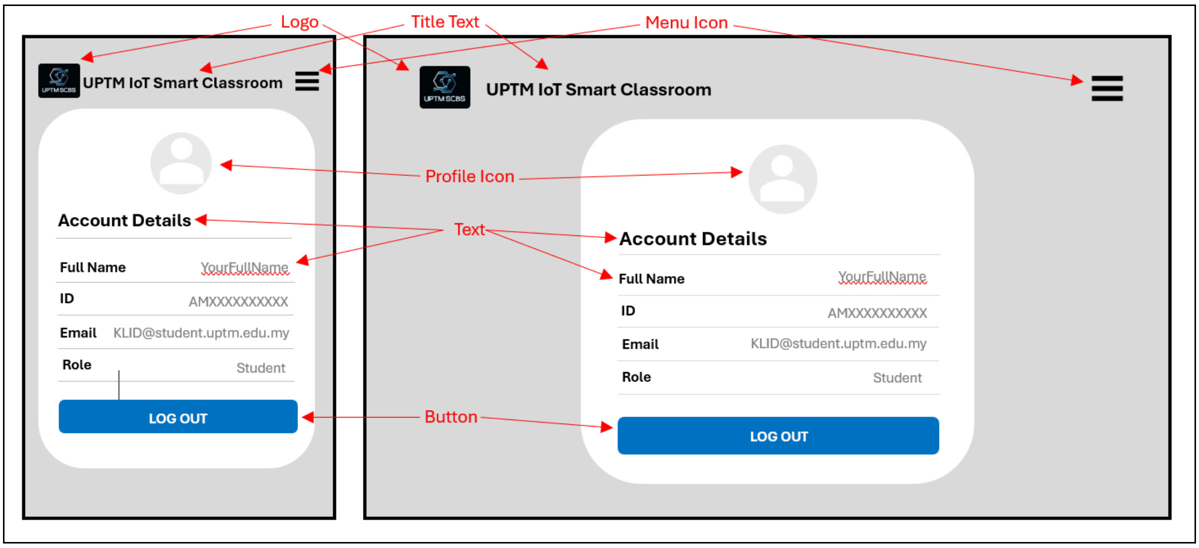


Figure 4.6: Account Interface Design (Ain,2026)

The account interface enables the user to monitor and maintain personal data of the SCBS. On this page, the user profile information such as name, user id, email address and department or course association are displayed. The data is given in read only format in a clean format with an log out profile button which allows the user to log out from the account. This interface will respond to the privacy and personalization needs that were developed during the questionnaire analysis as the people believed that they needed secure authentication and data protection. The system reassures user experience and ensures security by giving the users the ability to control their profile information and their notification preferences.

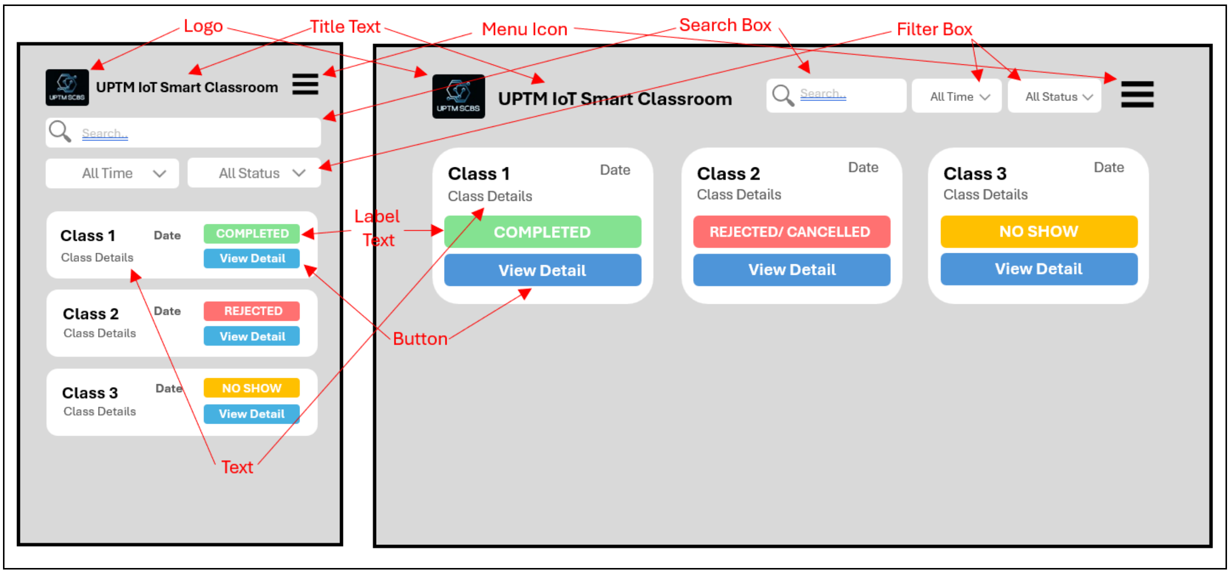


Figure 4.7: History and Upcoming Interface Design (Ain,2026)

The history interface will provide the user with detailed history of all his past bookings in the classroom. The booking is displayed on this page in chronological list, or table format with separate visual separation using coloured label between rejected, cancelled, no show reservations and completed reservations. Each entry of the history includes such fundamental data as classroom name, date, time slot and booking status (confirmed, pending, completed, cancelled or rejected). Filter and search are available, which allow the user to streamline the listing based on the date, classroom, or the status. In the case of upcoming bookings, it will be like the history interface. The action button of upcoming page will provide the same entry entails and will enable the user to cancel and process the reservation without visiting the administrative counters, which will satisfy the needs of the respondents who requested this option. The interface history also planned to satisfies the foresight of the requirement of accessing the history that had been identified in the survey of the users who desired to access history. This openness will enable the users to track the inclinations of their classroom use and accountability of the resource use.

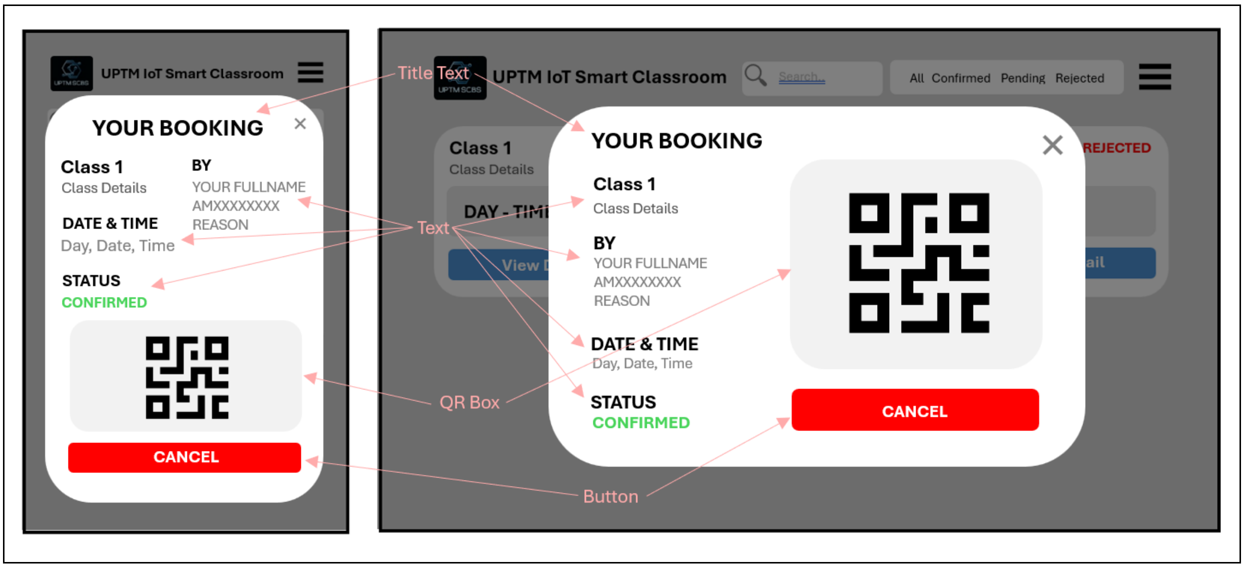


Figure 4.8: View Detail Interface Design (Ain,2026)

The view detail interface consists of detailed information concerning a particular booking when it is chosen in the history or upcoming pages. In this page, all the pertinent information has been posted in an easily readable format. In the following, the important details will be provided in the labelled sections: Classroom Details (name, code, capacity, facilities), Booking Schedule (date, start time, end time), Purpose (reason provided by the user when booking), and QR Code (generated unique code used to access the classroom). In the right-hand side of the desktop page and the bellow of the mobile page, the QR code is visible. They were scaled correctly so that they can be scanned by IoT devices at the entrance of the classroom. The QR code can be opened at the time of booking. There are action buttons located at the bottom which enable the user to make changes or cancel the reservation in case they are still pending and have the time frame to do so. This interface is the digital evidence of a reservation and access credential, which achieves the QR code-based access control goal in the project goals.

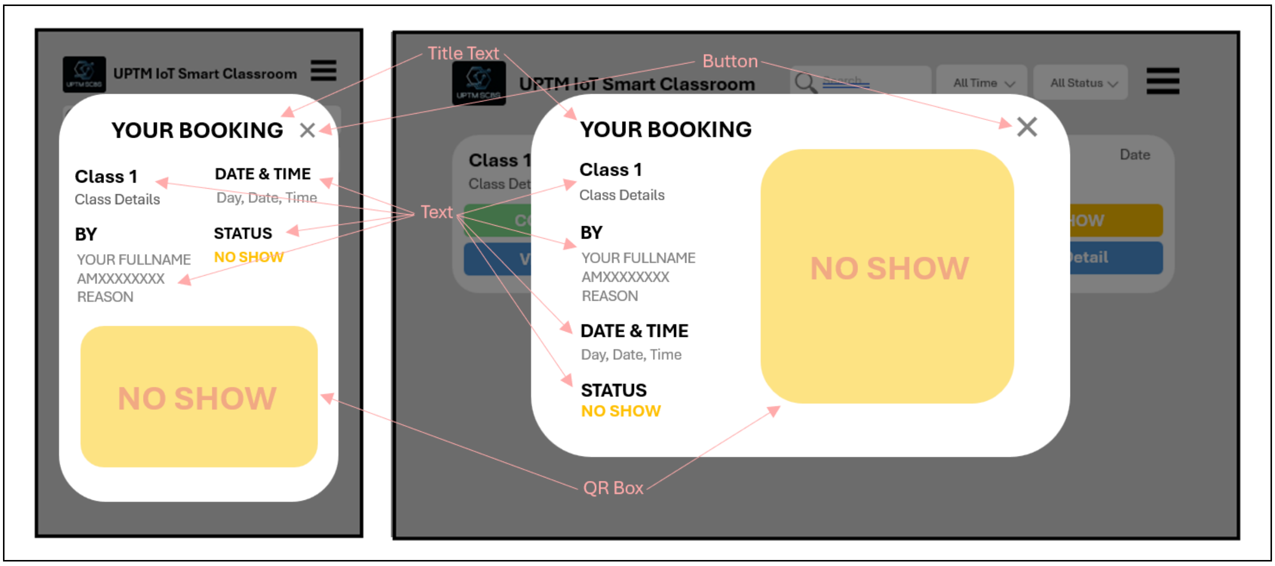


Figure 4.9: View Detail Interface Design for Invalid Booking (Ain,2026)

It is an interface that is used to show the error state when a user tries to access a booking that is invalid, expired, or not available anymore. The layout is similar to the regular view detail page, but the booking data is substituted with clear error messages. One of the striking warning signs and messages like "NO SHOW" or "CANCELLED" will instantly inform the user of the problem. In this modal, the information that is important is provided in labelled sections as it was on the history page and QR code will not appear on the history page. This error handling design proves to have a careful consideration of the user experience, such that when an error arises, the user is guided in the right direction instead of being confused. The interface supports the usability concerns found in the questionnaire where 98 percent of the participants highlighted the need to have a mobile friendly and easy to use interface that manages errors in a graceful manner.

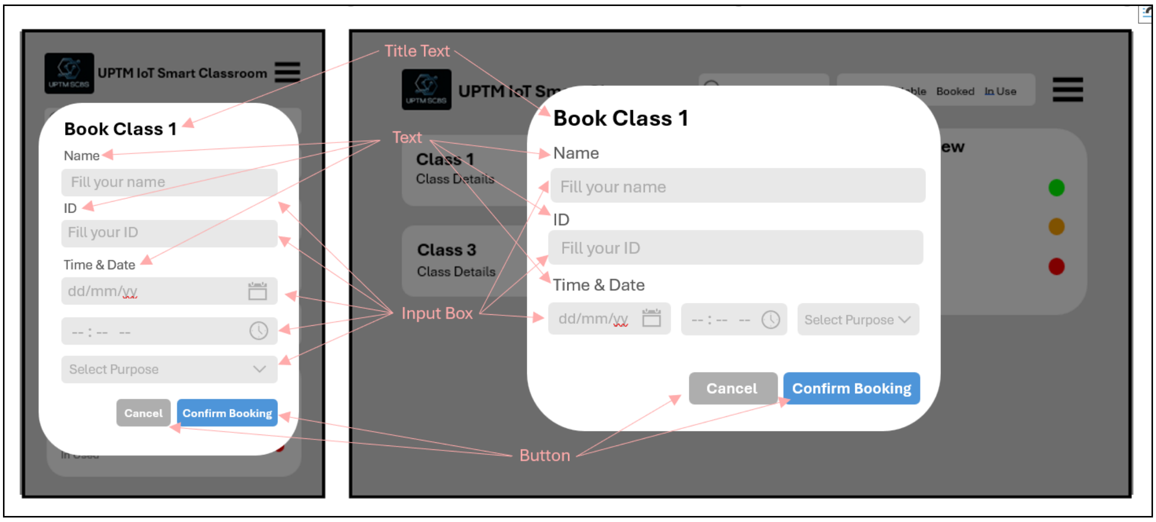


Figure 4.10: Booking Form Interface Design (Ain,2026)

The booking form interface provides a structured way for users to create new classroom reservations. This form is triggered when the “Book” button is pushed. The form is designed with clarity and efficiency in mind, guiding users through each required piece of information. At the top, a heading such as "Booking (Class Name)" establishes the purpose. The form is organized into logical sections, User Detail Selection (input box for the username and ID), Date Selection (calendar picker with visual indicators of availability), Time Selection (start and end time pickers with validation to ensure logical scheduling), and Purpose (dropdown for describing the reason for booking, such as lecture, group study, or project meeting). Real-time validation checks availability as the user selects date and time, preventing submission of conflicting bookings. A "Check Availability" button may update the display dynamically. Once all fields are completed, a "Submit Booking Request" button sends the reservation for administrator approval. This form directly addresses the requirement for an online booking system that eliminates the inefficiencies of manual paper-based processes, supporting the respondents who wanted online modification capabilities and the need for automated conflict prevention.

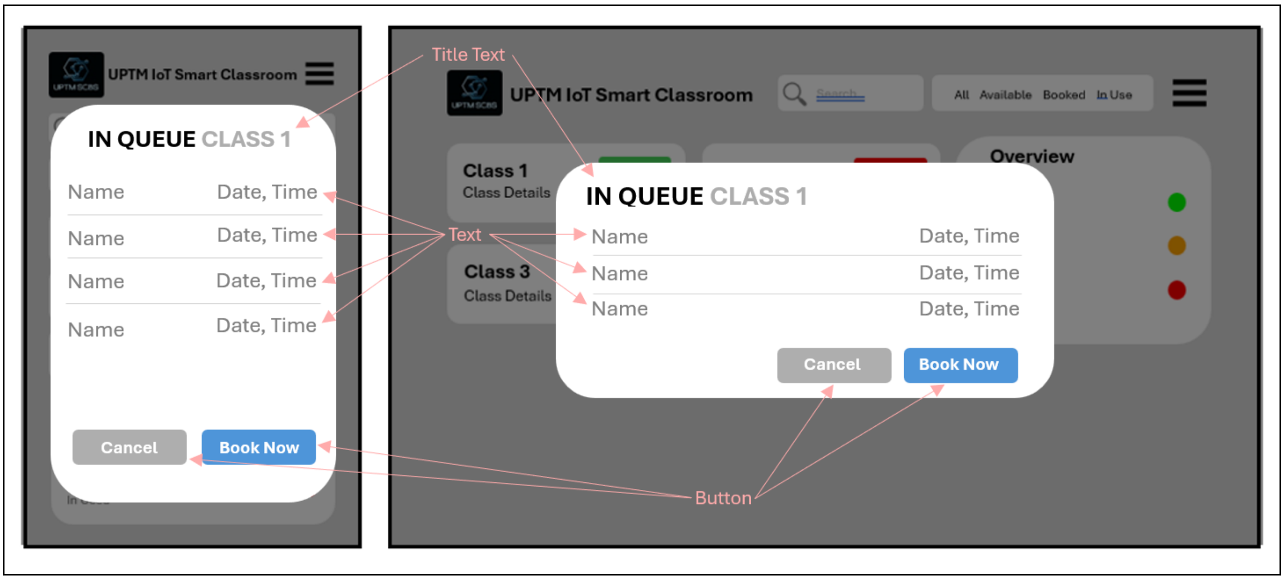


Figure 4.11: In Queue Interface Design (Ain,2026)

The in-queue interface is shown when the status of the current class is status in use or booked. This screen gives all the confirmed in a particular classroom, to ensure that who is in turn then goes to the booking form should they desire. The user is informed of the detail of user (username) and confirmed date and time of booking by explanatory text below. The user can make proper plans of their next booking by having clear status communication.

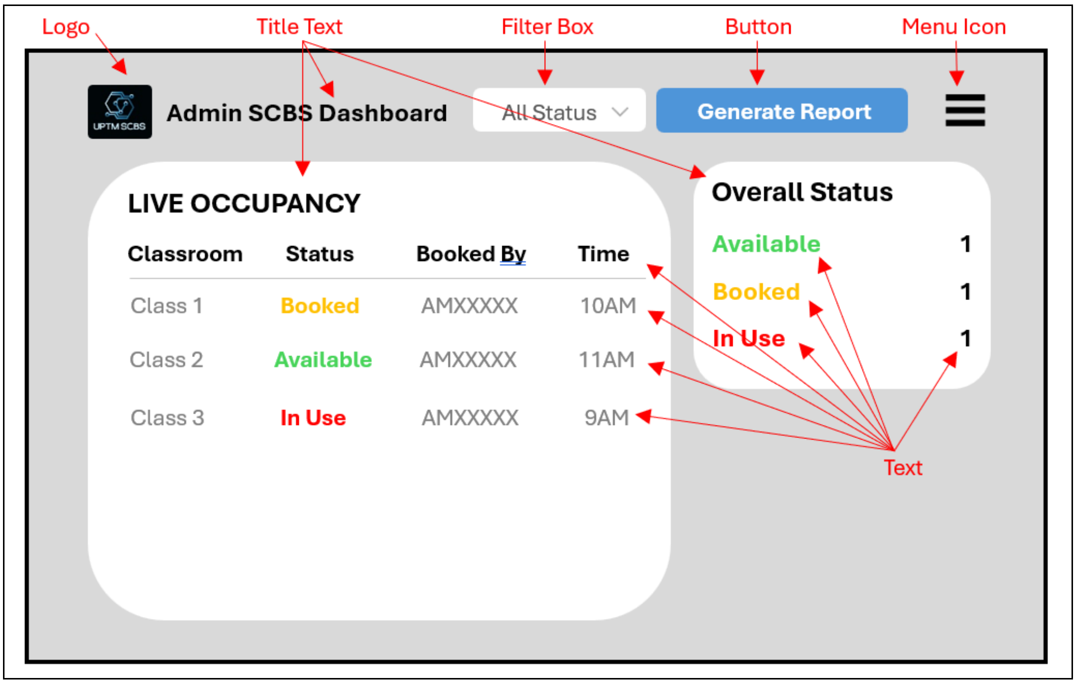


Figure 4.12: Admin Dashboard Interface Design (Ain,2026)

The system administration control panel or the admin dashboard is the central control panel of the system administrators and has extensive monitoring and administration capabilities. This interface is efficiency-oriented and data visualization-oriented to allow administrators to monitor all classroom events in brief depending on the real time. The dashboard has been divided into some important areas. On the left, there are summary cards with important metrics up to date on the status of current classrooms (available, booked, in use). To the right, an occupancy grid is presented in real-time, which demonstrates all the classrooms with their status (available, booked, in use) based on IoT sensors and the specificity of the confirmed occupancy of the booking (user detail, book time). The admin can also filter and search any desires classroom he or she would like to monitor at the top. The generate report button is also given at the top as optional under the admin summary report. The purpose of this extensive dashboard is to achieve the goal of administrative monitoring platform that will improve transparency, accountability, and informed decision-making in classroom allocation and facility planning.

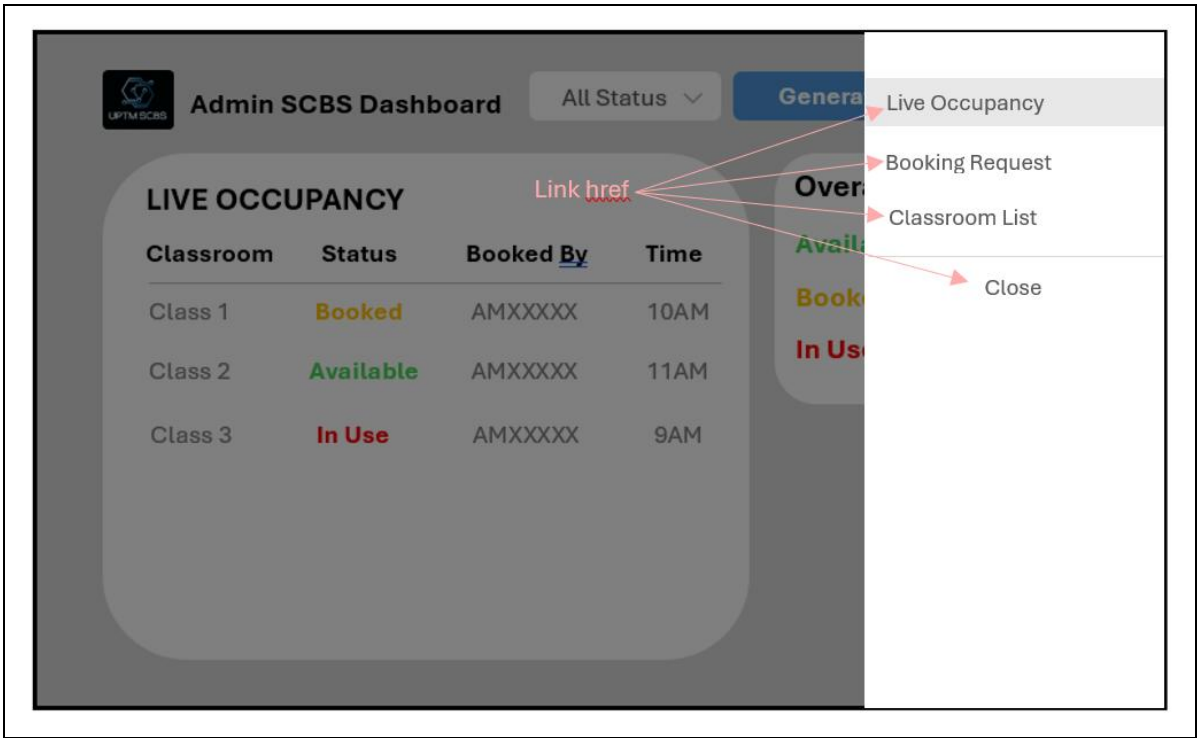


Figure 4.13: Admin Menu section Interface Design (Ain,2026)

The SCBS has an admin menu section, which is used to navigate administrator-specific functions. The menu is also more elaborate compared to the typical user menu in line with the increased duties of the administrative personnel. The menu is grouped into rational sections: Live Occupancy (go back to main administration panel), Booking Request (accept/deny requests, see all the booking request), and Classroom List (add, edit, delete classrooms). Every menu item has text labels to make it readable and the active part is also highlighted to indicate the present position. In the case of the admin features, it only supports desktop view because of professionalism. The menu is designed to enable administrators to move easily between the numerous management tools necessary to manage the whole classroom booking ecosystem to aid the overall control needs that were realized in the objectives of the project.

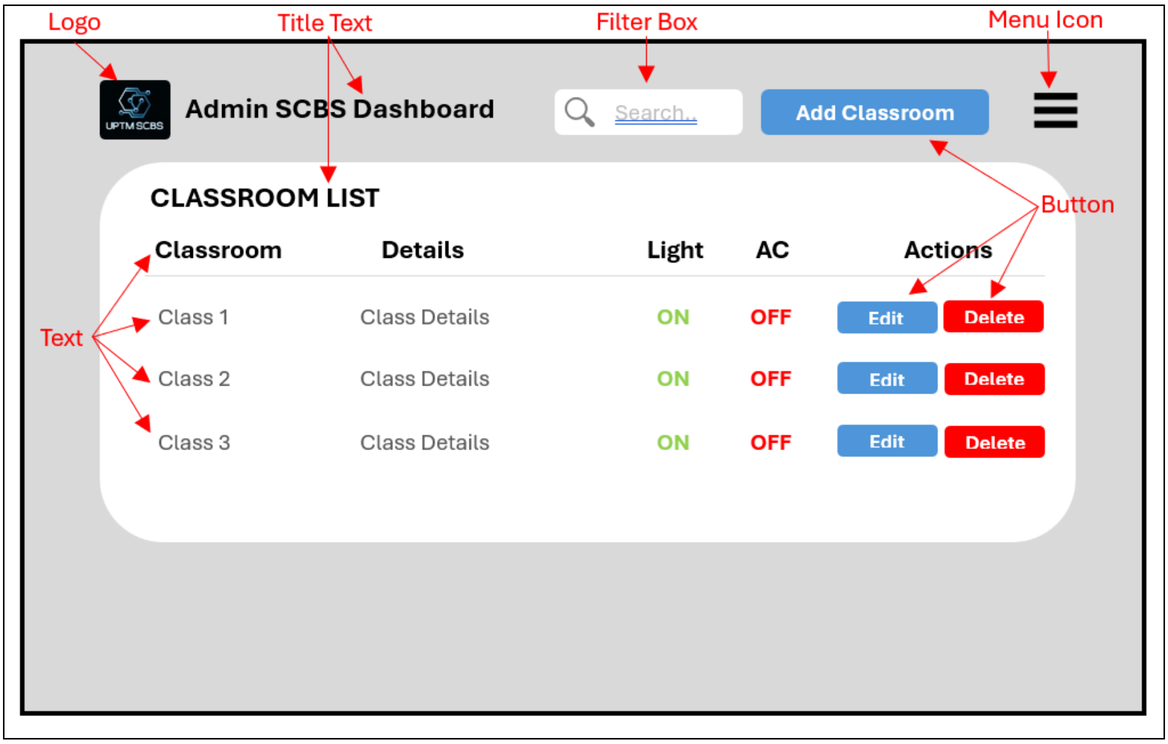


Figure 4.14: Classroom List Interface Design (Ain,2026)

The classroom list interface provides administrators with a comprehensive view of all classrooms in the system. This page displays classrooms in a sortable, filterable table format with key information for each room. Columns include Classroom Name, Classroom Details (Capacity, Facilities), Lights and Air Conditions Status (online/offline), and Action buttons for editing or deleting. Search and filter options allow narrowing the list by building, capacity range, or available facilities. A prominent "Add New Classroom" button at the top enables quick addition of new rooms to the system. The interface supports pagination for large numbers of classrooms, ensuring performance and usability. This page fulfills the requirement for administrators to manage classroom data effectively, including the ability to add, edit, or delete rooms as facilities change. By providing a centralized view of all classrooms, the system ensures that the resource list remains accurate and up to date.

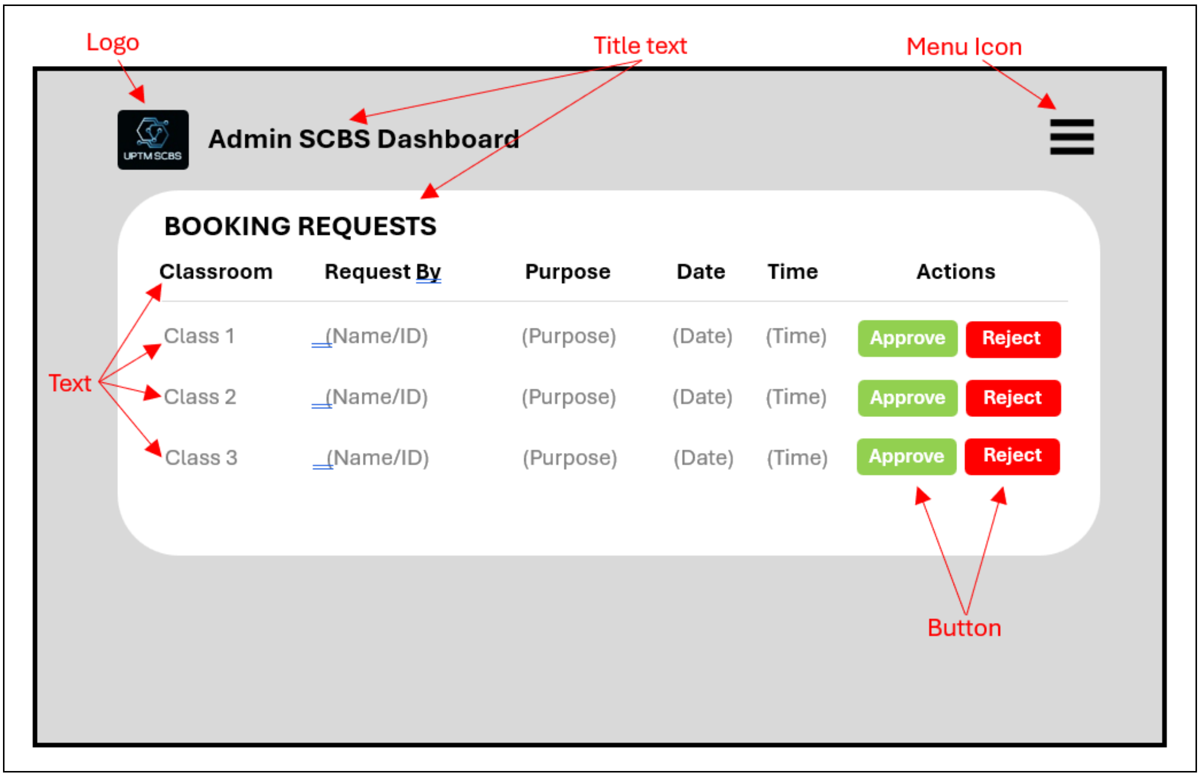


Figure 4.15: Booking Request Interface Design (Ain,2026)

The booking request interface shows all the pending reservations that will have to be approved by the administrators. The page follows an effective reviewing and decision-making structure serving requests in a straightforward, doable format. Request is normally shown in a table or card format in a chronological order of the oldest pending requests shown first. The main information is provided in each entry, the name of the person requesting it (student/lecturer), the classroom requested, the purpose, date and time slot. There are action buttons against each request which are "Approve" and "Reject" and the latter will open a dialog where a reason will be given and sent to the user. In times of high volume, batch approval can be used so that an administrator can approve several non-conflicting requests at once. Conflict clues indicate any requests that can possibly conflict with existing approved reservation, however, the automated conflict prevention of the system should reduce such instances. This interface also responds to the requirement of having efficient administrative oversight, as the manual and paper-based approval process is replaced with the simplified digital workflow, which minimizes delays and enhances transparency.

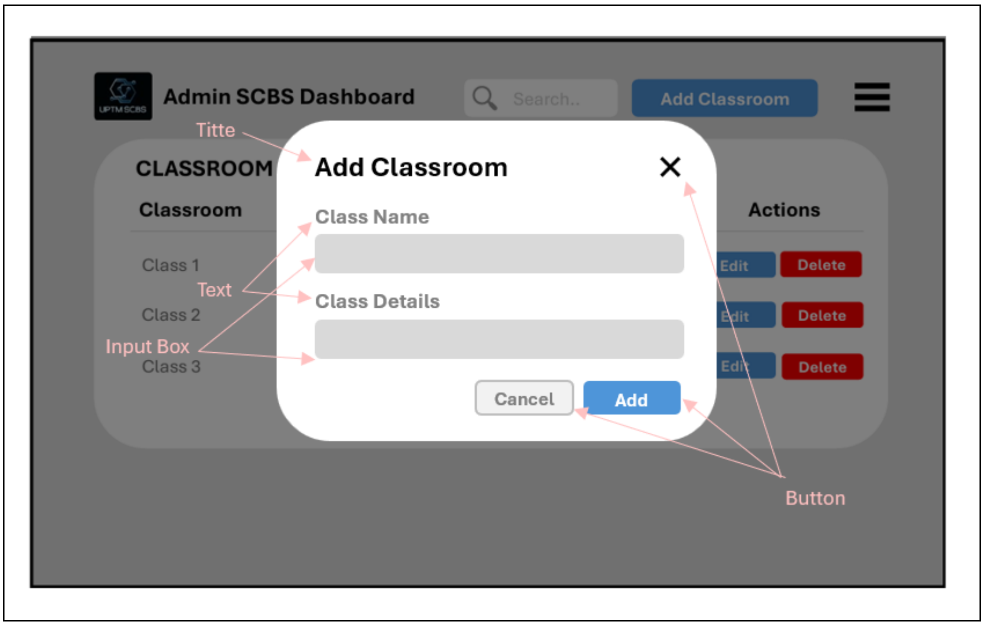


Figure 4.16: Add Classroom Interface Design (Ain,2026)

The add classroom interface offers a structured interface through which the administrators can add new classrooms to the SCBS database. This form will ensure that all the required information is documented in a uniform manner with regards to each new room. On filling the form, there is an add button to submit the data and cancel to go back to the classroom list without storing the information. This interface allows the effective extension of the system with the addition of new classrooms or the renovation of the existing ones to provide a complete and accurate database on classrooms to users who make their booking decisions.

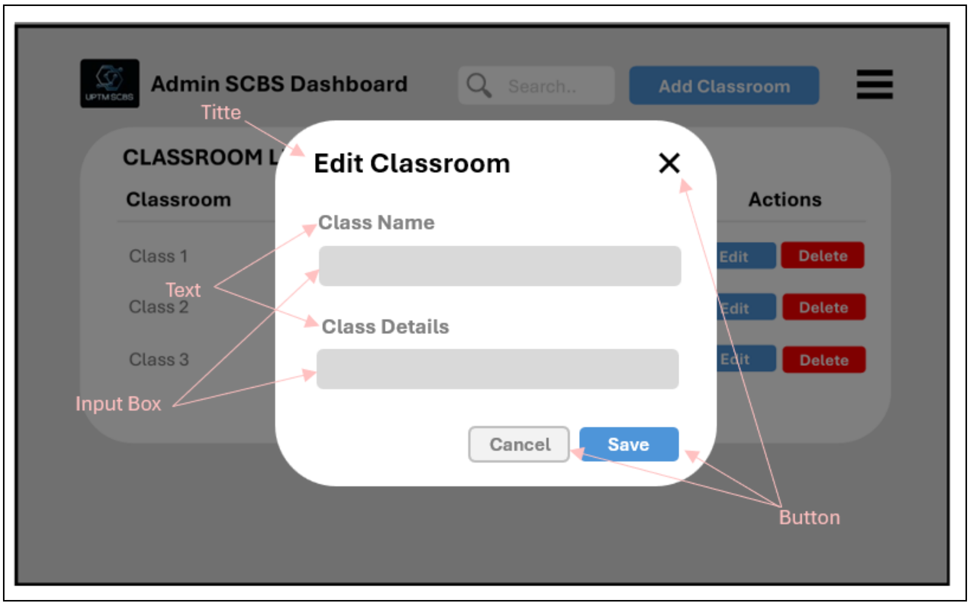


Figure 4.17: Edit Classroom Interface Design (Ain,2026)

The edit classroom interface enables an administrator to edit the information of current classrooms in the system. This type is pre-filled with the existing data at the classroom level, which allows updating it quickly without having to add all the data. The interface is modeled after the add classroom, and there are tabs with the Basic Information, Physical Specifications, Facilities and Equipment, IoT Configuration, and Status Settings. Any field can be changed by administrators, and validation ensures that there is no loss of data integrity (i.e. room code is not duplicated on change). It has an interface with a "Save" button to save and a cancel button to drop changes. The system allows administrators to maintain the resource database up to date without technical assistance since it has easy editing features.

4.3 Database Design

4.3.1 Data Dictionary

The data dictionary serves as the foundation of the Smart Classroom Booking System (SCBS) database design. It defines the structure of each table, the attributes contained within them, and the relationships that exist across the system. By documenting the data dictionary, the project ensures consistency, clarity, and integrity during implementation. Each table is designed to support the functional requirements of classroom booking, IoT monitoring, and administrative control, while also maintaining normalization to reduce redundancy.

The users table is central to the system, as it stores all information related to students, lecturers, and administrators. Attributes include id_user as the primary key, which is defined as a VARCHAR(20) to accommodate alphanumeric identifiers such as STU001 or ADM001. Other attributes include name_user, email, password, course_or_department, and role. The role field differentiates between students, lecturers, and administrators, ensuring that the system can enforce role-based access control. This table is critical because it links directly to the booking and administrative logs, providing accountabilities for all actions performed within the system.

Table 4.4.1: Data Dictionary of User Table

Table	Field	Data Type	Description
users	id_user	VARCHAR(20), Primary Key	Unique ID for each user
	name_user	VARCHAR(100)	Full name of the user
	email	VARCHAR(100)	Email address used for login
	password	VARCHAR(255)	Encrypted password
	course/departm	VARCHAR(100)	User's academic course or

	ent		department
	role	ENUM	Role of the user: student, lecturer, admin

The Booking table records all reservations made by users. Its attributes include id_booking as the primary key, id_user as a foreign key referencing the users table, and id_class as a foreign key referencing the classroom table. Other attributes include name_user, purpose, start_time, end_time, status_ booking, qr_code, check_in and check_out. The QR code field stores the unique code generated for each booking, which is later used for classroom access. This table is essential for managing reservations, preventing double-booking, and maintaining a historical record of classroom usage.

Table 4.2: Data Dictionary of Booking Table

Table	Field	Data Type	Description
booking	id_booking	INT, Primary Key	Unique ID for each booking
	id_user	VARCHAR(20), Foreign Key	Links to the user who made the booking
	id_class	INT, Foreign Key	Links to the classroom being booked
	name_user	VARCHAR(100)	Name of the person booking (redundant but shown)
	purpose	TEXT	Reason for booking the classroom
	time (start/end)	DATETIME	Start and end time of the booking
	status_booking	ENUM	Booking status: pending, confirm, reject, completed, cancel
	qr_code	VARCHAR(255)	QR code for access control
	check_in	DATETIME	Time when user checks into the classroom
	check_out	DATETIME	Time when user checks out of the classroom

The classroom table documents the physical classrooms available for booking. Its attributes include id_class as the primary key, name_class, code, detail_class and status_class. The status field indicates whether a classroom is available, booked, or currently in use. Additional attributes iot_device may include IoT device references, such as sensor IDs, which connect the classroom to occupancy monitoring and energy control systems. This table ensures that each classroom is uniquely identified and can be managed effectively within the booking process.

Table 4.3: Data Dictionary of Classroom Table

Table	Field	Data Type	Description
classroom	id_class	INT, Primary Key	Unique ID for each classroom
	name_class	VARCHAR(100)	Name of the classroom
	code	VARCHAR(50)	Room code (e.g., CR 3.04)
	detail_class	TEXT	Description or features of the classroom
	status_class	ENUM	Current status: in use, booked, available
	iot_device	VARCHAR(100)	IoT device linked to the classroom (sensor/light/AC)

The Energy Action Log table records all energy control operations performed either automatically by the IoT system or manually by administrators. Attributes include id_energy as the primary key and id_class referencing the classroom and action_energy (ights on/off, AC on/off). This table provides a record of how energy resources are managed, supporting sustainability goals and enabling administrators to analyze consumption patterns.

Table 4.4: Data Dictionary of Energy Action Log Table

Table	Field	Data Type	Description
energy_action_lo g	id_energy	INT, Primary Key	Unique ID for each energy action
	id_class	INT, Foreign Key	Links to the classroom affected
	action_energy	ENUM	Type of action: AC/light on/off

The IoT Sensor Log table stores data collected from occupancy and environmental sensors. Attributes include id_iotsensor as the primary key, id_class referencing the classroom, and sensor readings such as montion_detect. This table integrates IoT functionality into the SCBS, allowing the system to monitor classroom conditions in real time and trigger automated energy actions.

Table 4.5: Data Dictionary of Iot Sensor Log Table

Table	Field	Data Type	Description

Iot_sensor_log	id_iotsensor	INT, Primary Key	Unique ID for each sensor log
	id_class	INT, Foreign Key	Links to the classroom being monitored
	montion_detect	TINYINT(1)	1 if motion is detected, 0 if not

Finally, the Admin Log table captures actions performed by administrators. Attributes include id_adminLog as the primary key, id_user referencing the admin role from user table, and id_booking referencing the booking being managed. Additional fields include action_admin (approve, reject, modify). This table ensures accountability by recording all administrative decisions, which can later be audited for transparency.

Table 4.6: Data Dictionary of Admin Log Table

Table	Field	Data Type	Description
admin_log	id_adminLog	INT, Primary Key	Unique ID for each admin action
	id_user	INT, Foreign Key	Links to the admin user
	id_booking	INT, Foreign Key	Links to the booking being acted on
	action_admin	ENUM	Action taken: approve or reject

4.3.2 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) of the Smart Classroom Booking System (SCBS) is a diagram that describes how data and interactions are supposed to flow between the users, system modules, and the IoT parts. It acts as a graphical guide in the processing of information within the system so that all the functional requirements are logically linked and effectively implemented. The diagram consists of four primary actors Lecturer, Student, Admin, and IoT Components and shows the flow of operations between the login and booking confirmation and access validation.

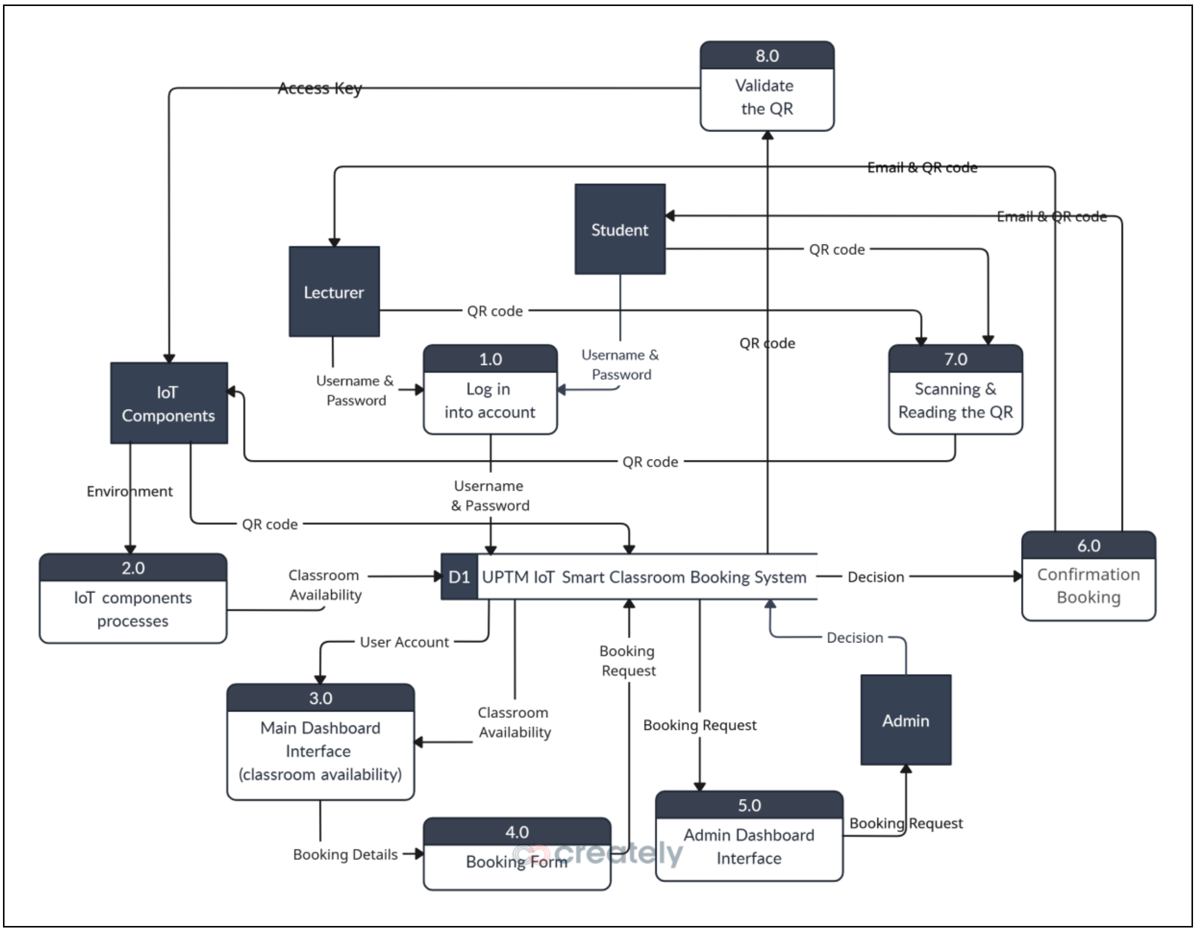


Figure 4.18: Data Flow Diagram of UPTM SCBS (Ain,2026)

The procedure starts with Module 1.0: Log in into account, whereby users identify themselves through the use of their access key, username, and password. This module compares the credentials with the user database and redirects users to their corresponding dashboard depending on the roles. The students and lecturers are guided to the main dashboard, and administrators have a special interface and monitoring and approval options. Role-based authentication will provide the user with access to the functions that he/she is authorized to perform based on his/her duties.

The Module 2.0: IoT Components Processes is turned on immediately after the login. This module signifies the unrelenting background tasks of the IoT system that measures the classroom settings with the help of the sensors like PIR motion detectors, temperature sensors, and light sensors. These elements gather real time information concerning the occupancy and the environmental conditions of the classroom. The system uses the data to check whether a classroom is occupied and to automatically perform energy saving measures, like switching off lights or air conditioning when no movement is detected. This integration will make sure that the system helps in achieving sustainability agendas and minimizing on unnecessary energy use.

After the authentication stage and the activation of the IoT monitoring, the users can communicate with the Module 3.0: Main Dashboard Interface, where real-time classroom availability is shown. This module reads classroom table information and combines the occupancy status of IoT sensors. The user can see which classrooms are available, booked or in use and therefore makes informed choices before making a booking request.

Module 4.0: Booking Form deals with the booking process, during which the users provide information about booking, including the choice of a classroom, purpose, date, and time. On submission, the system will create a booking request and save it on the booking table. At the same time, a QR-code is developed and sent to the user through email that will be utilized in the future when verifying access. This module makes sure that all booking information is properly captured and is associated with appropriate user and classroom.

Administrators deal with such requests using Module 5.0: Admin Dashboard Interface, which allows them to look at the wait list and decide to approve or disapprove a booking. This module records the action in the booking status and in the table of the log of actions of the administration. Verified reservations will cause confirmation messages and enable the QR code to be opened. There is also the admin dashboard with the tools to monitor the use of classrooms and the use of energy to assist transparency and accountability.

After the confirmation of a booking, the user goes to Module 6.0: Confirmation Booking that completes the reservation and prepares the system to provide access control. Upon entering the classroom, the users will work with Module 7.0: Scanning & Reading the QR, during which a QR code is read by an IoT-enabled device like ESP32-CAM.

This scan is transmitted to Module 8.0: Validate the QR that verifies the code with the booking database to verify its authenticity and timing. This will make sure that only the authorized users can enter into the classroom at the time they have been allocated.

All the information handled by these modules is stored in Data Store D1: UPTM IoT Smart Classroom Booking System that can be considered the central repository of user accounts, booking history, classroom conditions and sensor data, and administrative operations. This data store makes sure that the information within the system is consistent and intact as all the modules have access to the information and can update it in real time.

Summing up, the DFD has a clear and rational flow of operations, which starts with the user login and the IoT activation, then proceeds to the dashboard interaction, booking submission, administrative review, and access validation. It also over emphasizes interdependence of system modules as well as the significance of real-time data in the realization of SCBS goals of efficiency, transparency, and sustainability.

4.3.3 Entity Relational Diagram (ERD)

The Smart Classroom Booking System (SCBS) has six entities, User, Booking, Admin_Log, Classroom, Iot_Sensor_Log, and Energy_Action_Log, which are defined in the Entity Relationship Diagram (ERD). Every component has a niche in the system and connections among them will guarantee that booking processes, administrative functions, and IoT monitoring are combined and harmonized.

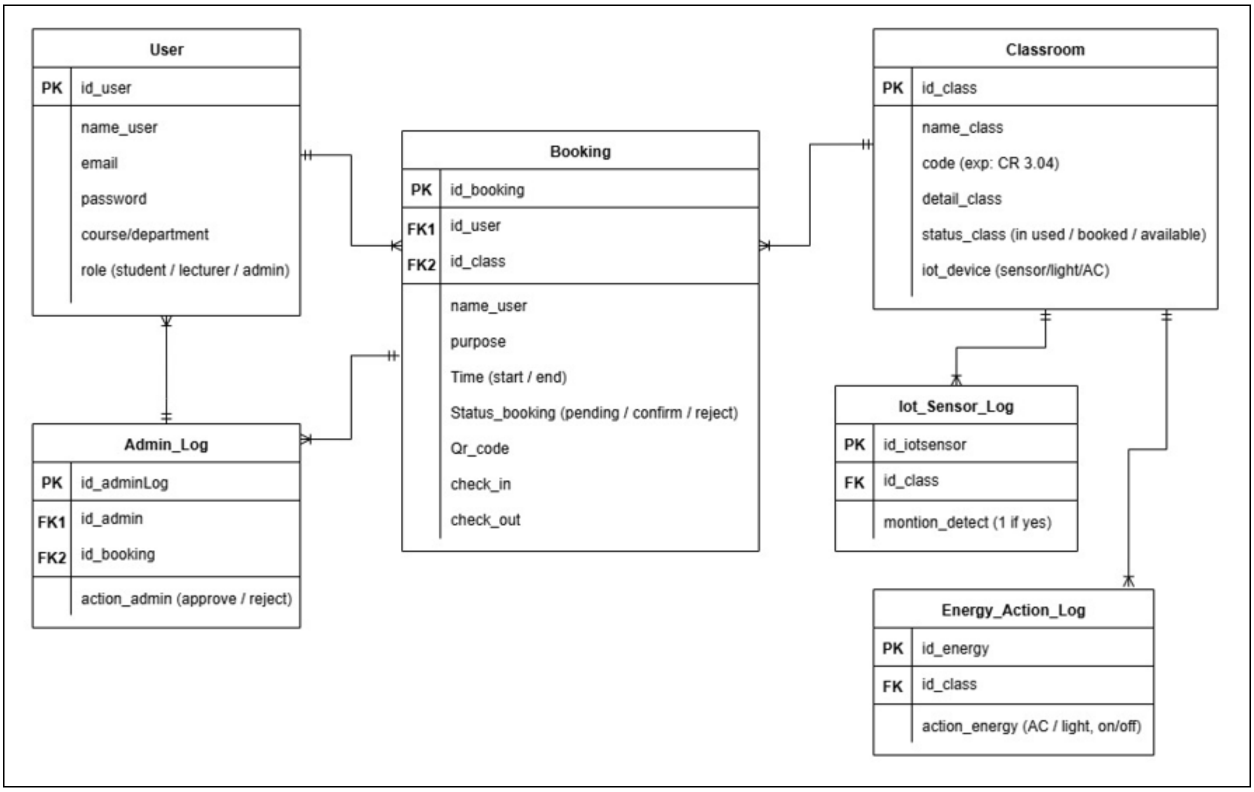


Figure 4.19: Entity Relational Diagram of UPTM SCBS (Ain,2026)

The User entity represents all individuals who interact with the system, including students, lecturers, and administrators. Each user is uniquely identified by id_user. A user can create many bookings, but each booking record belongs to one specific user. This relationship ensures that every reservation is tied to the person who made it. For administrators, the User entity also connects to the Admin_Log entity, meaning that one administrator can perform multiple actions on bookings, but each log entry is linked to a single administrator.

The Booking entity records all classroom reservations. Each booking is identified by id_booking and is linked to one user and one classroom. A booking can be reviewed multiple times by administrators, which creates multiple entries in the Admin_Log entity. This relationship ensures that all administrative decisions are tied directly to the bookings they affect. The Booking entity also contains details such as purpose, time slots, QR code, and check-in/out records, which provide accountability and support access control.

The Classroom entity defines the physical classrooms available for booking. Each classroom is identified by id_class. A classroom can be reserved many times, but each booking record refers to one specific classroom. Classrooms also connect to IoT_Sensor_Log and Energy_Action_Log, meaning that one classroom can generate many sensor logs and many energy action records. This design makes the Classroom entity the structural hub of the system, linking physical spaces to both user reservations and IoT monitoring.

The Admin_Log entity documents actions taken by administrators. Each log entry is tied to one administrator and one booking. This means that an administrator can perform many actions, and a booking can have multiple actions recorded, but each log entry connects a single administrator to a single booking. This relationship ensures transparency and traceability of administrative decisions.

The IoT_Sensor_Log entity stores environmental data collected from classrooms. Each sensor log belongs to one classroom, but a classroom can generate many sensors logs over time. This relationship supports occupancy monitoring and allows the system to adjust energy usage based on real-time conditions.

The Energy_Action_Log entity records energy control operations. Each action log belongs to one classroom, but a classroom can have many actions recorded, such as turning lights or air conditioning on and off. This relationship ensures that energy management is documented and linked to the correct physical space.

In summary, the ERD demonstrates a well-structured and normalized database design where each entity plays a distinct role, and the relationships between them guarantee consistency and accountability. Users are linked to bookings, administrators are linked to booking actions, and classrooms serve as the central hub connecting reservations, sensor data, and energy actions. This relational model ensures that the SCBS can manage classroom reservations securely, monitor occupancy through IoT integration, and optimize energy usage effectively, fulfilling the system’s objectives of efficiency, transparency, and sustainability.

The User entity is the one which describes all the individuals who have interaction with the system, and these include all the students, lecturers and the administrators as well. Id user identifies each user uniquely. A user is allowed to make numerous bookings however, the booking record is unique to a particular user. Such a connection guarantees that each reservation is attached to the individual who placed it. To the administrators, the User entity also relates to the Admin_Log entity, that is, a single administrator can do several actions on the bookings, but each entry of the log is associated with one administrator.

All the classroom reservations are registered in the Booking entity. Every booking is defined by id booking and has one user and one classroom attached. Administrators can review a booking multiple times, and this adds several records to the Admin_Log entity. Such a connection means that the administrative decisions will be directly linked to the bookings they influence. Also in the Booking entity is the purpose, time slots, QR code and check-in/out records, which offer accountability and facilitates access control.

The classroom entity refers to the physical classroom that can be booked. id_class is used to identify each classroom. A classroom may be booked into many times, but the booking record is that of a single classroom. The classrooms are also linked to the IoT sensor log and energy action log, implying that a single classroom can produce numerous sensor logs and numerous energy action records. This architecture causes the Classroom entity to be the structural center of the system connecting physical space to user booking and IoT monitoring.

Admin_Log is an entity that records the administrative actions. The bookings are associated with one administrator and each log entry. This implies that a person in administration can do numerous things, and a booking can have numerous actions logged to it, but each log entry links only one administrator to one booking. This association provides openness and traceability of administrative rulings.

The IoT sensor log entity is where environmental data in the classroom is stored. A sensor log is associated with a single classroom, although a classroom may produce many sensor logs. This connection helps in tracking occupancy and enables the system to vary energy consumption depending on the real-time situations. Energy control operations are registered in the Energy_Action+Log entity. There are several actions logs in each classroom, and a classroom may have numerous actions logged, like turning lights or air conditioning on or off. This connection will make sure that energy management is recorded and connected to the appropriate physical space.

To conclude, the ERD illustrates a highly ordered and normalized database design with each entity having a clear role to play and the relationship between entities ensures consistency and accountability. Bookings and administrators have connections with users, and classrooms are the center of integration involving reservations, sensor data, and energy actions. This type of relational model will make sure the SCBS can create reservations of the classrooms safely, be able to monitor the occupancy with the help of IoT, and to achieve optimal energy consumption to meet the goals of the system which are efficiency, transparency, and sustainability.

4.4 Flow of the System

The flow of the Smart Classroom Booking System (SCBS) illustrates the interaction of modules to provide the entire booking and monitoring process. Although Chapter 4.3 showed the database design and Chapter 4.3.2 described the Data Flow Diagram (DFD), here the section describes the sequence of operation of the system in practice. It demonstrates the collaboration between users, administrators and IoT components to meet the goals of Chapter 3, which are secure classroom booking, transparent monitoring and sustainable management of energy.

4.4.1 Module 1.0: Login into Account

The first stage of contact between the users and the SCBS is the login module. It is also secure and role-based authentication which means that a user can only access the functions that are relevant to the duties that he or she is expected to perform. The credentials are checked against the User table and when authenticated, the students and lecturers are redirected to the main dashboard with administrators redirected to a specialized dashboard. This is a role-based redirection which ensures security and is aligned to the system requirement of avoiding unauthorized access.

4.4.2 Module 2.0: IoT Components Processes

Upon a successful login, the IoT components start to run in the background to track the classroom environments. The data of motion sensors, temperature sensors and light sensors is constantly updated in the IoT_Sensor_Log table providing the system with the opportunity to check the occupancy and environmental conditions on the spot. According to these readings, the system initiates automated energy behaviors like turning of lights or air conditioning on or off. The records of every action are written in Energy Action Log table meaning they are traceable. This module is thus able to help in proper availability of the classroom and also in the sustainability of the classroom through minimization of unneeded energy usage.

4.4.3 Module 3.0: Main Dashboard Interface

After authentication, users can communicate with the main dashboard, which shows the availability of the classrooms, integrating booking records and the data of the IoT sensors. This interface enables students and lecturers to see what classrooms are open, reserved or in use. The dashboard will offer transparency by incorporating real time sensor information to avoid scheduling conflicts and achieve the usability and reliability characteristics of the system.

4.4.4 Module 4.0: Booking Form

Upon a user deciding to book a classroom, they will be directed to the booking form. In this case, they will key in information like classroom choice, reason, date and time. This system checks the availability and then the booking is confirmed. After the approval, a QR code is created and sent to the user through email. This QR is a secure access token which will directly point to the Booking table attributes check in and check out. This module will make reservations safe, trackable and aligned with the functional specifications of the system.

4.4.5 Module 5.0: Admin Dashboard Interface

The management of the administrative actions is carried out by the admin dashboard. Administrators check the requests of booking and decide on their approval or disapproval. The records of their decisions are kept in the Admin_Log table, which makes them accountable. Bookings are approved, and the QR code is activated to access the bookings. This module indicates the system focus on transparency and control, which goes into the issue of unauthorized classroom usage that was mentioned in Chapter 2.

4.4.6 Module 6.0: Confirmation Booking

After the approval, the reservation will be made in the confirmation booking module. This module changes the status of a booking to the confirmed one and prepares the system to access control. The users are informed about their successful reservation and given the QR code to use in order to gain access. This measure is important to guarantee complete booking lifecycle and to provide the user with the required credentials to access the classroom.

4.4.7 Module 7.0: Scanning and Reading QR

At the time of classroom entry, users present their QR code to an IoT enabled device such as ESP32 CAM. The gadget reads the code and transmits the information to the system to be verified. This module will allow only authorized users to try to enter classrooms, which supports the security of the system.

4.4.8 Module 8.0: Validate the QR

All information that is processed by these modules is saved in the central repository, the UPTM IoT Smart Classroom Booking System database. This guarantees that user accounts, booking history, classroom availability, sensor history and administrative activities are always updated and viewable at any given time. The system ensures integrity with effective retrieval of information by connecting the flow of modules to the database design described in Chapter 4.3.

To conclude, the system process starts with login and IoT activation, then it continues with dashboard interaction, booking submission, administrative review, confirmation and QR validation. All modules have different functions and when combined they create a well-knit workflow that helps in efficiency, transparency and sustainability of classroom management. This flow is based on the requirements of Chapter 3 and the design aspects of Chapter 4.2 and 4.3; it shows how the SCBS will work after implementation to make sure the design planned is in line with the objectives of the system.

4.5 Implementation

The SCBS integrates some important functions that focus on the essence of the objectives that were identified in Chapter 1 and confirmed by the analysis of the questionnaires. Those are the most important innovations and value propositions of the system.

4.5.1 Execution Platform

The Smart Classroom Booking System (SCBS) will be installed and implemented in full in a local server setup using XAMPP without the need to use cloud hosting or external version control software. The approach guarantees full control over the development environment and makes it easier to deploy to institutional use.

4.5.2 Local Development and Production Environment

The system is designed and will be running on XAMPP (Cross-Platform Apache, MySQL, PHP, and Perl) which gives an inbuilt local server environment. All the components needed to make the SCBS work are included in XAMPP:

- **Apache Web Server:** Handles HTTP requests, serves web pages, and processes PHP scripts. Modular nature of Apache enables it to be customized and configured to meet the needs of an institutional setting in terms of security.

- **MySQL Database:** This stores all the data in the application such as user accounts, booking history, classroom history, sensor history and administrative history. MySQL is a local service meaning that the data is under the control of the institution.

- **PHP (Version 8.0+):** Executes the Laravel backend application, processing business logic, handling authentication, and managing database interactions. PHP's integration with Apache enables seamless web application delivery.

- **phpMyAdmin:** This is a database management system that has a graphical interface that enables database administrators to see, query and manipulate data when necessary.

To be deployed in UPTM, the XAMPP environment may be located in a dedicated institutional server or a high-performance workstation which is not offline. This local hosting model means that all the data is within the network of UPTM and any privacy or data sovereignty issues are solved, and also the costs associated with cloud hosting are removed every time.

4.5.3 IoT Hardware Platform

The implemented components of the IoT are based on the ESP32 microcontroller platforms, i.e. the ESP32 and ESP32-CAM boards. The devices are connected to the local network and access the XAMPP server through HTTP requests to Larval API endpoints. The classrooms have:

- **ESP32 Microcontroller**

  The ESP32 is the heart of the system, which is a link between the physical sensors and the digital network. It uses its built-in Wi-Fi and Bluetooth to connect to a central server to verify bookings and manages all the logic of all the hardware connected. It uses the input of the camera and motion sensors to make real-time decisions, like whether to unlock a door or turn off the lights, ensuring that the classroom is run in an efficient manner without having to be operated manually.

- **PIR Motion Sensors**

  Passive Infrared (PIR) sensors detect the presence of human beings by observing the changes in the infrared light inside the room. As the bodies that emit heat such as students or teachers move within the field of the sensor, it transmits a signal to ESP32 to signify the presence in the room. This will enable the system to automate energy saving measures such that utilities such as lights and air conditioning are only turned on when a person is physically present, thereby avoiding the use of electricity in empty rooms.

- **Relay Modules**

Relay modules are electrically powered switches which enable the low-voltage ESP32 to safely operate high-voltage appliances such as air conditioners and lighting systems. The microcontroller has no ability to support the large electrical currents of building utilities directly, so the relay physically isolates the fragile digital circuitry on the main power grid with galvanic isolation. The relay is activated when the ESP32 emits a trigger, which is used to switch on the devices in the room, which is a reliable way to manage the power supply in the room automatically.

- **ESP32-CAM**

ESP32-CAM is a specialized module that has a small camera lens to serve the purpose of a digital gatekeeper of the system. It is situated at the entrance of the classroom, and it scans and decodes QR code provided by the user to confirm his identity and booking status. When the image is processed and the code is checked against the central database, the module will inform the rest of the system to allow or disallow entry instead of using keys, a secure, time-based digital access system will be used.

- **Servo Motor**

The physical actuator of the smart lock is the Servo motor that transforms the digital authorization into mechanical action. A servo can be programmed to turn to an exact angle, unlike with standard motors, such as 0 degree to lock and to 90 degrees to unlock. On a confirmed entry by the ESP32-CAM, the ESP32 will order the servo to move, literally pulling back a latch or bolt to open the door, which is a small and efficient implementation of automated physical security.

- **LED Indicators**

LED indicators are needed to provide users with necessary visual feedback, and to ensure that the status of the system can be instantly read. These lights enhance user experience by providing confirmation that the system has acknowledged their presence by different colors like green, which is used in cases of successful access, red, which is used in cases of invalid QR code, and blue, which is used in cases of system processing. This easy-to-use interface will not cause any misunderstanding at the entrance and will allow users to know whether the system is working or needs some attention.

The ESP32 devices are coded to request HTTP requests to the IP address of the local server, which keeps all the communication within the campus local area network without the need of having access to the internet.

4.5.4 Client Access

The SCBS can be accessed by using any web browser on the devices that are attached to the UPTM network. The system is entirely responsive, and it can operate using desktop computers, laptops, tablets and smart phones. The hosting of the institutional network can be limited to the institutional network by the hosting institution or by offering VPN connection to off-campus access when necessary.

4.5.5 Implementation Tools

The SCBS is created with the help of a well-chosen group of tools that will cooperate with the XAMPP local development environment. All the tools have been selected based on their local hosting compatibility and ability to suit the project needs.

4.5.6 Frontend Development Tools

- HTML5, CSS3, and JavaScript form the core frontend technologies, providing the structure, styling, and interactivity for all user interfaces. These standard web technologies ensure broad browser compatibility without requiring external dependencies.

- Bootstrap 5 will be used as the major CSS framework to speed up the development and provide consistent and professional style throughout all pages. Bootstrap offers ready-made features and responsive grid-systems that facilitate the development of mobile-friendly interfaces. The framework is locally provided either through CDN or download files, which are useful even in the absence of the internet.

- JavaScript (Vanilla and jQuery) handles client-side interactivity, form validation, and asynchronous requests to the backend. jQuery simplifies DOM manipulation and AJAX calls, though modern vanilla JavaScript is also used where appropriate.

- Figma was the offline prototyping and design tool used in the process of planning. The interface mockups were designed in the desktop version of Figma, which enabled the stakeholders to validate them prior to the start of the coding.

4.5.7 Backend Development Tools

- **Laravel PHP Framework (Version 8)**

The backend framework that will be used in the SCBS is Laravel PHP Framework (Version 8). Laravel offers a sound MVC framework that divides business logic and presentation, which makes the codebase sustainable and extendable in the XAMPP platform. Some of the important Laravel features used are:

- **Eloquent ORM**

Eloquent ORM is Laravel's built-in tool that allows you to interact with your database using PHP syntax instead of writing raw SQL. It treats every database table as a "Model," enabling you to manage complex relationships such as linking a specific student to a classroom booking with simple, readable code. This abstraction makes the development process faster and ensures that your data interactions are more secure and easier to maintain.

- **Blade Templating Engine**

The Blade Templating Engine is a powerful tool used to design the front-end views of your application using clean and expressive syntax. It allows you to create reusable layouts and components, such as a standard navigation bar or a booking button, which can be shared across multiple pages. Blade helps keep your code organized by separating the complex application logic from the visual presentation, making it easier to build responsive dashboards for students and admins.

- **Built-in Authentication**

Laravel's Built-in Authentication provides a secure foundation for managing user identity, which you have customized for Role-Based Access Control (RBAC). This system ensures that different users (students, lecturers, and admins) have specific permission. For example, a student can only view their own bookings, while an admin can manage the entire building's hardware. It handles the complexities of password hashing and session management automatically, keeping user data safe.

- **CSRF Protection**

CSRF (Cross-Site Request Forgery) Protection is a vital security feature that automatically defends your application against malicious attacks that trick users into performing unwanted actions. Laravel achieves this by generating a unique "token" for every active user session, which must be included in any request that modifies data. This ensures that every command like booking a room or opening a lock is legitimate and originated from your actual application, not a third-party site.

- **Migrations**

Migrations act as a version-control system for your database schema, allowing you to define your table structures directly in your code. Instead of manually creating tables in a database manager, you write migration files that describe the columns and data types needed for your IoT system. This makes it incredibly easy for your team to stay in sync, as anyone can recreate the exact same database structure on their local machine with a single command.

- **Seeders and Factories**

Seeders and Factories are used to automatically populate your database with dummy data during the development and testing phases. While factories define the "blueprint" for what a user or a booking looks like, seeders inject hundreds of these records into the system instantly. This allows you to test how your smart classroom dashboard handles large amounts of information without having to manually type in test entries.

- **RESTful API Resources**

RESTful API Resources allow your application to communicate with external hardware by creating standardized endpoints that your IoT devices can consume. In your project, the ESP32-CAM and other microcontrollers use these APIs to send sensor data or verify QR codes against your central server. By organizing these endpoints into a RESTful structure, you ensure that the communication between your hardware and software is fast, reliable, and easy to scale.

- **MySQL (via phpMyAdmin)**

The database runs locally within XAMPP, ensuring fast query performance and complete data control. The database management system used is MySQL (through phpMyAdmin) where all the application data is stored. The database is locally run on XAMPP, which is fast in query execution and gives full control over data.

- **Postman**

In the development phase, API testing on postman is also used to fully test the endpoints of the internet of things communication before integration into hardware.

- **IoT Development Tools**

The ESP32 and ESP32-CAM are developed and uploaded with the help of Arduino IDE. The Arduino IDE supports C / C + programming on microcontrollers and offers libraries on:

- Wi-Fi connectivity
- Communication (through HTTP) to send requests to the local server.
- The use of JSON parsing to process API responses.
- Servo motor to operate door locking devices.
- QR code capture and processing camera operations.

The Arduino IDE is configured with ESP32 board support by adding the Espressif board manager URL in the preferences. The AI Thinker ESP32-CAM board is selected for programming the camera module.

- **Development and Testing Tools**

  - Visual Studio Code is the main code editor, which has syntax highlighting, debugging, and built-in terminal access to execute Artisan commands.

  - Chrome Developer Tools are helpful in frontend debugging, testing responsive designs, and monitoring network requests.

  - XAMPP Control Panel is used to control the environment of the local server, whereby Apache and MySQL services are easily started and stopped on demand. XAMPP Control Panel is a control panel which can enable the local server environment so that the Apache and MySQL services could be easily started/stopped when required.

- **IoT Hardware Components**

The SCBS integrates various hardware components to enable automated energy control, occupancy monitoring, and QR code-based access validation. The following components are used in the IoT implementation:

Table 4.7: Iot Hardware Component

Component	Function in SCBS
ESP32-CAM Development Board	Core processing unit for QR code scanning at classroom entrance. Captures and decodes QR codes presented by users, communicates with the local server for validation, and controls door access.
ESP32 Camera Module Base	Mounting and connectivity board for the ESP32-CAM, providing easier access to programming pins and stable power connection.
PIR Sensor	Occupancy detection inside the classroom. Detects human presence to trigger automated lighting and air conditioning control, and updates real-time classroom status.
Servo Motor	Smart lock actuator for door access control. Rotates to unlock/lock the classroom door when QR code validation is successful.
LEDs (Red, Green, White)	Visual status indicators: Red indicates denied access or errors, Green indicates successful validation, White indicates system active status.
Fan (DC5V)	Indicates Air Condition in classroom.
Resistors (220Ω, 1kΩ)	Current limiting for LEDs and pull-up/pull-down resistors for sensor input stabilization.
Breadboard	Prototyping platform for circuit assembly during development and testing before permanent installation.
Jumper Cables	Connection cables for linking components to the ESP32 on the breadboard.
Relay Module	Acts as an electrically operated safety switch. Its primary function is to bridge the gap between the system's low-voltage digital brain and the classroom's high-voltage electrical appliances.

The ESP32 devices are configured to interact with the local XAMPP server by means of the RESTful API endpoints. Upon a user scanning their QR code at the classroom doorway, the ESP32-CAM will capture an image, extract the QR code data and send that data to the server to be validated. Access grant or denial can be displayed by the server and results in the proper LED indicators, sounding buzzer, and servo motor control.

As an occupancy sensor, the PIR sensor is used to continuously detect the presence of humans. On detecting motion, the ESP32 makes an HTTP POST request to the API endpoint of the server with the classroom code and the motion status. This data creates an update on the database and activates automated energy control measures, including powering on of lights and air conditioning when a room is in use and powering off after 15 minutes of inactivity.

4.5.8 System Interface

The actual system interfaces have been successfully implemented according to the designs presented in Section 4.2 (Figures 4.1 through 4.17), with all functionalities fully operational within the local XAMPP environment. During the development process, several interfaces underwent thoughtful redesign to improve usability, simplify complex workflows, and enhance the overall user experience based on practical implementation considerations and initial user feedback. The following figures present screenshots of the working system, demonstrating the successful translation of design concepts into a functional application that meets the requirements identified through stakeholder consultation and questionnaire analysis.

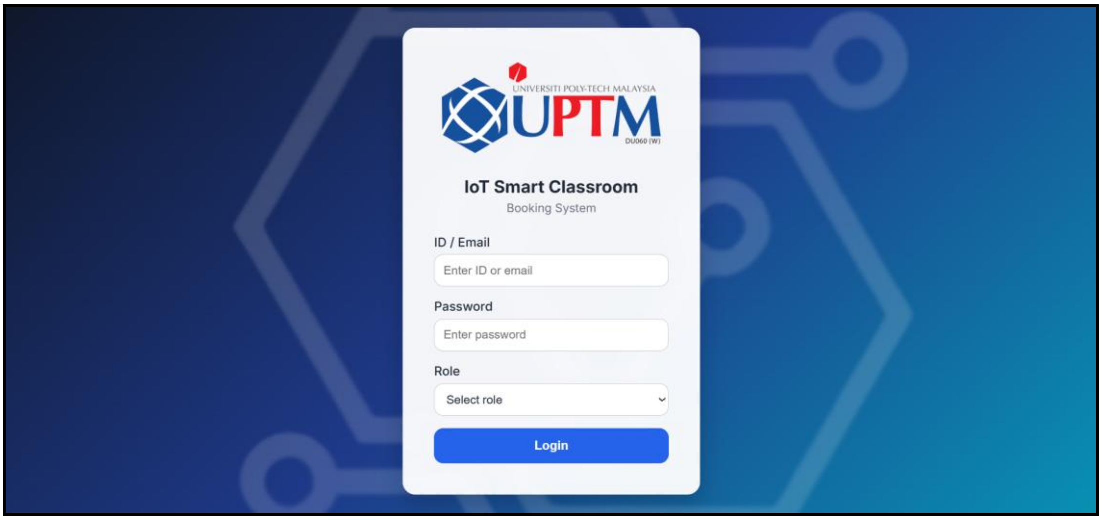


Figure 4.20: Unified Login Desktop Interface (Ain,2026)

.

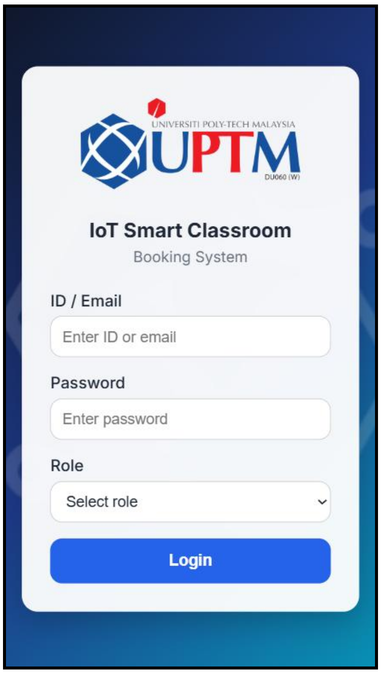


Figure 4.21: Unified Login Mobile Interface (Ain,2026)

The login interface presents a clean, centred form that unifies access for all user types in a single screen, representing a significant redesign from the planned approach of separate login pages for students, staff, and administrators. This consolidation was implemented to reduce complexity and provide users with a single memorable entry point rather than requiring different URLs based on their role. The interface displays the system branding at the top, followed by three input elements: an "ID / Email" field accepting either institutional ID or registered email, a "Password" field with masked characters for security, and a dropdown "Role" selector with options for Student, Staff, or Administrator to ensure users are directed to the appropriate dashboard after authentication. A prominent "Login" button submits credentials for validation against the local MySQL database, with the form including Laravel's CSRF protection and clear error messages for incorrect entries. The interface is fully responsive, adapting seamlessly to mobile screens as requested by the questionnaire respondents, successfully balancing security, usability, and simplicity while maintaining robust role-based access control.

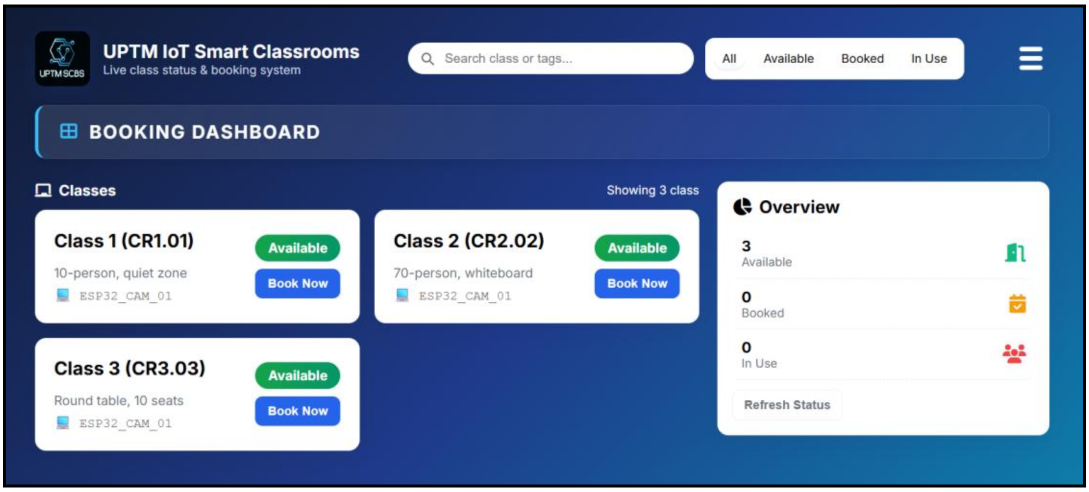


Figure 4.22: Booking Dashboard Desktop Interface (Ain,2026)

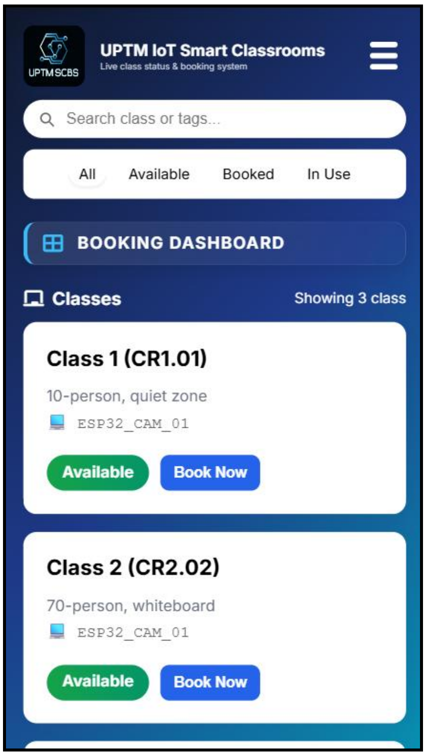


Figure 4.23: Booking Dashboard Mobile Interface (Ain,2026)

The main dashboard provides students and lecturers with a comprehensive real-time overview of classroom availability, representing a significant evolution from the planned design that originally featured a more complex layout with extensive data visualizations. Based on user feedback indicating that quick access to availability information was the primary need, the actual implementation focuses on clarity and immediacy by organizing classroom listings into distinct status-based sections. Each classroom appears in a clean card format displaying room code, capacity, facilities, and associated IoT devices, with available rooms highlighted separately for immediate booking opportunities. The right sidebar maintains the status overview showing "4 Available," "0 Booked," and "0 In Use" counts with a "Refresh Status" button, that works alongside the automatic 30-second AJAX refresh. This implementation successfully balances completeness with efficiency, addressing user needs for a straightforward booking process. The interface is fully responsive, adapting to mobile screens by stacking cards vertically and adjusting the status summary for smaller displays.

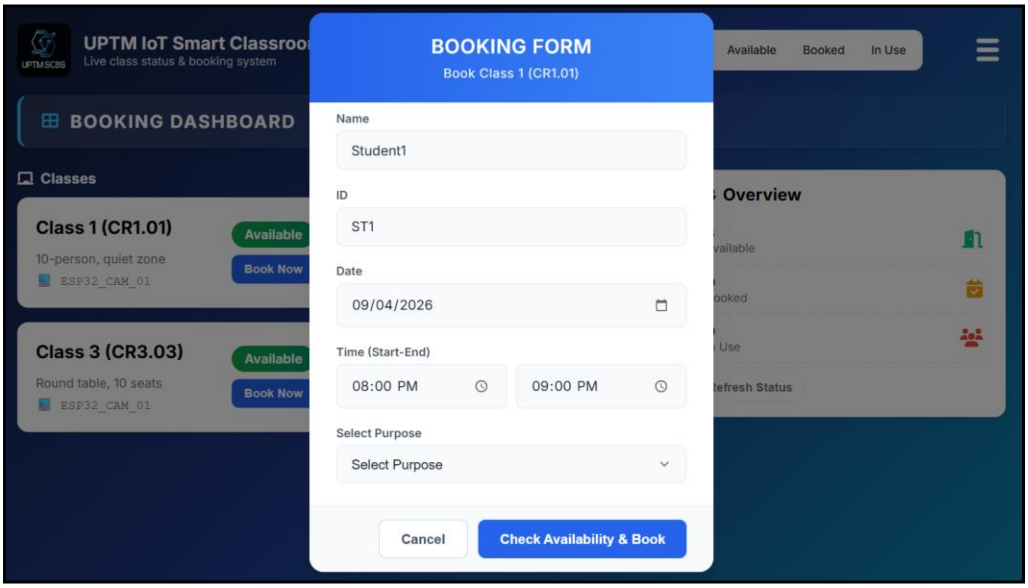


Figure 4.24: Booking Form Desktop Interface (Ain,2026)

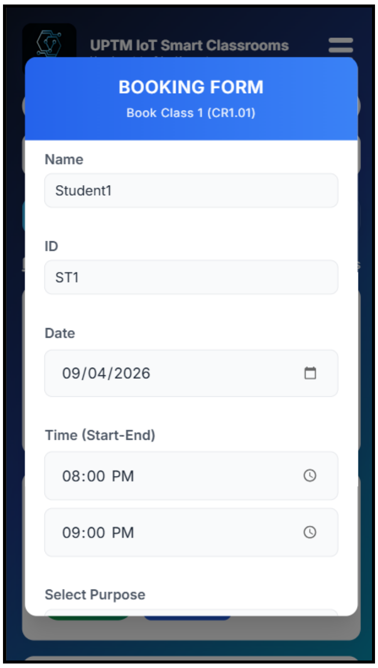


Figure 4.25: Booking Form Mobile Interface (Ain,2026)

The booking form interface appears when a user clicks the "Book Now" button from the main dashboard, specifically showing a pre-filled form for Class 1 (CR1.01). This implementation represents a streamlined approach compared to the planned design shown in Figure 4.10, which originally included a more complex multi-step wizard. Based on user feedback indicating preference for completing bookings quickly without multiple page loads, the actual implementation consolidates all necessary fields into a single, intuitive form. The interface maintains the main dashboard interface with a clear overlay "BOOKING FORM" platform identifying "Book Class 1 (CR1.01)" to eliminate confusion. Five essential fields are presented vertically: "Name" and "ID" fields automatically pre-fill with the logged-in user's information to reduce typing errors, while "Date," "Time (Start-End)," and a "Select Purpose" dropdown allow users to specify their booking details. Two action buttons provide clear choices: "Cancel" returns to the dashboard, and "Check Availability & Book" validates the selected time slot against existing bookings before submission.

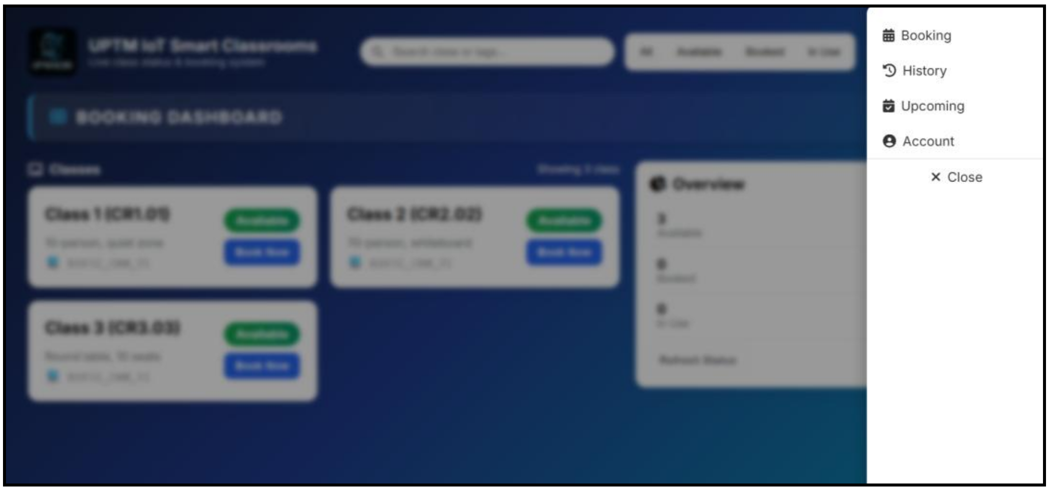


Figure 4.26: Menu Section Desktop Interface (Ain,2026)

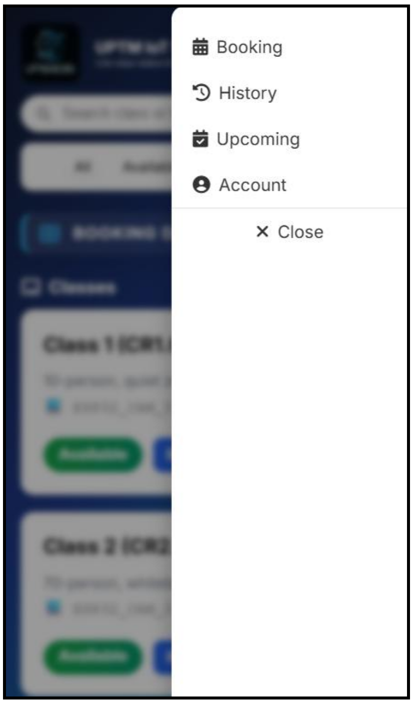


Figure 4.27: Menu Section Mobile Interface (Ain,2026)

The navigation menu appears as a sidebar on the left side of the dashboard interface, providing users with consistent access to all system functions throughout their session. This implementation represents a redesign from the planned top navigation bar with dropdowns, which was changed after testing revealed that users preferred having all navigation options visible at once without needing to hover or click through dropdown menus. The menu presents clear options including "Booking," "History," “Upcoming”, and “Account” that would appear when triggered. The menu remains fixed as users scroll through classroom listings, ensuring navigation is always accessible. This sidebar approach simplifies the user experience by providing immediate visibility of all available sections, reducing the cognitive load of remembering where specific functions are located.

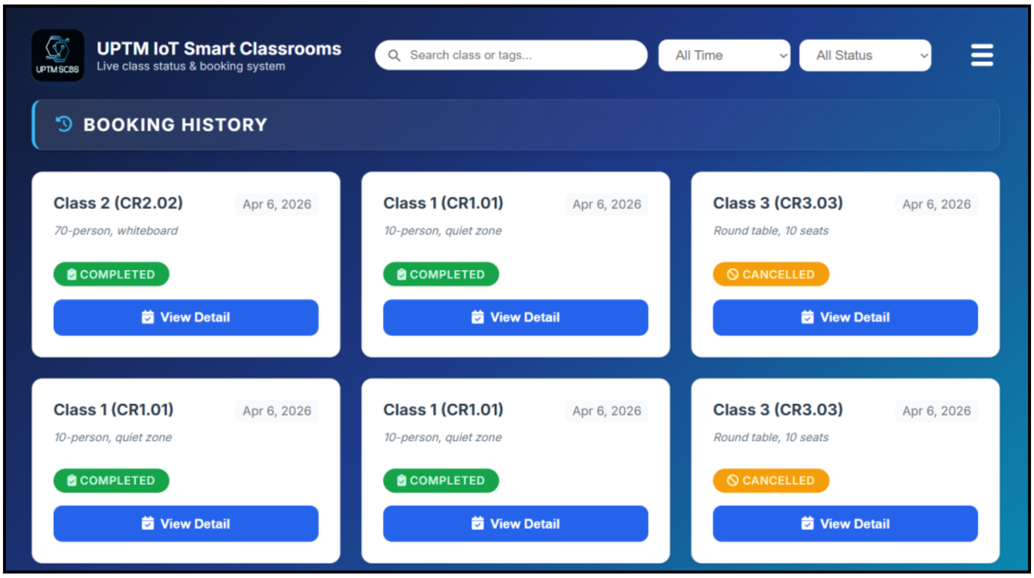


Figure 4.28: History Desktop Interface (Ain,2026)

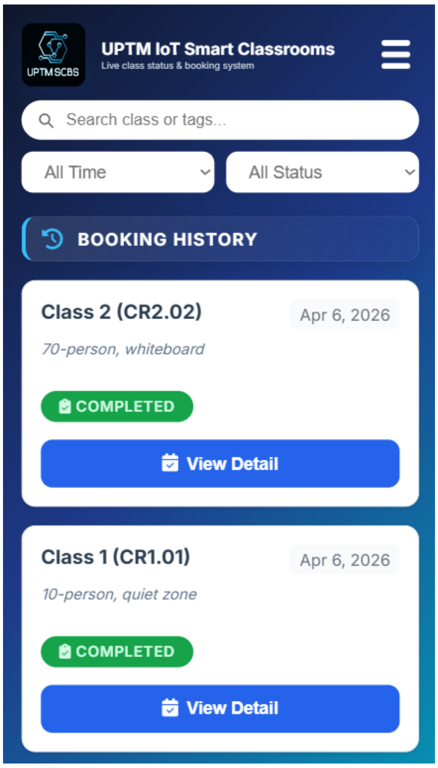


Figure 4.29: History Mobile Interface (Ain,2026)

The history interface displays all past bookings for the logged-in user in a chronological list format, representing a refinement from the planned table design which was changed after testing revealed that users preferred seeing more contextual information in a card-based layout. Each history entry appears as a distinct card showing the classroom name and code, room details such as "Round table, 6 seats" or "10-person, quiet zone," and a prominent status indicator highlighting outcomes like “COMPLETED”, “REJECTED”, “CANCELLED”, “NO SHOW” in bold text and different colours for immediate recognition. A "View Detail" button on each card allows users to access complete booking information including dates, times, and booking status. This card-based approach makes it easier for users to scan their booking history and quickly identify the status of each past reservation.

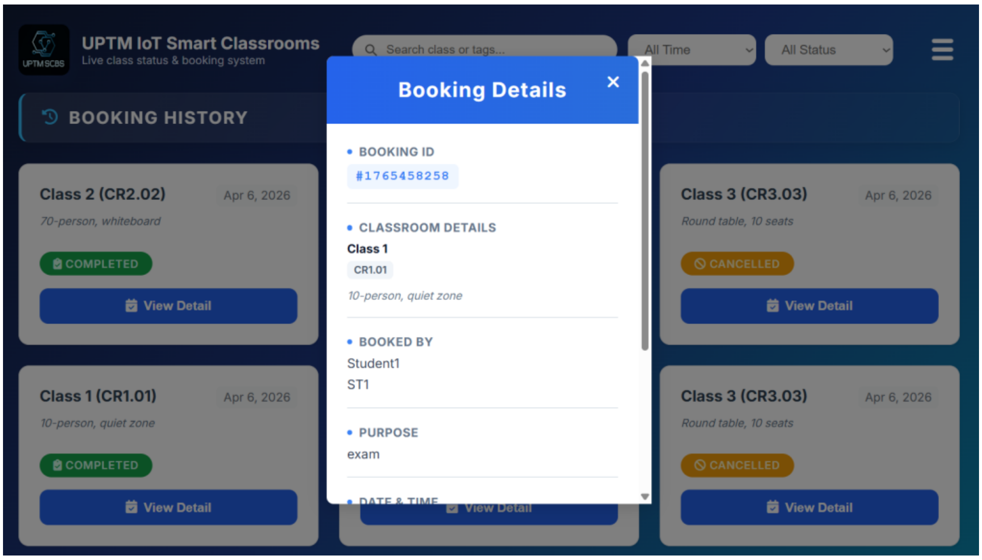


Figure 4.30: Booking Details of History Desktop Interface (Ain,2026)

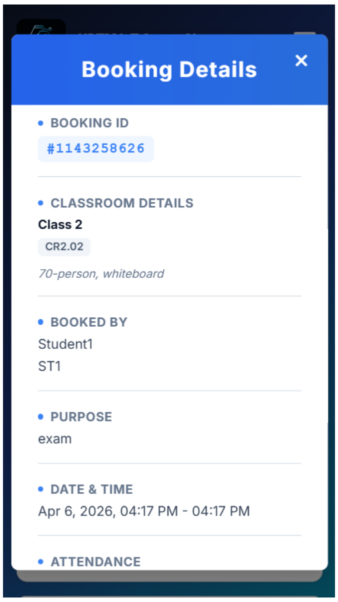


Figure 4.31: Booking Details of History Mobile Interface (Ain,2026)

The booking detail page displays comprehensive information when a user clicks "View Detail" from the history interface, presenting all relevant booking data in a structured, easy-to-read format. The page prominently shows the Booking ID, such as "#1388998668," and classroom identifier "Class 3 (CR3.03)" at the top for immediate reference. Information is organized into clearly labeled sections including "CLASSROOM DETAILS" showing name, code, and description like "Round table, 6 seats"; "BOOKED BY" displaying the user's name and ID; "BOOKING PURPOSE" indicating the reason such as "exam"; "BOOKING PERIOD" showing the specific date and time slot; and "ATTENDANCE" tracking check-in and check-out status. Most importantly, the overall "STATUS" is displayed prominently as "CANCELLED" (or other statuses like confirmed, completed, or no-show), providing immediate clarity on the booking outcome. This comprehensive view ensures users have access to all details about past and upcoming bookings for reference and planning purposes.

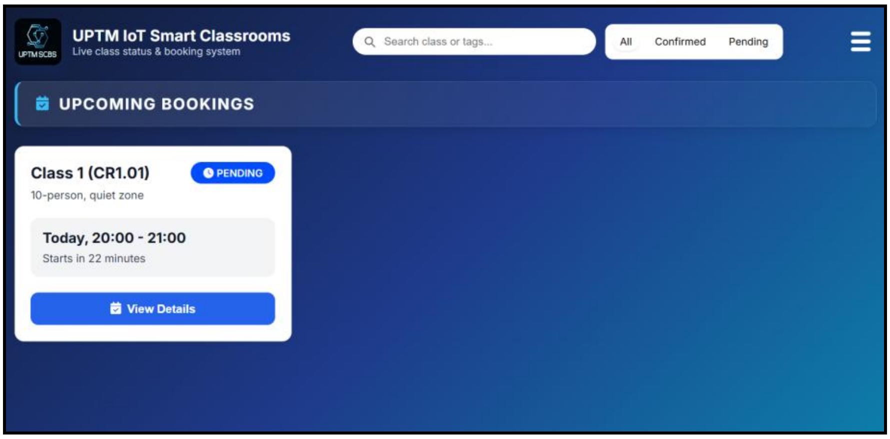


Figure 4.32: Upcoming Desktop Interface (Ain,2026)

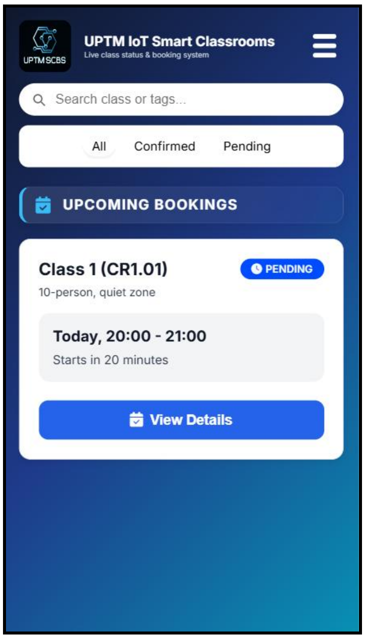


Figure 4.33: Upcoming Mobile Interface (Ain,2026)

The upcoming bookings page displays all confirmed future reservations for the logged-in user in a clean, countdown-focused format. Each booking card shows the classroom name and code, room details such as "10-person, quiet zone" or "70-person, whiteboard," followed by the scheduled date and time slot like "Tomorrow, 12:00 - 13:00" or "Tuesday 3 Mar, 15:00 - 16:00." A dynamic countdown timer displays time remaining until the booking starts, such as "Starts in 21 hours 3 minutes" or "Starts in 48 hours 3 minutes," providing users with immediate awareness of upcoming commitments. A "View Details" button on each card allows access to complete booking information including the QR code for classroom access. This implementation helps users track their scheduled sessions and prepare accordingly, addressing the need for easy access to upcoming reservation information.

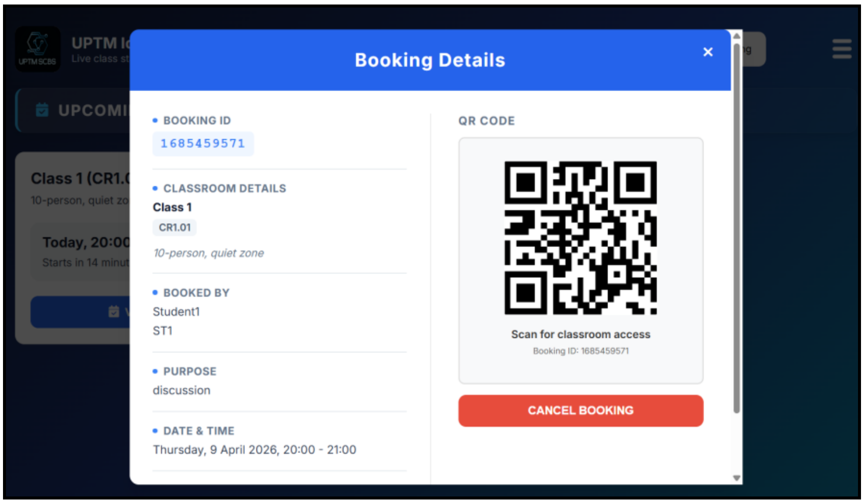


Figure 4.34: Confirmed Booking Detail Platform of Upcoming Interface (Ain,2026)

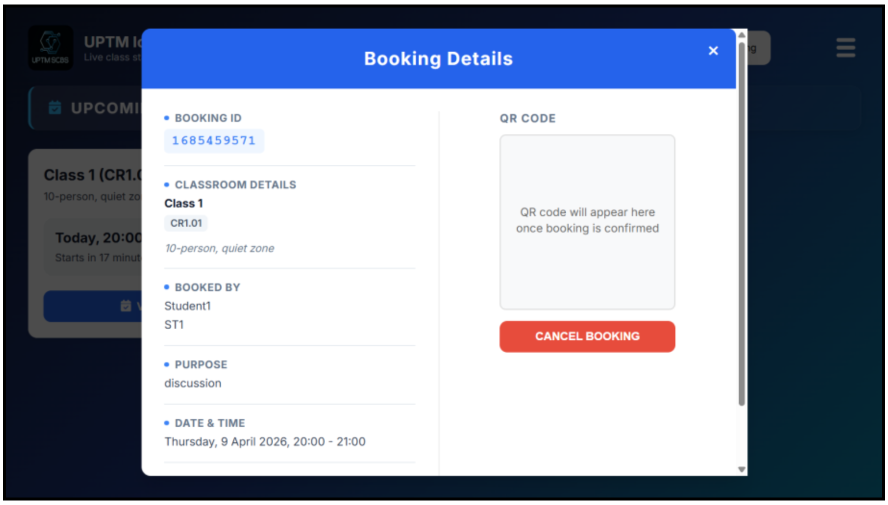


Figure 4.35: Pending Booking Detail Platform of Upcoming Interface (Ain,2026)

The upcoming booking detail page displays comprehensive information when a user clicks "View Details" from the upcoming bookings interface, with the content dynamically adapting based on the booking status. For confirmed bookings, the page shows all reservation details including classroom "Class 1 CR1.01" with its description, booker information, purpose, scheduled date and time, and a prominent "CONFIRMED" status indicator. Most importantly, a generated QR code is displayed prominently with instructions and the Booking ID for reference, providing the physical access credential needed for entry. For pending bookings, the page shows similar reservation details including classroom "Class 2 CR2.02," purpose, and scheduled time but with a "PENDING" status indicator and a placeholder message stating "QR code will appear here once booking is confirmed." This dynamic display appropriately manages user expectations by only providing access credentials for approved bookings while still allowing users to view and track their pending requests.

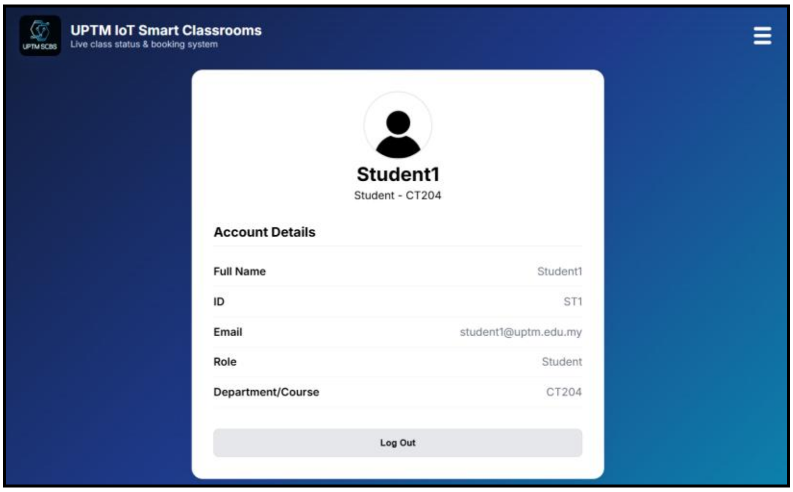


Figure 4.36: Account Desktop Interface (Ain,2026)

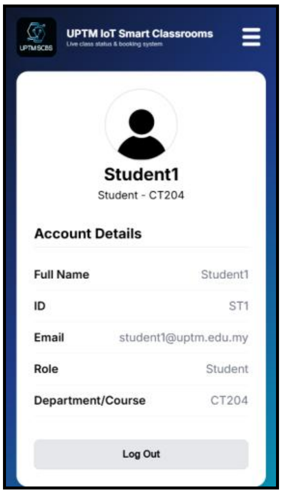


Figure 4.37: User Account Mobile Interface (Ain,2026)

The account interface displays the logged-in user's profile information in a clean, organized table format. The page header shows the user's name and role with course information "Student - CT204" for immediate context. Account details are presented in a two-column table with Field and Description columns, showing essential information including Full Name, ID, Email, Role, and Department/Course. This structured presentation makes it easy for users to verify their registered information briefly. A prominent "Log Out" button at the bottom allows users to securely end their session. The interface focuses on read-only display of information since core identity details are managed by the institution's central records, with email being the only editable field available through a separate edit function. This implementation addresses user needs for profile visibility while maintaining data integrity.

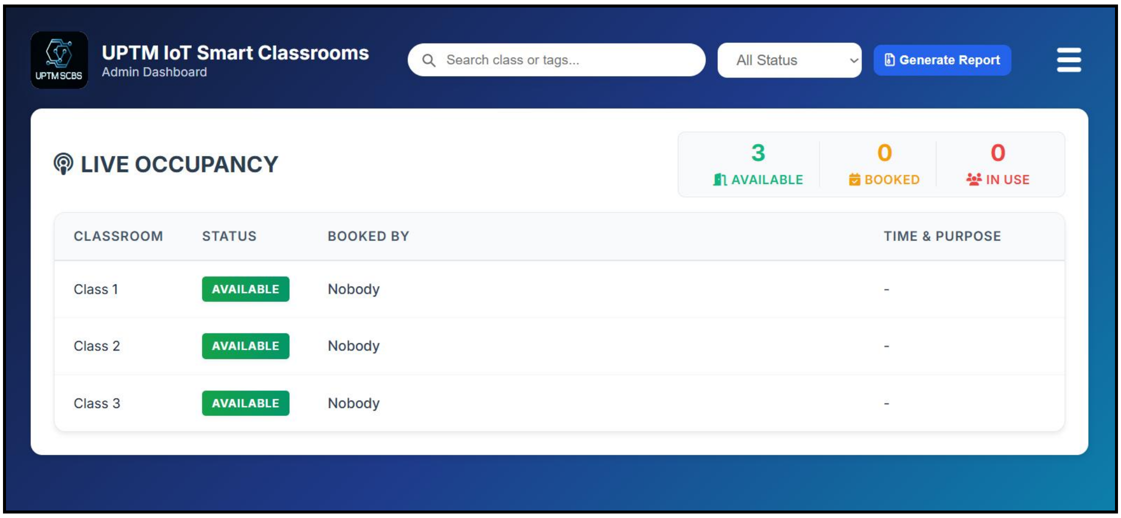


Figure 4.38: Admin Live Occupancy Desktop Interface (Ain,2026)

The admin dashboard presents a comprehensive live occupancy view designed for desktop use, focusing on real-time monitoring of all classrooms. The header displays system logo and "UPTM IoT Smart Classrooms" with the "Admin Dashboard" subtitle for clear context. The main feature is a detailed table showing "LIVE OCCUPANCY" with columns for CLASSROOM, STATUS, and BOOKED BY, listing all rooms such as Class 1, Class 2, Class 3, and Test, each currently showing "AVAILABLE" status with no occupant based on the IoT integration. Above the table, search and filter tools include a "Search class or tags..." field and an "All Status" dropdown for quick filtering, alongside a "Generate Report" button for exporting occupancy data. Status summary buttons display "4 AVAILABLE," "0 BOOKED," and "0 IN USE" for at-a-glance awareness. A time slider allows administrators to view historical occupancy or future bookings by adjusting the timeline. This implementation provides facility managers with powerful monitoring tools to track classroom usage, identify patterns, and make data-driven decisions about resource allocation.

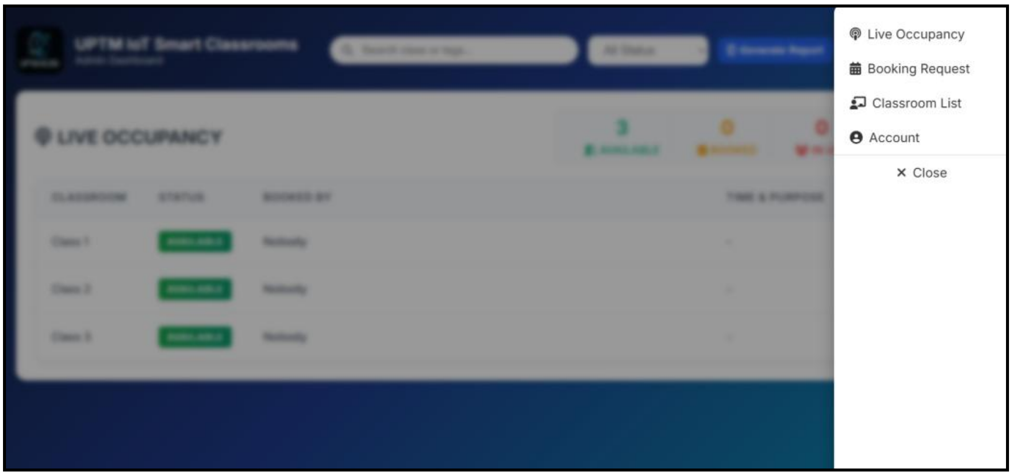


Figure 4.39: Admin Menu Section Interface (Ain,2026)

The admin navigation menu appears as a sidebar providing access to all administrative functions, with "Live Occupancy" highlighted as the current active section. The menu presents clear options including "Live Occupancy" for real-time classroom monitoring, "Booking Request" for managing pending reservations, "Classroom List" for editing room details, and "Account" for profile management. A "Close" button at the bottom allows administrators to collapse or log out from the menu. This implementation groups related administrative functions together for efficient workflow, with the active section visually indicated to help administrators maintain orientation while navigating between different management tasks.

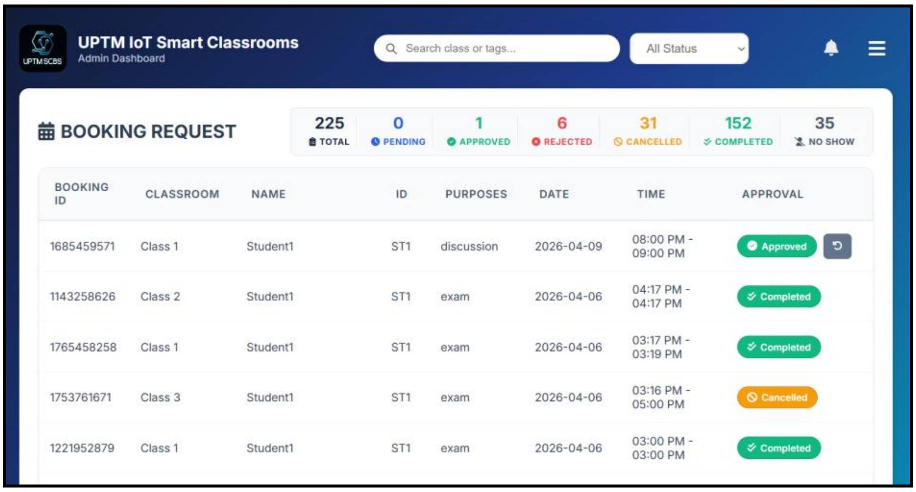


Figure 4.40: Admin Booking Request Interface (Ain,2026)

The booking request page displays all reservation requests in a comprehensive table format for efficient administrative review. The interface presents a detailed table with columns for ID, CLASSROOM, NAME, ID, PURPOSES, DATE, TIME, and APPROVAL, showing each request's complete information briefly. The approval column contains action buttons, with pending requests showing "Approve" and "Reject" options, while processed requests display outcome labels such as "Approved," "Cancelled," "Rejected," or "No Show" in different colours. For example, request ID 1851551338 shows pending approval for Student1 booking Class 2, while ID 1895698131 already shows "Cancelled" status. This implementation allows administrators to efficiently review, approve, or reject bookings while maintaining a complete history of all requests and their outcomes for accountability and reference.

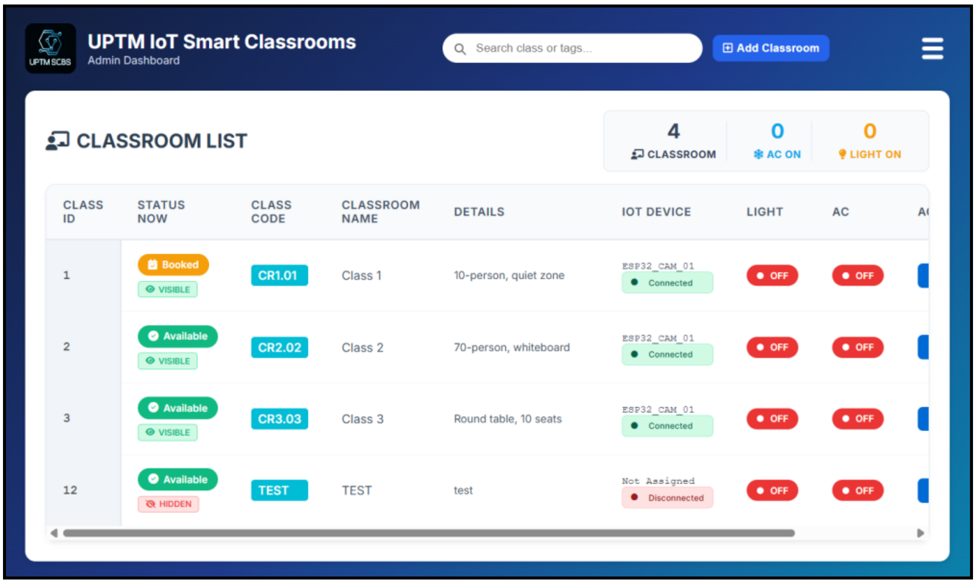


Figure 4.41: Admin Classroom List Interface (Ain,2026)

The classroom list page displays all classrooms in a comprehensive table format for efficient management. The table shows essential information including STATUS NOW with color-coded indicators, CLASS CODE, CLASSROOM NAME, DETAILS with room specifications, IOT DEVICE showing assigned sensors and connection status (Connected/Disconnected), and real-time LIGHT and AC switch (ON/OFF). The class status is also labelled with visible or hidden indicates the class is in ready condition or not. Each row includes Edit and Delete action buttons for classroom management. Above the table, a "Search class or tags..." field allows filtering, while an "Add Classroom" button enables quick addition of new rooms. Status summary buttons display "4 CLASSROOM," "0 AC ON," and "0 LIGHT ON" for at-a-glance awareness of current resource usage. This implementation provides administrators with complete visibility and control over classroom configurations and real-time device status.

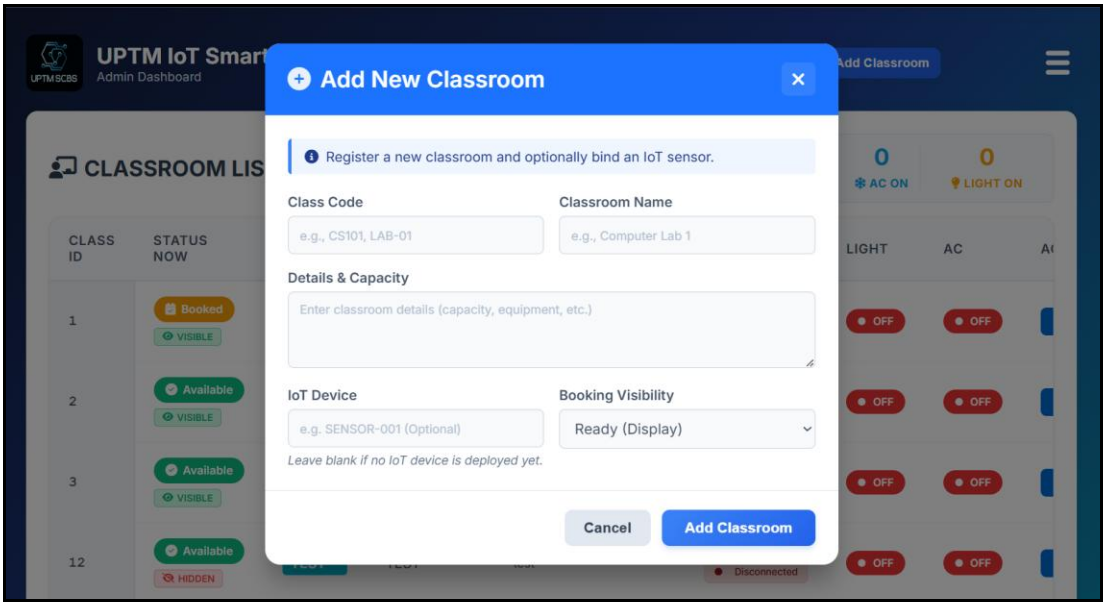


Figure 4.42: Admin Add Classroom Platform (Ain,2026)

The add classroom form provides a clean, focused interface for adding new rooms to the system. The form is organized with clear labels and helpful placeholder text guiding administrators through each field. "Class Code" requires a unique identifier such as "CS101" or "LAB-01" with instructional text explaining its purpose. "Classroom Name" captures the display name like "Computer Lab 1." "Details" allows entry of room specifications including capacity and equipment. An optional "IoT Device ID" field lets administrators associate sensors with the classroom, with guidance to leave empty if no device is assigned. The admin can also choose the class visibility between ready or not ready. This will help in case the class need a maintenance and must be temporary unavailable from book. Two action buttons at the bottom provide "Cancel" to return to the classroom list or "Add Classroom" to save the new room. This implementation simplifies the classroom creation process while ensuring all necessary information is captured consistently.

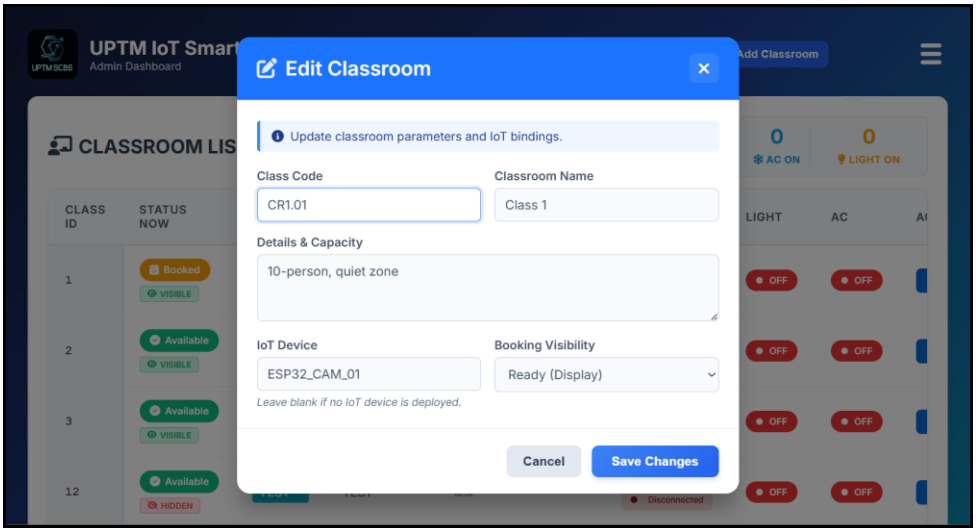


Figure 4.43: Admin Edit Classroom Platform (Ain,2026)

The edit classroom form pre-populates with existing classroom data, allowing administrators to modify information efficiently. The form displays all fields with their current values, including Class Code showing "CR1.01," Classroom Name showing "Class 1," Details showing "10-person, quiet zone," and IoT Device ID showing "Sensor1, Scanner1" for easy reference and modification. Two action buttons at the bottom provide "Cancel" to return to the classroom list without saving or "Save Changes" to update the database with any modifications. This implementation streamlines the updating process by showing existing information directly in the input fields, eliminating the need to re-enter data and reducing the potential for errors when making changes.

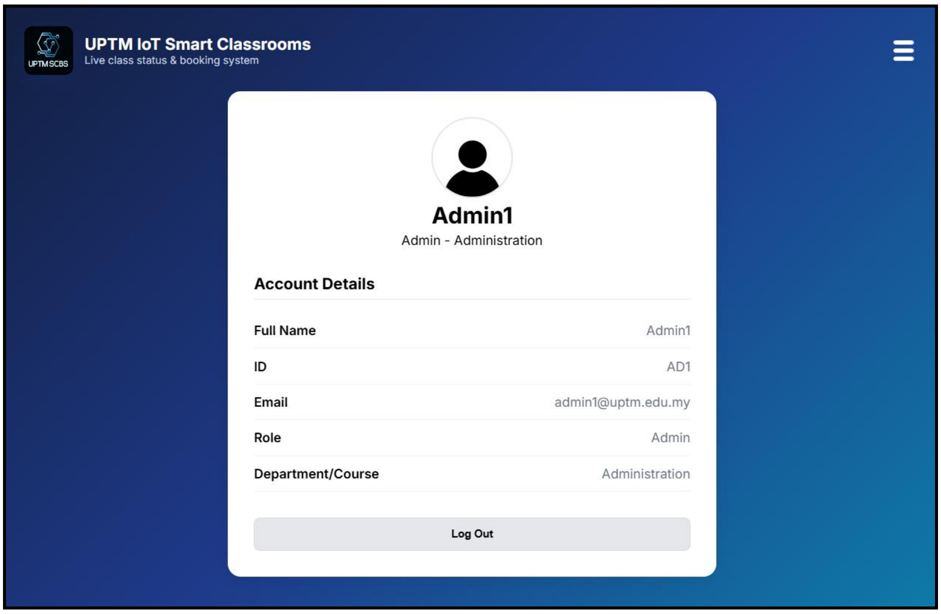


Figure 4.44: Admin Account Interface (Ain,2026)

The admin account interface displays the logged-in administrator's profile information in a clean, organized format. The page header shows "Admin1" with role designation "Admin - Administration" for immediate context. Account details are presented showing Full Name "Admin1," ID "AD1," Email "admin1@uptm.edu.my," Role "Admin," and Department/Course "Administration" in a clear, readable layout. A prominent "Log Out" button at the bottom allows administrators to securely end their session. This implementation provides administrators with quick access to their profile information while maintaining the same consistent design language used throughout the system.

4.5.9 Significant Functions

The Smart Classroom Booking System (SCBS) incorporates several significant functions that are implemented entirely within the local XAMPP environment, addressing the core objectives identified in Chapter 1 and validated through the comprehensive questionnaire analysis conducted with 50 UPTM community members. These functions work in harmony to create a seamless, secure, and efficient classroom management ecosystem that transforms the manual paper-based system into an automated intelligent platform.

4.5.9.1 Significant Functions Role-Based Access Control (Security Element)

```
// C:\xampp\htdocs\CODING\SCBS\login.js
document.addEventListener('DOMContentLoaded', function() {
    const loginForm = document.getElementById('loginForm');
    const loginBtn = document.getElementById('loginBtn');
    const messageDiv = document.getElementById('message');

    if (!loginForm) {
        console.error('Login form not found!');
        return;
    }

    loginForm.addEventListener('submit', async function(e) {
        e.preventDefault();

        // Get form values
        const username = document.getElementById('username').value.trim();
        const password = document.getElementById('password').value;
        const role = document.getElementById('role').value;

        // Simple validation
        if (!username || !password || !role) {
            showMessage('Please fill in all fields', 'error');
            return;
        }

        // Disable button and show loading
        loginBtn.disabled = true;
        const originalText = loginBtn.innerHTML;
        loginBtn.innerHTML = 'Logging in...';

        try {
            // Create FormData
            const formData = new FormData();
```

Figure 4.45: Login Role Based Access Control Part A (Ain,2026)

```
try {
    // Create FormData
    const formData = new FormData();
    formData.append('username', username);
    formData.append('password', password);
    formData.append('role', role);

    // Send to PHP
    const response = await fetch('login.php', {
        method: 'POST',
        body: formData
    });

    // Parse response
    const data = await response.json();

    if (data.success) {
        showMessage(data.message, 'success');

        // Store user data in localStorage
        localStorage.setItem('currentUser', JSON.stringify(data.user));

        // Redirect after 1 second
        setTimeout(() => {
            if (data.user.role === 'admin') {
                window.location.href = 'adminDashboard.html';
            } else {
                window.location.href = 'bookDashboard.html';
            }
        }, 1000);
    } else {
```

Figure 4.46: Login Role Based Access Control Part B (Ain,2026)

The SCBS implements comprehensive role-based access control (RBAC) as a fundamental security element that operates entirely within the local environment, ensuring that all user data and system functions remain protected within UPTM's network infrastructure. Three distinct user roles are defined with clearly separated privileges to maintain system integrity and appropriate access levels:

- **Students** are identified by their matrix numbers and authenticated against the local user's table. They are granted the ability to view classroom availability in real-time, make bookings for their academic activities such as group discussions, project meetings, and study sessions, manage their own reservations including modifications and cancellations without requiring administrative intervention, and access their complete booking history for reference and planning purposes. This self-service approach empowers students while significantly reducing the administrative burden on staff who previously handled all booking requests manually.
- **Lecturers or staff members** have identical privileges to students in terms of booking functionality for their lectures, tutorials, and academic meetings, but they are identified separately in the database using their staff ID credentials. This separation allows for future privilege differentiation should institutional policies

require expanded access for academic staff, such as priority booking during certain periods or access to departmental scheduling features that might not be appropriate for student users.

- **Administrators** possess full system access with elevated privileges that include approving or rejecting pending booking requests to ensure appropriate use of facilities, managing all classroom data including adding new rooms, editing existing room details, or removing decommissioned classrooms from the system, viewing all reservations across the institution regardless of who made them for comprehensive oversight, generating detailed reports on usage patterns, utilization rates, and energy consumption for informed decision-making, and monitoring all IoT devices and sensor data for troubleshooting and optimization purposes to ensure the hardware infrastructure remains reliable and effective.

The RBAC implementation leverages Laravel's built-in authentication and authorization features, specifically using middleware that checks user roles before allowing access to protected routes. This means that before any controller method executes, the system verifies that the authenticated user has the appropriate role to access that specific functionality, ensuring that unauthorized users cannot access restricted functions even if they manually navigate to protected URLs. This security element directly addresses the privacy requirements identified in the questionnaire, where an overwhelming 94% of respondents emphasized the critical need for secure authentication and data protection. All user data, including personal information, booking records, and access logs, remains securely stored within the local MySQL database, never leaving the UPTM network and thus maintaining complete institutional control over sensitive information.

4.5.9.2 Automated QR Code Generation and Validation

```php
try {
    if (mysqli_stmt_execute($stmt)) {

        // Generate QR Code if approved
        if ($action === 'approve') {
            // Data to encode: Simple verifiable string
            // Store raw QR data string so ESP32 scanner matches DB exactly
            $qr_data = "SCBS-BID-" . $booking_id . "-UID-" . $_SESSION['user_id'] . "-TIME-" . time();

            // Note: URL generation happens dynamically on frontend/viewing side now
            // $qr_url = "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=" . urlencode($qr_data);

            // Use a separate connection or just simple update query
            $qr_sql = "UPDATE booking SET qr_code = ? WHERE id_booking = ?";
            $qr_stmt = mysqli_prepare($conn, $qr_sql);
            if ($qr_stmt) {
                mysqli_stmt_bind_param($qr_stmt, "si", $qr_data, $booking_id);
                mysqli_stmt_execute($qr_stmt);
                mysqli_stmt_close($qr_stmt);
            }
        }

        // Clear any previous output (warnings, etc)
        ob_end_clean();

        echo json_encode([
            'success' => true,
            'message' => "Booking updated to $new_status",
            'id' => $booking_id,
            'status' => $action == 'approve' ? 'approved' : ($action == 'pending' ? 'pending' : 'rejected')
        ]);
    } else {
        throw new Exception("Execute failed: " . mysqli_stmt_error($stmt));
```

Figure 4.47: QR Backend Logic (Ain,2026)

```javascript
// 1. QR Code Logic
if (qrCodeContainer) {
    if (activeStatuses.includes(normalizedStatus)) {
        const qrData = booking.qr_code || `SCBS-BID-${booking.id_booking}`;
        const qrUrl = qrData.startsWith('http') ? qrData : `https://api.qrserver.com/v1/create-qr-code/?size=250x250&c
        qrCodeContainer.innerHTML = `
            <div style="text-align: center; padding: 50px;">
                <div style="padding: 15px; display: inline-block;">
                    <img src="${qrUrl}" alt="Booking QR Code" style="width: 200px; height: 200px; display: block; marg
                    <p style="color: #333; font-size: 14px; margin-top: 15px; font-weight: 600;">Scan for classroom
                    <p style="color: #666; font-size: 11px;">Booking ID: ${booking.id_booking}</p>
                </div>
            </div>
        `;
    } else {
        qrCodeContainer.innerHTML = `
            <p style="color: #666; text-align: center; padding: 20px;">
                QR code will appear here<br>
                once booking is confirmed
            </p>
        `;
    }
}
```

Figure 4.48: QR code image generated using an external API (Ain,2026)

A significant and innovative function of the SCBS is the generation and validation of unique, cryptographically secure QR codes for each confirmed booking, all processed locally without dependency on external services or cloud-based QR generators that could introduce security vulnerabilities or reliability issues. When an administrator approves a pending booking request, the system automatically initiates a multi-step process that transforms the digital approval into a physical access credential:

- **Unique token generation** begins with creating a cryptographically secure string that combines the booking ID, user ID, classroom ID, current timestamp, and a random salt using PHP's hash() function with the SHA-256 algorithm. This ensures that each QR code is mathematically unique and practically impossible to duplicate or predict through brute force attacks, as even two identical bookings made milliseconds apart will generate completely different tokens due to the timestamp and random salt components.

- **QR code image creation** follows using a PHP QR code library that runs entirely on the local server without external API calls. The library generates a standardized QR code in PNG format from the token string, with appropriate sizing and error correction levels to ensure reliable scanning even under less-than-ideal conditions such as poor lighting, slightly damaged codes, or scanning at slight angles.

- **Token storage** saves the generated token in the qr_code field of the booking table within the local database for later validation during access attempts, creating a permanent record linking the physical QR code to the digital booking record that can be referenced whenever the code is presented, even months later for audit purposes.

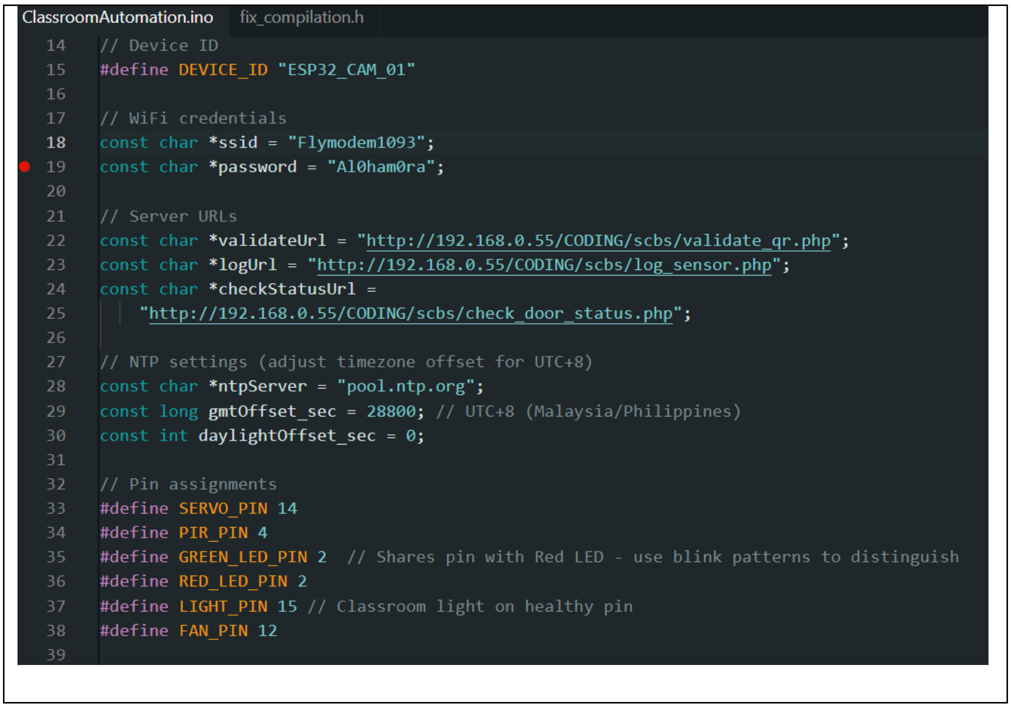
- **Interface display** makes the QR code available for viewing and download on the booking detail page within the web application, allowing users who prefer to access it through the system or need to display it on their mobile devices at the classroom entrance to do so easily.

At the classroom entrance, an ESP32-CAM device scans the presented QR code and sends an HTTP POST request to the local API endpoint. The validation endpoint queries the local database to find a matching booking, verifies the booking is confirmed, checks the current time against the booking window (with 15-minute grace period), and returns a JSON response indicating access grant or denial. All validation occurs locally without internet dependency, ensuring reliable operation even if external connectivity is unavailable.

4.5.9.3 IoT-Based Automated Energy Control

```php
<?php
require_once 'config.php';

// Get QR code from ESP32
$qrCode = isset($_POST['qr_code']) ? trim($_POST['qr_code']) : '';
$deviceId = isset($_POST['device_id']) ? trim($_POST['device_id']) : '';

if (empty($qrCode)) {
    echo json_encode(['success' => false, 'message' => 'No QR code provided']);
    exit;
}

// Get classroom ID from device
$classSql = "SELECT id_class FROM classroom WHERE iot_device = ?";
$classStmt = $conn->prepare($classSql);
$classStmt->bind_param("s", $deviceId);
$classStmt->execute();
$classResult = $classStmt->get_result();
$classroom = $classResult->fetch_assoc();

if (!$classroom) {
    // Log for debugging
    error_log("[" . date("Y-m-d H:i:s") . "] Validation failed: Device '$deviceId' not found in classroom table.\n", 3,
    echo json_encode(['success' => false, 'message' => 'Device not registered']);
    exit;
}

// Check booking with QR code — no classroom filter since one ESP covers all rooms (prototype)
$sql = "SELECT b.*, u.name_user, u.role
        FROM booking b
        JOIN users u ON b.id_user = u.id_user
        WHERE b.qr_code = ?
        AND b.status_booking = 'confirm'
```

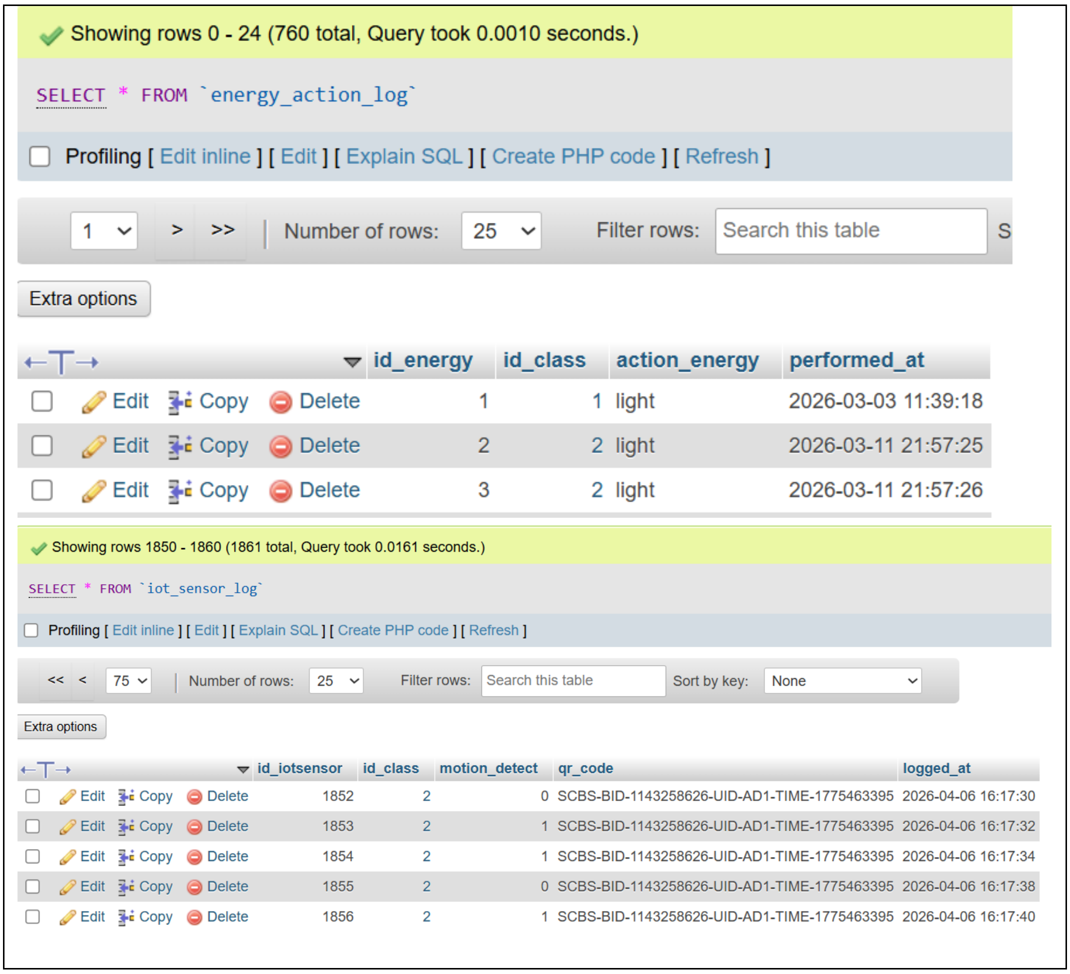
```php
if ($result->num_rows > 0) {
    $logQrStmt->bind_param("is", $booking['id_class'], $qrCode);
    $logQrStmt->execute();

    // Get current device statuses to sync with ESP32 immediately
    $statusSql = "SELECT light_status, ac_status FROM classroom WHERE id_class = ?";
    $statusStmt = $conn->prepare($statusSql);
    $statusStmt->bind_param("i", $booking['id_class']);
    $statusStmt->execute();
    $statusResult = $statusStmt->get_result();
    $statusRow = $statusResult->fetch_assoc();

    echo json_encode([
        'success'       => true,
        'message'       => $message,
        'user_id'       => $booking['id_user'],
        'user_name'     => $booking['name_user'],
        'user_role'     => $booking['role'],
        'action'        => $action,
        'id_class'      => $booking['id_class'],
        'end_time_unix' => strtotime($booking['end_time']), // Unix timestamp for ESP32 timer
        'light_status'  => intval($statusRow['light_status']),
        'ac_status'     => intval($statusRow['ac_status']),
        'open_lock'     => true
    ]);
} else {
    // Log failures to help track down why a scan was rejected
    error_log("[" . date("Y-m-d H:i:s") . "] Validation failed: No active booking found for QR '$qrCode'.\n", 3, "debug
    echo json_encode([
        'success' => false,
        'message' => 'No valid booking found'
    ]);
```



ClassroomAutomation.ino fix_compilation.h

```
// Device ID
#define DEVICE_ID "ESP32_CAM_01"

// WiFi credentials
const char *ssid = "Flymodem1093";
const char *password = "Al0ham0ra";

// Server URLs
const char *validateUrl = "http://192.168.0.55/CODING/scbs/validate_qr.php";
const char *logUrl = "http://192.168.0.55/CODING/scbs/log_sensor.php";
const char *checkStatusUrl =
    "http://192.168.0.55/CODING/scbs/check_door_status.php";

// NTP settings (adjust timezone offset for UTC+8)
const char *ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 28800; // UTC+8 (Malaysia/Philippines)
const int daylightOffset_sec = 0;

// Pin assignments
#define SERVO_PIN 14
#define PIR_PIN 4
#define GREEN_LED_PIN 2  // Shares pin with Red LED - use blink patterns to distinguish
#define RED_LED_PIN 2
#define LIGHT_PIN 15 // Classroom light on healthy pin
#define FAN_PIN 12
```

Figure 4.49: Interaction between System and IoT Devices (Ain,2026)



Showing rows 0 - 24 (760 total, Query took 0.0010 seconds.)

SELECT * FROM `energy_action_log`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Number of rows: 25 Filter rows: Search this table

Extra options

	id_energy	id_class	action_energy	performed_at
Edit Copy Delete	1	1	light	2026-03-03 11:39:18
Edit Copy Delete	2	2	light	2026-03-11 21:57:25
Edit Copy Delete	3	2	light	2026-03-11 21:57:26

Showing rows 1850 - 1860 (1861 total, Query took 0.0161 seconds.)

SELECT * FROM `iot_sensor_log`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

<< < 75 | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	id_iotsensor	id_class	motion_detect	qr_code	logged_at
Edit Copy Delete	1852	2	0	SCBS-BID-1143258626-UID-AD1-TIME-1775463395	2026-04-06 16:17:30
Edit Copy Delete	1853	2	1	SCBS-BID-1143258626-UID-AD1-TIME-1775463395	2026-04-06 16:17:32
Edit Copy Delete	1854	2	1	SCBS-BID-1143258626-UID-AD1-TIME-1775463395	2026-04-06 16:17:34
Edit Copy Delete	1855	2	0	SCBS-BID-1143258626-UID-AD1-TIME-1775463395	2026-04-06 16:17:38
Edit Copy Delete	1856	2	1	SCBS-BID-1143258626-UID-AD1-TIME-1775463395	2026-04-06 16:17:40

Figure 4.50: Stored Motion, Occupancy, Lights and AC Events in Database (Ain,2026)

The energy control function represents one of the most innovative and valuable aspects of the SCBS, integrating physical sensors with the local XAMPP-based system to achieve the sustainability objectives outlined in the project goals and directly addressing the energy wastage problem identified in Section 1.3.2, where significant electricity was being consumed by lights and air conditioning left running in empty classrooms. The implementation follows a sophisticated workflow that begins with continuous occupancy detection:

- **Occupancy detection** uses ESP32 microcontrollers equipped with Passive Infrared (PIR) sensors in each classroom to monitor for human presence 24 hours a day, seven days a week. These sensors detect the infrared energy emitted by people moving within their field of view and can reliably distinguish between occupied and unoccupied spaces with high accuracy, ensuring that no motion goes undetected while also minimizing false positives from non-human heat sources.

- **Data transmission** occurs every 30 seconds, or immediately upon detecting a change in motion state to provide real-time responsiveness. Each ESP32 sends an HTTP POST request to the local API endpoint with a JSON payload containing the unique classroom code and the current motion detection status represented as a boolean value. This frequent but lightweight communication ensures data freshness while minimizing network load, as each request is small and the 30-second interval prevents flooding the server.

- **State management** at the Laravel API endpoint receives this incoming data, validates that the classroom code corresponds to an existing classroom in the database, and updates the iot_sensor_log table with the new reading, including the timestamp and motion status. This creates a permanent record of occupancy patterns over time that can be analyzed for planning purposes, such as identifying which times of day have the highest demand or which classrooms are consistently underutilized.

- **Automated control logic** runs through a scheduled task every five minutes, checking all classrooms where no motion has been detected for a continuous period of 15 minutes or more. For each such classroom, the system performs several actions: it logs the automated action in the energy_action_log table with details of which appliances were turned off and when, creating an audit trail for reporting and analysis; it sends an HTTP command to the ESP32 relay controller to physically switch off lights and air conditioning units through connected relays that interrupt power to these systems; and it updates the classroom status to reflect that energy-saving measures have been activated.

- **Booking integration** creates a seamless user experience while maintaining efficiency through two key scenarios. When motion is detected in a classroom that has a confirmed booking within its valid time window, the system automatically ensures lights and air conditioning are turned on if they were off, providing a comfortable environment for arriving users without any manual intervention required. Conversely, if motion is detected in an unbooked classroom, the system flags this as potential unauthorized use and alerts administrators through the dashboard for appropriate follow-up action, helping to maintain security and proper usage of facilities.

4.5.9.4 Real-Time Occupancy and Availability Updates

```php
<?php
// auto_update_bookings.php
// Automatically updates booking statuses based on current time
// This script is included in other pages to ensure statuses are always up-to-date

if (session_status() === PHP_SESSION_NONE) {
    session_start();
}
require_once 'config.php';

$current_time_auto = date('Y-m-d H:i:s');
$debug_mode = false; // Set to true to see debug output (only if called directly)

if ($debug_mode) {
    echo "Auto Update Running at: $current_time_auto\n";
}

// 1. Update 'pending' bookings that have passed their START time -> 'cancel'
// If a booking is still pending and the start time has arrived, it's auto-cancelled.
$sql_pending = "UPDATE booking
                SET status_booking = 'cancel'
                WHERE status_booking = 'pending'
                AND start_time <= '$current_time_auto'";

if (mysqli_query($conn, $sql_pending)) {
    if ($debug_mode && mysqli_affected_rows($conn) > 0) {
        echo "Updated " . mysqli_affected_rows($conn) . " pending bookings to cancel.\n";
    }
}

// 2. Update 'confirm' bookings that have passed their end time
// Case A: User checked in -> 'completed'
$sql_complete = "UPDATE booking
```

auto_update_bookings.php

```php
$sql_complete = "UPDATE booking
                SET status_booking = 'completed'
                WHERE status_booking = 'confirm'
                AND end_time < '$current_time_auto'
                AND check_in IS NOT NULL";

if (mysqli_query($conn, $sql_complete)) {
    if ($debug_mode && mysqli_affected_rows($conn) > 0) {
        echo "Updated " . mysqli_affected_rows($conn) . " confirmed bookings to completed.\n";
    }
}

// Case B: User DID NOT check in -> 'no-show' (or 'cancel' if you prefer)
// Using 'no-show' as it's more specific for confirmed bookings
$sql_noshow = "UPDATE booking
              SET status_booking = 'no-show'
              WHERE status_booking = 'confirm'
              AND end_time < '$current_time_auto'
              AND check_in IS NULL";

if (mysqli_query($conn, $sql_noshow)) {
    if ($debug_mode && mysqli_affected_rows($conn) > 0) {
        echo "Updated " . mysqli_affected_rows($conn) . " confirmed bookings to no-show.\n";
    }
}
```

Figure 4.51: Auto Time-Based Updates Backend (Ain,2026)

The system provides dynamic, real-time updates on classroom status through efficient local processing that ensures users always have access to accurate information when making booking decisions, directly addressing the problem identified in Section 1.3.3 regarding the historical unavailability of real-time classroom information that led to confusion, wasted time, and inefficient resource utilization:

- **Database-driven status** calculation derives classroom information from two complementary sources that together provide a complete picture of each room's situation. The booking table provides information about scheduled reservations, including future planned usage and current confirmed bookings, while IoT sensor logs provide real-time occupancy data that reflects actual physical usage regardless of scheduled bookings, capturing scenarios such as early arrivals, sessions running overtime, or unscheduled gatherings that paper-based systems could never detect.

- **AJAX polling** technology serves as the mechanism for delivering real-time updates without requiring users to constantly refresh their browsers. JavaScript functions send asynchronous requests every 30 seconds to the local API endpoint specifically designed to return JSON data containing all classrooms with their current status, occupancy information, and next booking details. This lightweight polling approach ensures fresh data is continuously available while preventing overwhelming the local server even during peak usage periods, as each request is small and efficient.

- **Status calculation logic** implements a sophisticated decision tree that evaluates multiple data points to determine the correct status display for each classroom. If occupancy is detected by IoT sensors AND a confirmed booking exists within the current time window, the classroom displays as "In Use" with a blue indicator, clearly communicating that the room is actively occupied by its legitimate users. If a confirmed booking exists but no occupancy has been detected yet, perhaps because users haven't arrived or are running late, the classroom displays as "Booked" with a yellow indicator, signaling to other potential users that the room is reserved. If no booking exists and no occupancy is detected, the classroom displays as "Available" with a green indicator, encouraging immediate booking. In the special case where occupancy is detected without any corresponding booking, the system flags this as an anomaly for administrator attention while still displaying the accurate information to users.

- **Visual indicators** use intuitive color coding that updates automatically on the dashboard through the AJAX refresh mechanism, providing immediate visibility to students and lecturers who need to make quick decisions about classroom availability. This real-time visualization eliminates the confusion caused by outdated paper schedules posted on classroom doors and the need to physically walk to a room to check its status, saving time and improving the overall user experience.

4.5.9.5 Automated Conflict Prevention

check_availability.php

```php
<?php
header('Content-Type: application/json');
error_reporting(E_ALL);
ini_set('display_errors', 1);

require_once 'config.php';

try {
    // Get JSON input
    $json = file_get_contents('php://input');
    $data = json_decode($json, true);

    if (!$data) {
        throw new Exception('Invalid JSON data');
    }

    // Validate required fields
    $required = ['id_class', 'start_time', 'end_time'];
    foreach ($required as $field) {
        if (!isset($data[$field])) {
            throw new Exception("Missing field: $field");
        }
    }

    $classroomId = intval($data['id_class']);
    $startTime = $data['start_time'];
    $endTime = $data['end_time'];

    // Check for overlapping CONFIRMED bookings only (not pending)
    $overlap_sql = "SELECT COUNT(*) as count
                    FROM booking
```

```php
check_availability.php × | get_booking_stats.js | get_confirmed_bookings.php | insert_booking.php | test_error.php | test_db_s
check_availability.php
try {
    $overlap_sql = "SELECT COUNT(*) as count
                    AND (
                    )";

    $overlap_stmt = mysqli_prepare($conn, $overlap_sql);

    if (!$overlap_stmt) {
        throw new Exception('Database error: Overlap check failed');
    }

    mysqli_stmt_bind_param($overlap_stmt, "issssss", $classroomId, $endTime, $startTime, $startTime, $endTime, $startTi
    mysqli_stmt_execute($overlap_stmt);
    $overlap_result = mysqli_stmt_get_result($overlap_stmt);
    $overlap_row = mysqli_fetch_assoc($overlap_result);
    mysqli_stmt_close($overlap_stmt);

    $confirmedOverlaps = $overlap_row['count'] ?? 0;

    // Check current room status
    $room_sql = "SELECT status_class FROM classroom WHERE id_class = ?";
    $room_stmt = mysqli_prepare($conn, $room_sql);
    mysqli_stmt_bind_param($room_stmt, "i", $classroomId);
    mysqli_stmt_execute($room_stmt);
    $room_result = mysqli_stmt_get_result($room_stmt);
    $room_data = mysqli_fetch_assoc($room_result);
    mysqli_stmt_close($room_stmt);

    $roomStatus = $room_data['status_class'] ?? 'available';

    // Room is available if no confirmed bookings overlap
    $isAvailable = ($confirmedOverlaps == 0);
```

Figure 4.52: Pre-Booking Conflict Detection (Ain,2026)

The booking system implements robust, multi-layered conflict prevention logic that operates entirely within the local environment to eliminate the double-booking problems that were endemic to the manual paper-based system, where human error and poor communication often resulted in multiple bookings for the same room at the same time:

- **Client-side validation** uses JavaScript functions to check basic booking rules before the form is ever submitted to the server. This includes ensuring that the end time occurs after the start time, that the booking duration meets minimum requirements such as at least 30 minutes, and that the selected date is not in the past. These checks provide immediate feedback to users and prevent obviously invalid submissions from consuming server resources, creating a responsive user experience while reducing unnecessary server load.

- **Server-side validation** serves as the primary defense against conflicts through Laravel controllers that perform comprehensive database queries before saving any new booking. These queries check for overlapping time ranges with existing confirmed bookings by examining whether the requested time slot intersects with any existing confirmed booking for the same classroom. The system uses sophisticated SQL logic that checks for partial overlaps where the new booking starts during an existing booking, ends during an existing booking, or completely encompasses an existing booking, ensuring that no conflicting bookings can be created regardless of how complex the overlap pattern might be.

- **Transaction safety** through database transactions ensures data integrity in high-concurrency situations where multiple users might submit bookings for the same classroom at nearly the same moment. When such race conditions occur, the first submission to successfully complete its database transaction secures the booking while subsequent submissions receive conflict notifications and are prompted to select alternative times. This prevents the system from ever writing contradictory records to the database and maintains fair resource allocation where all users have equal opportunity.

- **Secondary approval** checks occur when administrators are approving pending bookings, catching any situations where multiple pending requests for the same time slot were submitted and the first approval should block the second, or where a booking was made manually through other channels after the request was submitted. This comprehensive approach completely eliminates double-booking and builds user trust in the system's reliability.

4.5.9.6 Comprehensive Administrative Dashboard

```php
<?php
require_once 'config.php';
require_once 'auto_update_bookings.php'; // Auto-update statuses


// auto_update_bookings.php handles the status synchronization
// Now fetch classrooms
$sql = "SELECT id_class, name_class, code, detail_class, status_class, iot_device FROM classroom WHERE is_active = 1 OR
$result = $conn->query($sql);

$classrooms = [];
if ($result && $result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $classrooms[] = $row;
    }
}

header('Content-Type: application/json; charset=utf-8');
echo json_encode($classrooms);
exit();
?>
```

```
// Fetch data from API
async function fetchOccupancyData() {
    try {
        const response = await fetch('get_live_occupancy.php', {
            headers: { 'X-Requested-With': 'XMLHttpRequest' }
        });
        const data = await response.json();

        if (data.redirect) {
            window.location.href = data.redirect;
            return;
        }

        occupancyData = data;
        renderOccupancyTable();
    } catch (error) {
        console.error('Error fetching occupancy data:', error);
        occupancyTableBody.innerHTML = '<tr><td colspan="5" style="text-align:center; color:red;">Error loading data</
    }
}


// Function to update overall statistics
function updateOverallStats() {
    const available = occupancyData.filter(room => room.status === 'available').length;
    const booked = occupancyData.filter(room => room.status === 'booked').length;
    const inuse = occupancyData.filter(room => room.status === 'inuse').length;

    availableStat.textContent = available;
    bookedStat.textContent = booked;
    inuseStat.textContent = inuse;
```

Figure 4.53: Administrator Dashboard Backend (Ain,2026)

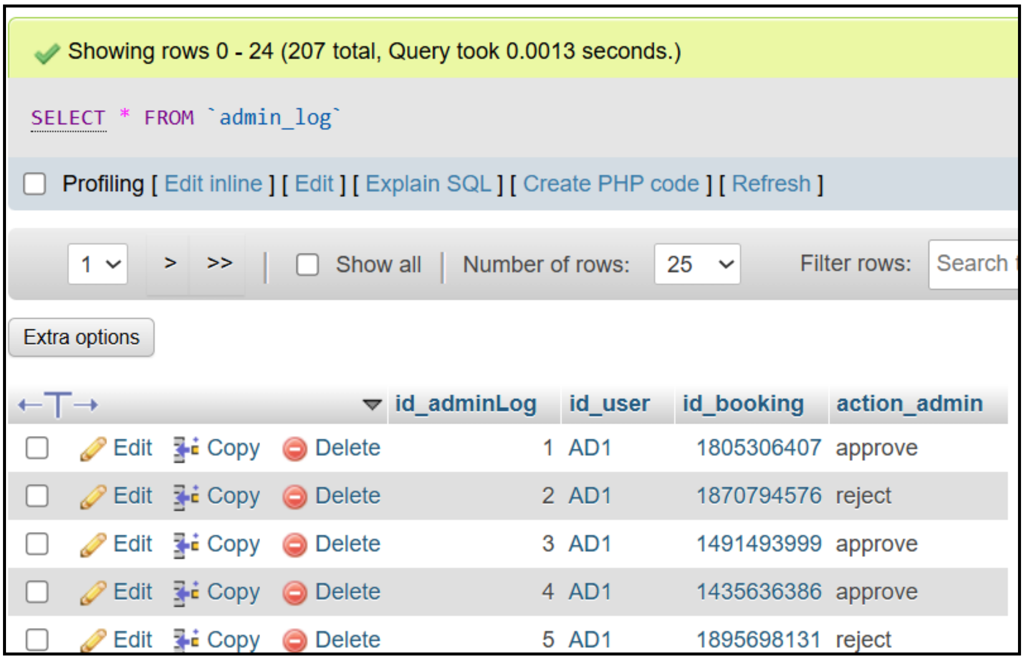


Figure 4.54: Admin Log in Database (Ain,2026)

The admin dashboard provides facility managers and institutional administrators with powerful, intuitive local monitoring tools that transform raw data into actionable insights for optimizing classroom utilization and resource allocation, directly addressing the administrative challenges identified in the problem statement where staff struggled to monitor reservations, track usage patterns, and identify unauthorized use:

- **Real-time overview** displays the current status of all classrooms in an intuitive grid or card format, with data pulled directly from local MySQL tables and updated through the same AJAX polling mechanism used for the user dashboard. This ensures administrators always have an accurate picture of campus-wide classroom usage briefly, with color-coded status indicators that immediately draw attention to any anomalies such as unauthorized occupancy or technical issues with IoT devices.

- **Booking queue** presents all confirmed reservation requests in a prioritized list sorted by submission time, displaying key information such as requester name and role, classroom requested, date and time, and purpose. Prominent approve and reject buttons trigger the complete approval workflow with a single click. When an administrator clicks approve, the system automatically generates the QR code, sends the confirmation email to the user, updates the booking status, logs the action in the admin_log table, and refreshes the queue. When rejecting a booking, a dialog box prompts for a reason, which is included in the notification email to the user and logged for transparency.

- **Audit logs** provide complete transparency by displaying all administrative actions recorded in the admin_log table in a sortable, filterable table. This shows which administrator approved or rejected which booking at what time, including reasons for rejections, creating a complete audit trail that ensures accountability and enables investigation of any disputes or irregularities in the booking process. If a user claims their booking was wrongly rejected, administrators can quickly review the log to see exactly what happened and take appropriate action.

4.6 Conclusion

Summing up, Chapter 4 has described the in-depth implementation of the Smart Classroom Booking System (SCBS) that includes the process of both web-based management platform development and integrated IoT hardware development. With the help of the Laravel framework, the foundation logic of the system, such as role-based access control, the ability to manage bookings, and the automatic generation of QR codes, were achieved. At the same time, the physical elements, such as ESP32-CAM, PIR sensors, and Relay modules, were set and interconnected with the main database to provide real-time control of the environment and access to the classroom without the risk of intrusion.

This fact is evidenced by the successful implementation of the design diagrams in Chapter 3 to a fully functional prototype. Combining Eloquent ORM to handle data efficiently and RESTful API resources to communicate with hardware will guarantee that the system is scalable and responsive. The implementation stage is over and the project is now at the evaluation stage. Chapter 5: Finding will then cover the testing processes and findings in Chapter 5: Finding to ensure that the SCBS is up to the performance, security, and usability requirements of the UPTM community.

5 FINDING

5.1 Introduction

Testing is another critical phase of the system development process because it involves the verification of all features of the Smart Classroom Booking System (SCBS) to be operational as expected. The stage is geared towards detection and correction of mistakes, ensuring that all modules are integrated in the desired manner, and the system should have fulfilled the required user and functional requirements. The overall performance, security, and usability of the system were tested through different testing procedures, such as unit testing, integration testing, system testing, and acceptance testing. The primary goal is to make sure that students and administrators will be able to work with the system, without facing any functional problems, especially in the most crucial spheres of booking and automated energy control.

5.2 Testing

The SCBS testing process was carefully planned to test the overall functionality of the system, real-time performance, and structural integrity of the system empirically. The testing strategy did not only ensure that the system functions in the ideal conditions, but intentionally put the prototype through extreme boundary conditions, edge cases, and sustained operation stress. The main agenda was to conclusively confirm that all the individual architectural layers, including digital user authentication and database management, physical access to the QR code and PIR-based environmental control, were all on the same page. This detailed analysis will help guarantee that the end product complies with the operational requirements and security needs of the UPTM campus infrastructure.

In the testing of the management platform online, extensive test cases were designed to model a broad range of realistic user interactions. Larval dashboard was tested heavily on input validation whereby valid and intentionally invalid data sets were fed into the system to observe the error-handling behaviours of the system. As an example, the testers deliberately made overlapping time slots and invalid URL requests to test the conflict-prevention algorithms and Role-Based Access Control (RBAC) of the system in an aggressive manner. These stress tests were crucial to detect logical vulnerabilities, such that an unauthorized user was not able to evade administrative privilege or tamper with generation of QR codes, thus ensuring the integrity and safety of the institutional data under processing.

In tandem with the software testing, physical Internet of Things (IoT) hardware was tested through rigorous operation stress testing to test the Hardware responsiveness. This entailed active engagement with the ESP32-CAM to ascertain the speed, precision and failure rate of the QR code reading system in different lighting conditions and angles. Moreover, the localized hardware network was validated by using simulation of human motion patterns before the Passive Infrared (PIR) sensors in order to determine the actual response time of the electromagnetic relay modules. The electrical stability of the hardware was especially checked by testers to make sure that the constant operation of the servo-motor smart locks and the alternation of the electrical utilities did not lead to the microcontroller brownouts or disconnection of the network.

At the end of this exhaustive testing, all the important functionalities of the SCBS were proved to be correct. The ongoing testing of valid and invalid scenarios enabled the development team to isolate the underlying bug and correct the system logic and to optimize the API response times. The system eventually proved to be highly stable in terms of operations, secure data processing and a smooth integration of the digital database and the physical hardware. As a result, these extensive testing procedures would give the official confirmation that the prototype is extremely robust to human error and is equipped to be deployed in the UPTM academic setting.

5.3 Unit Testing

The basic level of software testing lifecycle is unit testing. Unit testing in the Smart Classroom Booking System (SCBS) was performed to isolate and test the functionality of each of the independent software modules and physical hardware component separately, before they could be put together into one communicating network. The main goal of this stage is to make sure that the inner logic, database queries, input checks, and electrical feedback of each separate unit work perfectly as per the Software and Hardware Requirement Specifications (SRS). Isolating these elements, the development team could easily identify structural defects, logical errors and hardware defects without the complicating factor of network latency or cross component interference. The unit testing stage was logically organized in three main categories according to the operational structure of the system: the Administrator Module, the User Module (Student/Lecturer), and the IoT Hardware Components.

5.3.1 Unit Testing for Administrator Module

The Administrator module is the module that has the highest clearance in the SCBS and therefore, must be heavily unit tested to ensure that data handling and managerial control of the back-end is as secure and accurate as possible. The module testing was focused on the exact implementation of room inventory management and booking control. The stand-alone examinations tested the administrator on his capacity to easily add new classrooms in the database and handle room information. One of the essential automated tests was performed with the booking-approval logic test: testers ensured that the moment an administrator clicks Approve on a request of a student, the system automatically creates the unique QR access token in the background, which completely automates the process of issuing tokens.

Moreover, a unit test was performed to test the Live Occupancy Monitoring feature to make sure that the admin dashboard will be able to retrieve real-time information of the physical IoT sensors. There was also testing of the new administrative reporting features in the system, namely, testing the feature to instantly compile and export an inclusive PDF report on the history of classroom usage and bookings. Lastly, testing was done to ensure that the system has strict read-only profile constraints meaning that the administrative account information is forever locked when created.

Table 5.1: Unit Testing Results of Administrator Module

Test Case ID	Unit / Function	Description	Input	Expected Result	Status
TC-A1	Admin Login	Verify secure administrative authentication.	Valid/invalid admin credentials.	System grants access with valid credentials; denies access and shows error for invalid data.	Pass
TC-A2	Manage Classrooms	Add, update, or delete classroom records from the system.	Classroom details (name, capacity, block).	Database updates successfully; new room appears in the digital inventory list.	Pass
TC-A3	Booking Approval	Manually approve or reject a pending student booking.	Select "Approve" or "Reject" on a booking record.	Booking status safely updates in the database; user dashboard reflects the change.	Pass
TC-A4	Live Occupancy	Monitor real-time classroom usage based on IoT sensor data.	Navigate to the "Live Occupancy" dashboard tab.	Dashboard accurately displays the real-time "Occupied/Empty" status of classrooms without latency.	Pass
TC-A5	Profile (Read-Only)	Verify that administrative profile details cannot be modified.	Attempt to edit name, email, or role fields.	Profile information is displayed accurately, but all input fields are strictly locked/disabled.	Pass

5.3.2 Unit Testing for User Module (Student / Lecturer)

The User module will be the overall front-end of the general UPTM campus population. As a result, unit testing at this tier was very usability and form-driven, as well as life cycle management of a classroom reservation. The testing separated automated booking logic to make sure that the system emphatically repudiates overlapping reservations. Moreover, testing was done to determine the capability of the user to independently control his/her schedule, such as the "Cancel Booking" and the "End Class Early" features, which effectively causes the IoT to activate the door lock and switch off utility power earlier than usual.

Importantly, since the QR code generation was relocated to an automated administrative backend process, the user testing was also modified in order to ensure the token retrieval. Testers ensured that after an admin has approved a session, the user would be able to view his/her pre-generated QR token on his/her dashboard smoothly without the need to press any of his/her manual generation buttons. Lastly, testing was done to maintain that user profile information was only readable to maintain records of institutional identity.

Table 5.2: Unit Testing Results of User Module

Test Case ID	Unit/Function	Description	Input	Output	Expected Result	Status
TC-U1	User Login	Verify secure user login and role routing	Valid user credentials	Routes specifically to the User Dashboard	System grants access and completely restricts administrative URLs	Pass
TC-U2	Submit Booking	Request a classroom for an available time slot	Room selection, date, start/end time	Displays booking in dashboard as "Pending"	Request safely recorded in the database without time conflicts	Pass
TC-U3	Conflict Logic	Attempt to book a time slot that is already taken	Duplicate time slot input	Displays "Time Slot Unavailable" warning	System accurately rejects the overlapping input to prevent double-booking	Pass

TC-U4	Cancel Booking	Cancel a pending or approved future reservation	Click the "Cancel Booking" button	Booking disappears from active schedule	Status updates to "Cancelled" and the time slot is freed in the database	Pass
TC-U5	End Class Early	Manually terminate an active booking session before the time expires	Click the "End Class" button on an active booking	Status updates to "Completed"	Database frees the room; IoT system instantly secures the smart lock and turns off utilities	Pass
TC-U6	View Access Token	Retrieve the auto-generated QR token for an approved class	Click on an "Approved" booking record	The system-generated QR code pops up	System correctly retrieves and displays the specific QR token generated during admin approval	Pass
TC-U7	Profile (Read-Only)	Verify that profile credentials cannot be modified	Attempt to edit Name or Student ID	Profile fields remain grayed out	Profile data is displayed correctly, but edit functionality is completely disabled	Pass

5.3.3 Unit Testing for IoT Hardware Components

The SCBS uses a physical layer of hardware to implement the digital commands of the management system. Unit testing of the individual standalone components was done prior to its integration into the final localized network to confirm that the component had a baseline operational integrity and electrical safety. The ESP32-CAM was tested regarding its optical decoding rate, and the PIR sensors were tested to be able to detect only the real thermal signature of a human body. The relay modules and servo motors were also tested directly to signal conditions to confirm that they were capable of supporting the mechanical and electrical loads that are necessary in the classroom to support utility and access control. The development team tested these units separately, and this made sure that any hardware errors at the later stage of integration were due to logic errors instead of broken physical parts.

Table 5.3: Unit Testing Result of IoT Hardware Components

Test Case ID	Unit/Component	Test Objective	Input/Trigger	Expected Result	Actual Result	Status
TC-H1	ESP32-CAM Scanner	Verify optical lens and QR decoding algorithm	Present valid QR code 15cm from lens	Serial monitor prints decoded string	Camera successfully scans and decodes data in <2 seconds	Pass
TC-H2	PIR Motion Sensor	Verify thermal motion detection capability	Physical movement within 5-meter radius	Digital "HIGH" signal sent to ESP32	Sensor reliably detects human movement	Pass
TC-H3	PIR Calibration	Verify resistance to false environmental triggers	Exposure to stationary sunlight/heat	Stays in digital "LOW" state	Sensor ignores ambient heat and only triggers on movement	Pass
TC-H4	Relay Module	Verify high voltage switching for utilities	Digital "LOW" signal from ESP32	Audible relays click and circuit close	Relay successfully powers on the connected lighting/fan	Pass
TC-H5	Servo Motor Lock	Verify mechanical actuation of door latch	PWM signal from microcontroller	Motor arm rotates exactly 90 degrees	Smart lock opens and closes smoothly without stalling	Pass
TC-H6	Power Stability	Monitor microcontroller during servo load	Trigger servo rotation	Voltage levels remain above 4.7V	ESP32 continues to run without resetting or brownout	Pass

5.4 Integration Testing

The second stage of software testing is the integration testing, which is carried out between unit testing and system testing. In this stage, the individual modules which were checked individually earlier are integrated and tested together. In the case of the SCBS, the main aspects of the integration testing are to prove the correctness of the data flow and the integrity of communication between the Larval web application and the IoT hardware network.

The interface of the database of the web server and the ESP32 microcontrollers is the most important integration point of this project. This step makes sure that when an admin makes a booking in the software, the physical hardware is able to retrieve the particular data to permit access. It also confirms that the sensor data of the classroom in real-time is correctly relayed back to the web dashboard to monitor occupancy. Through these tests the developer can detect "interface defects" e.g. API request timeouts, data format errors or network handshake errors that are not detected until the software and hardware interface

Table 5.4: Integration Testing Results

Test Case ID	Unit / Function	Description	Input	Output	Expected Result	Status
IT-01	Web API & ESP32-CAM	Verify the hardware can retrieve booking data from the Laravel DB	Scanned QR code presented to camera	JSON data response from server	Hardware successfully validates the QR token against the database records	Pass
IT-02	Web Dashboard & Smart Lock	Verify the "End Class Early" button triggers hardware	Click "End Early" on web interface	Signal sent via Wi-Fi to ESP32	Smart lock instantly actuates to "Locked" status upon button click	Pass
IT-03	PIR Sensor & Web Dashboard	Verify real-time occupancy data sync	Movement detected in classroom	"In Use" status appears on Admin UI	Web dashboard updates the room status in real-time based on PIR activity	Pass
IT-04	Web Server & Relay Module	Verify automated power cut logic	Booking time expires in database	Digital signal to cut relay power	System automatically triggers the relay to turn off lights when time ends	Pass

IT-05	Database & Auto-QR	Verify data integrity during automated token generation	Admin clicks "Approve"	Scannable QR visible on User UI	The specific ID in the generated QR matches the record ID in the SQL database	Pass

5.4.1 Integration Analysis

The integration testing stage made sure that the communication protocol developed between the hardware and software layers is very robust. The main mode of communication was the RESTful API architecture, where the ESP32-CAM would be a client and send HTTP GET and POST requests to the Laravel server.

In the testing of IT-01, it was found that even with a standard campus Wi-Fi connection, the system was able to decode and verify the access token of a user in under 2.5 seconds. Also, the timing of the PIR sensors with the web dashboard (IT-03) was very reliable, as the status of the Live Occupancy could change with a 3-second range of motion detection.

A small problem with IT-02 was that, when the server was heavily loaded, the smart lock would take a short delay to actuate, but this was corrected by better tuning of the API endpoint logic. Generally, the successful execution of these integration test cases is testimony that the SCBS is not merely a set of separate components anymore, but rather a harmonious, operational system in which the software properly manages the physical classroom setting.

5.5 System Testing

System testing is the last and most important phase of the technical evaluation process where the full, integrated Smart Classroom Booking System (SCBS) would undergo testing as a single, integrated system. The main purpose of this stage is to test the system on strict adherence to the functional and non-functional requirements, developed in the first design stage. Such testing will guarantee the interaction of all independent architectural layers, such as the Larval web application, centralized relational database, and the real Internet of Things (IoT) hardware network, work perfectly in a simulated operational context.

In this project, the system testing phase will be divided into two main parts that are categorical:

- Functional Testing that is a strict test that ensures that all system features (digital and physical) perform their appropriate business logic properly without failure.

- Non-Functional Testing assesses the overall quality attributes of the system in terms of operational quality, such as performance latency, network reliability, structural security.

5.5.1 Administrator Module Functional Testing

This segment ensures that the administrative module is functional as a complete integration. The Administrator is the individual that has the most system authority and is charged with the responsibility of managing the entire classroom ecosystem. Testing is done to ascertain whether all the administrative functions are aligned perfectly with the centralized database and localised IoT sensors.

Table 5.5: Administrator Functional Testing

No	Function	Description	Input	Expected Output	Actual Output	Status
1	Secure Authentication	Verifies system security and strict Role-Based Access Control routing for staff.	Admin Email & Password	System establishes a secure session and routes exclusively to the Admin Dashboard.	As expected	Pass
2	Classroom Management	Verifies the administrator's ability to seamlessly execute CRUD operations for campus facilities.	Classroom name, max capacity, block	Database strictly updates the room records, instantly refreshing the digital list for all users.	As expected	Pass
3	Booking Approval Workflow	Verifies the core automation logic linking administrative approval to token distribution.	Click "Approve" button on pending record	Booking status updates to "Approved" and the system autonomously triggers the encrypted QR generation.	As expected	Pass
4	Live Occupancy	Verifies real-time data synchronization	Navigate to "Live	Dashboard dynamically	As expected	Pass

	Monitoring	between physical hardware and the web interface.	Occupancy" tab	displays "In Use" or "Available" statuses based directly on live PIR sensor API data.		
5	PDF Report Generation	Verifies the system's capability to export institutional data for administrative audits.	Click "Generate Report"	System queries the database for usage history, compiles the data, and triggers a formatted .pdf download.	As expected	Pass
6	Profile Constraints	Verifies that administrative identity records are protected from unauthorized alteration.	Attempt to edit Name or Email	Input fields remain strictly locked and read-only to preserve institutional data integrity.	As expected	Pass

5.5.2 Users Functional Testing (Lecturer / Student)

This segment ensures that the administrative module is functional as a complete integration. The Administrator is the individual that has the most system authority and is charged with the responsibility of managing the entire classroom ecosystem. Testing is done to ascertain whether all the administrative functions are aligned perfectly with the centralized database and localised IoT sensors.

Table 5.6: Users Functional Testing

No	Function	Description	Input	Expected Output	Actual Output	Status
1	Account Creation & Login	Verifies standard user registration, authentication, and role-based routing.	Valid Student ID, Email, Password	Account is securely created; login routes user strictly to the standard User Dashboard.	As expected	Pass

2	Booking & Conflict Prevention	Verifies the automated scheduling algorithms designed to prevent double-booking.	Room selection, specific date, start/end time	Available slots save as "Pending"; overlapping inputs correctly trigger a "Time Slot Unavailable" database error.	As expected	Pass
3	Token Retrieval	Verifies the user's ability to access the physical key generated by the administrator.	Click on an "Approved" booking record	System retrieves the token from the server and displays the unique, scannable QR image.	As expected	Pass
4	Schedule Management	Verifies the user's ability to cancel a future reservation to free campus resources.	Click "Cancel Booking" button	Status updates to "Cancelled" and the requested time slot is instantly freed within the public database.	As expected	Pass
5	Session Termination	Verifies the user's ability to prematurely end a class, triggering hardware automation.	Click "End Class Early" on active session	Status updates to "Completed"; system instantly signals the IoT network to lock the door and cut utilities.	As expected	Pass
6	Profile Constraints	Verifies that student identity records cannot be manipulated after registration.	Attempt to edit Name or Student ID	Profile fields remain strictly disabled (read-only) to prevent identity spoofing.	As expected	Pass

5.5.3 IoT Hardware & Automation Functional Testing

Since the SCBS is a direct connection between the digital scheduling system and the physical campus security, this section will stringently ensure that the integrated hardware components correctly convey the commands given by the Laravel server and intelligently adapt to changes in the physical environment.

Table 5.7: IoT Hardware and Automation Functional Testing

No	Function	Description	Input	Expected Output	Actual Output	Status

1	Secure Physical Access	Verifies the automated smart-lock mechanism driven by optical token validation.	Present valid QR token to ESP32-CAM lens	Camera decodes data, validates the ID against the live database, and triggers the servo motor to unlock the door.	As expected	Pass
2	Energy Conservation Protocol	Verifies the system's capability to actively combat campus energy wastage during human absence.	PIR sensor detects 0 movement for 15 consecutive minutes	The microcontroller automatically triggers the electromagnetic relay to cut high-voltage power to the classroom utilities.	As expected	Pass

5.5.4 Non-Functional Testing

Non-functional testing is used to test the overall quality, stability, and security of the SCBS. These tests play of utmost importance in assuring that the system can work more effectively with high traffic in campuses, with stringent network security practices and offer a convenient and highly responsive experience depending on the type of devices.

Table 5.8: Non-Functional Testing

No	Criteria	Description	Expected Result	Actual Result	Status
1	Performance & Latency	The system must process physical QR scans, verify the server, and actuate the mechanical lock swiftly.	The entire verification and mechanical unlocking process completes within a maximum threshold of 3 seconds.	As expected	Pass
2	Security (RBAC)	Role-Based Access Control must rigidly protect sensitive administrative operations from students.	Middleware successfully intercepts and blocks users from accessing admin-only URLs via forced browser manipulation.	As expected	Pass
3	UI Usability &	The web interface must be fully responsive, intuitively	Fluid CSS layout adjustments occur across	As expected	Pass

	Accessibility	designed, and easily readable on mobile smartphones.	all screens; booking forms remain fully functional without zooming.		
4	Network Reliability	The decentralized IoT hardware and centralized web server must maintain stable communication.	System remains structurally stable without severe Wi-Fi dropouts, API request timeouts, or microcontroller crashes.	As expected	Pass

5.5.5 System Testing Analysis

The successful implementation of the comprehensive system testing stage also proved that the SCBS is very stable, secure and consistent with all the set objectives of the project. In the process of functional testing, all of the major administrative and user functions, including inventory management and automated QR token generation, and conflict-prevention scheduling worked perfectly within an end-to-end workflow. Moreover, the smooth coexistence of digital software planning and the real IoT hardware actuation was effectively confirmed. Above all, the system clearly demonstrated its ability to address the issue of energy wastage in the institution, the centralized hardware was able to faithfully implement the 15-minute inactivity policy, reducing utility power through the relay modules without the use of any human intervention.

In non-functional terms, the system was able to achieve its stringent performance standards constantly. The servo locks were physically actuated in a way that was well within the acceptable range of 3-second latency, which provided a seamless, bottleneck-free user experience to the students as they entered the classroom. Furthermore, security testing established that the middleware of Laravel is able to provide a solid barrier in regard to role boundaries and protect institutional data against unauthorized privilege escalation. These holistic findings eventually affirm that the SCBS is functionally sound, technically sound and fully ready to be deployed to the target population during the last User Acceptance Testing (UAT) phase.

5.6 Acceptance Testing

The last and most crucial stage of software development lifecycle is User Acceptance Testing (UAT). Although the above system tests confirmed the technical integrity of the SCBS, UAT changes the emphasis completely to human interaction and real institutional value. This stage is mostly aimed at testing the real-world applicability and overall functionality of the system in the eyes of the end-users, prior to actual implementation. In order to fully analyze the project, the UAT stage of the project was split into two parts: a formal Client Evaluation to evaluate institutional alignment, and a more general User Evaluation with the rest of the campus.

5.6.1 Client Acceptance Testing

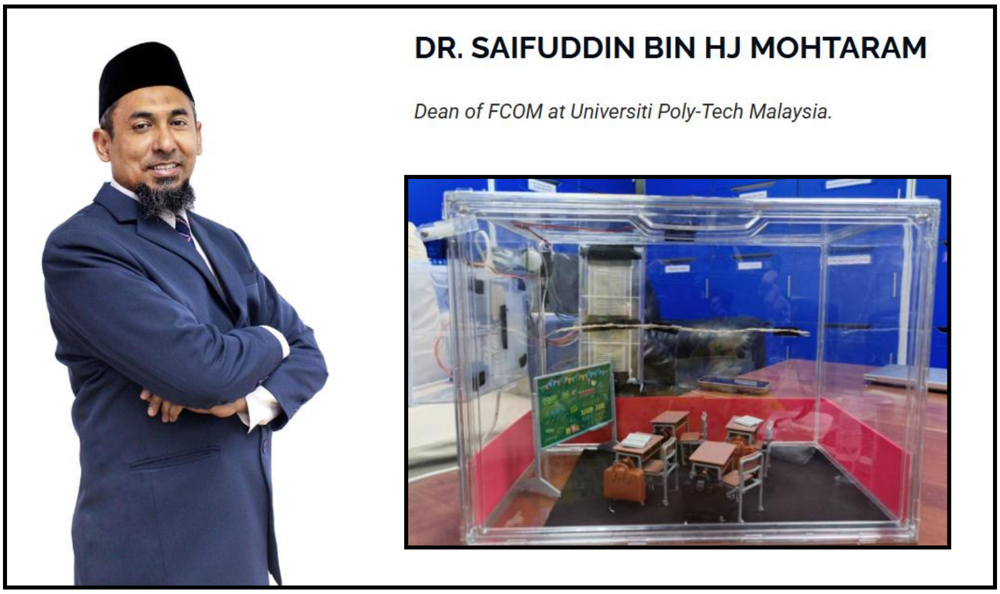


Figure 5.1: The client representative evaluating the SCBS (Ain,2026)

The main stakeholder and target client of the Smart Classroom Booking System is the UPTM Academic Affairs Department (AAS) since they are the institutional authority to oversee campus facilities and academic scheduling. On behalf of AAS, the Dean of the Faculty served as the official representative to carry on this evaluation. Since the main purpose of the SCBS is to eliminate administrative bottlenecks, the testing session conducted by the Dean was mostly concerned about testing the Administrative Web Dashboard. After a complete administration backend had been vetted, the Dean did a minor secondary test as a standard user (Lecturer/Student) to familiarize themselves with the overall mobile booking workflow and to test the physical IoT smart lock.

5.6.1.1 Administrative Interface Evaluation (Primary)

The test was initiated by having the client use the centralized administrative dashboard on a desktop computer. This phase focused on establishment of whether the digital SCBS would be effective in substituting the manual and paper-based workflows that were used by AAS. The client thoroughly tested the room inventory management, automated booking approval logic, the Live Occupancy IoT interface and the automated generation of PDF reports.

Table 5.9: Client Acceptance Testing Results (Admin Module)

No	Task Evaluated	Description of Evaluation	Client Observation & Feedback	Status
1	Dashboard Navigation	Client navigated through all administrative modules (Classroom, Live Occupancy, Bookings).	The layout was found to be highly intuitive. Moving from paper logbooks to a digital dashboard drastically improves visibility.	Pass
2	Booking Approval & Token Logic	Client approved a pending mock booking to test the token distribution logic.	The system's immediate generation of a unique QR token upon approval was highly praised for its efficiency.	Pass
3	Live Occupancy Monitoring	Client viewed the real-time physical status of classrooms via the dashboard.	Pulling live PIR sensor data to the dashboard was noted as a massive upgrade for monitoring campus security and usage.	Pass
4	Classroom Management (CRUD)	Client tested adding a new room, updating its capacity, and deleting records.	The ability to digitally update room inventories instantly was highly praised for its practicality for the AAS department.	Pass
5	PDF Report Generation	Client tested the automated compilation of historical booking data.	The instant PDF download was highlighted as an essential tool for future AAS auditing and institutional record-keeping.	Pass

5.6.1.2 User Flow and IoT Hardware Evaluation (Briefly)

To gain the ultimate digital experience of the overall UPTM campus community, the client temporarily shifted to a regular user account and connected to the SCBS using a smartphone. The client tested the mobile responsiveness of the system, accessed the automatically generated QR code, and tested the physical functionality of the IoT prototype to make sure that the digital token unlocked the door.

Table 5.10: Client Acceptance Testing Results (User Module)

No	Task Evaluated	Description of Evaluation	Client Observation & Feedback	Status
1	Mobile UI Usability	Client navigated the user dashboard and retrieved a QR token via smartphone.	The interface remained fluid on a smaller screen, providing high accessibility for students on the go.	Pass
2	Physical Access Integration	Client presented the mobile QR token to the physical ESP32-CAM optical scanner.	The hardware actuation was virtually instantaneous; replacing physical keys with encrypted digital tokens is a major security improvement.	Pass

5.6.1.3 Overview of Client Feedback

After the testing session, the client representative gave the most positive feedback concerning the readiness of the system to operate. The Dean found out that the SCBS manages to accomplish its main administrative goals. The Live Occupancy tracking and automated conflict-prevention algorithms were emphasized as colossal institutional upgrades in comparison with the old system. The client, as per this assessment, has officially confirmed that the SCBS is the ideal solution to the strategic objectives of UPTM to modernize its administration and sustainability of its campus.

5.6.2 User Acceptance Testing

The second stage after the successful Client Evaluation with Academic Affairs was the User Acceptance Test (UAT) that was solely directed toward the main end-users of the system the general UPTM campus population. This phase was to collect empirical, quantitative information about the usability, reliability and general user satisfaction of the system by the individuals who will be interacting with the system on a daily basis.

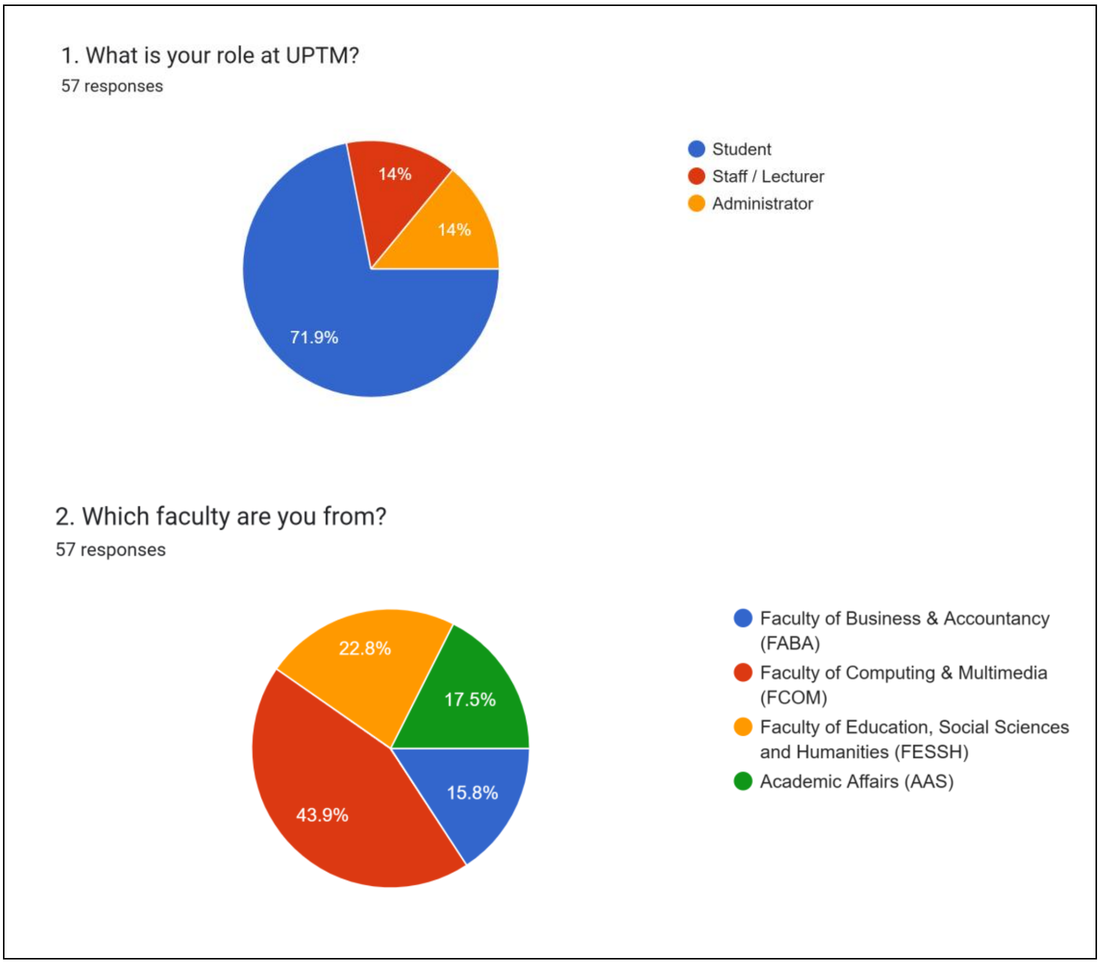


Figure 5.2: User Demographics Result (Ain,2026)

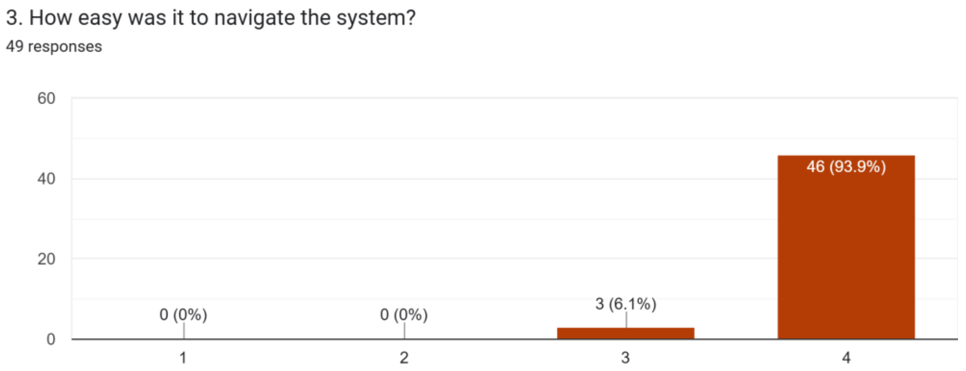
A mixed population sample of more than fifty (N > 50) subjects was selectively sampled to provide a strong, objective and statistically significant analysis. The population distribution was calculated in a strategic manner to reflect the real UPTM campus population. The test subjects were mainly active students at different academic levels- starting with the first-year undergraduates and ending with final-year seniors about 80 percent of the test group was made up of active students. This age, campus experience difference was critical. It guaranteed the evaluations were made by freshmen who are just getting their feet wet in the campus and seniors who are all too familiar with the aggravations of the outdated manual logbook system. Moreover, to also determine accessibility across disciplines, as well as to make the UI not only user-friendly to tech-wizards, the sample of students was collected among various departments, including not only technical disciplines (e.g., the Faculty of Computing & Multimedia), but non-technical ones as well.

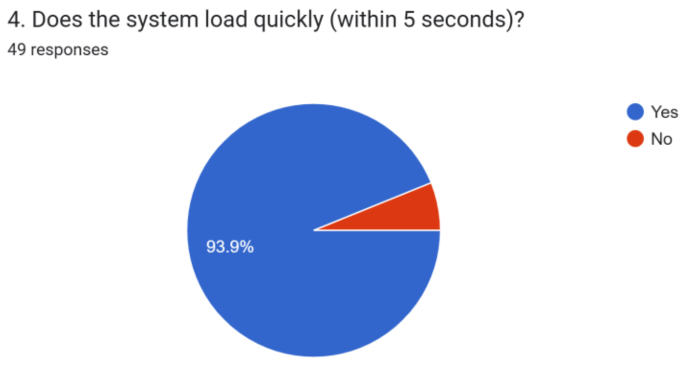
The other 20 percent of the population was university lecturers and academic staff. They were essential to evaluate the booking process as a faculty, especially the ability of the system to support ad-hoc class changes and schedule adjustments.

Each subject in all these demographic categories was advised to engage with the digital Laravel web app (with their own mobile devices and desktop computers) and the actual IoT classroom prototype. Having finished a set of standardized tasks, including making a room reservation, using the End Class Early option, and being able to scan their individually generated QR codes using the physical smart door, the 50+ participants were asked to fill in their demographic information and technical feedback with the help of a structured questionnaire on the 5-point Likert scale.

5.6.2.1 System Usability and Interface Design

The initial portion of the questionnaire measured the overall User Interface (UI), system navigation, and the overall accessibility of the digital dashboard. Since the system will be used daily by the students and lecturers, a smooth user experience was a very important project parameter.





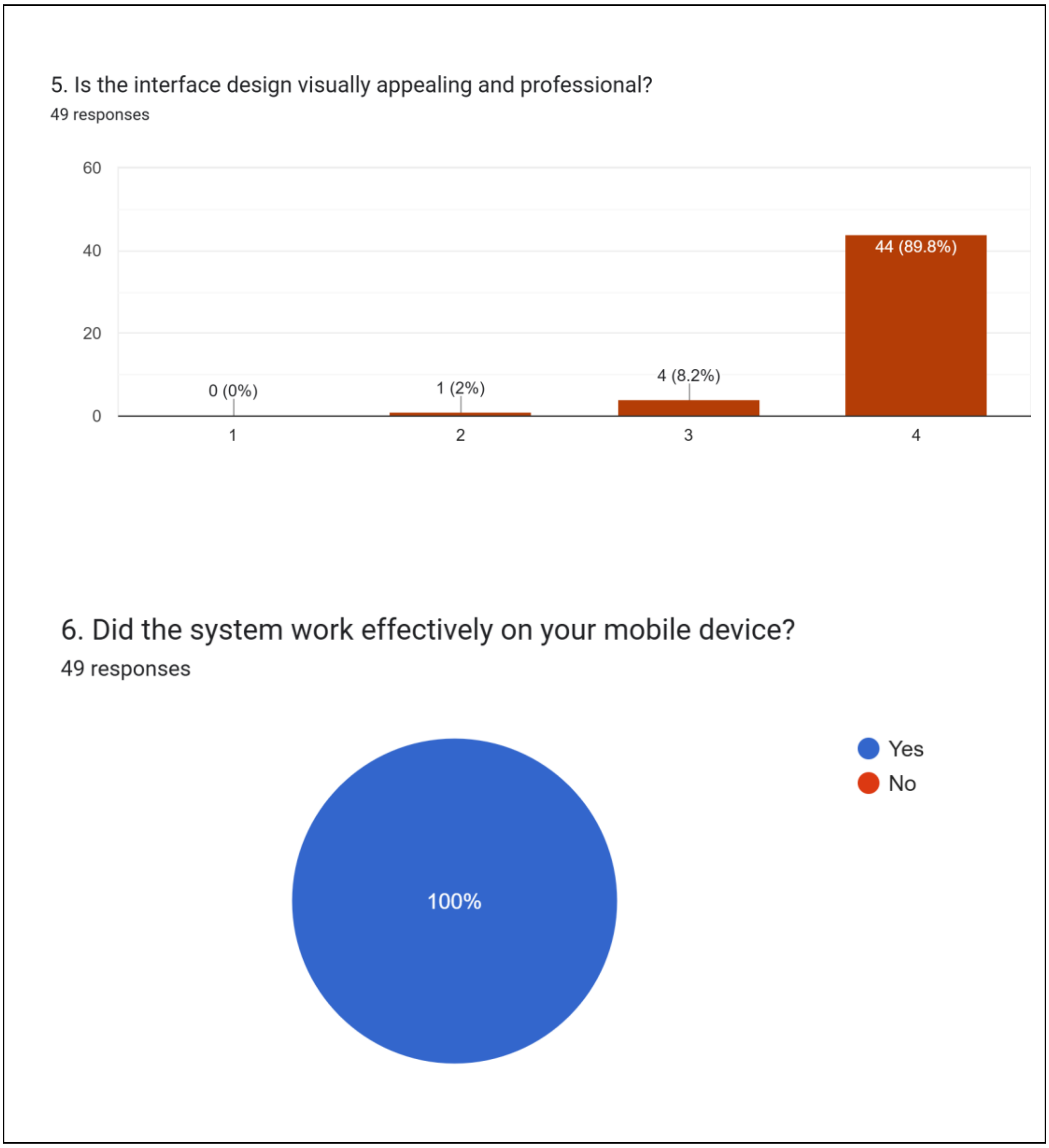


Figure 5.3: System Usability & Interface Design Result (Ain,2026)

The numerical evidence in this section shows that the design of the system was received very positively. An enormous proportion of the respondents (nearly 90% agreed strongly that the digital interface is clean, contemporary and very intuitive to navigate. It was particularly noted by users that the Laravel application was mobile responsive, which means that they could read their schedules and access the dashboard with ease on smaller smartphone screens without having to zoom or scroll horizontally. Moreover, the process of authentication had been considered highly safe and easy. The survey confirms that the SCBS greatly minimizes the cognitive load and time that users need to operate their campus activities by the fact that it replaces the archaic paper logbook with a ubiquitous digital dashboard.

5.6.2.2 Core Booking Credibility and Resolution of conflicts

The overall goal of the SCBS is to simplify the reservation pipeline and decisively remove the past problem of the classroom double-booking. Thus, the survey placed much importance on the confidence of the users on the automated scheduling logic of the system.

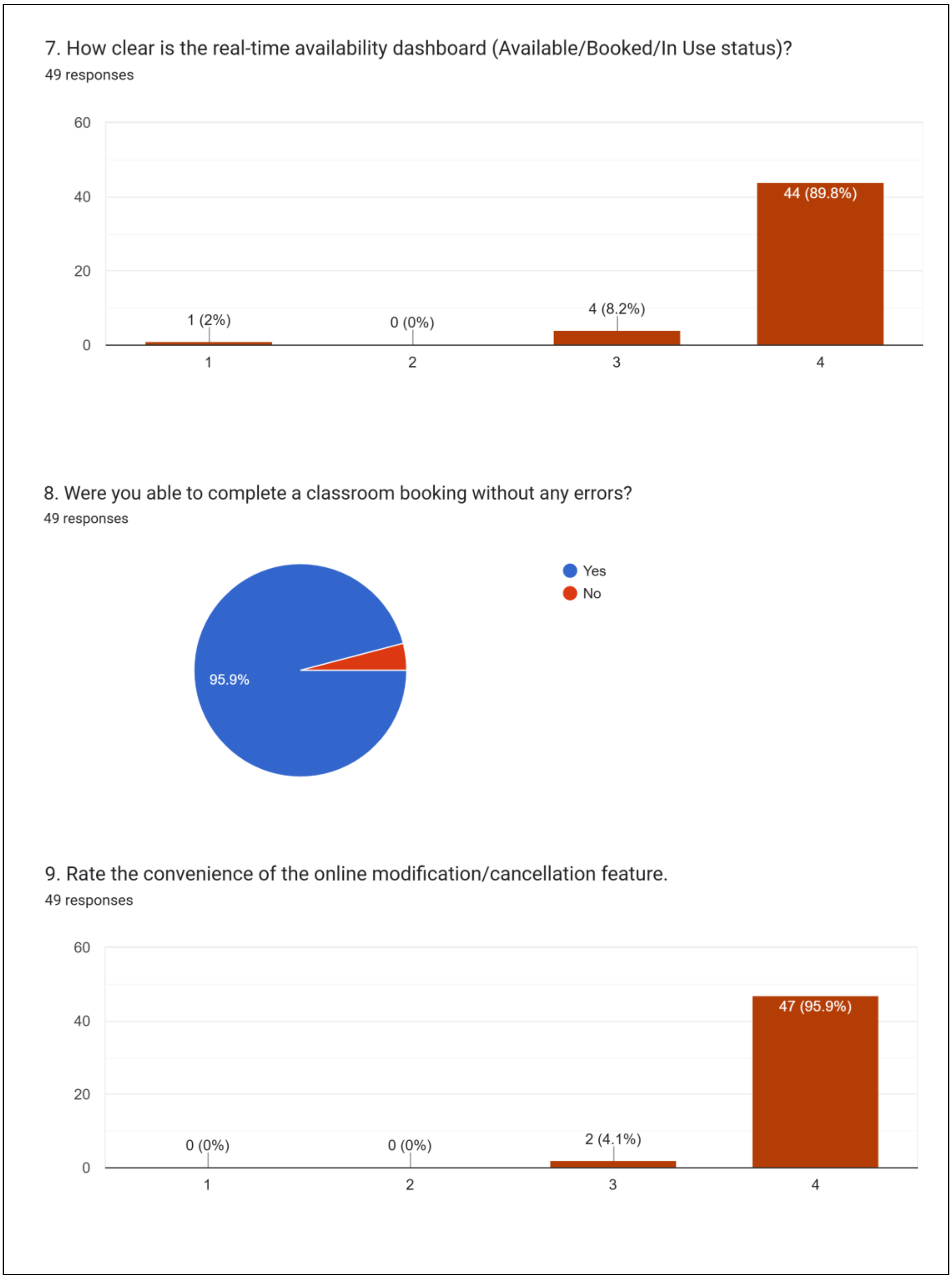
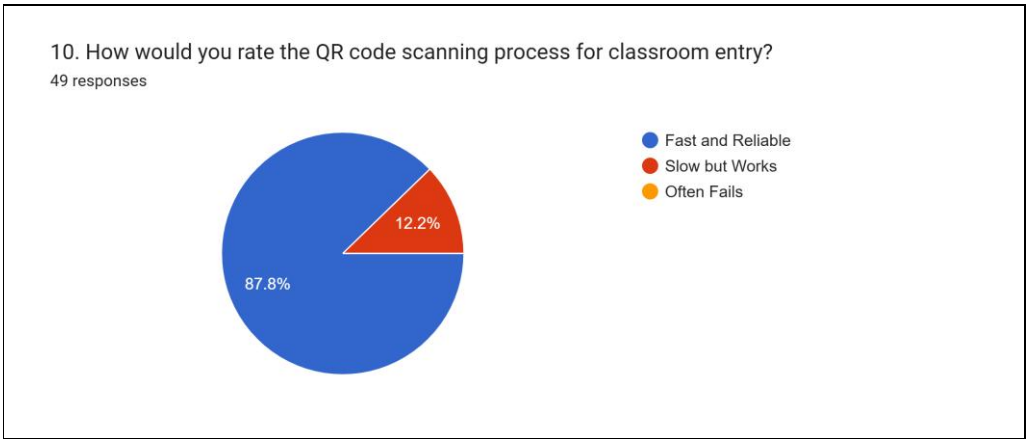


Figure 5.4: Booking Functionality Result (Ain,2026)

This section of the empirical research proves the overwhelmingly positive user approval of the automated conflict-prevention algorithms. In response to how effective the system was in intercepting and rejecting overlapping time slots, 90% of the respondents rated the feature as very effective, and that the feature has completely eliminated the frustrating aspect of arriving at a classroom and discovering it was occupied. Secondly, the new features of the "Cancel Booking" and "End Class Early" options were also rated among the most satisfied during the whole UAT phase. The respondents were complimentary about the system as it had given them control over their work schedules. They were happy with the one-click system that would automatically update the database when they had a class that completed on time. The users identified that real-time physical campus resource digital release would significantly enhance classroom availability to their peers.

5.6.2.3 IoT Interaction with Hardware and Physical Access

Since the SCBS is the interface between a digital software program and a physical campus setup, it was important to consider the comfort and success rates of users as well as their perceived security in dealing with the ESP32-CAM itself and the automated door system.



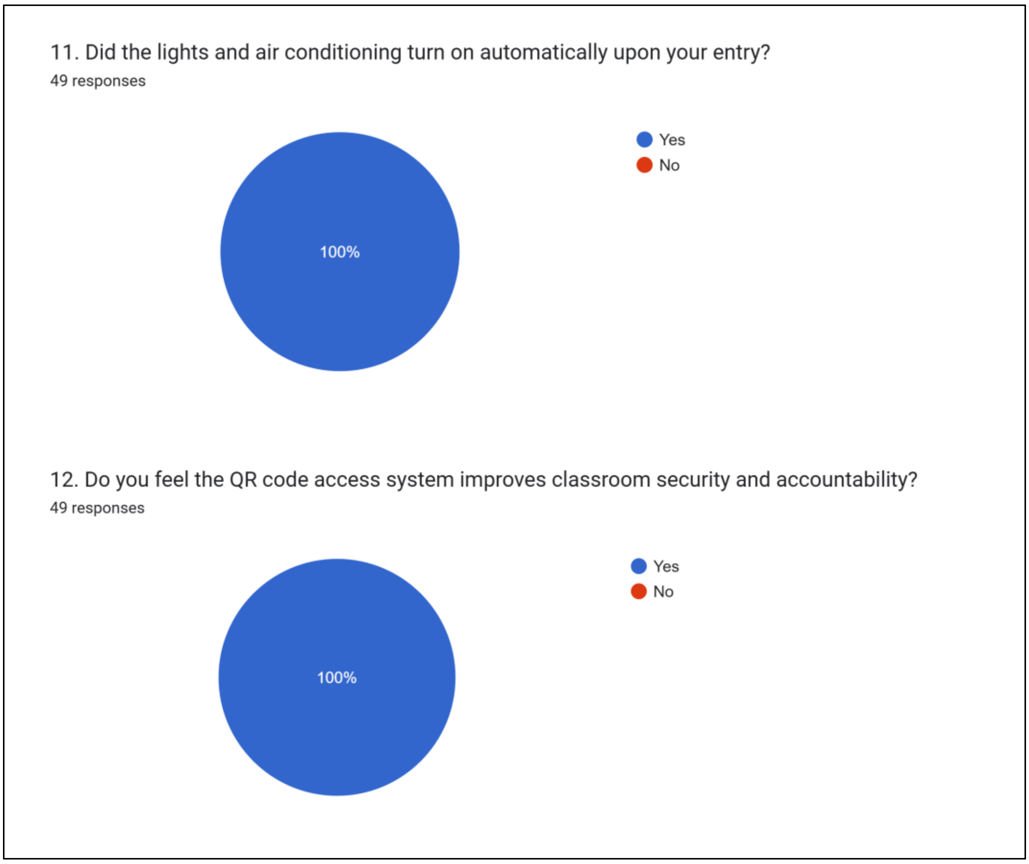


Figure 5.5: IoT & Access Control Result (Ain,2026)

The feedback information shows that the campus populace acclimatized at an extremely high rate to the new physical entry practice. The respondents found it mind-blowing that an impressive 87% of them found it far much better to access the auto-generated QR code on their mobile dashboard and hand it over to the physical scanner compared to the traditional way of accessing and handing over physical metal keys in an administrative office. The optical scanning was very reliable as reported by users. Although a small percentage of users (12%) reported that they had to short-term modify the brightness of their smartphone display, so that the camera could scan the token, the mechanical response of the smart lock was evaluated as exceptionally rapid and quick. Finally, the survey confirms that the substitution of physical keys that are easy to duplicate with encrypted, time specific digital tokens added a lot of perception of campus security amongst the users.

5.6.2.4 Overview of User Acceptance Testing Results.

In order to sum up all the results of the user questionnaire, the results of the feedback of the 50+ participants have been summed up into the table below. The statistics clearly indicate that there is a high end-user acceptance level in all the parameters considered.

Table 5.11: Summary of User Evaluation Survey Results

No	Evaluation Criteria (Survey Focus)	User Feedback Summary	Acceptance Status
1	Interface Accessibility: Ease of navigating the web dashboard on both mobile and desktop screens.	The vast majority found the UI intuitive and highly accessible, praising the modern mobile layout.	Highly Accepted
2	Booking Efficiency: The ease of submitting requests and the effectiveness of conflict-prevention.	Users felt highly confident that the system would securely lock their time slots and successfully prevent double-bookings.	Highly Accepted
3	Schedule Autonomy: The usefulness of the "Cancel" and "End Class Early" features.	Rated as the most practical scheduling feature, allowing users to efficiently manage their time and dynamically release unused rooms.	Highly Accepted
4	Hardware Reliability: The speed and convenience of the physical QR code scanner at the door.	Users overwhelmingly preferred the digital token system over traditional keys, noting the hardware response time was impressively fast.	Highly Accepted
5	Overall Preference: Preference for the SCBS over the legacy manual logbook system.	Near unanimous agreement among the 50+ participants that the SCBS should permanently replace current manual processes at UPTM.	Highly Accepted

5.7 Conclusion

The full implementation of Chapter 5 has completely tested the Smart Classroom Booking System (SCBS) using multi-level test validations. Unit testing was done to verify the correctness of individual code units and separated hardware components. Integration and System testing revealed that the digital Laravel architecture integrates perfectly with the localized IoT network, and the automation of classroom access and energy management has been achieved. Lastly, User Acceptance Testing (UAT) that was performed by the Academic Affairs client representative and over 50 general campus users also established that the system is not only technical but also extremely practical, easy to use and tackle the administrative bottlenecks at UPTM. The project is now ready to be deployed with the full testing and acceptance of the system by its target demographic.

6 CONCLUSION

6.1 Introduction

The latter chapter is a summary and critical reflection of the development, implementation, and evaluation of the UPTM IoT Smart Classroom Booking System (SCBS). The main aim of this chapter is to present the reflective analysis of the overall project lifecycle in terms of the effectiveness of the project schedule implementation and the possibility of the risk’s reduction during the software and hardware development stages. Moreover, this chapter appraises the degree to which the project has achieved its original goals as a testament to the functionality of the system and the effect it has on the real-life institutional inefficiencies. It also critically points out the limitations and constraints that were inherent in the current prototype that were revealed in the testing phase. Lastly, the future improvement strategic plan is offered in the form of recommendations in order to make sure that the SCBS will be able to further develop as a feasible, scalable, and long-term solution to the UPTM campus ecosystem.

6.2 Project Schedule

The project schedule that was used to achieve the successful deployment of the SCBS was strict and highly disciplined in that it was created with the specific purpose of balancing conventional web application development with the challenges of physical Internet of Things (IoT) hardware integration. The timeline adhered to by the project allowed ensuring that the critical path, such as backend Laravel API development, database schema migration, and physical sensor calibration, are carried out in a systematic manner without harming the academic deadline. The schedule was actively managed and dynamically changed to meet the unavoidable debugging stages that are necessary in the process of synchronizing the local web servers with ESP32 microcontrollers.

6.2.1 Work Breakdown Structure

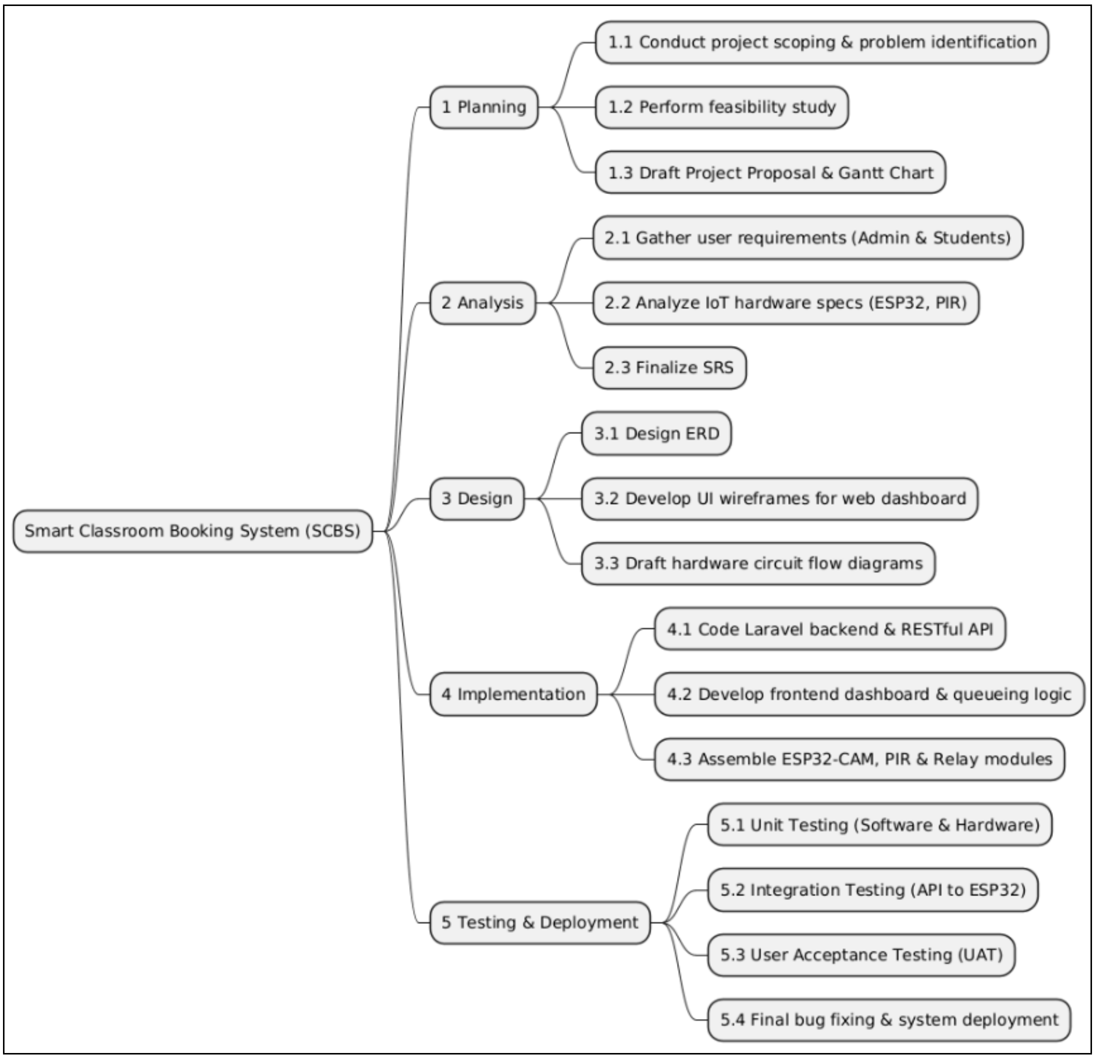


Figure 6.1: Work Breakdown Structure of SCBS (Ain,2026)

The Work Breakdown Structure was a crucial hierarchical breakdown of the project that was a complex integration of software and hardware into five manageable development phases: Planning, Analysis, Design, Implementation, and Testing. This was a step-by-step process which kept the physical hardware completed and ready to be tested precisely when the software endpoints were completed, to avoid any developmental bottlenecks.

The project lifecycle began with the first stage, Planning, where there was a clear foundation scope. The first step in this phase was to carry out a thorough scoping and identification of problems to capture in writing the real institutional problems of manual room booking and energy waste. Subsequently, a technical and operational feasibility analysis was conducted to determine whether the proposed IoT solution was feasible to be developed within the academic time frame. The formal project proposal was drafted, and a final Gantt chart was drawn which formally terminated the planning phase and gave the project the necessary direction in terms of the development timeline.

Moving into Phase 2, Analysis, the developmental emphasis changed to the accumulation of specific functional requirements. This entailed an active extraction of detailed user requirements of both the administrative and students to ensure that the end web dashboard would meet their workflow requirement. At the same time, the task of analyzing the hardware requirements was also taken very seriously, with a strong emphasis placed on the choice of the most efficient IoT devices, including the ESP32 microcontrollers and PIR motion sensors. Such analytical work eventually led to the completion of the ultimate document of the Software and Hardware Requirement Specifications (SRS) which was the ultimate plan of the entire system.

Phase 3, Design, involved taking the theoretical plans developed in the analysis stage and converting them into working technical formulations. The backend architecture of the software was clearly drawn using the modeling of Entity-Relationship Diagrams (ERD) to graphically represent the database schema and table relation relations. At the same time, User Interface (UI) wireframes were prepared to visualize the design and flow of the web dashboard. In the case of the physical IoT components, detailed circuit flow charts were developed to trace the exact electrical routes of electrical connections among the microcontrollers, sensors, actuators, and power supply and ensure safe and rational hardware integration before assembling.

Implementation, Phase 4, was the most important implementation phase of the project since both the digital and physical parts were being built. On the software front, the Laravel backend framework was developed to develop the main server logic, APIs, and database models. This was soon followed by the frontend dashboard which was developed to have the required automated queueing and booking logic. The physical hardware assembly occurred parallel with the software development. This is a very technical process that incorporated the wiring, soldering and casing of the ESP32-CAM, PIR motion sensors, and electromagnetic relay modules into a functional prototype model of a smart-lock.

The last stage, Testing, was fully devoted to the strict system testing, debugging, and quality control. This stage was initiated with isolated Unit Testing to confirm individual software functionality, including the production of QR codes, and individual hardware reactions. After validation of the isolated components, Integration Testing was done with the aim of ensuring smooth, real-time flow of data between the physical IoT sensors and the localized web API. This was followed by formal User Acceptance Testing (UAT) on a sample population of UPTM users to obtain practical feedback on the usability of the system and interface design. The phase and the project lifecycle in general, ended formally with the last system deployment, and this was the successful transfer of the SCBS out of the experimental prototype into a fully functioning model.

6.2.2 Gantt Chart

The Gantt chart served as the major time map and overall plan of the SCBS development, the beginning and the conclusion dates of each stage of the project were well-defined in two semesters of the academic year. To address the enormous complexity of implementing a web application with real-world IoT hardware, the schedule was planned in two macro-phases in total, Final Year Project 1 (FYP 1) in the second half of 2025 and Final Year Project 2 (FYP 2) in the first half of 2026. This comprehensive, bi-semester proposal allowed a gradual, step-by-step implementation of the system starting with the first idea and theoretical endorsement of the system to the actual physical implementation.

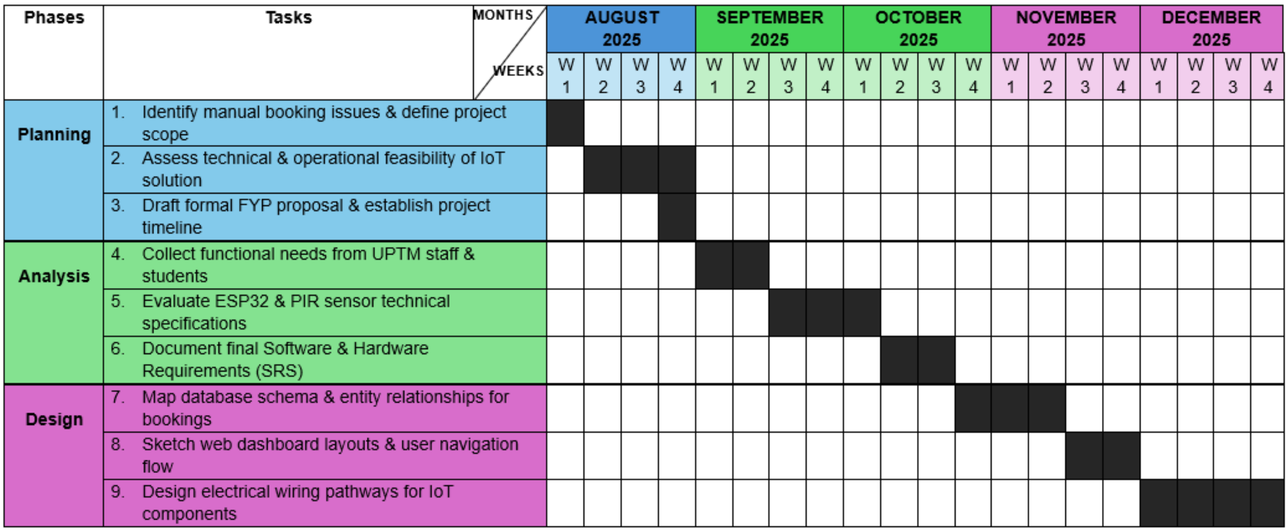


Figure 6.2: Gantt Chart of 2025 (Ain,2026)

The project started officially in September 2025 and the whole fall semester of the project was dedicated to the foundation research, planning of the feasibility and academic documentation. The plan was set to start the first two weeks of September with the intensive preparation and completion of the Final Year Project title so that the proposed system would be up to date with the needs of UPTM regarding the development of a computer application. In the very next week, starting the third week of September and ending the second week of October, the main emphasis was on preparing the comprehensive project proposal. Within the framework of this four-week block, the main problem statements, namely the inefficiency of manual room booking and local energy wastage, were officially written down, and the main system goals were defined.

It was by the third and fourth week of October that this proposal was formally presented to the academic panel to be critically reviewed and evaluated. The presentation was one of the key project milestones, as it made the scope of the IoT integration realistically attainable. According to the positive feedback of the panel on the issue of hardware limitation and software functionality, the initial three weeks of November were strictly planned as the time of implementing needed corrections. This refinement process was important, as it defined the approach and technical limits of the project. The 2025 plan formally ended with a highly-documented period that began during the last week of November and went on throughout the month of December. The foundational academic chapters, Chapter 1 (Introduction), Chapter 2 (Literature Review) and Chapter 3 (Methodology), were drafted, finalized and approved in a systematic manner in this five week block, before any active development had taken place.

Phases	Tasks	MONTHS / WEEKS	JANUARY 2026				FEBUARY 2026				MARCH 2026				APRIL 2026			
			W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Implementation	10. Develop server logic, APIs, & role-based access control		■	■	■													
	11. Implement automated queueing & real-time UI updates					■	■	■	■									
	12. Wire, solder, & assemble ESP32-CAM & smart lock model									■	■	■	■	■				
Testing	13. Test isolated software functions & individual hardware parts													■	■			
	14. Validate real-time data sync between IoT sensors & API															■		
	15. Gather usability feedback from a sample group of users															■		
	16. Resolve final bugs & transition prototype to live model																■	
Deployment	17. Deploy the system and final prototype to live local environment																	■
	18. Finalize system handover and user documentation																	■

Figure 6.3: Gantt Chart of 2026 (Ain,2026)

By the start of January 2026, the project was well underway in the active technical implementation of FYP 2, having successfully moved out of the theoretical model planning phase. The critical design stage was allocated the whole month of January. These four weeks block was necessary in terms of theory to practice where the structural architecture of the database (ERD), the wireframes of the user interface of Laravel, and the complex electrical circuit diagrams of the ESP32 microcontrollers were scrupulously drawn. Under these very specifications, the four weeks of February 2026 were totally dedicated to the physical engineering and development of the hardware. This included buying the parts, assembling the circuitry via soldering and attaching the ESP32-CAM, PIR motion sensors and relay modules into a working localized smart-lock prototype. The allocation of a month to hardware meant that the physical infrastructure was in good ground before software commands were introduced.

Afterwards, all the month of March was devoted to software development. In these four weeks, Laravel web framework was developed, Role-Based Access Control (RBAC) was added, and the automated booking queue was created. Importantly, this month also entailed the programming of the RESTful APIs that would enable the newly developed web dashboard to securely communicate with the hardware prototype created in February. Since the development cycle was almost coming to an end, the initial two weeks of April were set aside to test and debug the system rigorously. This was when unit testing of the QR code generation, integration testing of the network connectivity between the ESP32 and the server and User Acceptance Testing (UAT) of the final system to eliminate any system faults were achieved. The last two weeks of April were the final stages in the project lifecycle, and were completely dedicated to the final documentation, Chapter 4, 5, and 6, and the final creation of the complete, functional SCBS prototype to be used during the final academic presentation.

6.3 Risk Management

Risk management was a must-have in the way of navigating the complexities that were bound to come with an IoT-based development project. The combination of physical electronics with software projects brings in special points of failure that are not present in software-based projects only. The identification of the possible technical risks, such as the network latency and hardware overheating, was done early in the lifecycle. Through the proactive risk register, the project did not suffer any serious delays and the integrity of the physical hardware and the data that the Laravel application was working with was not compromised during the development cycle.

Table 6.1: Risk Assessment and Mitigation Strategy

Risk Category	Identified Risk	Impact Level	Mitigation Strategy Executed
Technical / API	Network latency between the ESP32 and the local server causing delayed door access for users.	High	Optimized the RESTful API endpoints to process extremely lightweight JSON data, ensuring a response time of under 2 seconds.
Hardware	Overheating or burnout of Relay modules due to continuous high-voltage switching for the AC/Lights.	High	Selected physically isolated 5V Relays with galvanic isolation and implemented a physical cooling gap in the 3D-printed enclosure.
Data Security	Loss of the database schema or pre-registered student records during continuous system testing.	Medium	Utilized Laravel’s built-in Migration versioning and maintained daily automated SQL database backups.
Schedule	Delays in the procurement and shipping of specific IoT components (ESP32-CAM, Servo motors).	Low	Ordered all physical hardware immediately during the Phase 1 Planning stage and utilized virtual circuit simulators (Wokwi) while waiting for delivery.

6.4 Achievement

The ultimate success of the SCBS is highly justified by the complete realization of its three main research and development goals as set out in Chapter 1. Every goal accomplished will be a step toward unravelling the inefficiencies in the manual, paper-based system and wastage of energy that was formerly seen in the UPTM classroom management system.

6.4.1 To Design a Web-Based Management System

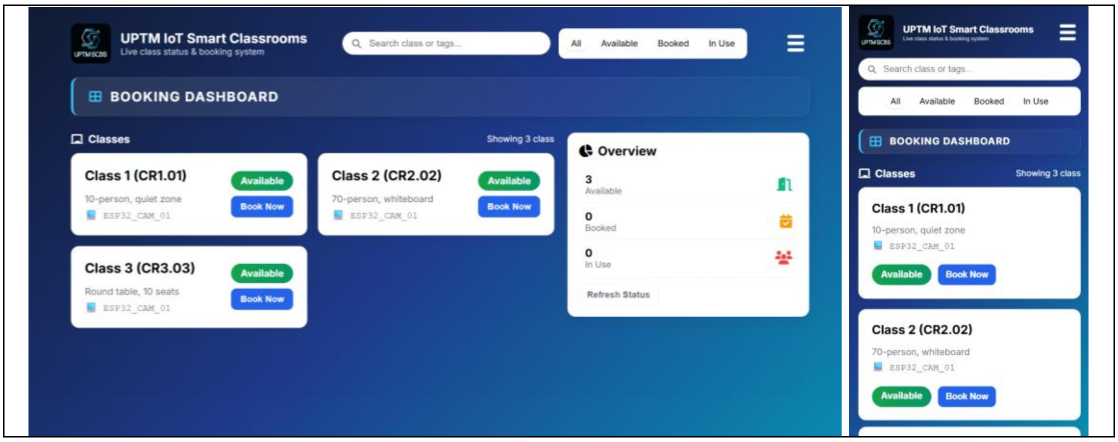


Figure 6.4: Web Based and Mobile Main Page (Ain,2026)

This was achieved by the design, development, and deployment of an advanced and highly responsive web architecture based on the Laravel PHP framework. The system offers a central digital platform through which administrative personnel can effortlessly control room inventories, track usage activity and process requests. At the same time, the students and lecturers will have an easy interface to view the room availability and post a booking request in real-time. The success of this goal is substantially demonstrated by the effective change of secure role-based access control (RBAC) and automated queuing logic that completely substitutes the former paper-based reservation and physical key-collection systems at UPTM.

6.4.2 To Develop an IoT-Integrated Access and Control System

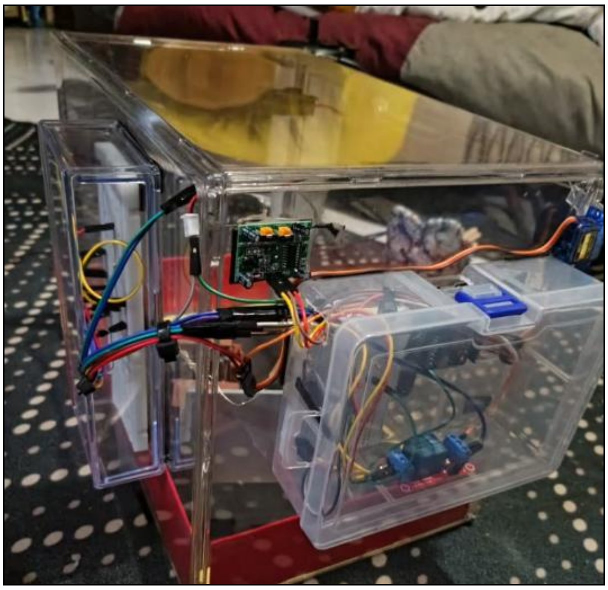


Figure 6.5: IoT Integration Implementation (Ain,2026)

The second goal was the extremely technical integration of the physical hardware actuators and sensors into the physical classroom space automation. This was done by developing a safe, dynamic QR-code scanning module using the ESP32-CAM, which was directly connected with motion-based power control using PIR sensors and electromagnetic relay modules. This success has been able to fill the gap between computer scheduling and campus presence. It is designed to provide access control to only authorized users during their designated time periods with the use of servo-motor smart locks, as well as automatically control the lighting and air conditioning conditions of the room due to the mere presence of active human occupancy.

6.4.3 To Evaluate System Accuracy and Energy Efficiency

Figure 6.6: Lights and AC Implementation (Ain,2026)

The last goal was met with careful, multi-level system testing and user acceptance testing (UAT). A comprehensive software testing revealed that the web platform is running with impeccable precision, namely, in its logical capacity to avoid the occurrence of scheduling conflicts, bar the overlapping of time slots and ensure the security of user data. At the same time, the hardware test revealed the energy efficiency of the system as it showed that it can automatically turn electrical power to utilities after fifteen minutes of detected room inactivity. The success of this objective demonstrates the practical value of the project in the magnitude of minimizing operational electrical expenses of UPTM and proactively enhancing the sustainability of the campus.

Table 6.2: Summary of Project Objectives and Achievements

No.	Project Objective	Achievement Status	Evidence of Completion
1	To design a web-based management system for classroom booking.	Achieved	Deployment of a fully functional Laravel dashboard featuring Role-Based Access Control (Admin/User) and conflict-prevention algorithms.
2	To develop an IoT-integrated access and control system.	Achieved	Successful physical integration of the ESP32-CAM for QR scanning and PIR sensors for real-time motion detection and door actuation.
3	To evaluate system accuracy and energy efficiency.	Achieved	System testing confirmed the prevention of double-bookings and reliably demonstrated the ability to cut utility power after 15 minutes of inactivity.

6.5 Constraint and Limitation

Although the SCBS is able to satisfy all its non-functional and functional requirements, the system cannot avoid a number of technical and environmental limitations that should be taken into consideration. These limitations are important to appreciate the extent of operations of the present prototype.

Table 6.3: System Constraints and Limitations

Limitation Type	Description of Constraint	Operational Impact on System
Power Supply Instability	The shared power source in the small-scale class model experiences consistent voltage	Causes microcontroller brownouts or servo stalls, resulting in the smart door lock failing

	drops during sudden electrical load spikes (when the servo motor actuates).	to fully open or close during a valid QR scan, requiring manual reboots.
Network Dependency	The ESP32-CAM requires a continuous, stable Wi-Fi connection to validate scanned QR codes against the central database.	Any local network outages or router fluctuations will completely disable the hardware's validation process, preventing legitimate user entry.
Deployment Scope	The current prototype database and web application are optimized solely for a localized server environment (XAMPP/Localhost).	Strictly restricts students and administrators from accessing the booking dashboard or generating QR tokens when they are connected to off-campus networks.

6.6 Future Work and Recommendation

In order to transform the SCBS to a better version than what it is at the moment and make it viable in the long term as a permanent campus installation, some strategic improvements are strongly suggested to be introduced in future versions. The recommendations deal with the existing constraints and give ways of scaling the technology.

Table 6.4: Proposed Future Enhancements

Component	Proposed Enhancement	Expected Institutional Benefit
Software Interface	Native Mobile Application Integration (iOS/Android).	Allows the system to send direct push notifications for booking approvals, schedule changes, and session expiration warnings.
Hardware / Sensors	Integration of Ultrasonic Sensors alongside the existing PIR sensors.	Drastically improves occupancy detection accuracy by combining thermal tracking with sound-wave tracking, eliminating stationary blind spots in large halls.
Cloud Infrastructure	Migration from local hosting to Cloud Servers (such as AWS, DigitalOcean).	Enables remote booking access for students from home and allows centralized multi-campus management for UPTM administrators.
Power Resilience	Localized Battery Backup and dedicated power supply for the IoT door modules.	Directly resolves the voltage drop limitation, ensuring the servo motors have sufficient current to actuate and remain secure during campus power failures.

6.7 Conclusion

To sum up, the UPTM IoT Smart Classroom Booking System (SCBS) is an extremely successful, very practical combination of Internet of Things technology and modern web development as an application. The project offers a highly scalable example of smart infrastructure in future smart campuses by directly tackling and resolving the most basic institutional inefficiencies of manual room booking systems and avoidable waste of electrical power. The codified, Agile-based development process, with its smooth transition between analytical requirement definition and strict physical hardware testing, will make the SCBS not just a functional academic prototype, but a secure, reliable and user-friendly administrative resource. Finally, this project can become a solid technical basis of future innovations in UPTM, which will conclusively demonstrate that the combination of software and hardware can play a crucial role in increasing the efficiency of the administration and the sustainability of the environmental situation in the institution.

Appendix A – Requirements Specification Document

This section outlines the complete functional and non-functional requirements, system constraints, and dependencies that guided the development of the Smart Classroom Booking System (SCBS). It also includes the raw data collection instruments utilized during the User Acceptance Testing (UAT) phase.

A.1 Functional Requirements

Functional requirements define the core behaviours, processes, and features that the SCBS must execute to operate successfully.

ID	Module	Requirement Description
FR-01	Authentication	The system must allow users to securely register, log in, and log out using encrypted credentials.
FR-02	Access Control	The system must enforce Role-Based Access Control (RBAC), restricting students from accessing the administrative dashboard.
FR-03	Room Management	The Administrator must be able to Create, Read, Update, and Delete (CRUD) classroom inventory records in the database.
FR-04	Booking Engine	Users must be able to select a room, date, and time slot to submit a booking request.
FR-05	Conflict Prevention	The system must actively cross-reference the database and reject any booking inputs that overlap with an existing approved session.
FR-06	Token Generation	Upon administrative approval, the system must autonomously generate a unique, time-sensitive QR code linked to the user's session.
FR-07	IoT Access Actuation	The ESP32-CAM must optically scan the user's QR code, validate it with the server, and actuate the servo motor to unlock the door.
FR-08	Energy Automation	The PIR sensor must monitor physical occupancy; upon 15 minutes of inactivity, the system must trigger the relay

		module to cut power to classroom utilities (lights/fan).
FR-09	Live Occupancy	The web dashboard must pull real-time data from the PIR sensors to display whether a classroom is currently "Occupied" or "Empty".
FR-10	Report Generation	The system must allow administrators to export historical booking and usage data as a formatted PDF document.

A.2 Non-Functional Requirements

Non-functional requirements define the quality attributes, performance metrics, and operational standards of the system.

ID	Quality Attribute	Requirement Description
NFR-01	Performance (Latency)	The physical IoT hardware (ESP32-CAM) must scan, verify the database, and unlock the door within a maximum threshold of 3 seconds.
NFR-02	Usability (Responsive)	The Laravel web interface must dynamically adjust its CSS layout to be fully readable and functional on both mobile smartphones and desktop PCs.
NFR-03	Security (Data Privacy)	All user passwords must be irreversibly hashed using the Bcrypt algorithm before being stored in the centralized SQL database.
NFR-04	Reliability (Network)	The IoT microcontrollers must be programmed to automatically attempt to reconnect to the Wi-Fi network in the event of a brief campus network dropout.

A.3 System Constraints

System constraints refer to constraints and boundaries that are experienced in the research, development and testing stages of the project.

- **Time Constraint:** The full lifecycle of the project, which covered software architecture, hardware soldering as well as UAT testing was ruthlessly forced within the length of the last year academic semester.

- **Hardware Scale Constraint:** The IoT smart-lock actuation was piloted and validated on a small, downsized prototype door because of budgetary constraints and campus property regulations, instead of being installed on large industrial deadbolts on real UPTM campus doors. Budgetary

- **Constraint:** The researcher funded all the IoT components (ESP32-CAM, relay modules, PIR sensors, and servo motors) himself, which did not allow him to afford expensive, enterprise-level optical scanners.

A.4 Assumptions and Dependencies

This part is a list of the outside influences that the SCBS depends on to work well in a realistic campus setting.

- **Network Dependency:** Localized IoT hardware is strictly reliant on stable, persistent 2.4GHz Wi-Fi connection to interact with the centralized Laravel server through API endpoints.

- **Power Supply Dependency:** ESP32-CAM and relay modules presuppose continuous and uninterrupted external power supply. Physical doors would have to be manually overridden with a key in case of a complete blackout of the campuses.

- **User Device Assumption:** It is assumed that every student and lecturer will have a functional smartphone whose screen is not broken such that the generated QR code can be easily seen on the optical scanner.

A.5 Research Survey Questionnaire

Research Survey: Developing a Smart Classroom Booking System for UPTM

This option provides more context about the problems and the proposed solution, which can help respondents give more thoughtful answers.

Description:

Project Title: UPTM IoT Smart Classroom Booking System (SCBS)
Researcher: Nurul Ain Shazwani Bt Ahmad Zaini (CT204 - Bachelor of Information Technology in Computer Application Development)

Introduction:
This research survey is conducted as part of my final year project to develop a Smart Classroom Booking System (SCBS) for UPTM. The current manual booking process often leads to issues like unavailable classrooms, energy wastage (lights/AC left on in empty rooms), and unauthorized usage.

Purpose of the Study:
This project aims to create an online, IoT-powered system that allows for real-time booking, automated energy control, and better monitoring of classroom usage. Your participation will help identify the key needs and preferences of the UPTM community to ensure the new system is effective and user-friendly.

Confidentiality & Time:
All responses will be kept **confidential** and used solely for academic purposes. This survey should take no longer than **7 minutes** to complete.

Your input is incredibly valuable. Thank you for helping to shape the future of classroom management at UPTM.

Indicates required question

Email *

Research Survey: Developing a Smart Classroom Booking System for UPTM

Please answer all the question below. Thank you !

1. What is your role in UPTM ? *

Mark only one oval.

- Student
- Lecturer

2. How long have you been studying/working at UPTM ? *

Mark only one oval.

- Less than a year
- More than a year

3. What is your primary method for booking a classroom at UPTM ? *

Mark only one oval.

- In-person at the administration counter (paper form)
- Through WhatsApp / Messaging platforms
- Through a lecturer or head of department
- I don't book classrooms. It's too difficult

4. Is the classroom always available whenever you want to book it ? *

Mark only one oval.

- Yes
- No

5. Does your classroom booking usually take a long time to be approved ? *

Mark only one oval.

- Yes
- No

6. When you enter the classroom, have you ever found the lights or air-conditioning already turned on even though nobody is inside ? *

Mark only one oval.

- Yes
- No

7. Have you ever used (or seen others using) a classroom without making a booking or for non-educational purposes ? *

Mark only one oval.

- Yes
- No

8. Do you agree if the classroom booking method is changed to an online system ? *

Mark only one oval.

	1	2	3	4	
Stro					Strongly Agree

9. Do you agree if the system shows classroom availability based on real-time updates (in used, available, booked)? *

Mark only one oval.

	1	2	3	4	
Stro					Strongly Agree

10. Do you agree if the system automatically turns lights and air-conditioning on/off based on motion detection whether someone is in the classroom or not ? *

Mark only one oval.

	1	2	3	4	
Stro	◯	◯	◯	◯	Strongly Agree

11. Do you agree that classrooms should be prioritized for those who booked in advance, rather than for walk-in users or those using the room for non-educational purposes? *

Mark only one oval.

	1	2	3	4	
Stro	◯	◯	◯	◯	Strongly Agree

12. Would you like the system to send you a booking confirmation through email or notification after you book a classroom? *

Mark only one oval.

◯ Yes
◯ No

13. Should the system automatically cancel your booking if you do not check in within 15 minutes after the scheduled time? *

Mark only one oval.

◯ Yes
◯ No

14. Would you like the system to allow you to modify or cancel your booking online (without going to the counter)? *

Mark only one oval.

- Yes
- No

15. Should the system show a live timetable of which classrooms are currently occupied, available, or reserved? *

Mark only one oval.

- Yes
- No

16. Do you want the system to keep a booking history (past and future bookings) that you can check anytime? *

Mark only one oval.

- Yes
- No

17. Should the booking system load quickly (less than 5 seconds when checking or booking a classroom)? *

Mark only one oval.

- Yes
- No

18. Do you think the system should require a secure login using your student/staff ID before booking a classroom? *

Mark only one oval.

- Yes
- No

19. Should the booking system be available 24/7 (not limited to office hours)? *

Mark only one oval.

- Yes
- No

20. Should the system be mobile-friendly and easy to use on both computers and smartphones? *

Mark only one oval.

- Yes
- No

21. Should the system protect privacy so that only authorized students, lecturers, and admins can view or manage bookings? *

Mark only one oval.

- Yes
- No

A.6 User Acceptance Testing (UAT) Questionnaire

Section 1: User Demographics

1. What is your role at UPTM? *

Mark only one oval.

- Student *Skip to question 4*
- Staff / Lecturer *Skip to question 4*
- Administrator *Skip to question 14*

2. Which faculty are you from? *

Mark only one oval.

- Faculty of Business & Accountancy (FABA)
- Faculty of Computing & Multimedia (FCOM)
- Faculty of Education, Social Sciences and Humanities (FESSH)
- Academic Affairs (AAS)

Section 2: System Usability & Interface Design

3. How easy was it to navigate the system? *

Mark only one oval.

	1	2	3	4	
Very	◯	◯	◯	◯	Very Easy

4. Does the system load quickly (within 5 seconds)? *

Mark only one oval.

- Yes
- No

5. Is the interface design visually appealing and professional? *

Mark only one oval.

	1	2	3	4	
Stro	◯	◯	◯	◯	Strongly Agree

6. Did the system work effectively on your mobile device? *

Mark only one oval.

◯ Yes

◯ No

Section 3: Booking Functionality

7. How clear is the real-time availability dashboard (Available/Booked/In Use status)? *

Mark only one oval.

	1	2	3	4	
Very	◯	◯	◯	◯	Very Clear

8. Were you able to complete a classroom booking without any errors? *

Mark only one oval.

◯ Yes

◯ No

9. Rate the convenience of the online modification/cancellation feature. *

Mark only one oval.

	1	2	3	4	
Poo	◯	◯	◯	◯	Excellent

Section 4: IoT & Access Control

1. 10. How would you rate the QR code scanning process for classroom entry? *

Mark only one oval.

- Fast and Reliable
- Slow but Works
- Often Fails

2. 11. Did the lights and air conditioning turn on automatically upon your entry? *

Mark only one oval.

- Yes
- No

3. 12. Do you feel the QR code access system improves classroom security and accountability?

Mark only one oval.

- Yes *Skip to question 17*
- No *Skip to question 17*

Section 5: Administrative Experience (Admins Only)

4. 13. Rate the efficiency of the booking request approval process.

Mark only one oval.

	1	2	3	4	
Poor	◯	◯	◯	◯	Excellent

14. Is the "Live Occupancy" table useful for monitoring unauthorized usage?

Mark only one oval.

- ◯ Yes
- ◯ No

15. How useful are the generated reports?

Mark only one oval.

	1	2	3	4	
Not	◯	◯	◯	◯	Very Useful

Section 6: Overall Satisfaction & Recommendations

16. Compared to the previous manual system, how much time does SCBS save you? *

Mark only one oval.

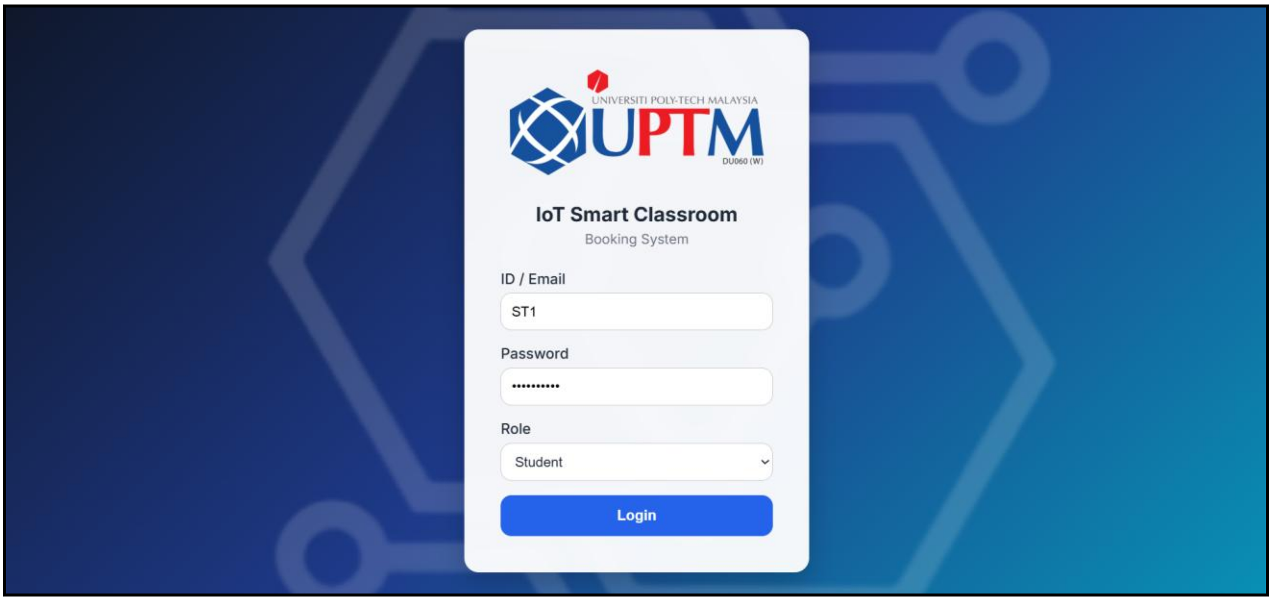
	1	2	3	4	
No c	◯	◯	◯	◯	Significant time saved

Appendix B – User Manual

This is an official User Manual of the Smart Classroom Booking System (SCBS). It includes step-by-step instructions of how to use the web application and control classroom bookings, as well as how to communicate with the physical IoT smart locks. The manual is categorized into two parts in accordance to the user roles, General Users (Students/Lecturers) and System Administrators (Academic Affairs).

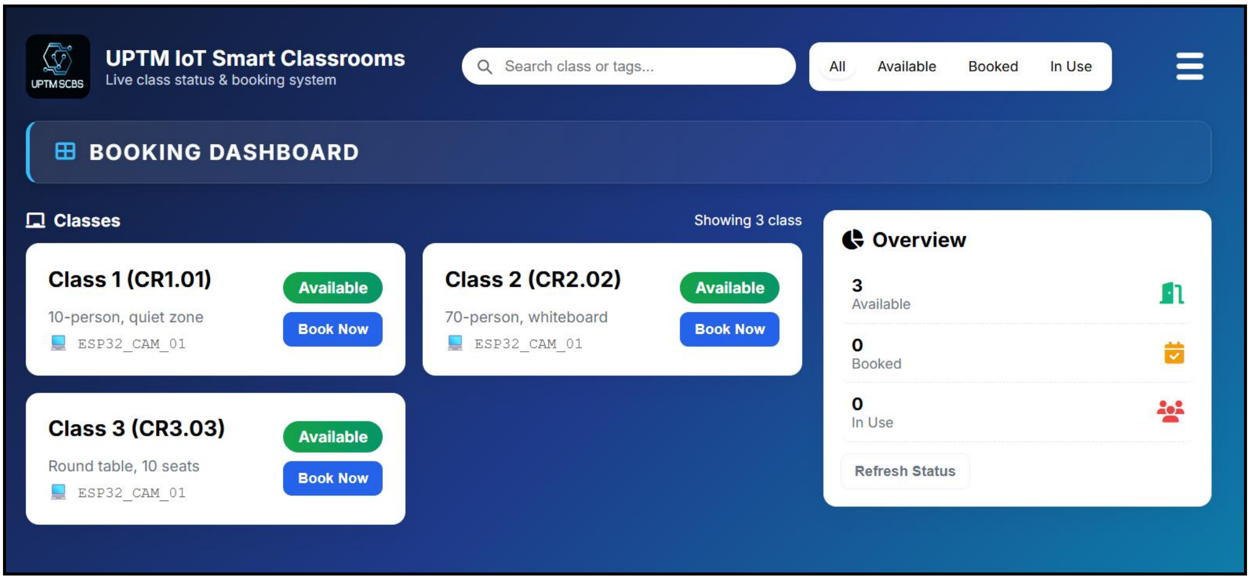
B.1 General User Guide (Students & Lecturers)

System Access and Login

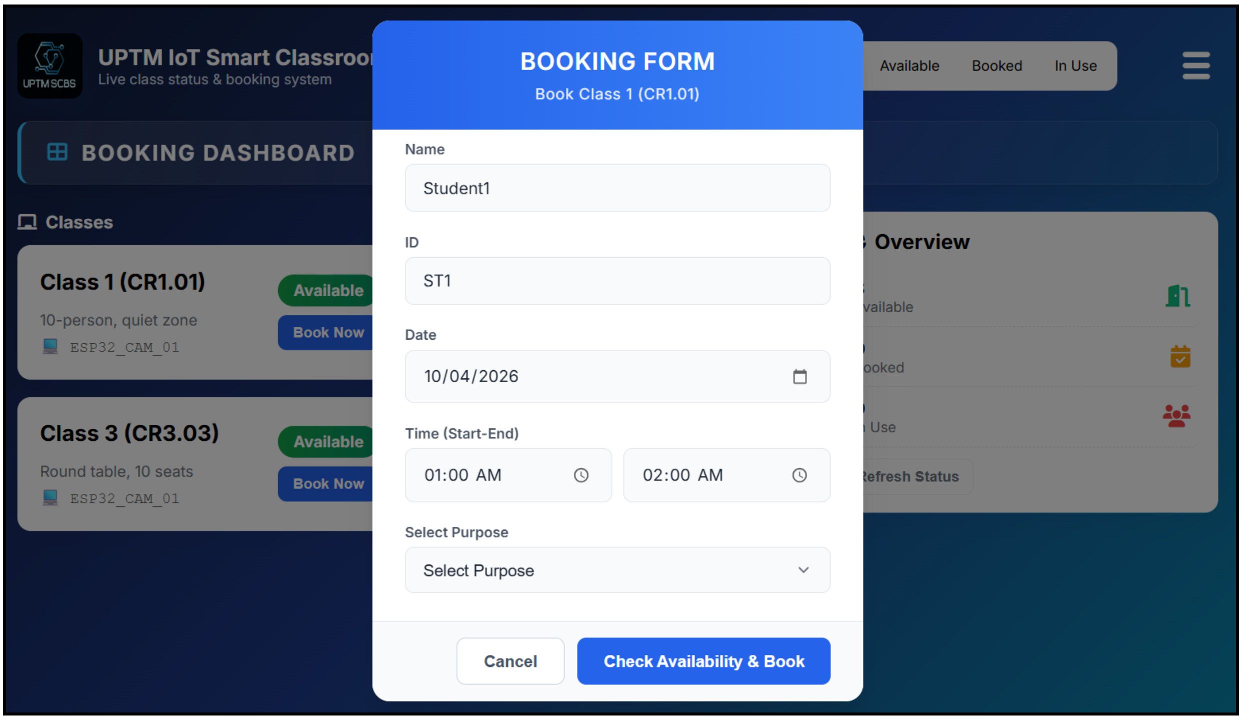


1. Navigate to the SCBS web portal via your smartphone or desktop browser.
2. Enter your official UPTM Student/Staff Email and your assigned Password on the secure login page. For demo:
   - ID: ST1 | Pass: student123
   - ID: LC1 | Pass: lecturer123
3. Pick your role (student or lecturer)
4. Click the "Login" button.
5. The system will authenticate your credentials and securely route you to your personal Booking Dashboard.

Submitting a Classroom Booking & Retrieving the QR Access Token

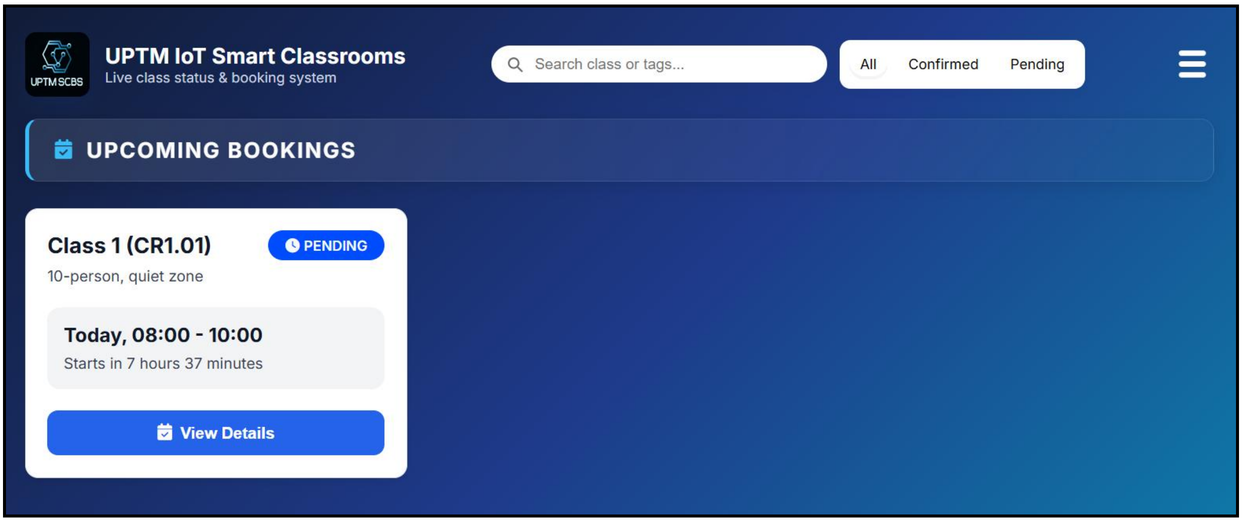


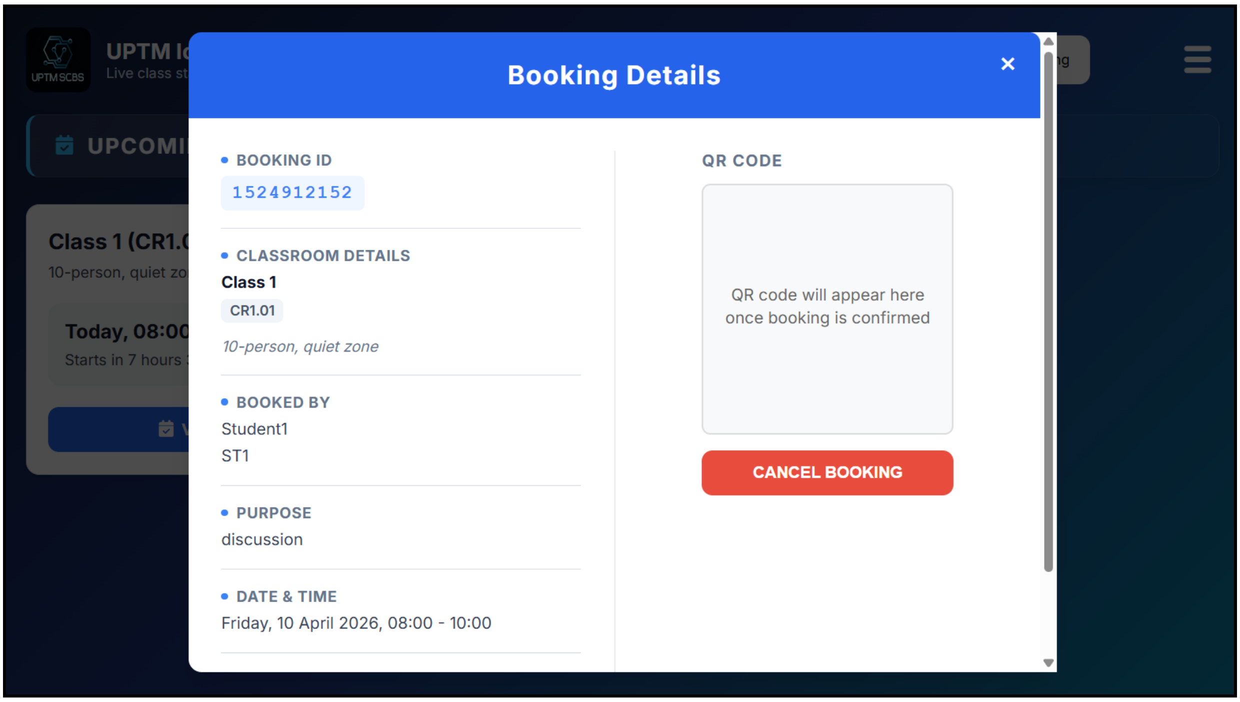
1. After the login verification, the Booking Dashboard will be displayed
2. Browse the list of available UPTM classrooms and select your desired room.



3. Choose the Date, Start Time, End Time and Purpose for your session.
4. Click "Check Availability & Book".

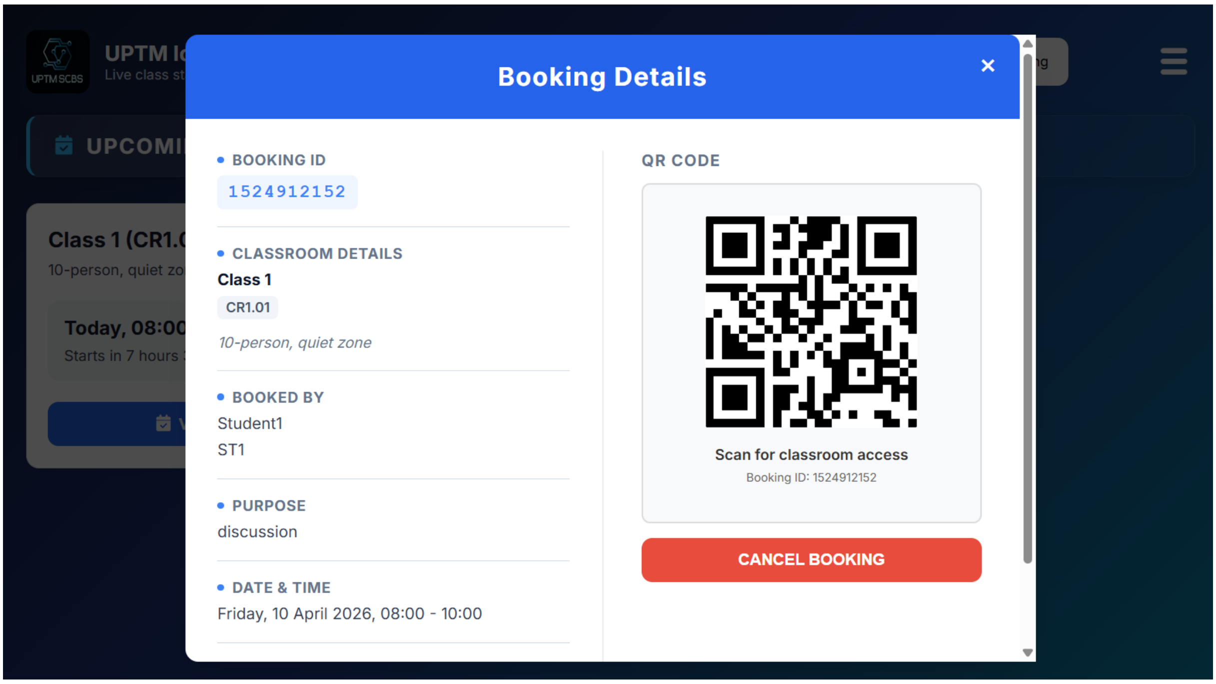
   Note: If your selected time slot overlaps with another user's session, the system will display a "Time Slot Unavailable" warning. You must select a different time.





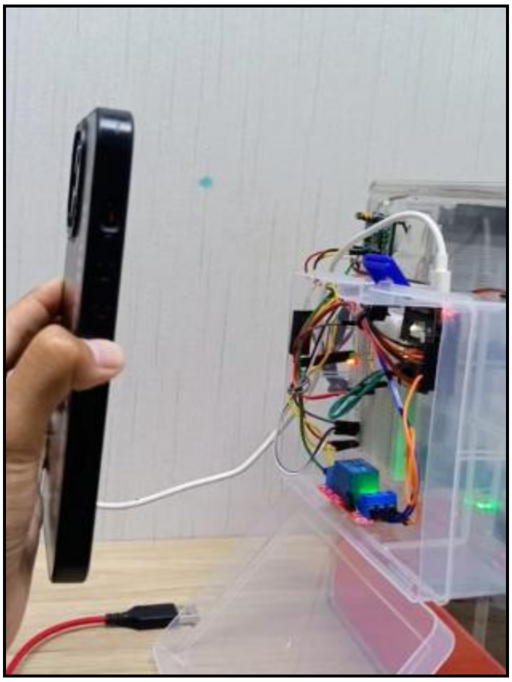
5. Your booking will now appear in your Upcoming page, awaiting Administrator approval.

   Note: You can go to other pages by clicking the menu icon on the top right corner of the page and this is where you can cancel your booking before scanning on the door.



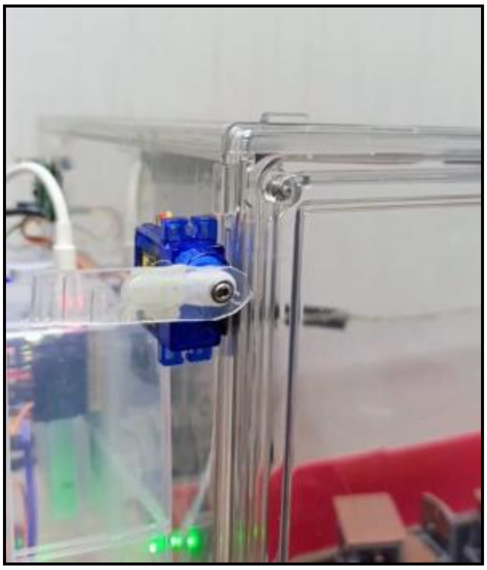
6. Once Academic Affairs approves your booking, a unique auto-generated QR Code will show up on the provided space. This is your digital key for the physical classroom.

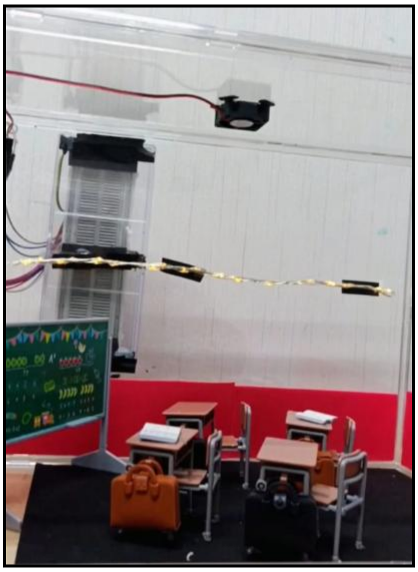
Scanning the QR Code at the Classroom Door



1. Arrive at the physical classroom during your approved time slot.
2. Open the QR code on your mobile phone screen. Turn your screen brightness up to maximum for the best results.

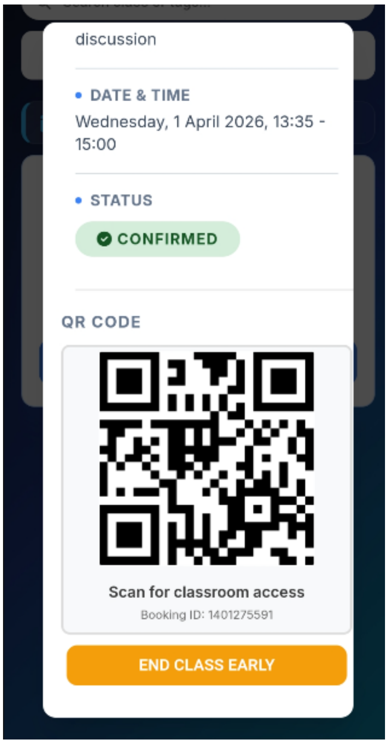
3. Hold your phone screen approximately 5 to 10 centimeters in front of the classroom's ESP32-CAM lens.





4. Once the camera flashes and reads the code, the system will verify your session. You will hear the servo motor unlock the door, allowing you to enter. The light and fan will be automatically on motion sensor.

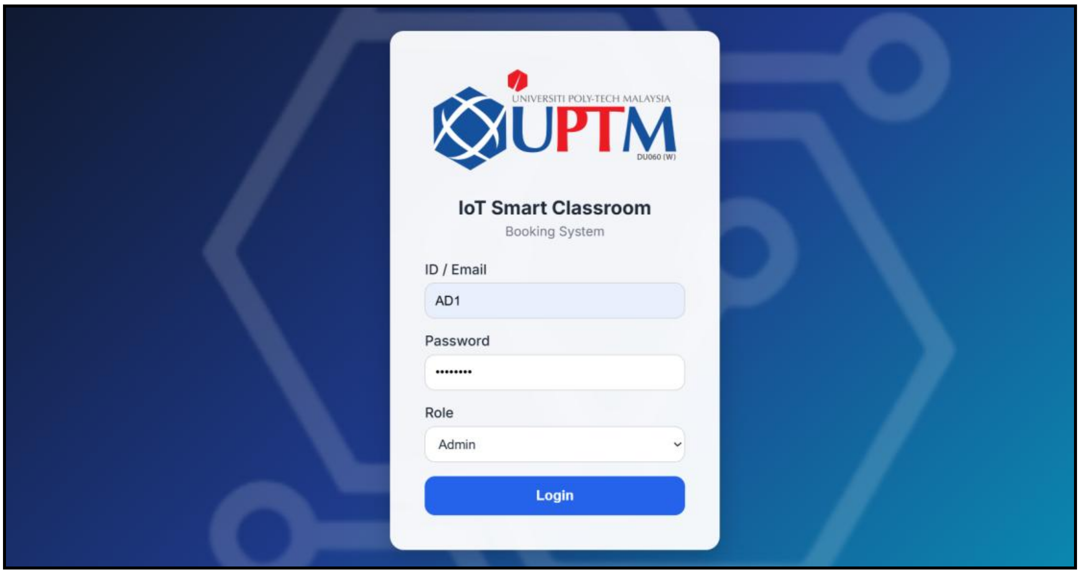
Managing a Session (Cancel & End Early)



If your class finishes before the scheduled end time, open your active booking and click "End Class Early" at where the previous “Cancel Booking” button. The system will update the database, lock the door, and cut the utility power to save energy.

B.2 Administrator Guide (Academic Affairs / Dean)

Admin Login and Dashboard Overview

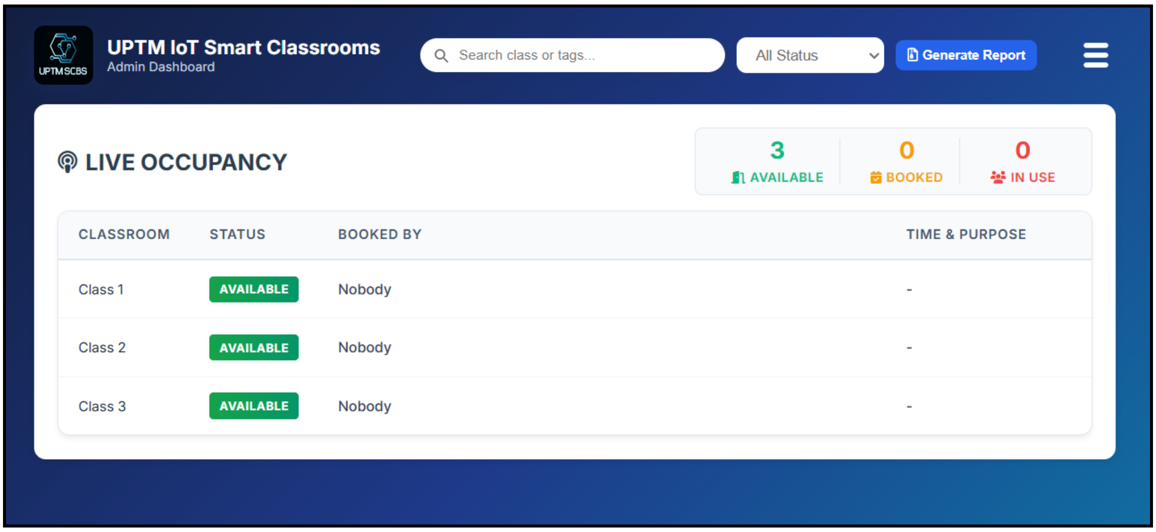


1. Navigate to the SCBS login page.

2. Enter your registered Administrative Email or ID, Password.and admin role
   - ➢ ID: AD1
   - ➢ Password: admin123

3. The system will recognize your administrative credentials and route you directly to the Admin Dashboard, granting you access to system-wide controls.

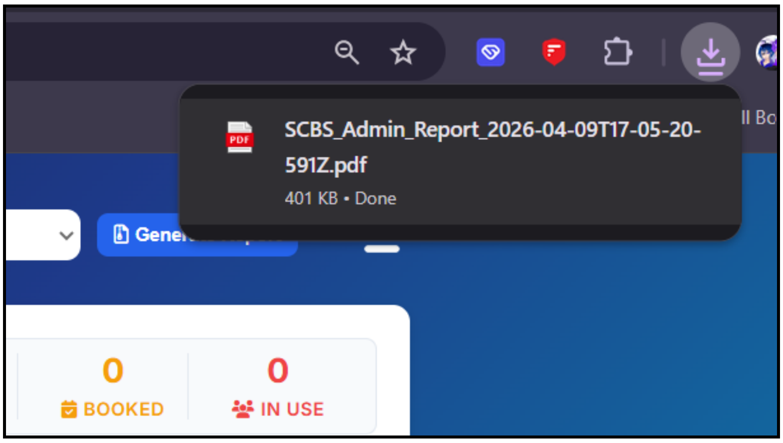
Live Occupancy Monitoring

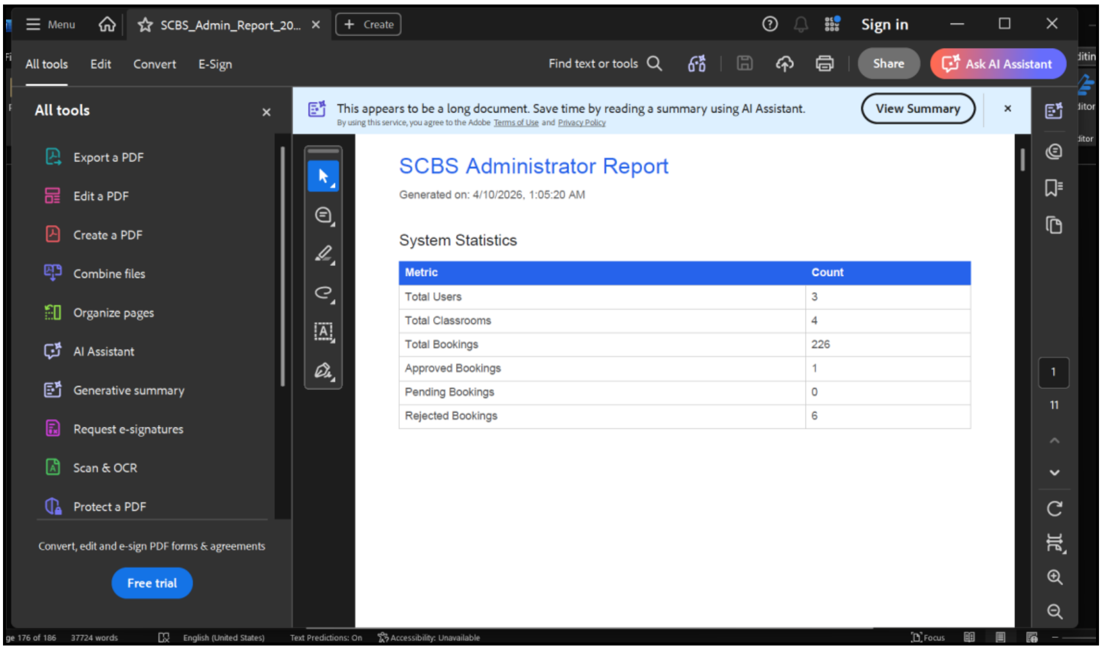


1. To monitor the physical security and usage of the campus, navigate to the "Live Occupancy" tab.

2. This page pulls live API data directly from the PIR sensors inside the classrooms.
3. The dashboard will dynamically display either "In Use" (movement detected), "Available" (no movement) for each integrated room or “Booked” for any confirmed booking and waiting to be entered.

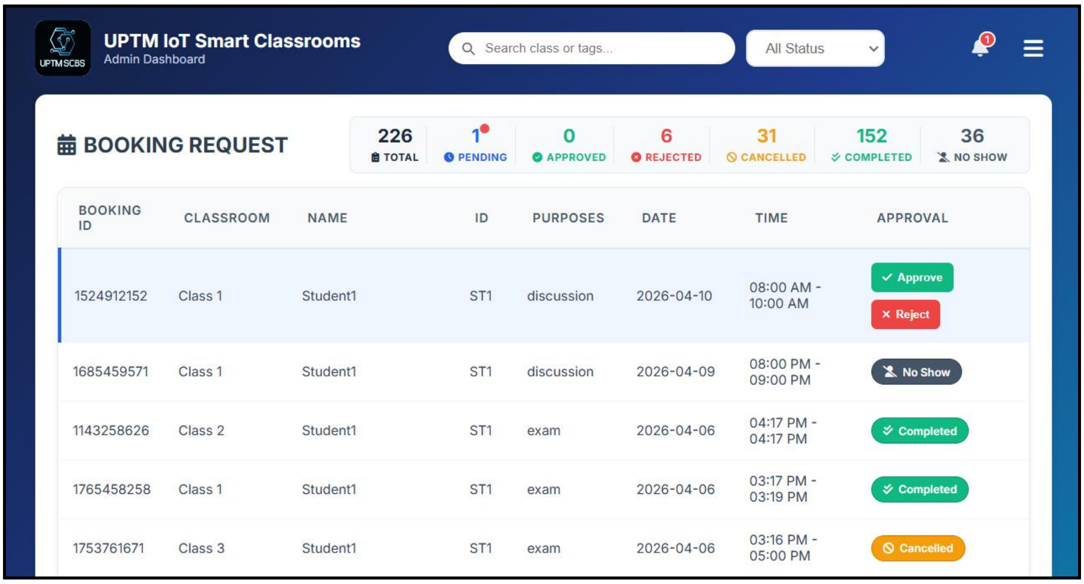
Generating Reports





1. For institutional auditing or end-of-semester reviews, navigate to the **"Live Occupancy"** module.
2. Click the **"Generate Report"** button on the top.
3. The system will compile all historical booking data, user statistics, and room utilization rates, and automatically download a formatted .pdf file to your computer.

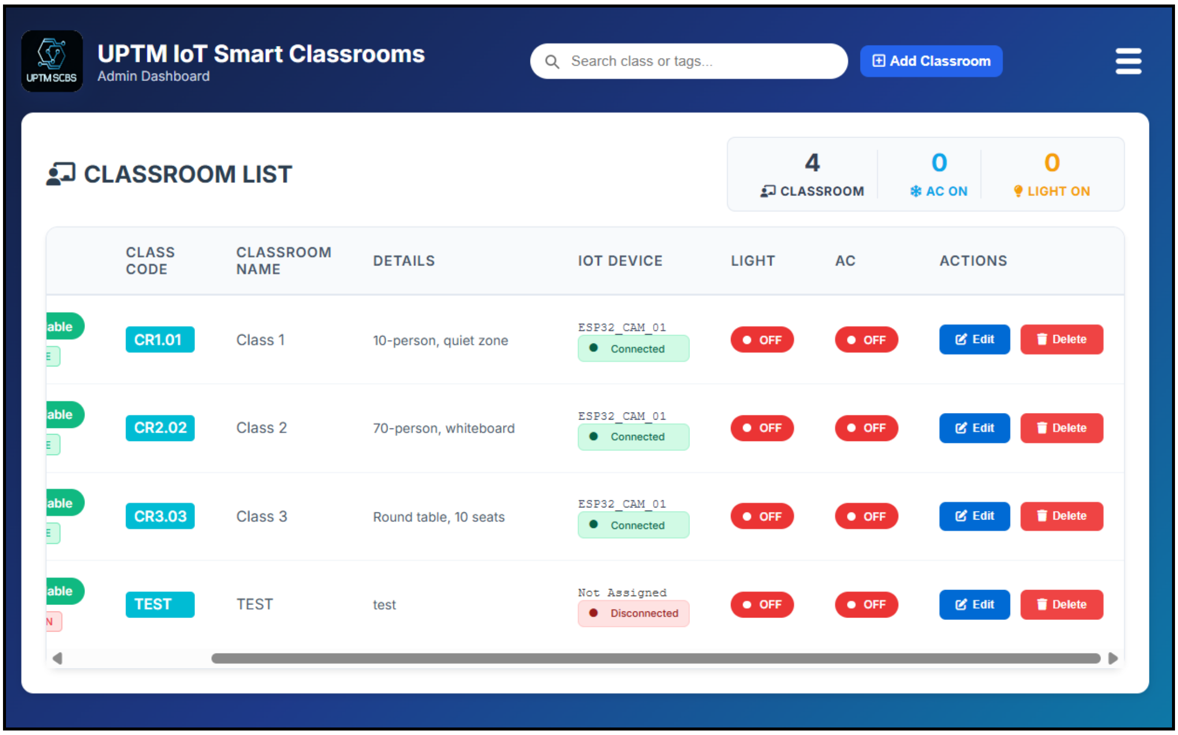
Approving Bookings & Triggering QR Codes



1. Navigate to the "Booking Request" tab to view all pending student/lecturer requests.
2. Review the requested room and time slot.
3. Click the "Approve" or “Reject” button.

   If approved, the system will automatically generate an encrypted QR code in the background and send it to the user’s dashboard. The status will change to "Approved".

Classroom Inventory Management (CRUD)



1. To manage campus facilities, navigate to the "Classroom List" tab.
2. To Add: Click "Add New Room", enter the Room Name, Block, and Maximum Capacity, then click Save.
3. To Edit/Delete: Locate an existing room on the list and click "Edit" to update its details, or "Delete" to permanently remove it from the system.

B.3 Project Source Code & Demonstration Video

Git Hub: https://github.com/Ain-Shazwani/FINAL-FYP-SCBS

Demo Video: https://youtu.be/7nxs1qAY2Yg

Appendix C – Turnitin Result

32% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

266 AI-generated only 32%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

REPORT SCBS (FULL)

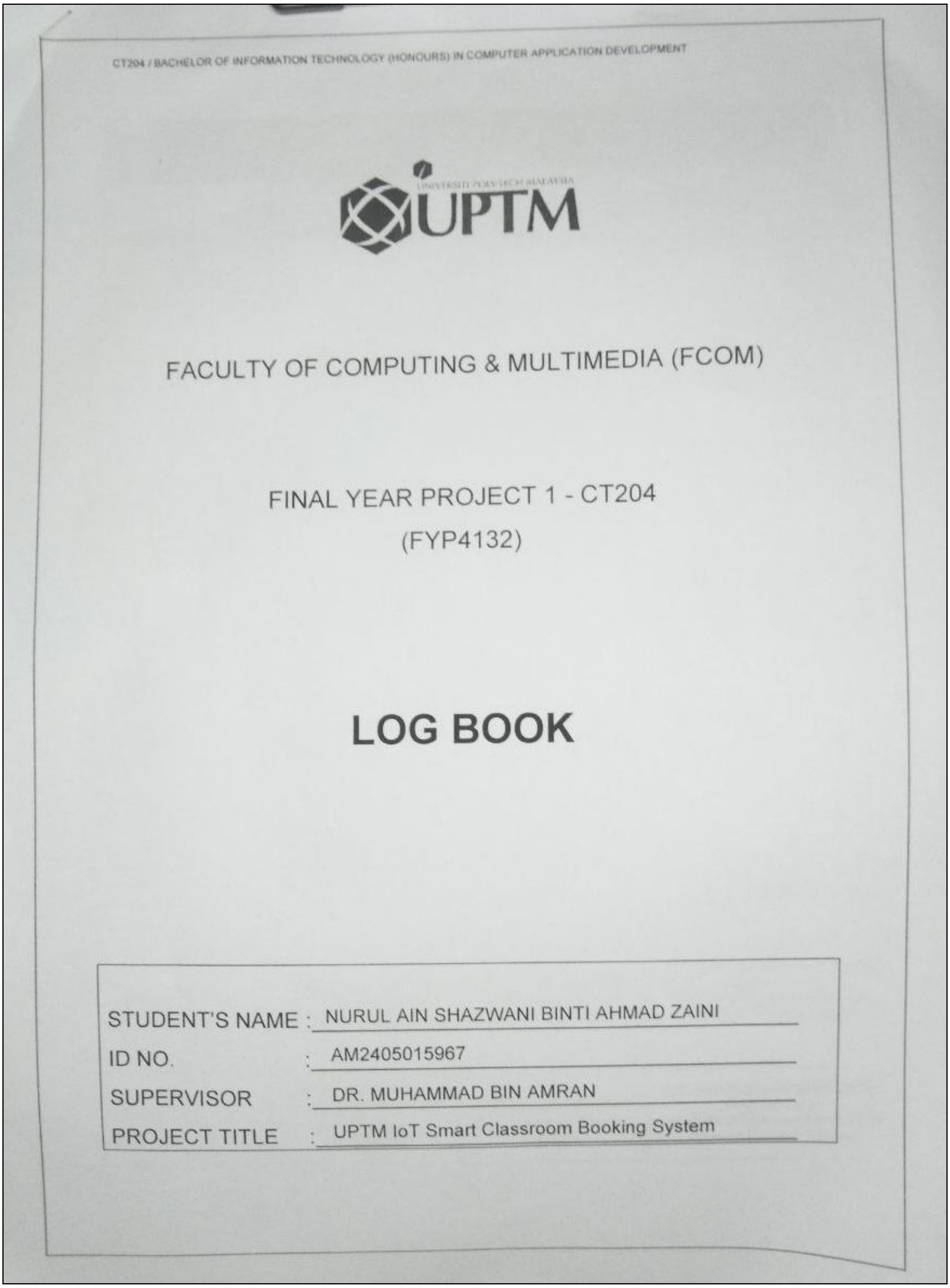
ORIGINALITY REPORT

4%
SIMILARITY INDEX

PRIMARY SOURCES

1	core.ac.uk Internet	123 words	< 1%
2	www.coursehero.com Internet	72 words	< 1%
3	www.techscience.com Internet	71 words	< 1%
4	ouci.dntb.gov.ua Internet	64 words	< 1%
5	umpir.ump.edu.my Internet	51 words	< 1%
6	odr.chalmers.se Internet	41 words	< 1%
7	acikerisim.sakarya.edu.tr Internet	37 words	< 1%
8	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publications	35 words	< 1%
9	journal.ilmudata.co.id Internet	32 words	< 1%
10	Faria, Nelson Filipe Miranda. "Cancer Detec - Lung Cancer Diagnosis Support System", Instituto Politecnico do Cavado e do Ave (Portugal), 2024 ProQuest	26 words	< 1%

Appendix D – Log Book


CT204 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLICATION DEVELOPMENT

UNIVERSITI POLY-TECH MALAYSIA

UPTM

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

FINAL YEAR PROJECT 1 - CT204

(FYP4132)

LOG BOOK

STUDENT'S NAME : NURUL AIN SHAZWANI BINTI AHMAD ZAINI

ID NO. : AM2405015967

SUPERVISOR : DR. MUHAMMAD BIN AMRAN

PROJECT TITLE : UPTM IoT Smart Classroom Booking System

CT204 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLICATION DEVELOPMENT

Date/ Week		Agenda	Next Agenda	Signature (Supervisor)
12/8/2025	1	Identify potential supervisors	Finalize research topic and title with supervisor	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
14/8/2025	2	Discuss research topic and title with supervisor	Prepare proposal (Problem Statement & Objectives)	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
21/8/2025	3	Proposal consultation with supervisor	Revise proposal (Problem Statement & Objectives)	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
28/8/2025	4	Consultation on revised proposal with supervisor & submission of proposal	Prepare questionnaires for users	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
4/9/2025	5	Consultation on questionnaires & feedback from proposal submission	Prepare system planning (layout, framework, flow)	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
11/9/2025	6	Consultation on system planning (layout, framework, flow)	Prepare detailed report.	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
18/9/2025	7	Consultation on report draft (Literature Review and Comparative Analysis table)	Revise comparative analysis table and draft system components diagram.	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
25/9/2025	8	Consultation on revised comparative table and system components requirements	Finalize system components diagram and begin database design	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

CT204 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLICATION DEVELOPMENT

1/10/2025	9	Consultation on initial database structure design for development phases	Finalize database schema and draft Topic 3 (Methodology) of the report	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
9/10/2025,	10	Consultation on Topic 3 (Methodology) of the report	Refine methodology phases and draft system use cases	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
14/10/2025	11	Consultation on the system use case diagram	Refine use case flows (for Student and Lecturer roles) and draft flowchart	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
23/10/2025	12	Consultation on revised use cases and flowchart diagram	Finalize flowchart, complete report (Topics 1-3), and prepare presentation slides	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
30/10/2025	13	Consultation on full report draft and presentation slides	Incorporate final feedback and rehearse presentation	DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
10/11/2025	14	Presentation		DR. MUHAMMAD BIN AMRAN Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

CT204 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN COMPUTER APPLICATION DEVELOPMENT

UPTM

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

FINAL YEAR PROJECT 2 - CT204
(FYP4144)

LOG BOOK

STUDENT'S NAME	:	NURUL AIN SHAZWANI BINTI AHMAD ZAINI
ID NO.	:	AM2405015967
SUPERVISOR	:	NORFAZLINA BINTI JOHAR
PROJECT TITLE	:	UPTM IoT Smart Classroom Booking System

Date/ Week		Agenda	Next Agenda	Signature (Supervisor)
30/12/25	1	Set up laravel environment & mysql DB	Develop web UI and Role Based Access Control	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
6/1/26	2	Coded the booking interface & conflict prevention logic	Begin IoT hardware set up	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
13/1/26	3	Programmed ESP32-CAM to scan QR codes & connect API	Add Servo Motor & PIR sensor components.	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
20/1/26	4	Finished hardware assembly. Programmed 15 min retg relay power cut	Integrate hardware with the web server	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
27/1/26	5	Conducted full system integration & fixed network bugs	Draft Chapter 4 documentation & diagrams	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
3/2/26	6	Finalized Chapter 4 write-up with supervisor	Draft Test Cases	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
10/2/26	7	Executed internal system testing (Admin, User & IoT)	Conduct client UAT with Academic Affairs (Dean)	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
17/2/26	8	Completed Client UAT. Gathered feedback on Admin Dashboard	Conduct UAT with general campus population	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia

24/2/26	9	Conducted User UAT with 50+ students & lecturers	Analyze the collected Google Form data	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
3/3/26	10	Analyzed UAT survey & ~~data~~ drafted chapter 5 graphs	Finalize the Chapter 5 write up	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
13 ~~10~~/3/26	11	Reviewed & corrected chapter 5 with SV	~~Dr~~ Draft chapter 6	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
20/3/26	12	Drafted chapter 6, finalized all appendices	Correct the appendices & turnitin check	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
30/~~2~~3/26	13	Corrected appendices & turnitin report	~~Prepr~~ Preparation for presentation & prototype	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
2/4/26	14	Presentation & Demonstration	minor formatting corrections.	NORFAZLINA BINTI JOHAR Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia

References

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly, 25*(1), 107–136. https://doi.org/10.2307/3250961

Ali, M., Zraqou, J., Maaita, A. A., & Alkhadour, W. (2022). Towards a smart campus: Smart classroom management. *International Journal of Advanced Research in Computer and Communication Engineering, 11*(6), 22–27. https://www.academia.edu/download/102099275/ijarcce.2022.pdf

Azizi, S., Nair, G., Rabiee, R., & Olofsson, T. (2020). Application of Internet of Things in academic buildings for space use efficiency using occupancy and booking data. *Building and Environment, 186*, 107355. https://doi.org/10.1016/j.buildenv.2020.107355

Bagitzhanov, B., & Myrzadiyar, U. (2023). *Booking of classrooms in KAZGUU*. Repository.kazguu.kz. http://repository.kazguu.kz/handle/123456789/1682

Bootstrap. (n.d.). *Bootstrap: The most popular HTML, CSS, and JS library in the world*. Retrieved November 25, 2024, from https://getbootstrap.com

Chen, L., & Ahn, J. (2021). Intelligent occupancy sensing for commercial buildings: A review and meta-analysis. *Sustainable Cities and Society, 71*, 102971. https://doi.org/10.1016/j.scs.2021.102971

Classroombookings. (2024). *Classroombookings: Open-source room booking system*. Retrieved November 25, 2024, from https://www.classroombookings.com

Garcia, M., & Lee, S. (2022). Beyond the smartboard: Re-defining the technological scope of the modern classroom. *Journal of Educational Technology Systems, 50*(3), 345-362. https://doi.org/10.1177/00472395221079231

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

Hassan, R., & Salauddin, S. (2020). *Implementation of IoT based smart classroom: Room occupancy estimation, automatic seat allocation and environment monitoring* (ID: 14321012). EasyChair.

Inflectra Corporation. (2024). *Hybrid agile methodology: Spiral, V-Model & more*. https://share.google/Tmcoqgyu5Jyyd6pwj

Lee, H., & Choi, K. (2019). The impact of smart classroom technologies on administrative efficiency in higher education. *International Journal of Educational Management, 33*(5), 1020-1035. https://doi.org/10.1108/IJEM-08-2018-0275

Marcotte, E. (2011). *Responsive web design*. A Book Apart. https://abookapart.com/products/responsive-web-design

Marikyan, D., Papagiannidis, S., & Alamanos, E. (2019). A systematic review of the smart home literature: A user perspective. *Technological Forecasting and Social Change, 138*, 139–154. https://doi.org/10.1016/j.techfore.2018.08.015

Mohammed, A., Singh, P., & Zhao, Y. (2020). Integrating sustainability into educational resource management: The next frontier for campus ERP systems. *Computers & Education, 151*, 103860. https://doi.org/10.1016/j.compedu.2020.103860

Noor, T. H., Atlam, E.-S., Almars, A. M., Noor, A., & Malki, A. S. (2023). An IoT-based energy conservation smart classroom system. *Intelligent Automation & Soft Computing, 35*(3), 3785–3799. https://www.techscience.com/iasc/v35n3/49426

OWASP Foundation. (2013). *SQL Injection*. OWASP. https://owasp.org/www-community/attacks/SQL_Injection

Owusu, E. A., & Antwi, K. (2020). The role of digital platforms in enhancing higher education services. *Education and Information Technologies, 25*(5), 4205–4225. https://doi.org/10.1007/s10639-020-10177-2

Patel, R. (2021). A comparative analysis of digital scheduling tools in academic institutions. *Journal of Institutional Research, 16*(2), 45-60. https://www.jstor.org/stable/10.2307/48731982

Rahman, A., Fong, S., & Ismail, M. (2021). Energy conservation in university buildings through IoT-based smart classroom systems. *Energy and Buildings, 251*, 111333. https://doi.org/10.1016/j.enbuild.2021.111333

Rodway, C. A. (2025). *Classroombookings*. Classroombookings. https://www.classroombookings.com

Sharma, P., & Kumar, R. (2020). Energy-efficient smart classrooms using IoT: A step towards sustainable campus. *International Journal of Innovative Technology and Exploring Engineering, 9*(5), 260–264. https://doi.org/10.35940/ijitee.E6171.039520

Sharma, P., & Kumar, V. (2020). Smart classrooms: A framework for active learning and operational efficiency. *Education and Information Technologies, 25*(4), 3007-3025. https://doi.org/10.1007/s10639-020-10112-8

Skedda. (2025). *Skedda*. Skedda.com; Skedda. https://www.skedda.com/solutions/classroom-scheduling-software

Skedda. (2025). *Skedda: Classroom scheduling software*. Retrieved November 25, 2024, from https://www.skedda.com/solutions/classroom-scheduling-software

Smith, T. (2022). *The case for progressive web apps in enterprise and education*. O'Reilly Media. https://www.oreilly.com/library/view/the-case-for/9781492093459/

Tailwind CSS. (n.d.). *Tailwind CSS: Rapidly build modern websites without ever leaving your HTML*. Retrieved November 25, 2024, from https://tailwindcss.com

Tan, K. X. (2024). *Smart booking classroom management system for UTAR Kampar* [Undergraduate thesis, Universiti Tunku Abdul Rahman]. http://eprints.utar.edu.my/6497/1/fyp_CN_2024_TKX.pdf

Thong-on, M., Boonying, J., & Kongpan, Y. (2024). Development of the online classroom booking system of the college of innovation and management Suan Sunandha Rajabhat University. *Proceeding of the International Conference on Multidisciplinary Innovation for Sustainability and Excellence*. https://icmiessproceeding.icmiess.co/index.php/2024/article/view/26/25

Uddin, M., Strachan, R., & Scanlon, E. (2020). A systematic review of the adoption of cloud computing in education. *International Journal of Educational Technology in Higher Education, 17*(1), 1–28. https://doi.org/10.1186/s41239-020-00236-1

Wafi Nasrudin, M., Nordin, N. A., Ismail, I., Jais, M. I., Rahim, A., & Mustafa, W. A. (2021). Smart classroom for electricity-saving with integrated IoT system. *Journal of Physics: Conference Series, 2107*(1), 012019. https://doi.org/10.1088/1742-6596/2107/1/012019