People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Frères Mentouri Constantine 1 University
Faculty of Technology Sciences
Department of Electronics

Order Number: ... ...

Master Thesis

Field: Automation
Specialty: Automation and Industrial Computing

Title

Smart Classroom Automation Using IoT Technologies - Ahmed HAMMANI Campus

Presented by

Rabah DJEBBES

Ouail MAOUCHE

Supervised by

Dr Karima BENZAID

Pr Nora MANSOURI

Academic Year: 2025/2026

Acknowledgments

This work would not have been possible without the support and encouragement of many people. We are deeply grateful to all of them.

First and foremost, we extend our sincere gratitude to our supervisors, ***Dr Karima BENZAID*** *and* ***Pr Nora MANSOURI****, for their expert guidance, constructive feedback, and constant availability. Their dedication to our work and their insightful suggestions pushed us to refine our ideas and persevere through challenges. We have learned immensely from their mentorship.*

We would also like to thank our families—our parents, siblings, and loved ones— for their unconditional love, their endless patience, and their quiet encouragement during the long hours of work. Their belief in us never wavered, and for that, we are forever grateful.

Our thanks also go to our friends and colleagues, who offered their support, shared their perspectives, and provided a sense of community throughout this journey. Their kindness and camaraderie were a constant source of motivation.

We are also grateful to the broader academic and professional community whose work and resources laid the foundation for our research.

Lastly, we gratefully acknowledge the members of the jury for their time, their careful examination of this work and their invaluable feedback.

With sincere appreciation,

Rabah DJEBBES

Ouail MAOUCHE

Dedication

This work is a reflection of the hands that shaped me, the hearts that carried me, and the souls that believed.

To my mother and father

For the quiet sacrifices I will never fully know, and the love that never once faltered. I am, and will always be, your gratitude.

To my brothers and sister

For the laughter that lightened my heaviest days, and the unwavering presence that reminded me I was never alone.

To my friends and colleagues

For the kindness that arrived without condition, and the encouragement that never tired. You made the journey lighter.

To all my family

For being the foundation upon which everything was built. Your love is the quiet constant of my life.

To every hand that lifted me, and every heart that believed— I carry you with me.

Rabah

Dedication

I dedicate this work, with immense gratitude and respect, to the people who have always been my source of strength, inspiration, and motivation.

To my beloved parents, whose unconditional love, endless sacrifices, unwavering support, and constant prayers have guided every step of my academic journey. Your belief in my abilities has been my greatest motivation.

To my mother, whose tenderness, patience, and sacrifices have illuminated my path since childhood. Your prayers have been my refuge in moments of doubt and my strength in moments of difficulty. This achievement is as much yours as it is mine.

To my father, whose wisdom, integrity, hard work, and determination have always inspired me to pursue excellence. Thank you for teaching me the value of perseverance and for always believing in my potential.

To my brothers, for your affection, encouragement, and constant presence. Your support has always motivated me to move forward and overcome every challenge.

To my friends and colleagues, thank you for the unforgettable memories, the mutual support, and the moments of friendship that made this journey meaningful and memorable.

To my professors and everyone who contributed to my education, thank you for sharing your knowledge and helping me grow both academically and personally.

Finally, to every student who believes that determination, patience, and hard work can transform challenges into achievements—may this thesis stand as a symbol of perseverance, learning, and the pursuit of knowledge.

With profound respect, gratitude, and appreciation.

Ouail

Abstract

This final-year project is part of the design and development of an intelligent classroom management system based on Internet of Things (IoT) technologies. The proposed solution relies on RFID technology for automatic identification and attendance recording, ESP32 microcontrollers for data collection and transmission, and a dedicated web platform for monitoring and managing the entire system. In this work, a complete and integrated architecture was designed and implemented to ensure efficient management of classroom users, teaching sessions, and attendance reports. The system also enables real-time communication between connected devices and the central platform through a lightweight communication protocol adapted to the constraints of IoT environments. The implementation of this solution helps improve the efficiency of educational space management by automating monitoring and control tasks while reducing manual operations. It therefore contributes to the creation of a smarter, more efficient, and better-organized educational environment.

Résumé

Ce projet de fin d'études s'inscrit dans le cadre de la conception et le développement d'un système intelligent de gestion des salles de classe basé sur les technologies de l'Internet des Objets. La solution proposée repose sur l'utilisation de la technologie RFID pour l'identification et l'enregistrement automatique des présences, de microcontrôleurs ESP32 pour la collecte et la transmission des données, ainsi que d'une plateforme web dédiée à la supervision et l'administration de l'ensemble du système. Dans ce travail, une architecture complète et intégrée a été conçue puis mise en œuvre afin d'assurer une gestion efficace des utilisateurs des salles de classe, des séances pédagogiques et des rapports de présence. Le système permet également la communication en temps réel entre les équipements connectés et la plateforme centrale grâce à un protocole de communication léger adapté aux contraintes des environnements IoT. La mise en œuvre de cette solution contribue à améliorer l'efficacité de la gestion des espaces pédagogiques, en automatisant les tâches de suivi et de contrôle, en réduisant les opérations manuelles. Elle participe ainsi à la création d'un un environnement éducatif plus intelligent, plus efficace et mieux organisé.

ملخـــــص

يُعد مشروع التخرج هذا جزءًا من عملية تصميم وتطوير نظام ذكي لإدارة الفصول الدراسية يعتمد على تقنيات إنترنت الأشياء. وتعتمد الحلول المقترحة على تقنية للتعرف التلقائي وتسجيل الحضور، والميكروكونترولرات لجمع البيانات ونقلها، ومنصة ويب مخصصة لمراقبة وإدارة النظام بأكمله. في هذا العمل، تم تصميم وتنفيذ بنية كاملة ومتكاملة لضمان الإدارة الفعالة لمستخدمي الفصول الدراسية، والجلسات التعليمية، وتقارير الحضور. كما يتيح النظام التواصل في الوقت الفعلي بين الأجهزة المتصلة والمنصة المركزية من خلال بروتوكول اتصال خفيف الوزن ومكيف ليتناسب مع قيود بيئات إنترنت الأشياء. ويساعد تنفيذ هذا الحل على تحسين كفاءة إدارة المساحات التعليمية من خلال أتمتة مهام المراقبة والتحكم مع تقليل العمليات اليدوية. وبالتالي، فإنه يساهم في خلق بيئة تعليمية أكثر ذكاءً وكفاءةً وتنظيمًا.

Contents

List of Acronyms and Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CSS	Cascading Style Sheets
DC	Direct Current
Django	High-level Python web framework
ESP32	Espressif 32-bit Microcontroller
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
Linux	Open-source operating system based on Unix
Mosquitto	Open-source MQTT message broker implementation
MQTT	Message Queuing Telemetry Transport
ORM	Object Relational Mapping
PC	Personal Computer
PDF	Portable Document Format
PoE	Power over Ethernet
PostgreSQL	An Object-Relational Database Management System
PSU	Power Supply Unit
Python	High-level programming language used for backend development
RAM	Random Access Memory
RBAC	Role-Based Access Control
RDBMS	Relational Database Management System
REST	Representational State Transfer
RFID	Radio Frequency Identification
SMTP	Simple Mail Transfer Protocol
SQLite	Lightweight relational database management system
SSD	Solid State Drive
SSE	Server Sent Event
U4.0	University 4.0
UI	User Interface
UID	Unique Identifier
UML	Unified Modeling Language
Ubuntu	Linux-based operating system distribution
URL	Uniform Resource Locator

List of Figures

List of Tables

General introduction

The development of information and communication technologies has significantly transformed many aspects of modern life, including education. Currently, smart systems are increasingly adopted in educational institutions in order to to enhance efficiency, automate administrative tasks, and improve the management of academic processes. In this context, traditional classroom management methods are becoming insufficient to meet the growing demands for monitoring, attendance tracking, efficient resource management and real-time supervision.

This project focuses on the design and development of a smart classroom management system based on IoT technologies. The proposed solution includes the use of RFID technology, ESP32 microcontrollers, MQTT communication protocol, and a web-based platform that together aim to automate and optimize classroom management processes.

By enabling real-time monitoring, reliable data exchange, and centralized system control, the solution enhances both operational efficiency and decision-making processes. Also, the aim of this project is to minimize manual intervention in classroom management through the implementation of an intelligent and automated system. By combining hardware and software components within a unified architecture, the proposed solution provides a more intelligent and automated approach to managing classroom activities and enhancing the overall educational environment.

The manuscript is organized into three chapters and a conclusion.

- The first chapter introduces the research context, presents the problem statement, and reviews existing solutions in the field.

- The second chapter describes the requirements analysis and the proposed solution, including project scopes, functional and non-functional requirements, and use case modeling.

- The third chapter focuses on system design and implementation, explaining how the proposed architecture is translated into a working technical system using appropriate technologies such as Django, MQTT, RFID, and ESP32.

In conclusion, this work presents a comprehensive overview of the development process of the smart classroom management system and highlights the main contributions and achievements. Furthermore, it discusses potential improvements and future developments that could be considered in subsequent versions of the system.

Chapter 1

General context and problem statement

I.1 Introduction

Digital transformation is reshaping higher education, yet many universities still rely on traditional management systems that are manual, slow, and inefficient. These systems lack automation, real-time monitoring, and centralized supervision, leading to challenges in attendance management, classroom security, energy consumption, reporting, and resource coordination.

This chapter presents an overview of these key challenges, introduces the concept of smart campuses with a focus on smart classrooms, reviews the enabling technologies, and provides a comparative analysis of existing solutions. The goal is to establish the need for a customized, cost-effective and scalable smart classroom system adapted to university requirements.

I.2 Problem statement

Conventional educational environments use traditional management systems within schools, colleges, and universities. However, these systems are often slow and inefficient in daily operations. They also provide limited automation and weak monitoring capabilities, which make it difficult to manage students, control learning resources, and ensure classroom security. As a result, educational institutions may face many problems related to organization, supervision, and resource management, particularly in large institutions with a high number of staff, where manual management becomes impractical.

In this project, we focus specifically on this problem within universities. The main challenges faced in traditional university educational systems are listed as follows [1]:

I.2.1 Manual attendance management

The manual system for recording student attendance is still commonly used in many classroom environments using paper-based methods. However, this process is often time-consuming as teachers have to record the attendance of students during each class session.

In addition, manual attendance systems are prone to human errors and may allow problems such as incorrect or fake attendance records. These limitations reduce the reliability of attendance tracking and increase the workload of teachers.

I.2.2 Unauthorized classroom access

Uncontrolled access to classrooms is quite common in many educational institutions due to the absence of access control systems in such environments. Without an effective system in place, people can freely enter and leave classrooms without proper authentication or monitoring.

However, uncontrolled access to classrooms poses various security risks, not only to students but also to the equipment present inside the classroom. Moreover, it becomes difficult, if not impossible, to keep track of people who enter the room because there is no adequate monitoring or control system.

I.2.3 Uncontrolled energy consumption

Energy consumption is another problem in many classroom settings because lights, fans, and other electronic devices are sometimes left ON even when not in use.

This usually happens because there is no automated system to control energy consumption according to classroom activity. As a result, a large amount of energy may be wasted, which can increase electricity costs. In addition, unnecessary use of equipment may lead to damage or a reduction in its useful lifetime.

I.2.4 Delays in administrative reporting

Reporting time lags occur due to the use of manual approaches to report generation in educational institutions. When large amounts of data must be collected and analyzed, the manual report generation process requires considerable time and effort. Consequently, information becomes available very late, and sometimes it is impossible to obtain it in a timely manner.

I.2.5 Low level of classroom security

Classroom insecurity is very prevalent in traditional educational environments, where security measures and devices are lacking. There are no sensors or other forms of security systems that can detect anomalies in the classroom setting.

In addition, no detection or alarm systems are in place to alert people in the event of an incident, such as smoke, fire, or any form of unauthorized entry into the classroom. This can lead to numerous safety and security threats that affect both students and the equipment used therein.

Furthermore, emergency response times will be significantly longer due to the absence of intelligent security systems capable of monitoring the classroom around the clock.

I.2.6 Lack of centralized supervision

Traditional universities often suffer from a lack of centralized supervision. Administrative and security monitoring is typically distributed across independent departments, leading to fragmented oversight and poor coordination. This problem is particularly evident when managing shared resources, equipment, and classrooms used by multiple departments, where scheduling conflicts, resource allocation, and access control become difficult to coordinate without a unified system.

I.3 Digital transformation in universities

To overcome these challenges, modern universities are transitioning toward centralized, technology-driven supervision systems that integrate access control, resource management, real-time monitoring, and automated reporting [2]. These systems enable efficient coordination of shared classrooms and equipment between departments, while enhancing security and operational efficiency .

In universities, automation makes the system more efficient by reducing repetitive activities and simplifying everyday administrative and management tasks. It also reduces a significant amount of manual work by handling time-consuming tasks automatically. Thus, automation plays a key role in this transition by enabling centralized systems to function effectively, reducing human intervention, and allowing administrators and staff to focus on higher-value academic activities rather than routine manual processes.

I.3.1 Concept of smart campus

A smart campus uses digital and intelligent technologies to improve university services and management [3]. IoT devices help monitor, control, and automate different operations in real time. Examples of smart campus services include smart parking systems, smart lighting systems, and smart classrooms. Smart campus systems may also include smart monitoring, smart access control, and centralized systems to improve security and resource management.

Among these intelligent services, smart classrooms represent an integral part of the smart campus infrastructure. As technology continues to evolve, universities are progressively transforming into connected and intelligent environments. This transformation supports the vision of university 4.0 (U4.0), where universities become fully connected, intelligent, and automated environments [4].

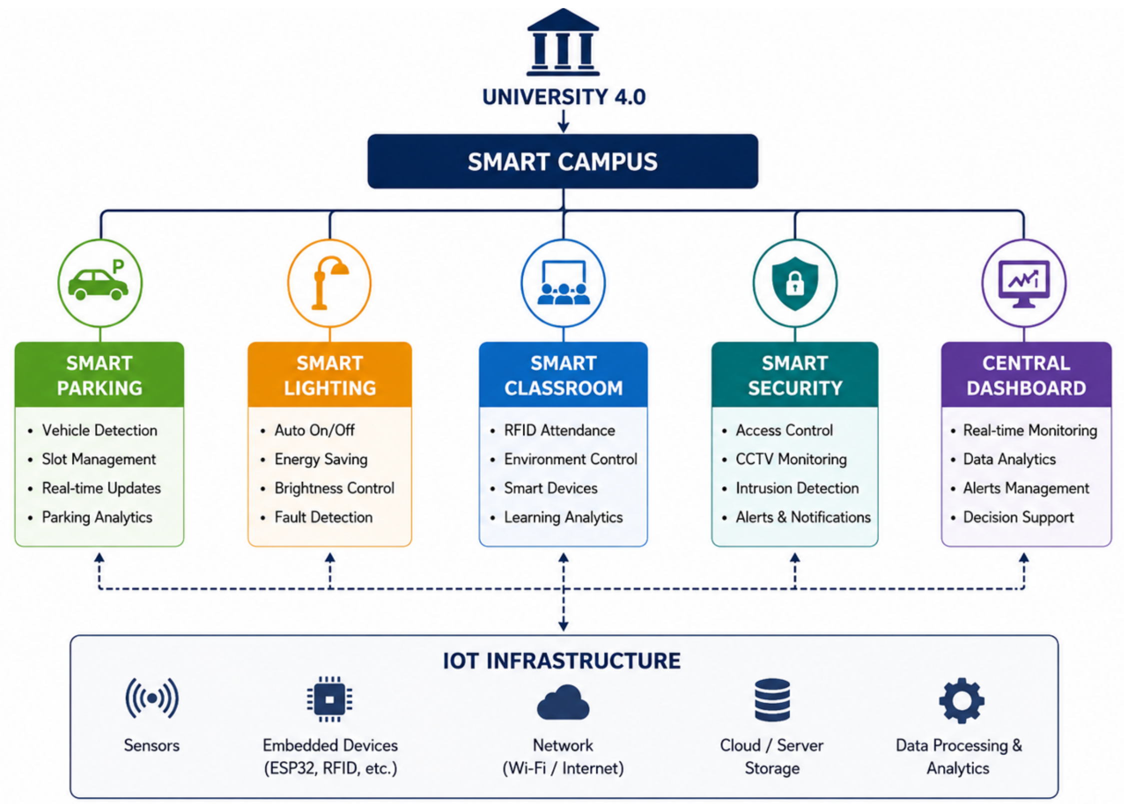


Figure 1.1: Smart campus ecosystem and its main components.

Figure 1.1 illustrates the smart campus ecosystem and its main components. It presents the integration of various intelligent technologies and services within a university environment. These interconnected subsystems work together to improve operational efficiency, resource management, security, and overall user experience across campus.

The proposed smart classroom system contributes to this vision by providing intelligent classroom management, automated monitoring, and connected educational services.

I.3.2 Importance of smart classrooms

The smart classroom is a core component of any smart campus. Although smart campus includes broad infrastructure such as energy grids, security networks, and administrative platforms, the classroom is where the most critical academic activities take place. Consequently, deploying smart technologies in classrooms - including smart sensors, automated access control, real-time monitoring, and intelligent resource management - is essential to achieve the goals of a smart campus [5]. Smart classrooms enable automated attendance tracking, lighting control, adaptive environmental control, remote equipment management, and immediate anomaly detection. When integrated into a centralized campus platform, they provide administrators with real-time visibility across departments, facilitating efficient resource sharing and rapid emergency response. Thus, smart classrooms serve as the functional units that transform a traditional campus into a truly smart one.

Figure 1.2 illustrates the concept of a smart classroom environment. It highlights the main components involved in the system, including the teacher, students, RFID cards, sensors, network infrastructure, and the supervision and management platform. These components work together to support intelligent classroom management, automated attendance monitoring, and connected educational services.

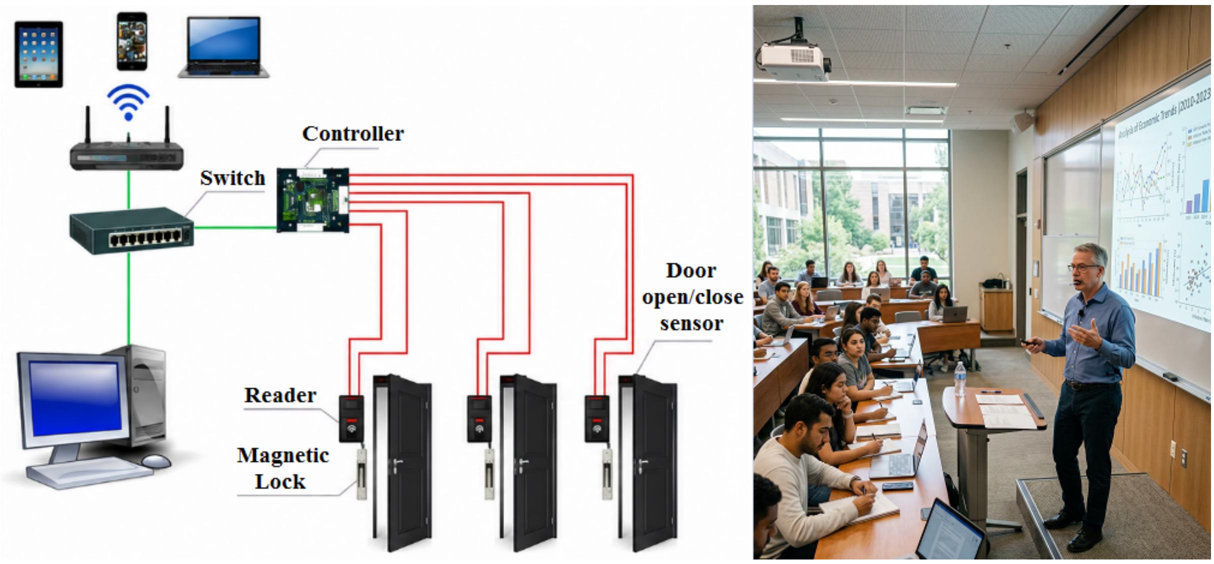


Figure 1.2: Smart campus concept.

I.4 Technologies used in modern educational settings

The development of modern smart educational systems employs various technologies that complement each other to optimize classroom management and automation. This section presents a general overview of these technologies.

I.4.1 Smart sensors and embedded systems

Smart sensors and embedded systems play an important role in modern intelligent systems [6]. Sensors are responsible for detecting different environmental conditions and events, including changes in temperature, light, motion, and human presence. After detecting these parameters, the information is processed by embedded systems to perform specific monitoring and automation tasks.

I.4.2 RFID technology

Radio Frequency Identification (RFID) refers to a system that helps to identify objects automatically through the use of radio frequencies. RFID technology is commonly used in university educational settings for staff identification in order to control access (to campus, parking, and classrooms), and provide automated attendance recording functions [7]. This type of identification is accomplished using RFID tags, cards, and readers, where the tags or cards contain information that is read by the RFID reader.

I.4.3 MQTT communication protocol

MQTT is a lightweight protocol designed for communication between interconnected devices through the exchange of information using a publish/subscribe model [8]. In this protocol, devices publish data to different topics, while other devices subscribe to these topics in order to receive information. MQTT is widely preferred in smart universities solutions because of its speed, efficiency, and low power consumption, especially in real-time IoT communication scenarios [9].

I.4.4 Web-based management platforms

Web-based management platforms refer to dashboards created to simplify system monitoring and administration. Users can use these platforms to monitor various data and system functions in real time. In addition, these tools provide functionalities such as data visualization, which facilitates data analysis and interpretation. They also offer centralized management of several interconnected devices and systems through a single platform.

In educational systems, web-based platforms aim to provide a user-friendly interface that enhances educational interaction between students, teachers, and administrators. Key functionalities include user registration, secure login, messaging, scheduling, PDF upload with search, real-time notifications, and robust data security [10].

Table 1.1: Technologies used in modern educational settings.

Technology	Main function	Application in smart classrooms
Sensors	Monitoring	Collect environmental and operational data for monitoring purposes.
ESP32, Microcontrollers, ...	Embedded control	Control connected devices and manages data transmission within the system.
RFID	Identification	Used for student attendance management and classroom access control.
MQTT	Data communication	Provides lightweight real-time message exchange between devices, servers, and applications.
Web platform	Management	Provides centralized supervision, reporting, classroom management, and data visualization.

I.5 IoT in modern education

The Internet of Things (IoT) refers to a network of devices (such as sensors, embedded systems, and many other devices) connected over the internet that can communicate and exchange data with each other automatically [11–14]. These devices can also interact with cloud systems that store and process information.

In the field of education, the technologies used in modern settings work together in the creation of a smart educational system. The IoT plays an important role in the connection of embedded systems (like ESP32 and micro-controllers), sensors, and software platforms to improve learning quality and simplify academic functions [15]. The data collected can be monitored and analyzed in real time, which helps to make educational systems smarter and more efficient. For example, IoT can be used for automatic attendance, smart classroom control, energy management, and better coordination between students, teachers, and administration.

Figure 1.3 illustrates the general architecture of an IoT-based smart classroom system. It presents the different functional layers of the system, from data acquisition through sensors and RFID devices to data processing, communication, storage, and visualization through the management platform.

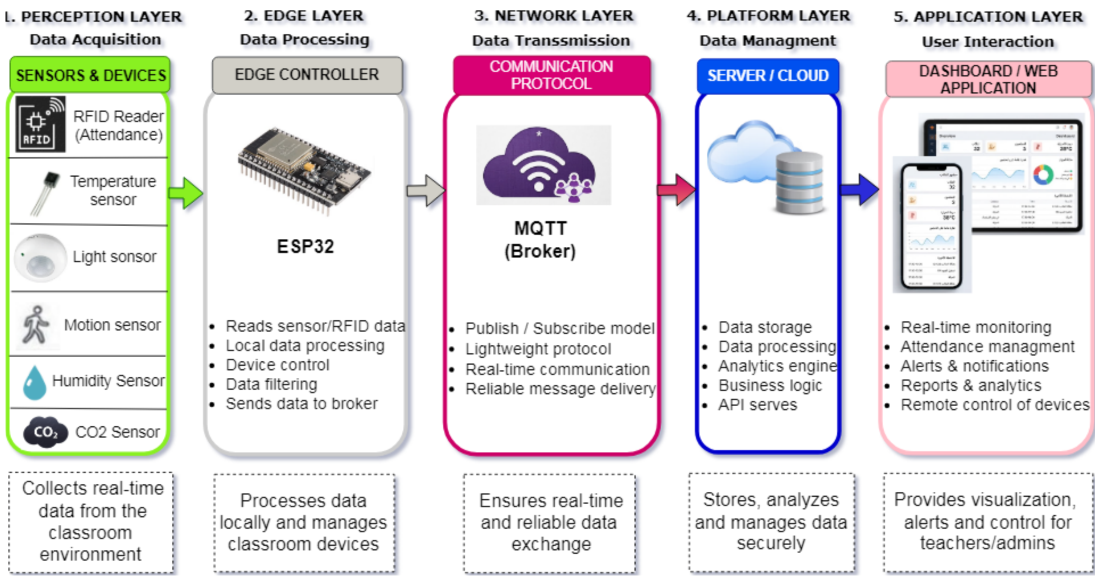


Figure 1.3: IoT architecture of a smart classroom system.

I.6 Comparative study of existing educational solutions

The educational systems of universities vary in their level of digitalization. Ranging from traditional to smart connected systems, they can be classified into three families as follows:

I.6.1 Traditional systems

Traditional systems include manual attendance records on paper, spreadsheets using Microsoft Excel, and other similar methods. With traditional systems, everything is done manually, without any form of automation whatsoever. This results in several drawbacks, such as inefficiency, human error, and ineffective data management. Furthermore, there is no way of monitoring classroom activities, which affects efficiency in supervision and control. Thus, traditional systems can be considered outdated and not appropriate by today's global university standards.

I.6.2 Basic RFID systems

Basic RFID systems can be found in many universities, where RFID cards or badges are used for access control or attendance records. In such cases, university staff simply use their RFID cards so that the access control system authorizes them to enter. In addition, students use them to register their attendance when entering the classroom. The advantage of this system is that it provides secure, traceable, and flexible authentication. It also automates student registration, reducing manual errors and saving time.

However, these types of RFID systems are used only for identification and attendance purposes. They lack intelligent functions such as data analysis, automated classroom management, or decision-making. Moreover, they do not integrate with other educational systems, making it impossible for them to contribute to a smart classroom environment [7]. Therefore, while basic RFID systems are useful for simple attendance tracking, they lack the intelligence and integration capabilities required to support comprehensive classroom management.

I.6.3 Existing smart classroom solutions

Smart classroom solutions can be defined as modern systems based on IoT technology. They use sensors, automation, and monitoring features to manage classroom environments in real time. These solutions offer numerous benefits, including automated management, real-time monitoring, and data collection. In addition, they contribute to increasing operational efficiency and enable intelligent classroom management through smart devices and control systems.

However, smart classroom solutions are often expensive and complex to deploy and are not necessarily part of a single fully integrated system [5,15]. Therefore, despite their many benefits, smart classroom solutions still present certain drawbacks that must be addressed.

Table 1.2 provides a summary of the comparative study on different aspects such as automation, monitoring capabilities, security, reporting, and energy management.

Table 1.2: Comparative study of existing educational solutions.

Criteria	Traditional systems	RFID systems	Ready-made smart classroom systems
Automation	No	Partial	Yes
Monitoring	No	Limited	Real-time
Security	Low	Medium	High
Reporting	Manual	Basic	Automated
Energy Management	No	No	Yes
Integration	No	Limited	Full integration
Cost	cheap	inexpensive	very expensive
Complexity	basic	moderate	very complex

Traditional systems are inefficient due to their lack of automation and real-time monitoring. In response to these limitations, RFID-based systems offer a significant improvement by providing access control and attendance tracking. However, they remain limited in their ability to automate operations and monitor classrooms effectively. To overcome these shortcomings, smart classroom systems present a more advanced solution, that integrates automation, monitoring, security, energy management, and IoT. Nevertheless, the complexity and cost of ready-made solutions remain significant barriers to deployment.

Overall, none of the systems discussed above can fully satisfy the requirements of a modern intelligent classroom. Consequently, a customized smart solution that is simple, cost-effective, scalable, and capable of comprehensively addressing these limitations is required. Moreover, such an approach would be better aligned with our university's specific requirements.

I.7 Conclusion

This chapter has highlighted the main limitations of traditional university management systems, including manual attendance, unauthorized access, energy waste, delayed reporting, poor security, and fragmented supervision. It also reviewed the concept of a smart campus and the technologies that enable smart classrooms. A comparative study of traditional systems, basic RFID systems, and ready-made smart classroom solutions revealed that none fully meets the needs of a modern intelligent classroom. Therefore, a customized, simple, cost-effective, and scalable solution is required. The following chapter will present the design and implementation of such a system.

Chapter 2

Requirements analysis and proposed solution

II.1 Introduction

The requirements analysis phase is a critical step in the development of any intelligent information system. It involves identifying the system's functionalities, constraints, user expectations, and technological requirements that will guide the entire development process. A well-defined set of requirements ensures that the final system meets the needs of its users and operates efficiently within its intended environment.

In this chapter, we present the requirements analysis for the proposed smart classroom management system. It details the project objectives, system scope, functional and non-functional requirements, hardware and software technologies, and the development methodology adopted to guide the implementation phase

II.2 Project objectives

This project focuses mainly on developing and implementing an IoT-based smart classroom management system that automates student attendance registration, provides real-time monitoring, and improves security and resource management within the classroom through intelligent access control and authentication.

The project is structured around the following objectives:

- Provide a secure access control mechanism for classroom access through authentication.
- Automate student attendance management using RFID technology.
- Enable real-time monitoring of classroom activities and environmental conditions.
- Automatically generate attendance and system reports for administrators.
- Improve monitoring and decision-making capabilities through a centralized web dashboard.
- Enable communication between IoT devices and the server using the MQTT protocol.

II.3 Proposed solution

The proposed solution is an intelligent smart classroom management system designed to automate traditional classroom operations using IoT technologies, RFID authentication, MQTT communication, and a web-based management system.

This solution aims to establish a fully automated platform capable of monitoring and managing classroom activities in real time. The system integrates both hardware and software components in order to ensure effective communication between users, devices, and the central server, while also providing centralized supervision of classroom activities through the unified management platform. Figure 2.1 provides an overview of the proposed system architecture, illustrating the communication flow between the central server, the network switch, and the various hardware components deployed in each classroom.

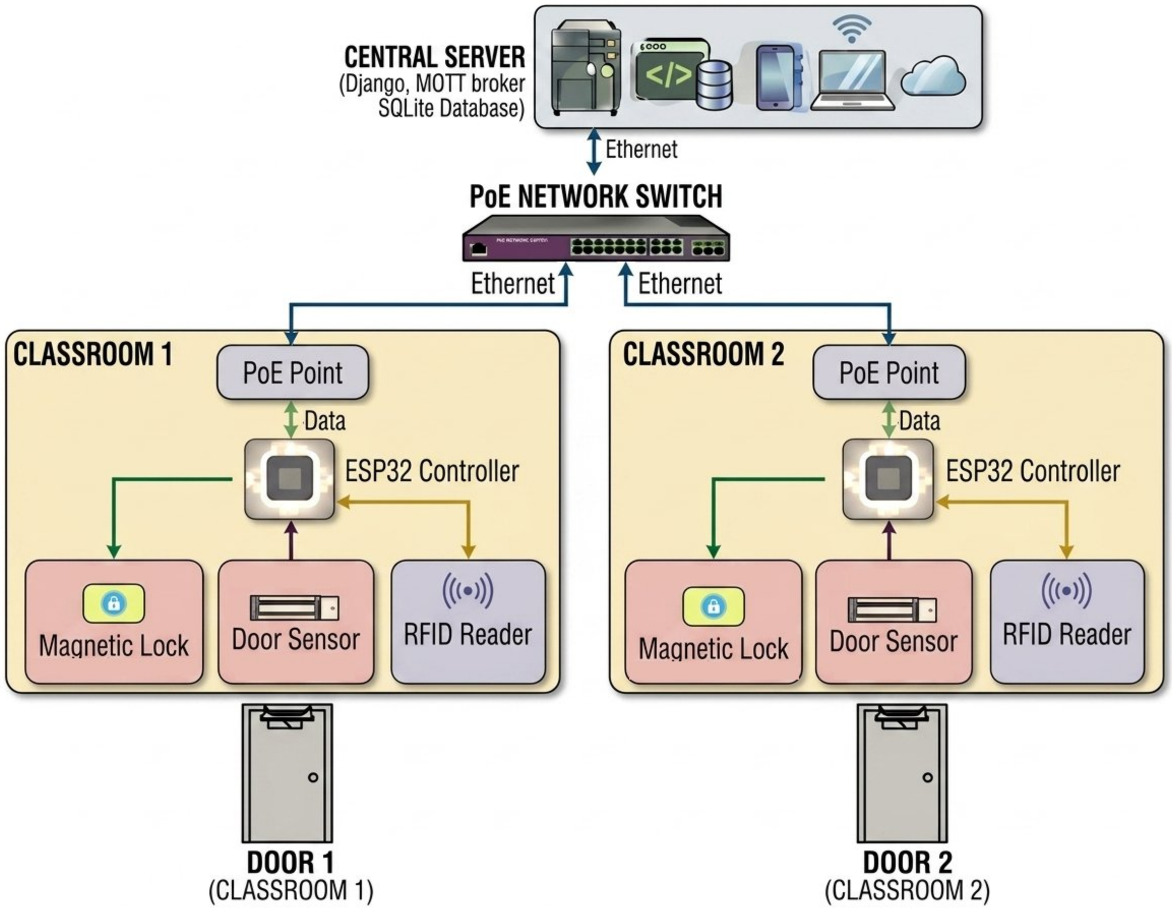


Figure 2.1: General architecture of the proposed smart classroom management system.

This project was developed in collaboration with a hardware team that was responsible for electrical installation and on-site equipment deployment. We, on the other hand, were responsible for software development, backend implementation, and system integration and connecting all devices to the PoE (Power over Ethernet) network infrastructure. The control room computer and router were installed in the university control room, serving as the central hub for the entire system.

II.4 Project scope

The scope of this project defines the boundaries of the smart classroom management system that is being developed, specifying the features included in the design. Establishing a clear scope is essential for distinguishing the system's capabilities from its limitations, particularly within the context of a prototype development framework.

II.4.1 Functional requirements

The following functionalities are included within the scope:

1. Authentication and access control

- Secure login interface with username and password authentication.
- Administrator access with full permissions to manage users, classrooms, and system settings.
- Staff access for maintenance and operational tasks.
- Teacher access for session initiation and classroom management.
- Management of student attendance using RFID authentication with unique card identifiers.

2. Classroom monitoring and supervision

- Real-time classroom status monitoring including occupancy, door status, and danger indicators.
- Automatic detection of classroom availability based on active session status.
- Door access control with out-of-hours scheduling capabilities.
- Organization of the schedule with support for Ramadan timing adjustments.
- Classroom activity logging and event tracking for audit purposes.
- Automatic session closure based on configured schedules and timings.

3. Session and attendance management

- Management of the session lifecycle (start, close, and auto-finish).
- Class sessions for teachers with student attendance collection.
- Inspection sessions for administrators (immediate closure, no students).
- Automatic attendance capture from RFID payloads for both teachers and students.
- Server-side timestamping of all attendance events.
- Attendance records linked to specific classroom sessions.
- Auto-finish engine that automatically closes expired sessions.

4. **Reporting and data visualization**
    - Automatic generation of PDF attendance reports with session summaries and student rosters.
    - Optional reporting email delivery through the SMTP configuration.
    - Dashboard with key performance indicator (KPI) badges for a quick status overview.
    - Real-time visualization of classroom occupancy and activity data.
    - Attendance history and session logs for monitoring and auditing.
5. **IoT and system integration**
    - Bidirectional communication between IoT devices and the backend server via the MQTT protocol.
    - MQTT event publishing from ESP32 devices (RFID scans, occupancy, door status).
    - MQTT response publishing from the Django backend (access approval, door delay configuration).
    - Real-time data synchronization between IoT devices and the web dashboard.
    - Device-to-dashboard event handling (room state updates, access requests, attendance requests, door delay requests).
    - Dashboard-to-device response handling (access responses, door delay responses).
6. **Administration and configuration**
    - Centralized web dashboard for administrative tasks and decision-making.
    - User management (create, edit, delete staff and student accounts).
    - Classroom management (add, edit, delete classrooms).
    - System configuration via the Settings page (auto-finish timing, pagination, KPI badges, compact mode).
    - MQTT broker configuration (host, port, topic wildcard, production/test mode).
    - SMTP configuration for automatic email delivery of reports.
    - Administrative panels for system monitoring and user supervision.

7. Additional platform features

- Pagination and bulk actions for efficient data management.
- UI density settings for customizable dashboard appearance.
- In-app MQTT integration guide for device developers.
- Management commands for system maintenance and testing.

All of the above functions constitute the complete feature set of the currently implemented smart classroom management system, providing a robust and scalable solution for automated classroom management.

II.4.2 Non-functional requirements

Non-functional requirements specify the performance characteristics of the smart classroom management system. They define how the system behaves by addressing aspects such as performance, reliability, security, availability, scalability, maintainability, and usability. These requirements play an important role in ensuring optimal system performance and guaranteeing secure and reliable operation for both users and connected IoT devices. The following functionalities are included within the scope:

1. Performance

The system needs to provide high real-time capability in order to process the information coming from IoT sensors with minimal delay. RFID authentication, attendance registration, and MQTT communication operations should be performed quickly to ensure efficient system operation during classroom sessions. The system must also function properly when multiple users access the platform simultaneously.

2. Security

Security plays an important role in the design of the proposed system. The platform should provide secure authentication for administrative users. The information transmitted between the IoT devices and the server should be protected against unauthorized access. In addition, user login credentials should be stored securely within the database.

3. Reliability

The system has to be highly reliable and free from unexpected failures. The IoT connectivity, attendance tracking, and data processing modules must be developed in a way that ensures maximum system reliability. In case of temporary network disruptions, data integrity should be maintained, and no data loss should occur.

4. Availability

The system must remain available during all classroom sessions. The availability of the web application and IoT services should be guaranteed in order to minimize system downtime.

5. Scalability

Scalability is another important design requirement that should be taken into account during system development. The system should support future expansion, including the addition of more classrooms, users, and IoT devices.

6. Maintainability

The system should be designed in a modular manner to support ease of maintenance. Maintenance of individual components such as RFID, MQTT, Django backend, and ESP32 modules should be simple and manageable.

7. Usability

The web application should provide an easy-to-use user interface. Users should be able to use and understand the system easily, even without specialized technical skills.

II.5 Hardware requirements

The hardware requirements define the physical components necessary for the deployment and operation of the smart classroom management system. These components are distributed in the control room and each classroom, forming the physical infrastructure that supports the software application and IoT communication.

II.5.1 Control room equipment

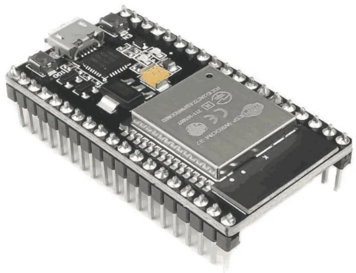
- **Control room computer:** A standard PC running the Linux operating system (Ubuntu 20.04 LTS or later) hosting the Django backend application, the MQTT broker (Eclipse Mosquitto) and the SQLite database. Minimum specifications include a multi-core processor, 8 GB RAM, and 256 GB storage.
- **Network router:** A switch providing stable connectivity between the control room computer, IoT devices, and users accessing the web dashboard over the local network.
- **Network switch:** A PoE (Power over Ethernet) switch to distribute both power and data to ESP32 devices installed in each classroom.

(a) Control room computer. (b) Network router. (c) Network switch.

Figure 2.2: Control room equipment.

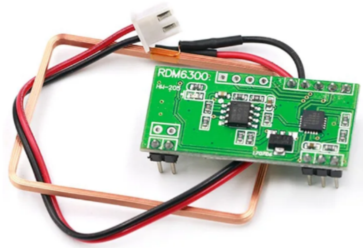
II.5.2 Per classroom equipment

- **ESP32 Microcontroller:** A low-power microcontroller with built-in Wi-Fi and Bluetooth capabilities. The ESP32 is responsible for reading RFID tags, controlling the magnetic lock, and communicating with the MQTT broker.
- **PoE access point:** The PoE access point is a device installed in each classroom that receives data over the Ethernet cable from the central PoE switch and provides network connectivity to the ESP32 microcontroller. Unlike a typical PoE setup, the access point is used exclusively for data transmission, as power is supplied separately through a dedicated 12V power supply unit. This separation ensures stable operation and simplifies troubleshooting.
- **RFID reader:** 125 kHz RFID reader module (e.g., compatible with EM4100). The reader captures the unique identifier (UID) from 125 kHz proximity cards presented by users and sends it to the ESP32 for processing.
- **Magnetic lock:** An electrically controlled door lock that is triggered by the ESP32 to unlock the door upon successful authentication.
- **12V Power Supply Unit (PSU):** A dedicated power supply providing 12V DC to the ESP32 microcontroller and connected peripherals. This ensures stable and sufficient power for the reliable operation of all classroom devices.
- **RFID cards:** 125 kHz passive proximity cards (EM4100 compatible), each containing a unique identifier (UID) assigned to students, teachers, and staff. These cards are widely used in access control systems due to their affordability, durability, and reliable read range.

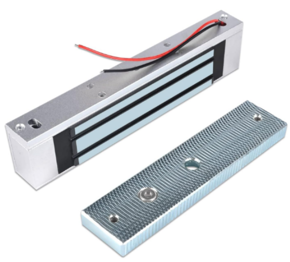


(a) ESP32 microcontroller.

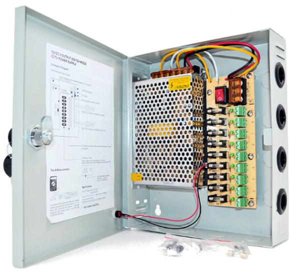
(b) PoE access point.



(c) RFID reader.



(d) Magnetic lock.



(e) Power supply unit.

(f) RFID cards.

Figure 2.3: Per classroom equipment.

Table 2.1 lists the hardware components required for the proposed system, indicating the quantity of each component and its role.

Table 2.1: Hardware components summary.

Component	Quantity	Description
Control room PC	1	Ubuntu 22.04, 8GB RAM, 256GB SSD
Network router	1	Provides local network connectivity
PoE switch	1	Distributes power and data to classrooms
ESP32	1 per classroom	Wi-Fi enabled microcontroller
PoE access point	1 per classroom	Data connectivity
RFID reader	1 per classroom	Reads RFID cards (EM4100 compatible)
Magnetic lock	1 per classroom	Electrically controlled door lock
Power supply unit	1	Provides constant 12v power
RFID cards	Per user	User identification (EM4100 compatible)

II.6 Software technologies used

The deployment of the smart classroom management system takes advantage of several modern technologies that have been chosen on the basis of their reliability, scalability, and compatibility with intelligent IoT systems. These technologies include backend technology, database technology, front-end technology, and IoT communication technology. Table 2.2 presents the main technologies used in the system, categorized by their purpose within the overall architecture.

Table 2.2: Summary of software technologies used in the proposed system.

Category	Technology	Purpose
Programming language	Python	Server-side logic, business rules, MQTT message processing
Backend framework	Django	Web application backend, authentication, ORM, admin interface
Database	SQLite	Lightweight, serverless data storage for prototype deployment
Frontend	JavaScript, CSS, HTML	Responsive web dashboard and user interface
Communication	MQTT	Lightweight publish/subscribe protocol for IoT messaging

II.6.1 Programming language

- **Python**

Python is the main programming language used for back-end development. Its simplicity, readability, and powerful libraries made it very convenient to implement business logic, process MQTT messages, and manipulate databases.

II.6.2 Backend framework

- **Django**

Django is a Python-based web framework that has been chosen for its various features available out-of-the-box (batteries included), such as user authentication, database management, and the administration panel which eases the process of managing the system. Django is the backbone of the system that manages all the server-side logic and API communication.

II.6.3 Database technology

- **SQLite**

SQLite is a light-weight and server-less RDBMS that fits well with the Django framework. The reason behind selecting it is due to its easy and configuration-free features, which help in building prototypes.

II.6.4 Front-end technologies

- **HTML, CSS, and JavaScript**

These core web technologies were used to build the front-end interface of the system. They enable the creation of dynamic, responsive and user-friendly dashboards that allow administrators and teachers to interact seamlessly with the system.

II.6.5 Communication technologies

- **MQTT**

MQTT is a lightweight publish/subscribe messaging protocol designed for IoT environments. It was selected for its low bandwidth consumption and efficient real-time communication capabilities. In this system, MQTT enables bidirectional data exchange between ESP32 IoT devices and the Django backend, facilitating real-time event processing and device control.

II.7 Development methodology

The design process of the smart classroom management system involved an iterative and incremental process, involving the use of Agile approach and continuous integration processes. This process was chosen due to the dynamic nature of IoT-based systems that require both hardware and software to be designed simultaneously.

II.7.1 Agile development approach

The project adopted the Agile methodology, which emphasizes iterative development, continuous feedback, and adaptation throughout the development life-cycle. This approach was particularly well-suited as it enabled parallel development between the software and hardware teams, provided flexibility to adapt to evolving requirements, and allowed for continuous integration and real-world testing in actual classroom environments.

II.7.2 Incremental development process

The system was built through multiple iterations, each adding new functionality:

- **Iteration 1**: Django backend setup, user authentication, and basic web dashboard.
- **Iteration 2**: MQTT broker integration and Django listener to receive events.
- **Iteration 3**: Bidirectional MQTT communication and session auto-finish engine.
- **Iteration 4**: PDF report generation, email delivery, and improved dashboard with real-time updates.
- **Iteration 5**: Full hardware integration with ESP32 devices, RFID readers, and magnetic locks.
- **Iteration 6**: Final polishing, security hardening, and deployment on the control room computer.

II.8 System architecture diagram

A use case diagram is a behavioral UML diagram that presents the functional interactions between actors and the system under discussion. The use case diagram provides a low-level view of the tools involved in achieving goals through interactions between users and external entities.

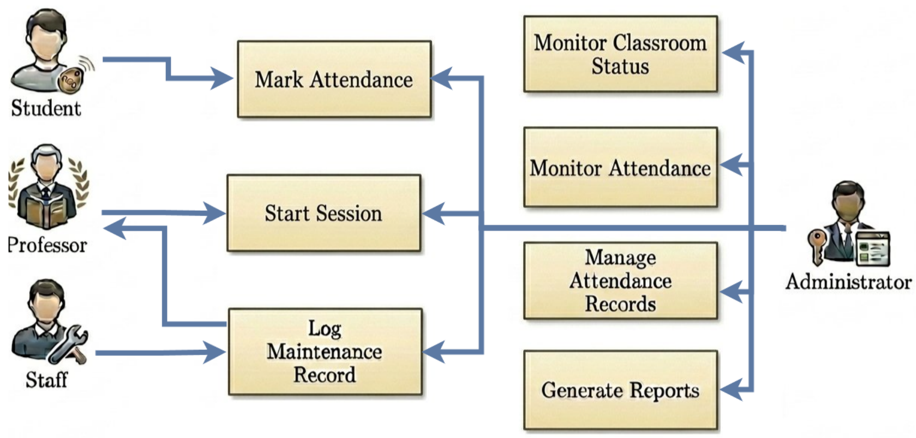


Figure 2.4: Use case diagram of the Smart Classroom Management System.

II.9 Conclusion

This chapter has defined the functional and non-functional requirements for the Smart Classroom Management System, along with the hardware and software technologies and the development methodology. This analysis provides a solid foundation for the system design and implementation presented in chapter 3.

Chapter 3

System design and implementation

III.1 Introduction

The system architecture and implementation phase marks an important phase in the development of the proposed smart classroom management system. In this phase, the analysis of requirements carried out in chapter 2 will be converted into a tangible solution. The goal of this phase is the design and implementation of a system that guaranties the smooth functioning of the hardware-software interaction based on functional and non-functional requirements.

In order to ensure the required functionality of the system, a number of technologies will be used, namely RFID tags for user identification, the ESP32 micro-controller for IoT devices control, MQTT protocol for real-time communication, the Django framework for backend development, SQLite database and web dashboard for management. All of these technologies combined will provide automated management of student attendance, session life-cycle management, reporting, and real-time classroom monitoring.

This chapter is divided into two major parts. In the first part, the architecture of the system will be discussed in terms of its global architecture, backend architecture, front-end architecture, data architecture, and security architecture. In the second part, the implementation of system components will be explained.

The key contribution of this chapter is to demonstrate how the requirements outlined in chapter 2 were successfully transformed into a fully functional smart classroom management system, ready for deployment and real-world operation.

III.2 General system architecture

The general architecture, depicted in Figure 3.1, provides a high-level view of the proposed smart classroom management system. It illustrates the interaction between the main hardware and software components used to support attendance management and classroom monitoring.

The system also integrates with an IoT layer comprising ESP32 micro-controllers, RFID readers, magnetic locks, and door sensors deployed in each classroom. These devices communicate with the application tier via the MQTT protocol over the local network, enabling real-time event processing and bidirectional device control.

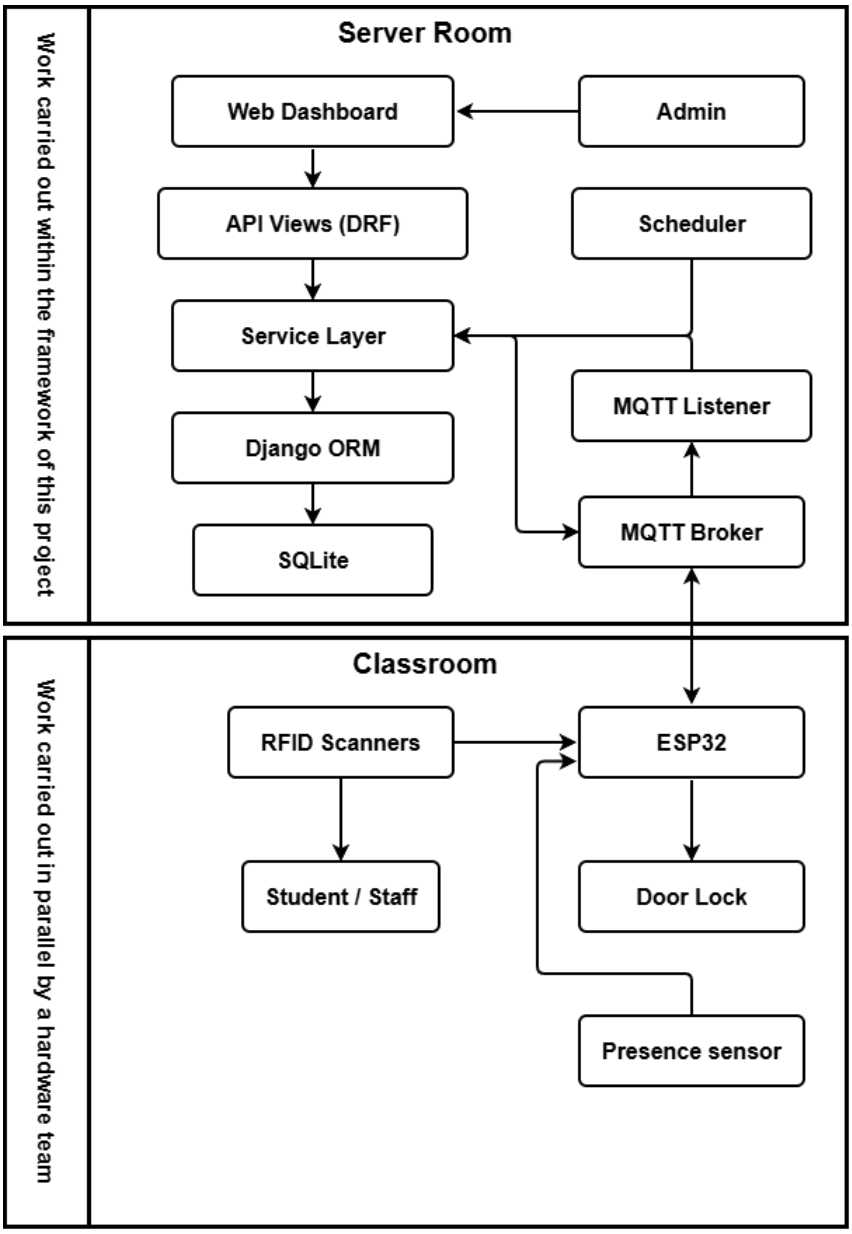


Figure 3.1: System architecture overview (Implementation view).

III.3 Backend and frontend architectures

This section provides the backend and frontend architecture of the Smart Classroom Management System. The chapter discusses how the system is designed to provide efficient data processing, real-time interaction, and communication.

III.3.1 Backend architecture

The backend of the proposed smart classroom management system is developed using the Django framework to provide a scalable and modifiable development environment to manage smart classroom logic, data management, and IoT interaction. This architectural approach is based on a layered structure consisting of models, views, services, and communication modules (see Figure 3.2).

Models are used to define the system data structure through the Django ORM module; in particular, the Student, Staff, Classroom, Session, and Attendance Report entities are defined with their relational mappings. Views manage HTTP interactions between the frontend and backend services, including operations such as authentication, session management, attendance processing, and report generation.

The service layer is where all the core business logic is located, including validation, attendance management, IoT device control, report generation, and other system operations. Real-time communication within the proposed IoT architecture is achieved using an MQTT listener that consumes IoT messages and forwards them to backend services for processing, .

Finally, the Scheduler and Auto-Finish Engine modules enable automated time-based actions, including session completion, attendance finalization, and updates to backend services. The Classroom State Synchronization mechanism ensures consistent real-time states between IoT devices, backend services, and the web application.

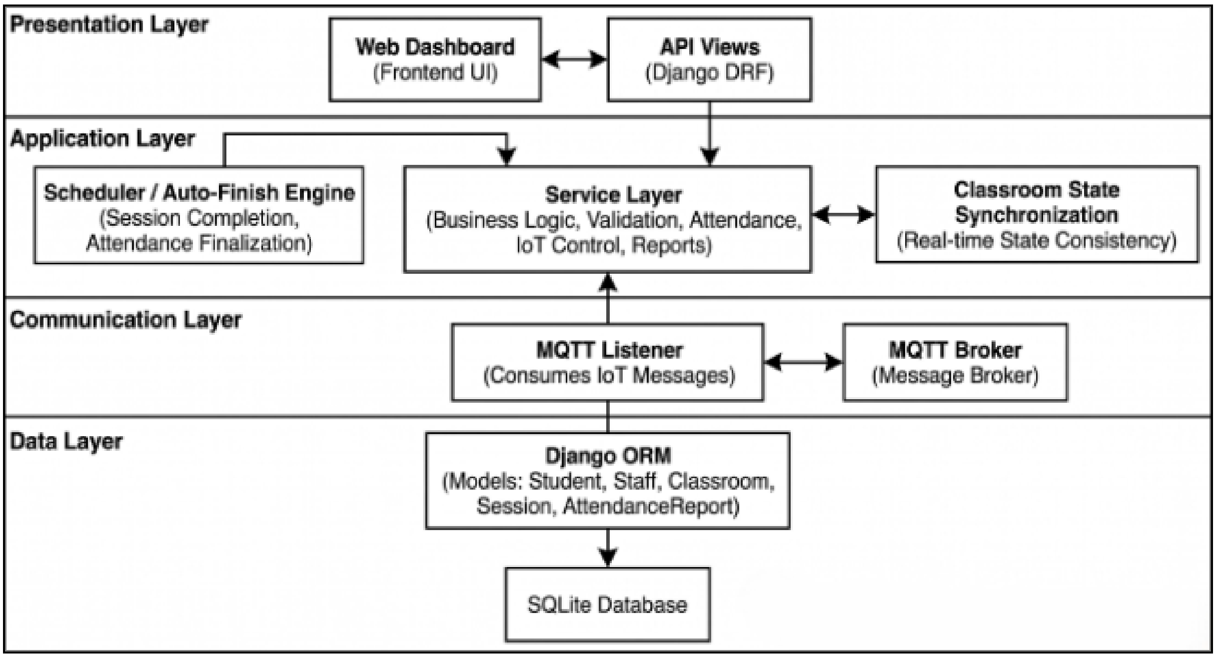


Figure 3.2: Backend architecture of the proposed smart classroom management system.

III.3.2 Frontend architecture

The frontend of the smart classroom management system is designed as a web-based dashboard that provides an interactive interface for monitoring and managing system operations. It communicates with the Django backend through REST APIs and HTTP requests to enable real-time data exchange within the system.

The frontend architecture is organized into several functional modules, including classroom management, student management, staff management, session management, system settings, and MQTT configuration. Each module provides interfaces for interacting with its corresponding data.

The dashboard acts as the central interface of the system, providing an overview of key information such as active sessions, classroom status, and attendance summaries. It ensures centralized access to system functionalities and supports efficient navigation between different modules. Figure 3.3 illustrates the frontend architecture of the proposed smart classroom management system.

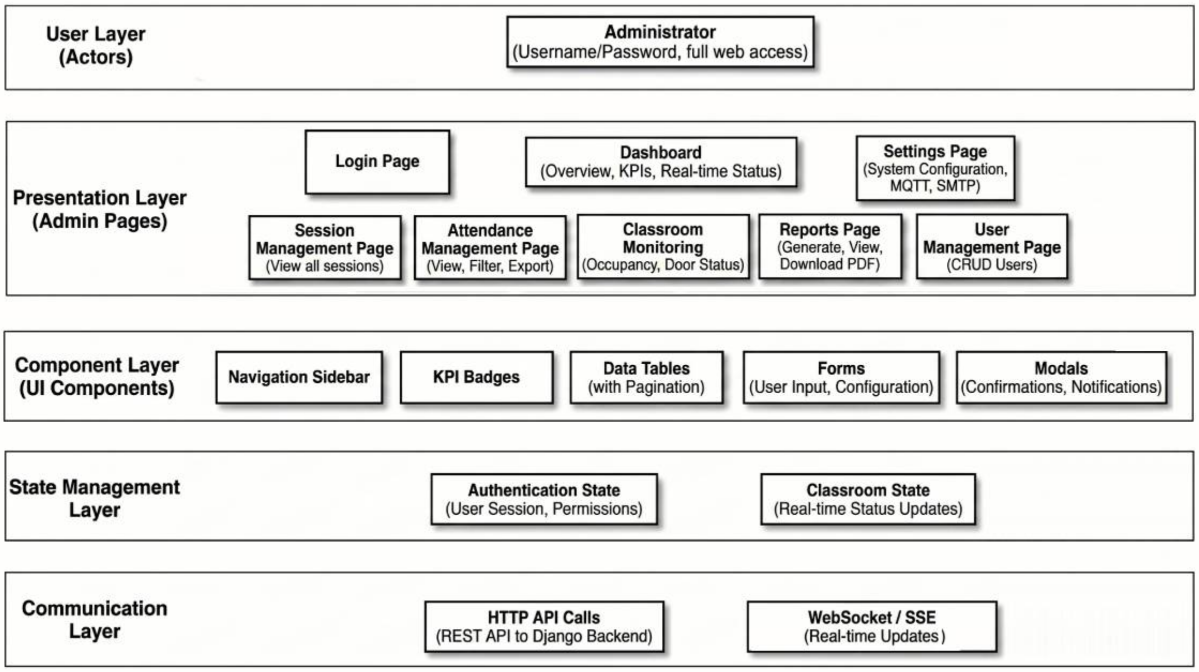


Figure 3.3: Frontend architecture of the proposed smart classroom management system.

III.4 Security architecture

A security-first architecture has been implemented for the smart classroom management system to ensure that educational resources and data remain confidential, accurate, and accessible. The architecture is composed of multiple security layers, including authentication, authorization, permission control, and RFID verification. Figure 3.4 illustrates the security architecture of the proposed smart classroom management system.

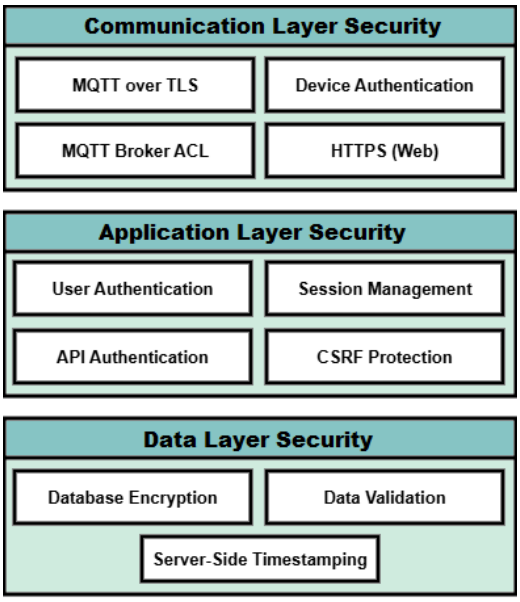


Figure 3.4: Security architecture of the proposed smart classroom management system.

The authentication layer represents the first line of defense in the system. It ensures that only authorized users can access system resources by verifying their identity before granting access to any services.

The authorization layer is based on role-based access control (RBAC). It defines the permissions assigned to each user role. Administrators have full access to system functionalities, while teachers are limited to attendance management, session control, and report consultation. This layer ensures that users can only perform actions that match their assigned privileges.

In addition, a permission management mechanism is implemented at the backend level to validate every critical operation before execution. This guarantees that no unauthorized modification of system data or services can occur.

For attendance security, RFID-based validation is used as an additional security layer. Each user is assigned a unique RFID tag, which is verified against the system database before any attendance record is accepted. If the RFID identifier is not valid or not registered, the system rejects the attendance request.

Through these combined security layers, the system ensures strong protection against unauthorized access, data manipulation, and misuse of resources.

III.5 Development environment

The development environment for the smart classroom management system includes the hardware configuration, the software platform, and the tools used in the development and deployment of the system. This environment is selected to ensure scalable, reliable and efficient communication between all components of the system.

III.5.1 Hardware environment

The hardware environment represents the physical infrastructure used to deploy and operate the smart classroom management system within the university environment.

In this project, the system is deployed on a standard computer located in the university control room. This machine hosts the Django backend application and is responsible for managing all system operations, including data processing, MQTT communication handling, and database management. A network router is used to ensure stable connectivity between system components, including IoT devices, the backend server, and users accessing the web dashboard.

Thus, the hardware environment is mainly based on a control room computer and network infrastructure that ensure the proper deployment and operation of the smart classroom management system. Figure 3.5 illustrates the hardware environment of the smart classroom management system, including the control room computer and network router used for system deployment.

Table 3.1: Hardware environment.

Component	**Description**
Control room PC	Hosts Django backend and system services
Network router / switch / Wireless N PoE Access Point	Provides connectivity between system components

(a) Control room. (b) Classroom (F9).

Figure 3.5: Hardware environment of the smart classroom management system.

III.5.2 Software environment

The software environment refers to the set of frameworks, programming languages, and tools required for the development of both frontend and backend components of the system.

The backend system is developed using the Django framework, which provides a structured environment for implementing business logic, authentication, database management, and API communication. Python is used as the main programming language due to its efficiency and simplicity.

The system uses SQLite as the relational database management system, which integrates easily with Django and supports the storage of user data, attendance records, and session information.

Communication between IoT devices and the backend system is achieved using MQTT, a lightweight publish/subscribe messaging protocol.

The system is deployed in a Linux-based environment running on the university control room computer to ensure stable server execution and reliable network communication.

Development and testing are performed using modern code editors such as Visual Studio Code and standard web browsers.

Development tools

The development process relies on several tools that support coding, version control, and system testing during the implementation phase.

Git and GitHub are used for version control and source code management, enabling efficient tracking of changes and collaboration.

Web browser developer tools are used for debugging and testing the frontend interface to ensure correct interaction with the backend services.

In addition, Django testing utilities and MQTT monitoring tools are used to validate system functionality and ensure correct real-time communication between IoT devices and the server.

III.6 Backend implementation

The backend implementation of the proposed smart classroom management system uses the Django framework and forms the main server-side module in charge of performing data processing, executing business logic operations, and coordinating other system components. Facilitates communication between IoT sensors, the database, and the frontend user interface while ensuring the security, scalability, and real-time capabilities of the whole solution.

The backend system is structured in a modular way based on different components, including project structure, database model, views, authentication service, business logic, MQTT communications and automation services. Such a design ensures easy maintenance and system scalability for further development.

The backend system continuously processes data received from IoT sensors via the MQTT protocol, stores it in the database, and maintains information consistency with the frontend application. It also performs automation tasks related to attendance processing, session management, and report generation.

III.6.1 Django project structure

The Django application is organized into a modular project structure, divided into several applications based on the system functionality. This modular approach ensures improved code organization, maintainability, and scalability of the application.

The project is composed of the following main applications:

- **Authentication App:** Manages user registration, login, logout, and password management.
- **Attendance App:** Handles student attendance marking, record management, and attendance reporting.
- **Classroom App:** Manages classroom entities, status monitoring, and scheduling.
- **Device Management App:** Manages IoT devices, MQTT communication, and device status monitoring.
- **Reporting App:** Handles the generation and export of reports in PDF format.
- **Session Management App:** Manages the classroom session life-cycle (start, close, auto-finish).

Each application follows a consistent internal structure consisting of:

- **Models:** Define the data structure and database schema using Django's ORM.
- **Views:** Handle HTTP requests and return responses (templates or API responses).
- **Services:** Contain the core business logic, including validation, attendance processing, and IoT control.
- **URLs:** Define the routing configuration for each application.
- **Templates:** Contain HTML templates for the web interface (when applicable).

This modular design enables independent development, testing, and maintenance of each application, facilitating parallel work and reducing the risk of regression bugs. It also allows for easy extension of the system by adding new applications as requirements evolve.

III.6.2 Database models

In the implementation of the database layer, Django ORM is used, whereby definitions of entities in the system are done by mapping objects of classes written in Python to relational database tables.

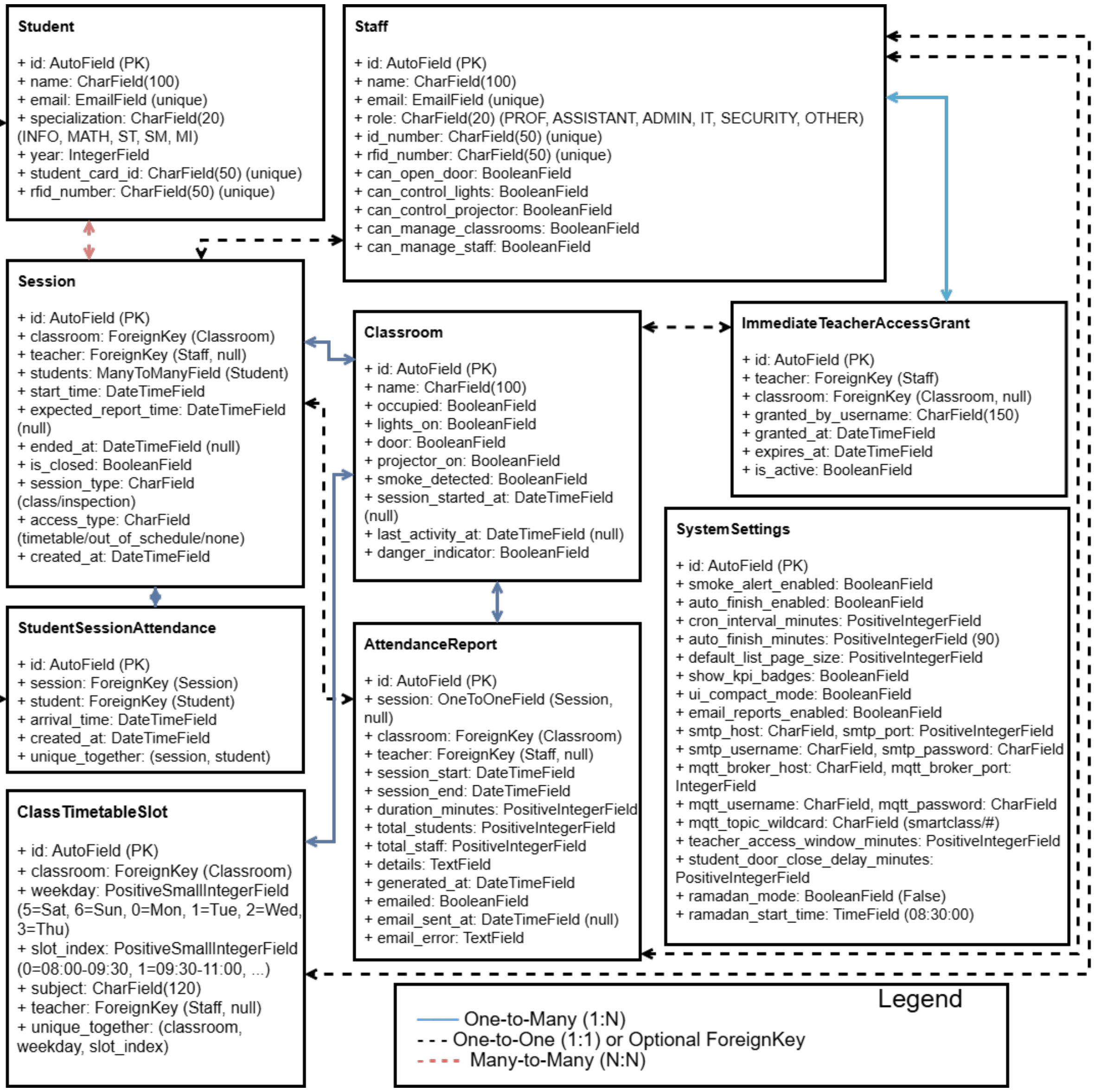


Figure 3.6: Django models of the smart classroom management system.

Figure 3.6 illustrates Django models of the smart classroom management system. Some of the primary models include Student, Staff, Classroom, Session, Attendance, and SystemSettings. These models describe the format in which the data is stored.

Foreign keys are used to define relationships between models, providing the ability to create relationships such as one-to-many relationships, where a student can have many attendance records and a classroom may conduct many sessions.

III.6.3 MQTT communication implementation

The MQTT protocol is used for communication between IoT devices and the backend server, where the publish/subscribe protocol was chosen for this purpose. and Eclipse Mosquito broker was installed on the server in the control room.

The ESP32 can play both roles of a pub/sub: receiving data such as students' delay tolerance and ESP config, or sending class events (light, presence, door, etc.) to specified topics, and the Django backend works as a subscriber to process these events instantly.

Device-to-backend communication (ESP32 → Django)

ESP32 devices publish events to the topic pattern:

```
smartclass/classrooms/<classroom_name>/events
```

The platform accepts four types of events from devices, each identified by an `event` or `command` field in the JSON payload:

A. Classroom State / Live Events Used for occupancy and room state changes:

```
{
  "event": "room_state",
  "classroom_name": "f9",
  "data": {
    "occupied": true,
    "door_unlocked": false,
    "danger_indicator": false,
    "teacher_rfid": "123456789123"
  }
}
```

The server ignores any client timestamp and records its own time for all events. Key fields include `classroom_name`, `occupied`, `door`, `teacher_rfid`, and `student_rfids`.

B. Teacher Access Request Sent when a teacher attempts to open a classroom session:

```
{
  "event": "access_request",
  "classroom_name": "f9",
  "data": {
    "teacher_rfid": "123456789123",
    "request_id": "door-req-001"
  }
}
```

The `teacher_rfid` field is required for this type of request.

C. Student Attendance Request Sent after a teacher session is open to mark student arrivals:

```
{
  "event": "attendance_request",
  "classroom_name": "f9",
  "data": {
    "student_rfids": ["1234567891254", "1234567891253"]
  }
}
```

The `student_rfids` field supports single or multiple students in a single payload.

D. Door Delay Request Sent to request the configured door-close delay for the classroom:

```
{
  "command": "door_delay_request",
  "classroom_name": "f9",
  "request_id": "delay-001"
}
```

Backend-to-device communication (Django → ESP32)

The Django backend sends responses and commands back to ESP32 devices using dedicated topics. The response topic pattern is as follows:

```
smartclass/classrooms/<classroom-name>/access/response
```

The access response payload:

```
{
  "event": "access_response",
  "request_id": "door-req-001",
  "approved": true,
  "reason": "authorized_in_time_window"
}
```

If the teacher is an administrator, the response reason is `authorized_admin_override` and the session becomes an inspection session. For door delay requests, the response is a plain number representing the delay in minutes:

```
10
```

Session rules

The system enforces the following session rules:

- Class sessions are open to teachers and can collect student attendance over a duration.
- Inspection sessions are only for administrators, are closed immediately, and do not use duration, reports, or student attendance.
- Attendance records are timestamped by the server, ignoring any client-provided timestamps.
- If there is no open class session, attendance requests are ignored with a warning.
- Teacher access requests first create the session, and then attendance requests fill that open session.

III.6.4 Scheduler and automation engine

The scheduler and automation engine are responsible for managing time-based activities within the smart classroom management system. It operates as a background service that automatically performs critical tasks without requiring manual user intervention.

The engine executes the following automated functions:

- **Session scheduling:** Automatically creates and schedules classroom sessions based on predefined timetables and configuration settings.
- **Session closure:** Monitors active sessions and automatically closes them after their configured expiration time, ensuring that classrooms are not left in an active state indefinitely.
- **Attendance finalization:** Finalizes attendance records when a session ends, locking them for further modifications, and preparing them for report generation.
- **System updates:** Updates classroom status, occupancy indicators, and other system states in real time based on session activity.
- **Ramadan timing support:** Adjusts the scheduling logic to accommodate Ramadan working hours when the corresponding setting is enabled.

The automation engine is implemented as a Django management command that runs at regular intervals, configurable through the system settings. This ensures that all time-based operations are performed consistently and reliably, contributing to the overall efficiency and autonomy of the smart classroom management system.

> **Technical note:** The scheduler interval can be configured via the `cron_interval_minutes` setting in the SystemSettings model, with a default value of 5 minutes. The auto-finish timeout is configured through the `auto_finish_minutes` setting, with a default of 90 minutes.

III.7 Frontend implementation

The frontend implementation of the proposed system refers to the graphical user interface layer of the system that allows users to communicate with the backend system. It consists of a web dashboard that offers intuitive control over classroom activities, system entities, and visualization of real-time data for administrators and teachers.

The frontend solution uses common web technologies and integrates with the Django backend through HTTP requests and API interfaces. Thanks to such integration, dynamic data exchange takes place, with real-time updates being delivered from IoT devices and backend subsystems. Frontend implementation is characterized by usability, responsiveness, and clarity, which makes it possible to ensure efficient system performance.

Generally, the frontend component serves as the presentation layer of the system, presenting all necessary data in dashboards, tables, graphs, and control forms.

III.7.1 Login interface

Figure 3.7 shows the login interface of the smart classroom management system. The interface serves as the point of entry to the system by authenticating a user's identity before granting access to system resources.

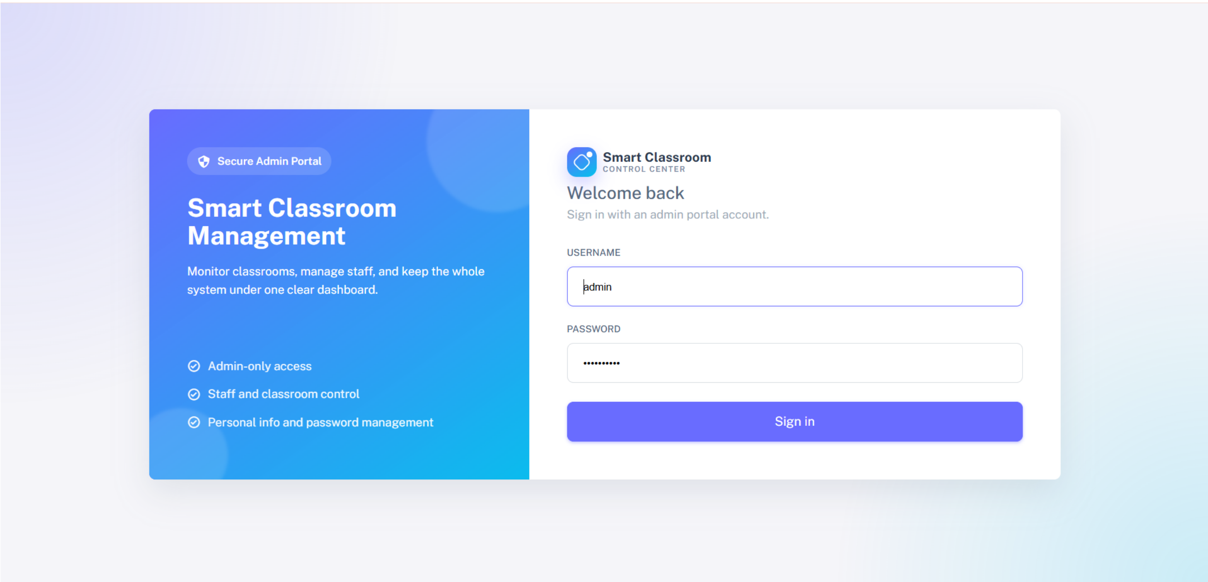


Figure 3.7: Login interface of the smart classroom management system.

The login page contains fields for entering a username and password, along with a sign-in button for submitting credentials. The interface also displays system information and security features such as admin-only access, staff and classroom control, and password management.

To access the system, users are required to enter valid credentials. Once authenticated, users are redirected to their respective dashboards according to their roles in the system.

The login interface improves system security by preventing unauthorized users from accessing protected resources.

III.7.2 Dashboard

Figure 3.8 shows the main dashboard of the smart classroom management system. The dashboard serves as the central hub for administrators, providing a comprehensive overview of the system status and key performance indicators.

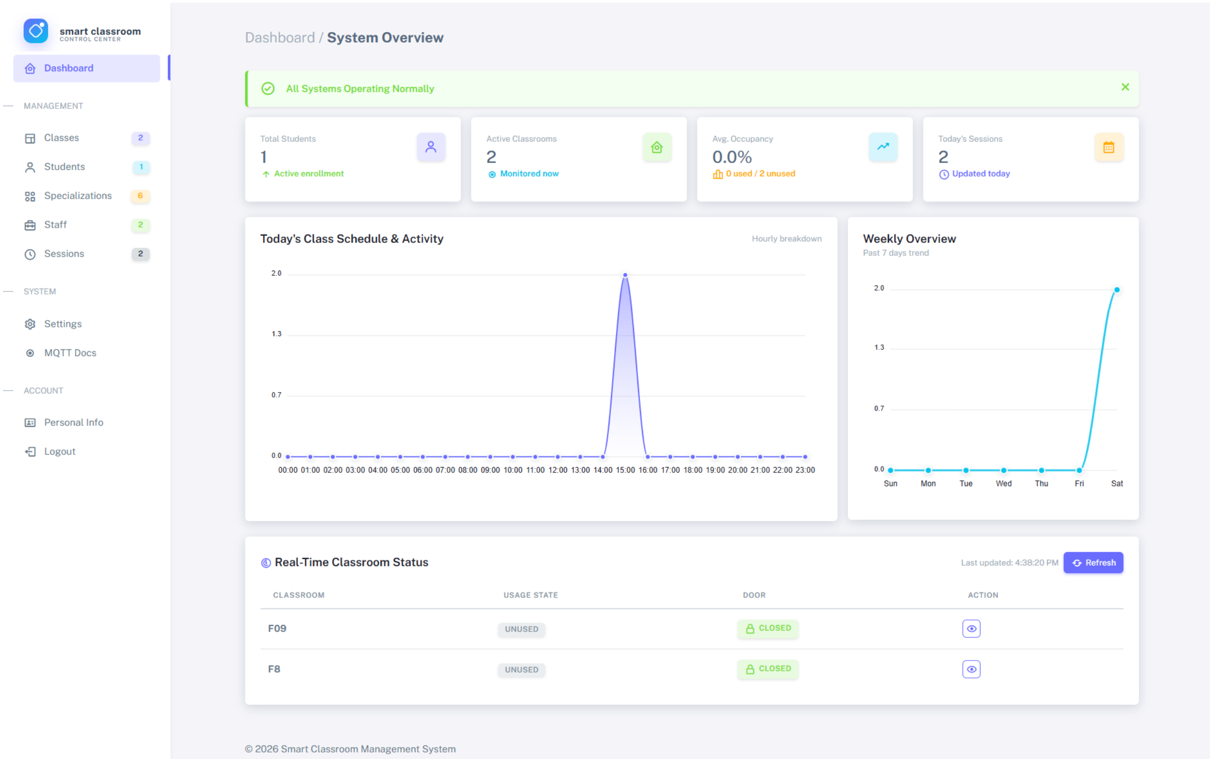


Figure 3.8: Dashboard of the smart classroom management system.

The dashboard displays KPI badges showing the total number of classrooms, active sessions, present students, and staff members. It also includes real-time status cards for each classroom, indicating occupancy, door status, and any active danger indicators. Recent sessions and attendance activities are displayed in tables for quick reference.

The dashboard enables administrators to monitor the entire system from a single interface and provides quick access to all management functions.

III.7.3 Classroom management

Classroom list

Figure 3.9 shows the classroom list page. This page displays all registered classrooms in the system in tabular format with pagination support.

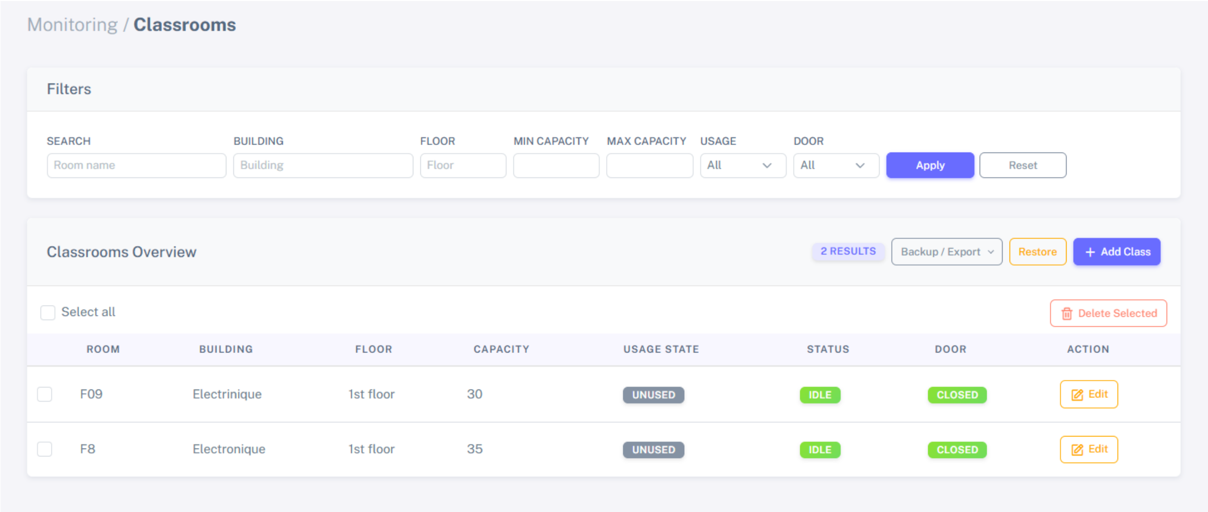


Figure 3.9: Classroom list page with export and restore functionality.

The table includes columns for classroom name, building, floor, capacity, occupancy status, door status, and current session information. Administrators can search, filter, and sort classrooms by various criteria including building, floor, capacity range, and occupancy status. Action buttons allow quick access to classroom details, editing, and deletion.

Additionally, the page includes an **export/backup button** and **restore button** that allow administrators to download the classroom list in Excel/JSON format for offline analysis and record-keeping, as well as load data from a backup JSON file.

Create classroom

Figure 3.10 shows the classroom creation interface. This form allows administrators to add new classrooms to the system.

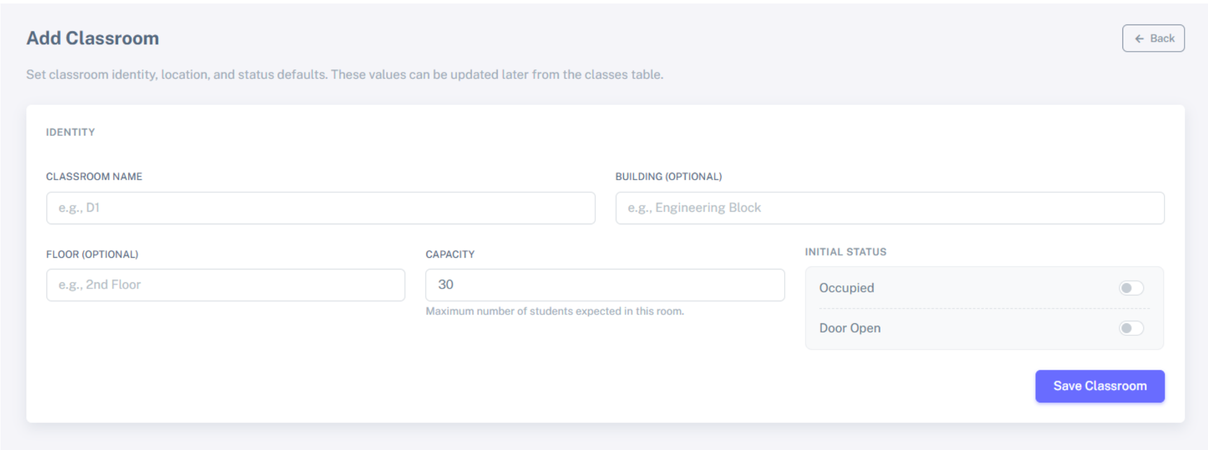


Figure 3.10: Create classroom interface.

The form includes fields for classroom name, building, floor, and capacity. Administrators can also set default status indicators. Upon submission, the system validates the input and creates the classroom record in the database.

Edit classroom

Figure 3.11 shows the classroom edit interface. This form allows administrators to modify existing classroom information.

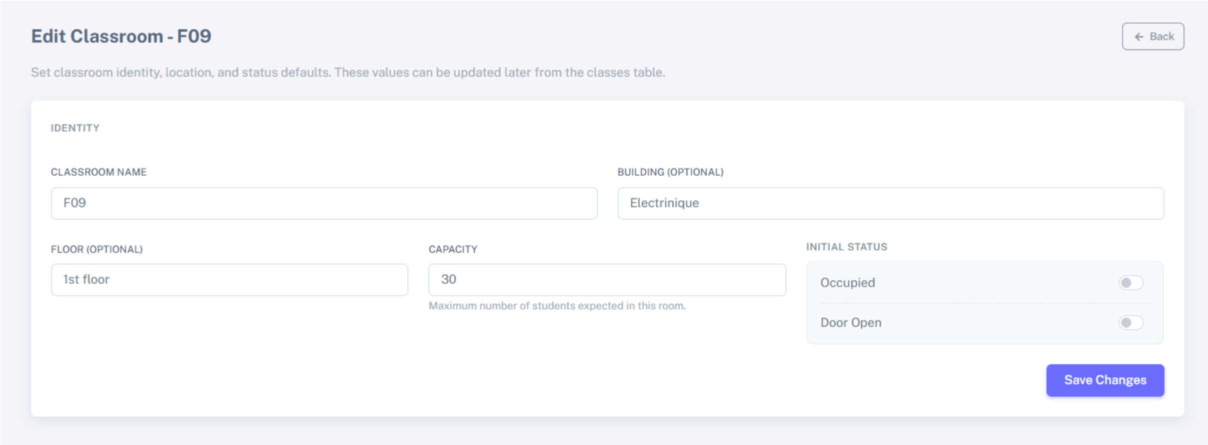


Figure 3.11: Edit classroom interface.

The form includes fields for classroom name, building, floor, and capacity. The form pre-populates with the current classroom data for easy editing. Upon submission, the system validates the input and updates the classroom record.

Classroom details

Figure 3.12 shows the classroom details page. This page provides a comprehensive view of a specific classroom, including its current status, active session, and attendance history.

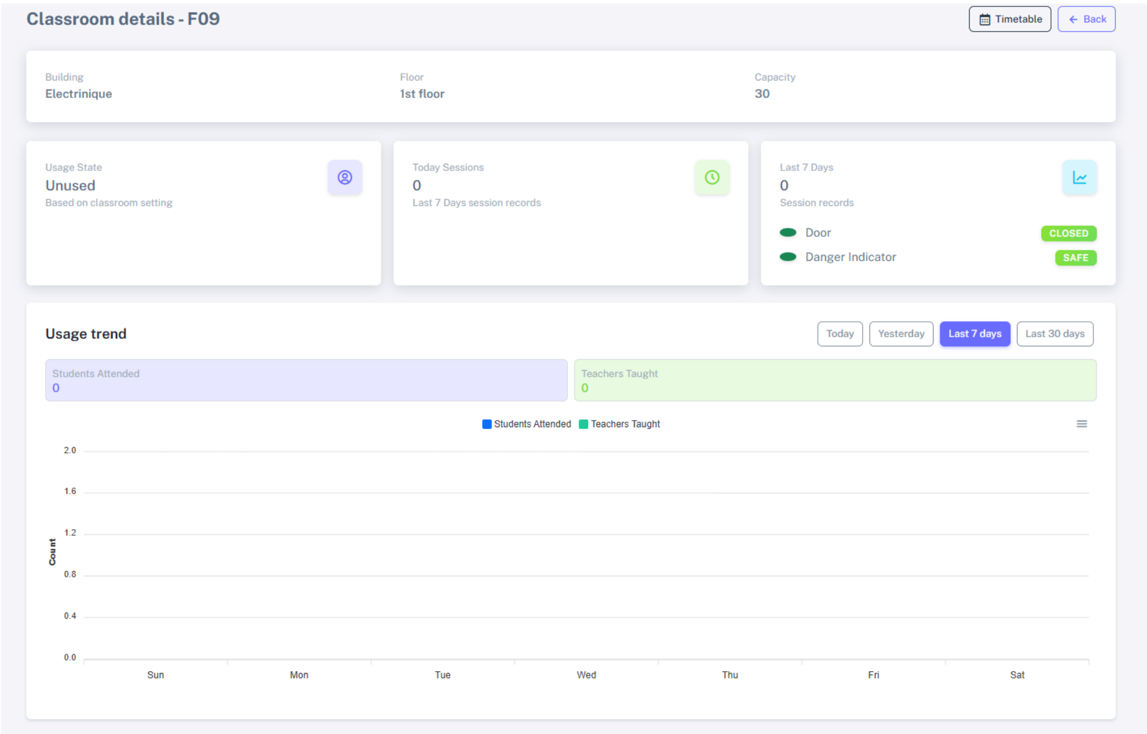


Figure 3.12: Classroom details page with timetable management.

The page displays real-time information such as occupancy, door status, danger indicators, building, floor, capacity, and usage trends. It also includes a **timetable button** that allows the administrator to set or modify classroom-specific timetables.

III.7.4 Session management

Session list

Figure 3.13 shows the session list page. This page displays all classroom sessions with filtering and pagination capabilities.

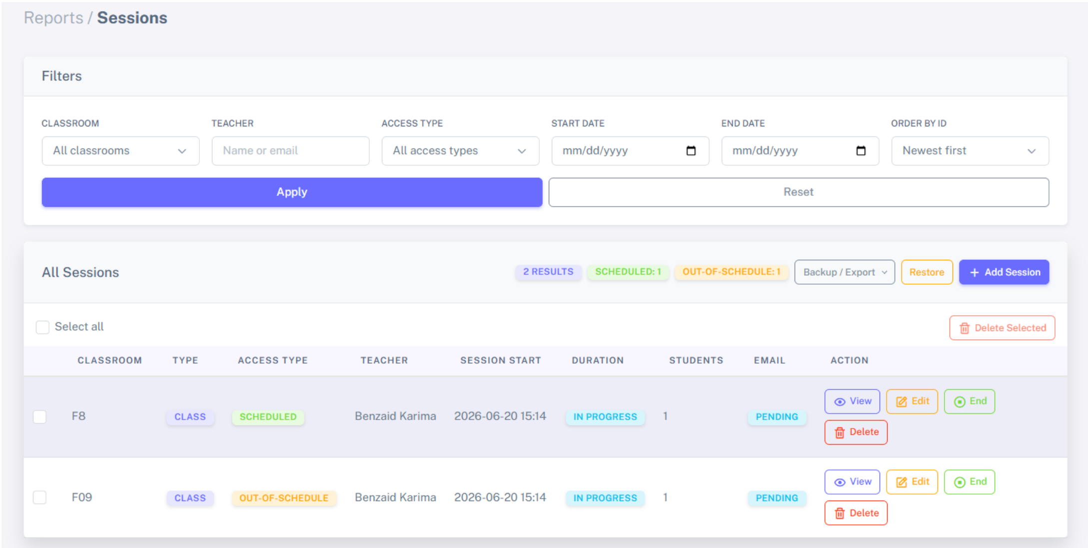


Figure 3.13: Session list page with access type filtering and export functionality.

The table includes columns for classroom, session type (class/inspection), access type (scheduled/out-of-schedule), teacher, start time, duration, number of students, and report email status. Administrators can filter sessions by date, classroom, access type, or status. Action buttons provide access to session details and reports.

The page includes KPI counters showing the number of scheduled and out-of-schedule sessions. Additionally, the page includes an **export/backup button** and **restore button** that allow administrators to download the sessions list in Excel/JSON format for offline analysis and record-keeping, as well as load data from a backup JSON file.

Create session

Figure 3.14 shows the session creation interface. This form allows administrators to manually start a new classroom session.

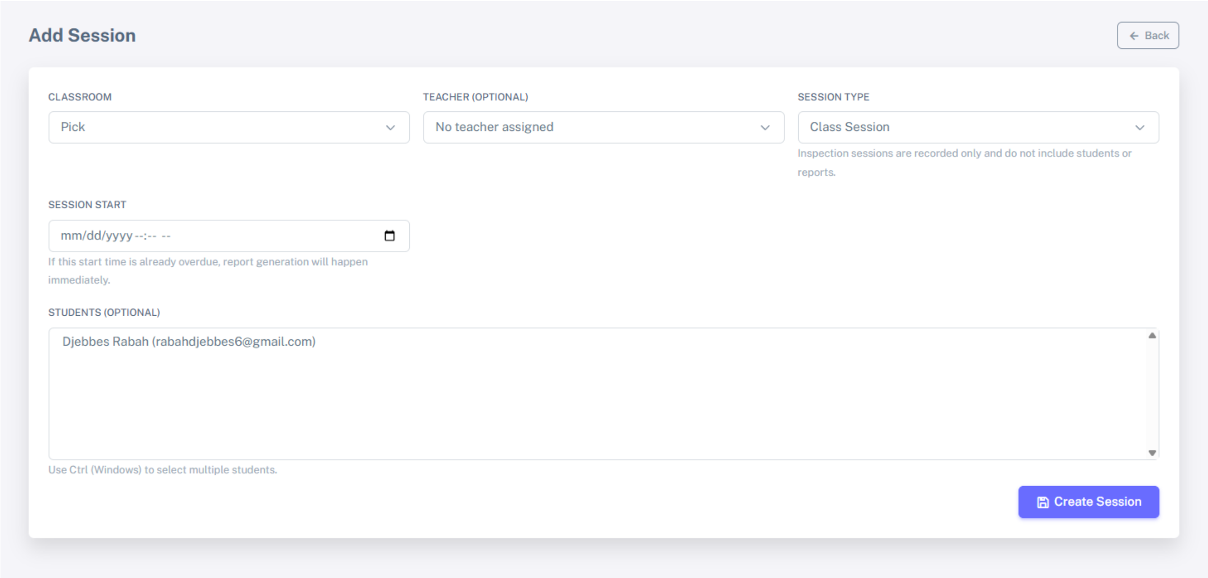


Figure 3.14: Create session interface.

The form includes fields to select classroom, teacher, session type (class or inspection), and access type (scheduled or out-of-schedule). Once created, the session becomes active and attendance recording is enabled.

Edit session

Figure 3.15 shows the session edit interface. This form allows administrators to modify the existing session details.

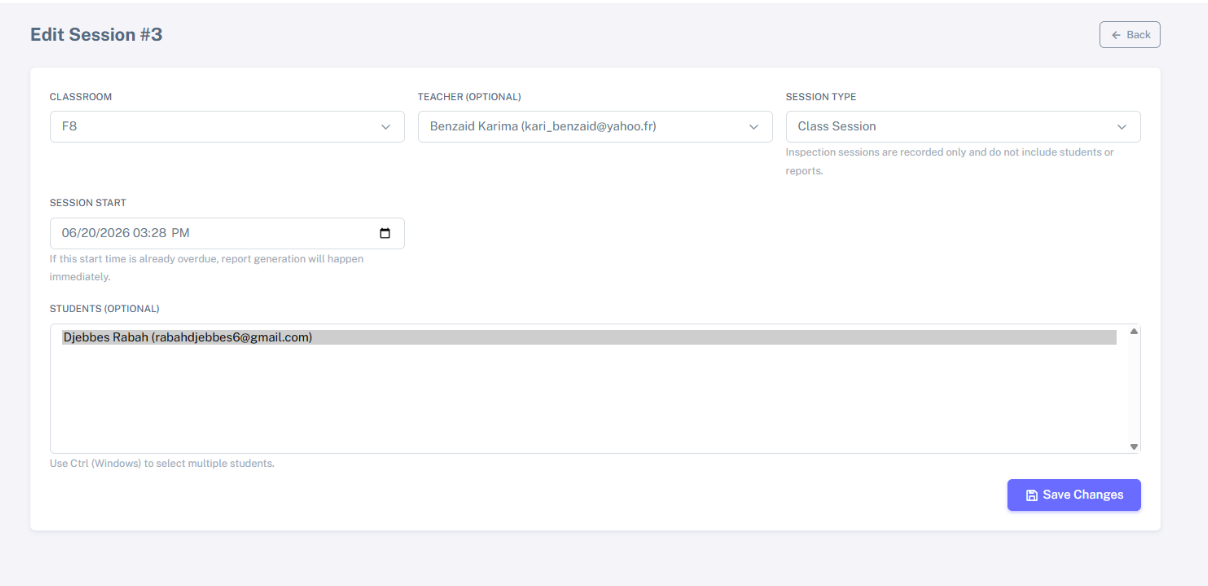


Figure 3.15: Edit session interface.

The form includes fields to update classroom, teacher, session type,. The access type is normalized to either scheduled or out-of-schedule using a centralized resolver.

View session

The system provides two distinct views for session details depending on the session status: active sessions and finished sessions.

- Active session view Figure 3.16 shows the active session details page. This page allows administrators to view comprehensive information about a currently active classroom session.

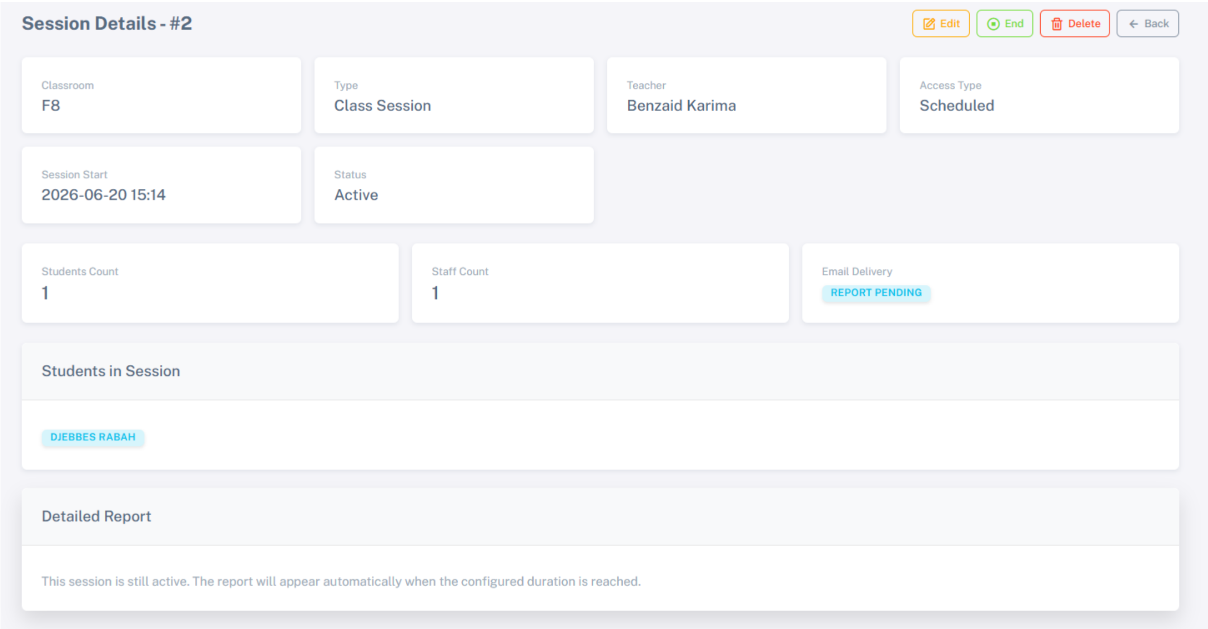


Figure 3.16: Active session details page.

The page displays key session details including classroom name, teacher, session type (class or inspection), access type (scheduled or out-of-schedule), start time, duration, and current status. The access type is normalized using a centralized resolver to ensure consistency across the system.

The page also shows a list of students who have already arrived, along with their arrival times. From this page, administrators can generate a session report, manually close the session, or navigate back to the session list.

- Finished session view Figure 3.17 shows the finished session details page. This page allows administrators to view comprehensive information about a completed classroom session.

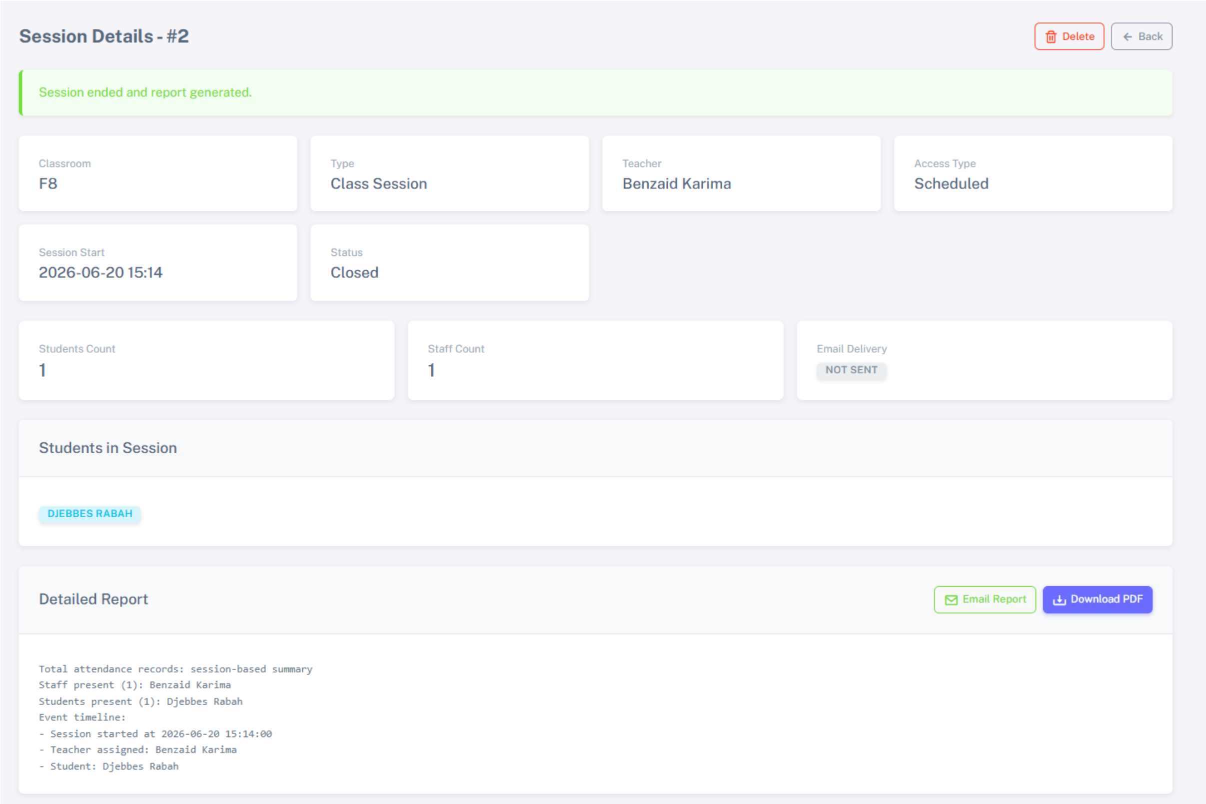


Figure 3.17: Finished session details page.

The page displays key session details including classroom name, teacher, session type, access type, start time, end time, total duration, and final status. The access type is normalized using a centralized resolver to ensure consistency across the system.

The page also shows a complete list of students who attended the session, along with their arrival times. From this page, administrators can generate a session report, view the final attendance summary, or navigate back to the session list.

III.7.5 Attendance Reports

Report design

Figure 3.18 shows the generated attendance report design. Each report is structured as a professional PDF document suitable for administrative and archival purposes.

Attendance Report

Session #3

Date	2026-06-20	**Classroom**	F8
Start Time	15:28:00	**End Time**	15:29:39
Teacher	Benzaid Karima	**Duration**	1 minutes
Generated At	2026-06-20 15:29:39	**Students**	1

Student Roster

Student ID	Student Name	Speciality
202034032186	Djebbes Rabah	Automatique et Informatique Industriel

Figure 3.18: Generated attendance report design.

The report includes the following sections and fields:

- **Report header:** Displays the report title "Attendance Report" along with the session identifier (e.g., Session #3).
- **Session details:** A summary table showing key session information including:
  - Date and name of the classroom
  - Start time and end time
  - Teacher name
  - Session duration
  - Total number of students present
  - Generation timestamp
- **Student roster:** A detailed table listing all students who attended the session, including:
  - Student ID (RFID number)
  - Student name
  - Specialization
- **Report footer:** Includes a confidentiality notice indicating that the report is for internal use only: "Smart Classroom Control System | Confidential".

The report is generated in PDF format, ensuring consistent formatting across all devices and platforms. It can be downloaded directly from the system or automatically sent via email to configured recipients through SMTP settings.

III.7.6 User management

Student list

Figure 3.19 shows the student list page. This page displays all registered students in the system.

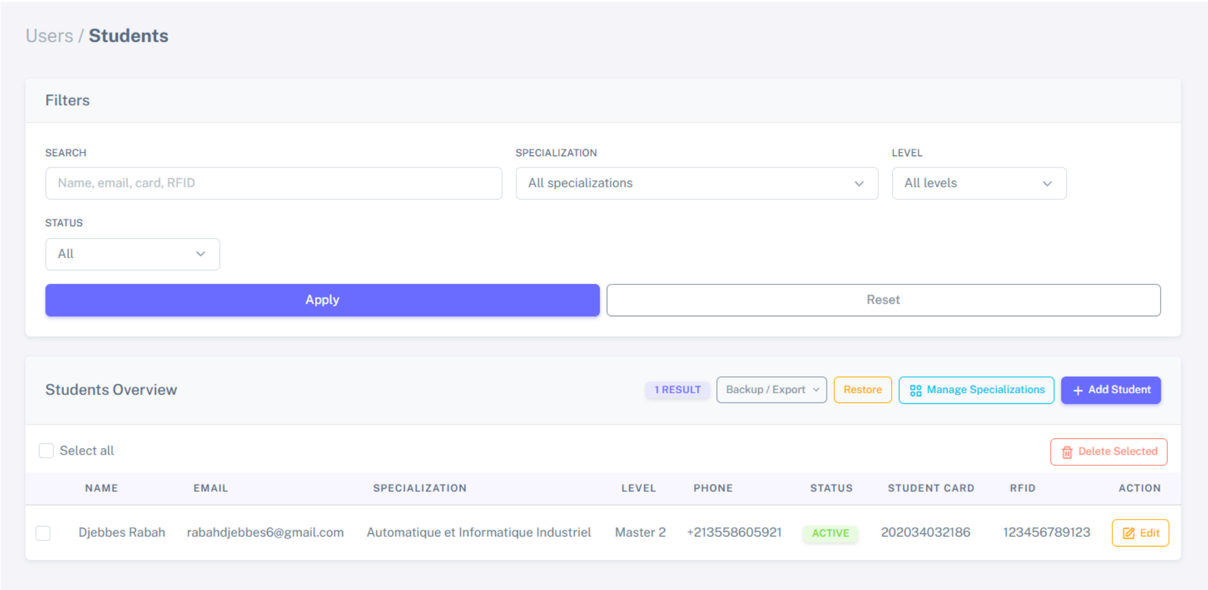


Figure 3.19: Student list page with level and status filtering.

The table includes columns for student name, email, specialization, level (L1, L2, L3, M1, M2), phone number, student card ID, RFID number, and active status. Administrators can search, filter by specialization, level, or status, and manage student records. Action buttons provide access to edit or delete student information.

Create student

Figure 3.20 shows the student creation interface. This form allows administrators to register new students in the system.

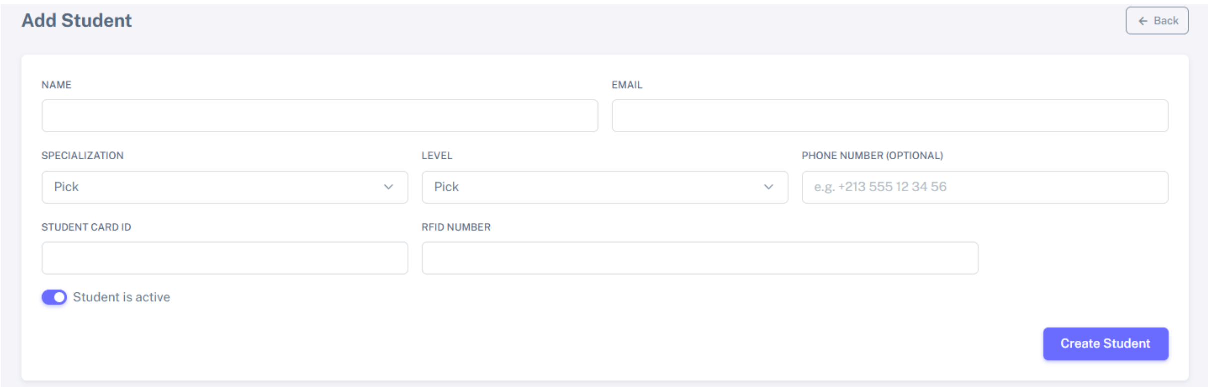


Figure 3.20: Create student interface with specialization catalog.

The form includes fields for student name, email, specialization (selected from the specialization catalog), level (L1, L2, L3, M1, M2), student card ID, RFID number and phone number. All fields are validated before the student record is created in the database.

Edit student

Figure 3.21 shows the student edit interface. This form allows administrators to modify existing student information.

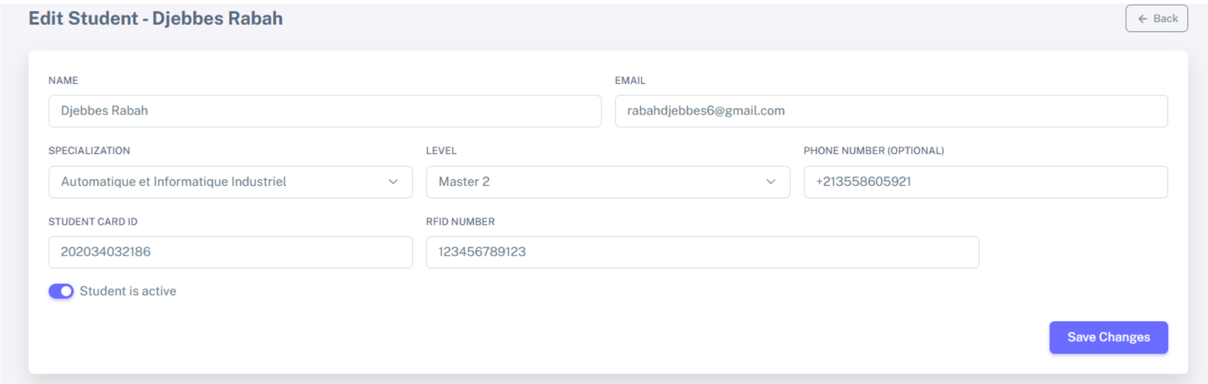


Figure 3.21: Edit student interface.

The form includes fields for student name, email, specialization (from catalog), level, student card ID, RFID number, phone number, and active status. The form pre-populates with current data for easy editing.

Student details

Figure 3.22 shows the student details page. This page provides a comprehensive view of a specific student.

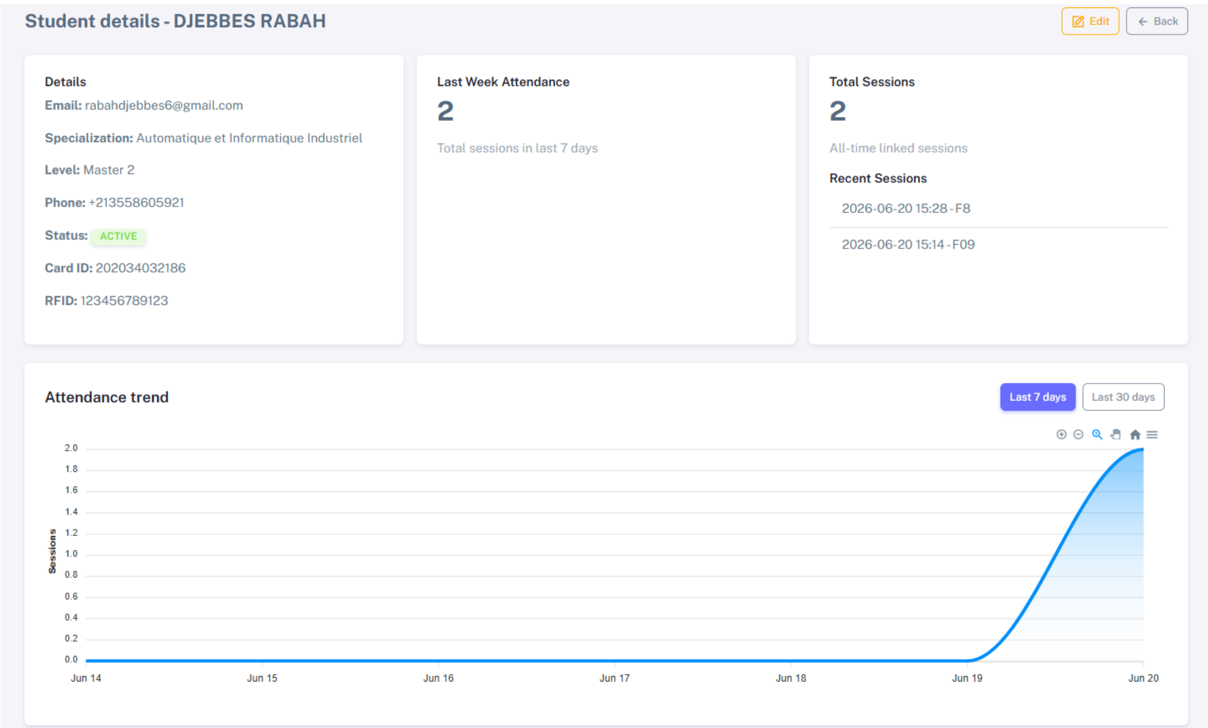


Figure 3.22: Student details page.

The page displays student information including name, email, specialization label, level label, phone number, status, and total sessions attended.

Staff list

Figure 3.23 shows the staff list page. This page displays all registered staff members in the system, including teachers, administrators, and support staff.

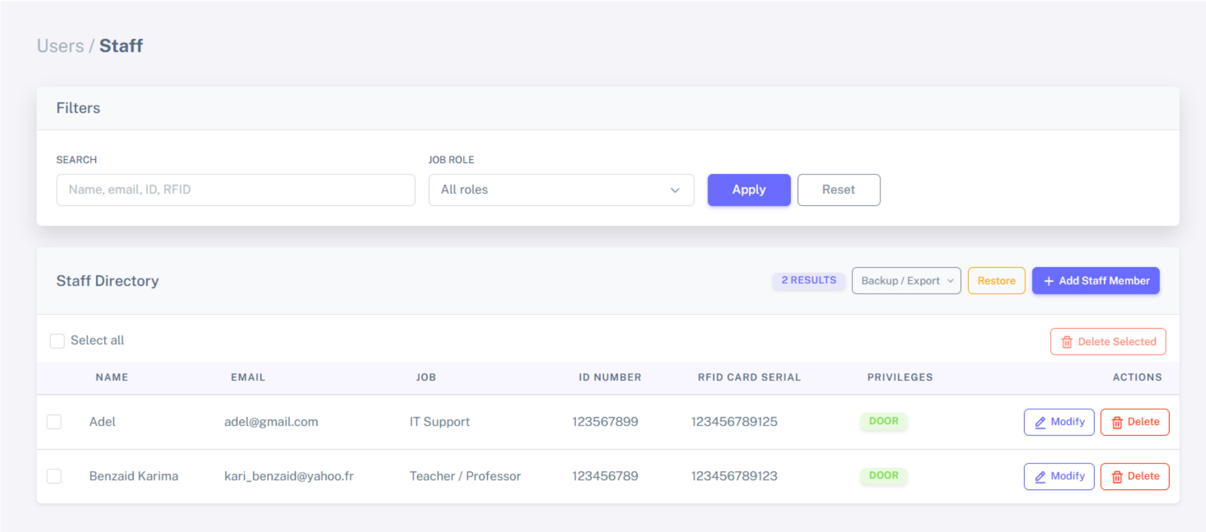


Figure 3.23: Staff list page.

The table includes columns for staff name, email, role, RFID number, and permissions. Administrators can search, filter, and manage staff records.

Create staff

Figure 3.24 shows the create staff interface. This form allows administrators to register new staff members in the system.

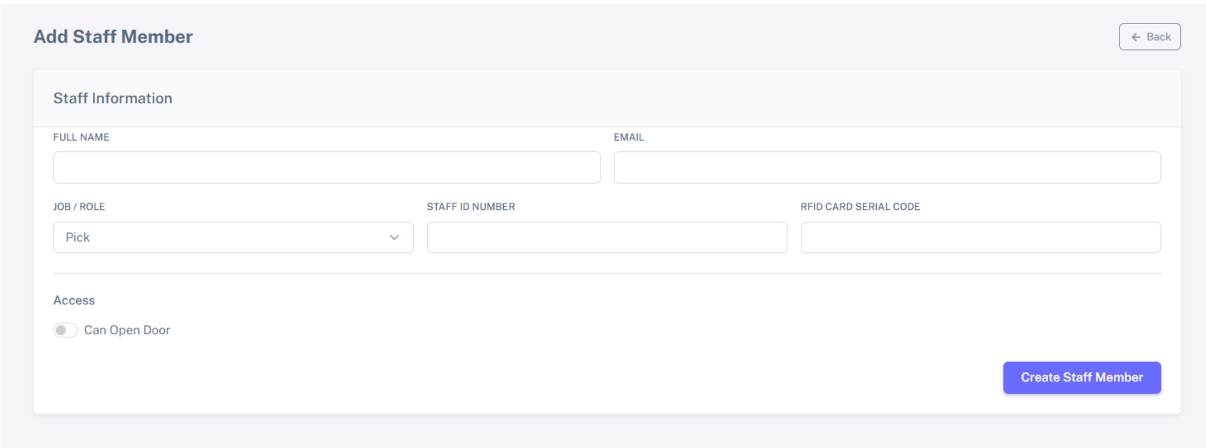


Figure 3.24: Create staff interface.

The form includes fields for staff name, email, role, ID number, and RFID number. Permission checkboxes allow administrators to grant specific capabilities such as door control for this implemented version.

III.7.7 Specialization management

Specialization list

Figure 3.25 shows the specialization catalog page. This page displays all registered specializations in the system.

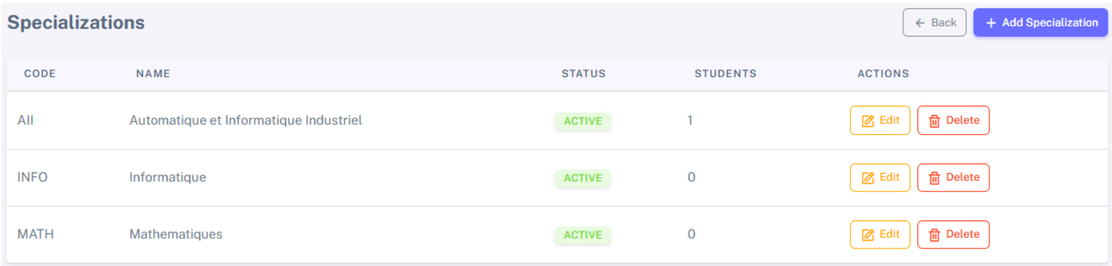


Figure 3.25: Specialization catalog page.

The table includes columns for specialization code, name, active status, and student usage count. Administrators can view, edit, or delete specializations. Deleting a specialization removes it from the catalog immediately, and students already linked to that code will keep the stored code value until manually reassigned.

Create specialization

Figure 3.26 shows the create specialization interface. This form allows administrators to add new specializations to the catalog.

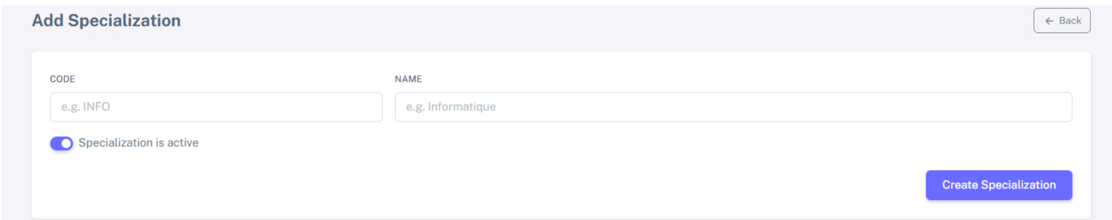


Figure 3.26: Create specialization interface.

The form includes fields for specialization code and name. The system checks for duplicates before creating the record.

Edit specialization

Figure 3.27 shows the edit specialization interface. This form allows administrators to modify existing specializations.

Figure 3.27: Edit specialization interface.

The form includes fields for specialization code and name. When the code is changed, the system automatically re-maps student specialization codes to the new value.

III.7.8 Settings

Figure 3.28 shows the settings page. This page allows administrators to configure all system parameters from a single centralized interface. The settings are organized into logical sections for ease of use and clarity.

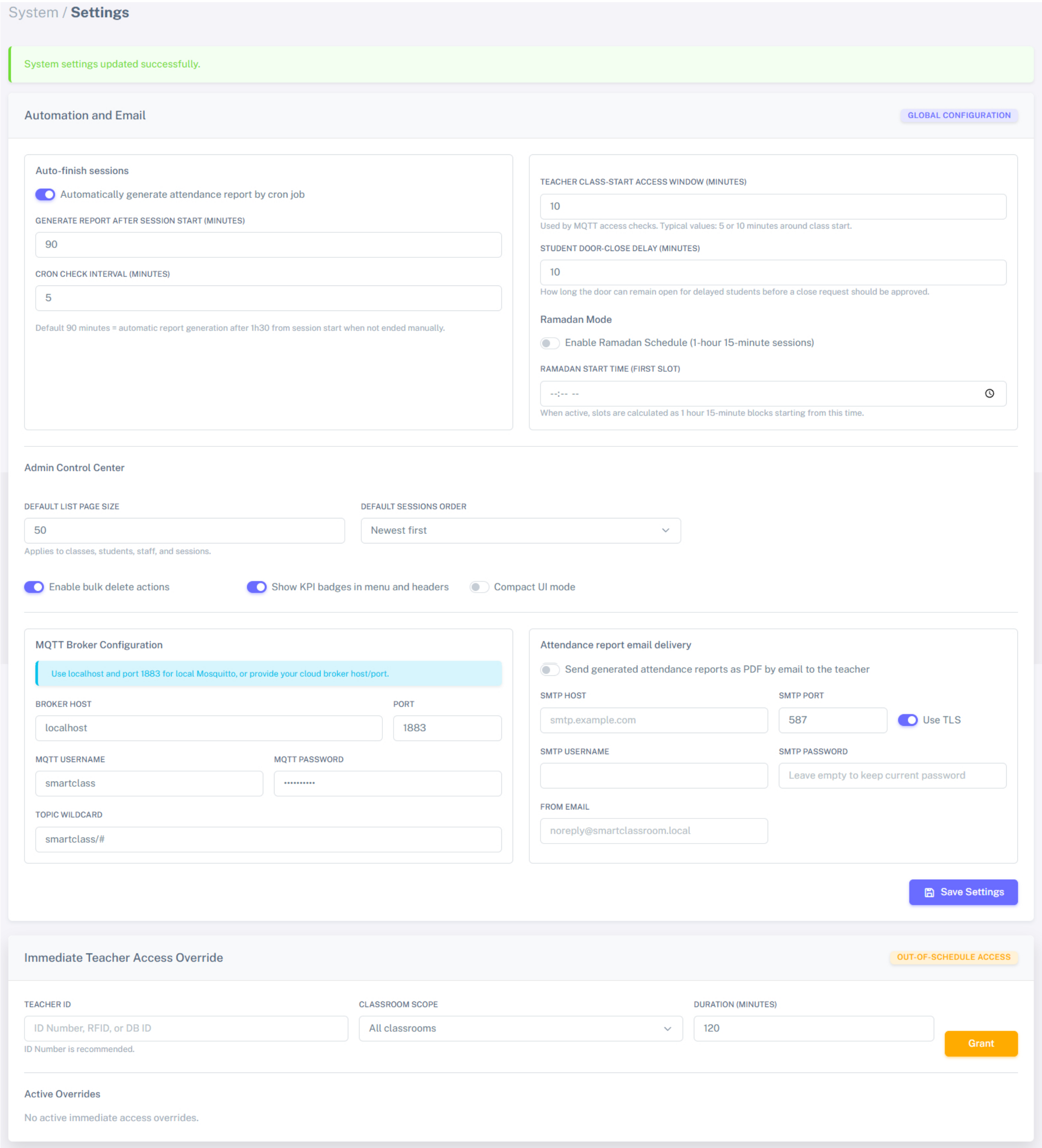


Figure 3.28: Settings page of the smart classroom management system.

Automation and email settings

This section controls the automated behavior of the system, including session auto-finish, cron job intervals, and email delivery of reports.

- **Auto-finish sessions:** enable automatic generation of attendance reports by the cron job. When enabled, sessions are automatically closed and reports are generated without manual intervention.
- **Generate a report after session start (minutes):** Defines the duration (in minutes) after which a session is automatically closed and a report is generated. The default value is 90 minutes, meaning that sessions are automatically finalized 90 minutes after they start.
- **CRON check interval (minutes):** Defines how frequently the system checks for expired sessions. The default value is 5 minutes, ensuring timely detection and closure of sessions.
- **Teacher class-start access window (minutes):** Defines the time window (in minutes) during which a teacher can start a scheduled session. The default value is 10 minutes.
- **Student door-close delay (minutes):** Defines how long the door can remain open for delayed students before a close request should be approved. The default value is 10 minutes.

Ramadan mode

This section provides special scheduling support for the holy month of Ramadan, during which working hours are typically reduced.

- **Ramadan mode:** When enabled, the system adjusts session scheduling to accommodate Ramadan working hours.
- **Ramadan start time (first slot):** Defines the start time of the first session slot during Ramadan. The default value is 08:30 AM. When active, slots are calculated as 1 hour 15-minute blocks starting from this time, instead of the standard 90-minute sessions.

Admin control center

This section allows administrators to customize the user interface and data presentation preferences.

- **Default list page size:** Defines the number of items displayed per page in list views (classes, students, staff, sessions). The default value is 50.
- **Default sessions order:** Defines the default sorting order for session lists. Options include "Newest first" or other ordering criteria.
- **Enable bulk delete actions:** When enabled, administrators can select and delete multiple records simultaneously from list views.
- **Show KPI badges:** When enabled, key performance indicator badges are displayed in menus and headers for quick status overview.
- **Compact UI mode:** When enabled, the interface uses a more compact layout, allowing more information to be displayed on screen.

MQTT broker configuration

This section configures the MQTT communication between the system and IoT devices.

- **Broker host:** The IP address or hostname of the MQTT broker. For local development, "localhost" is used. For cloud deployment, the cloud broker hostname is provided.
- **Port:** The port on which the MQTT broker is listening. The default port is 1883 for unencrypted communication, or 8883 for TLS-enabled communication.
- **MQTT username:** The username for authenticating with the MQTT broker (if required).
- **MQTT password:** The password for authenticating with the MQTT broker (if required).
- **Topic wildcard:** The topic pattern used for communication. The default wildcard is `smartclass/#`, which subscribes to all topics under the `smartclass/` hierarchy.

Email delivery settings

This section configures the SMTP email delivery system for sending attendance reports.

- **Attendance report email delivery:** When enabled, generated attendance reports are automatically sent as PDF attachments to teachers via email.
- **SMTP host:** The SMTP server hostname (e.g., `smtp.example.com`).
- **SMTP port:** The SMTP server port. The default port for TLS is 587.
- **SMTP username:** The username for authenticating with the SMTP server.
- **SMTP password:** The password for authenticating with the SMTP server.
- **From email:** The sender email address for outgoing reports (e.g., `noreply@smartclassroom.loca`

Immediate teacher access override

This section allows administrators to grant temporary access permissions to teachers.

- **Teacher ID:** The ID number, RFID, or database ID of the teacher receiving temporary access.
- **Classroom scope:** Defines which classrooms the teacher can access. Options include a specific classroom or "All classrooms".
- **Duration (minutes):** The duration (in minutes) for which the access override remains active.
- **Active Overrides:** Displays a list of currently active access overrides, including the teacher, classroom scope, and expiration time.

III.7.9 MQTT integration documentation

Figure 3.29 shows the MQTT integration documentation page. This page provides developers with comprehensive information on the MQTT communication protocol used by the system. It serves as a reference for hardware developers who integrate ESP32 devices with the platform.

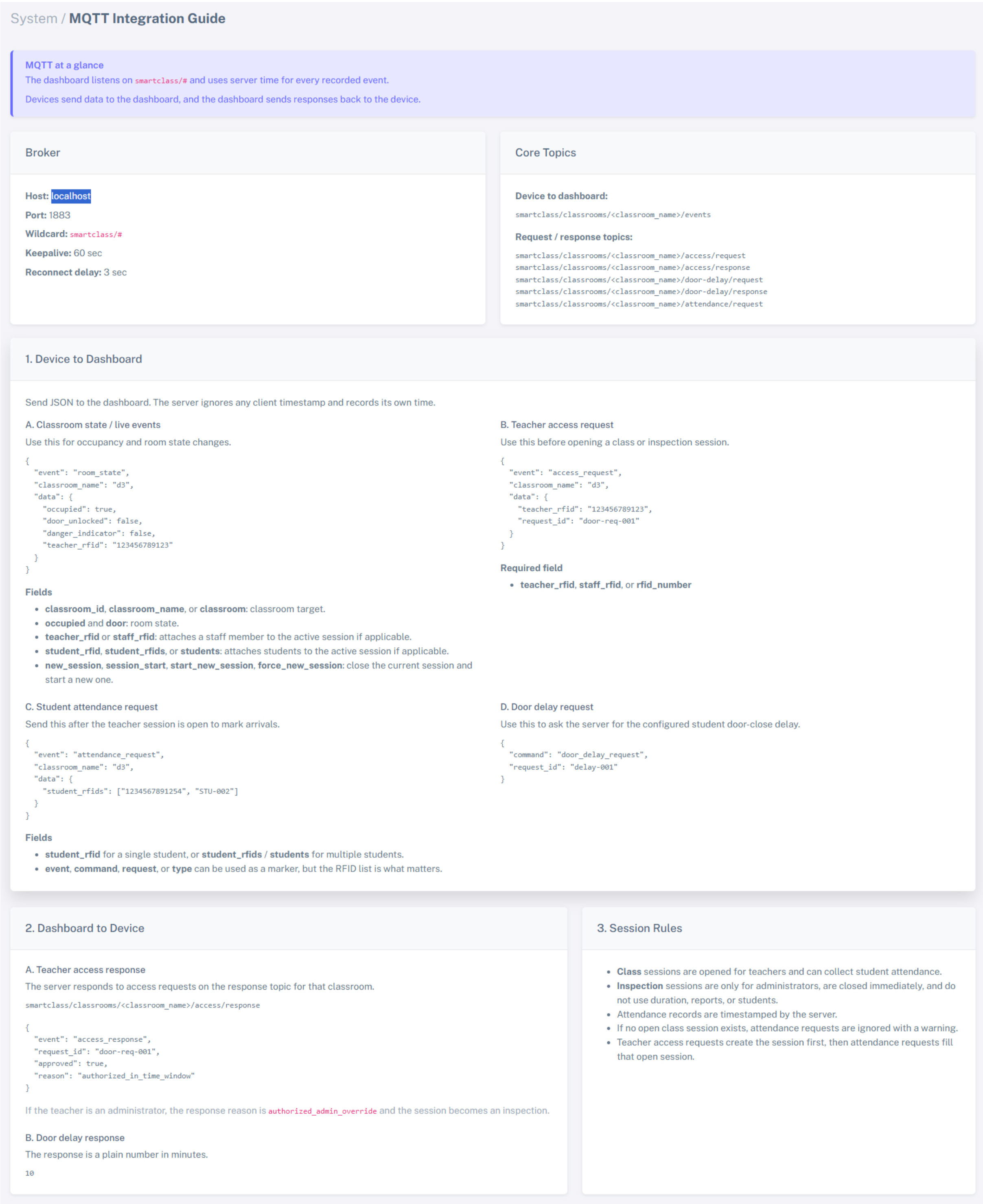


Figure 3.29: MQTT integration documentation page.

The MQTT integration follows a publish/subscribe model in which devices send data to the dashboard, and the dashboard sends responses back to the device. The dashboard listens to the topic wildcard `smartclass/#` and uses server time for every recorded event, ignoring any client-provided timestamps to ensure consistency and prevent tampering.

III.8 Technical discussion

In terms of the choice of protocols used in the proposed project for the interaction of components, HTTP and MQTT were selected to cover the needs of the system. Thus, HTTP will be used for communication between the user and the web dashboard related to the authentication process, managing the classroom settings, and generating reports, whereas MQTT will be used for communicating with the ESP32 sensors and actuators as well as between them and the backend server. MQTT provides benefits such as reduced bandwidth usage, lightweight messages, and a pub-sub architecture, which makes it an efficient option for real-time communication between devices in an IoT setting. Moreover, Django framework was decided to be used on the back end as opposed to Node.js, because it is equipped with security mechanisms, a powerful ORM, authentication and rapid development options. The platform was implemented using SQLite database and a local server, to make the deployment process easier and less costly. Although this choice makes system administration easy, it can cause limitations in scaling in larger educational organizations. Finally, RFID technology was selected due to its affordability and ease of use, although there are more automated alternatives, like face recognition.

III.9 System limitations

Although the system has been successfully implemented and tested, there are still some limitations in the designed system. Firstly, the existing system relies mainly on RFID card usage to track students' attendance and therefore requires students to bring and use their RFID cards properly. Lost, broken, or misplaced RFID cards can become obstacles to tracking students' attendance accurately. Secondly, the existing system uses a local server architecture with SQLite as the database; such architecture is suitable for small-to-medium-scale environments; however, it can face some challenges when scaling up the number of students, rooms, and users involved. Thirdly, the existing platform requires a local network for IoT devices to communicate with the backend server. Moreover, some advanced functions such as biometrics, remote smartphone connectivity, cloud deployment, and intelligent data analysis are not available in the existing system. Even though these limitations do not affect the primary function of the designed system, they might limit its scalability and versatility.

III.10 Conclusion

In this chapter, the design and implementation of the proposed smart classroom management system have been provided. It is shown how the requirements mentioned in chapter 2 are designed and implemented into a fully-fledged IoT system.

Firstly, the chapter introduced the components of system design, such as the global architecture, hardware architecture, data flow architecture, backend architecture, frontend architecture, and security architecture. They are critical parts of the system design and provide an overview of how the components in an IoT system work together.

Secondly, the implementation process, which includes the development environment, backend implementation, frontend interface, and system integration, is described in great detail. With the help of Django, the MQTT protocol, RFID technology, and ESP32 microcontrollers, the intelligent system can be developed to manage classrooms efficiently.

Lastly, the technical discussion, system limitations, and future work are mentioned to emphasize important design considerations, current problems in the system, and potential future work on improving the system.

In summary, this chapter shows that the architectural design in previous chapters has been successfully applied in developing the smart classroom management system.

General conclusion

This study discussed the development and design of a smart classroom management system that leverages the Internet of Things (IoT). The aim of designing such a system was to create an intelligent and automated solution that can help enhance classroom management processes by improving monitoring capabilities and increasing overall efficiency.

The development of such a smart system relies heavily on different kinds of technologies that can be used to meet the above-stated aims and objectives. Firstly, Radio-Frequency Identification (RFID) technology has been employed in order to identify students and automate attendance procedures in the classroom. The ESP32 is used as the main hardware component of the system in order to acquire data from sensors and control the operation of different devices used in classroom management. Communication between different IoT components has been achieved through the use of the MQTT protocol, which allows messages to be exchanged quickly and efficiently. On the other hand, the Django framework is used to build a sophisticated backend system for the application. Both functional and non-functional requirements have been taken into account while analyzing the requirements of the system, which helped define the behavior of the system and the constraints regarding its quality.

After the analysis, an overall architecture for the system was developed. The architecture includes hardware architecture, software architecture, communication architecture, security architecture, and data flow architecture. This architecture was then deployed and tested to evaluate its efficiency. The implemented system helps in performing tasks such as automated attendance management, classroom management, session management, user management, and report management, among others. These features help reduce human involvement in classroom management.

Nevertheless, despite the positive results obtained during the implementation process of the proposed IoT application, some limitations have been identified. First of all, the dependence of the system on RFID cards is one of its main drawbacks, since there is always a possibility that students or teachers may lose or damage their tags. In addition, the use of a local deployment environment and an SQLite database can become a limitation when working with a large number of devices and users. The necessity of a local network for communication between IoT devices and the web application represents another restriction on the accessibility of the developed system.

On the other hand, these limitations make it possible to consider several improvements in the future, including the development of a cloud-based system, mobile applications,database migration to PostgreSQL, the use of facial recognition systems for user identification, and the optimization of attendance data analysis using artificial intelligence. In general, it can be concluded that the smart classroom management system demonstrates the effective integration of IoT and web development technologies for creating a centralized educational management solution. This solution serves as a solid foundation for the further development of smarter and more scalable campus management systems.

Bibliography

[1] A. A. A. Sen, Y. M. Aljohani, A. A. H. Hakami, N. M. Bahbouh, A. Alkhodre, and M. S. Alghamdi, “A framework for moving from traditional to smart university,” in 2023 10th International Conference on Computing for Sustainable Global Development, 2023, pp. 562–569.

[2] S. Adipat and R. Chotikapanich, “Advancing higher education with the transition to smart universities: A focus on technology,” Shanlax International Journal of Education, vol. 12, no. 3, pp. 29–36, 2024.

[3] N. E. H. Dahmani and F. Kafi, “Digital transformation: Establishing smart university,” Finance and Business Economies Review, vol. 9, no. 1, pp. 42–50, 2025.

[4] K. Joshi, R. Kumar, S. Bharany, D. J. B. Saini, R. Kumar, A. O. Ibrahim, A. Abdelmaboud, W. A. E. Nagmeldin, and M. Medani, “Exploring the connectivity between education 4.0 and classroom 4.0: Technologies, student perspectives, and engagement in the digital era,” IEEE Access, vol. 12, pp. 24 179–24 204, 2024.

[5] P. S. P. Kumar, P. Negi and P. Mittal, “IoT-enabled smart classrooms for enhanced learning environments,” in 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), 2025, pp. 944–949.

[6] X. Zhang, Y. Ding, X. Huang, W. Li, L. Long, and S. Ding, “Smart classrooms: How sensors and AI are shaping educational paradigms,” Sensors, vol. 24, no. 17, 2024.

[7] H. E. Mrabet and A. A. Moussa, “IoT-school attendance system using RFID technology,” International Journal of Interactive Mobile Technologies (IJIM), vol. 14, pp. 95–108, 2020.

[8] N. M. Radwan and J. Alves-Foss, “MQTT in focus: Understanding the protocol and its recent advancements,” International Journal of Computer Science and Security (IJCSS), vol. 18, no. 1, pp. 1–14, 2024.

[9] P. A and V. T, “IoT-based smart university using MQTT protocol: A scalable and sustainable design approach,” in 2024 5th International Conference on Communication, Computing & Industry 6.0 (C2I6), 2024, pp. 1–7.

[10] W. M. Salama and M. H. Aly, “Web-based integrated school management system for enhanced student-teacher and resource accessibility,” in 2024 34th International Conference on Computer Theory and Applications (ICCTA), 2024, pp. 115–122.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[12] K. Ashton, "That "internet of things" thing," RFID Journal, 2009.

[13] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, 2014.

[14] P. Sethi and S. R. Sarangi, "Internet of things: Architectures, protocols and applications," Journal of Electrical and Computer Engineering, 2017.

[15] K. Umapathy and S. Chandra-Mohan, "Internet of things (IoT) enabled smart classroom with real-time data monitoring," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, pp. 4933–4937, 2019.