جامعة القدس
Al-Quds University

FACULTY OF ENGINEERING
DEPARTMENT OF COMUPTER ENGINEERING
ENGINEERING GRADUATION PROJECT- 0702592

"Smart Attendance System"

Using Facial & Fingerprint Recognition Technologies

Prepared by:

Waseem Abunada	22010182
Mohammad Ashour	22010558
Mohammad Herzallah	22011467

Supervised By

Dr.Imad Hamadeh

JERUSALEM – PALESTINE 2025

Table of Contents

List of Tables

List of Figures

Acknowledgment

Praise be to Allah, the Most Gracious, the Most Merciful, for His blessings and guidance throughout our academic journey.

First and foremost, we would like to express our deepest appreciation to our advisor, Dr. Imad Hamadeh, for his valuable guidance, mentorship and expertise throughout the entire project. His constructive comments and unwavering support were instrumental in shaping the direction and quality of our work.

We also extend our appreciation to the faculty members of the Department of Computer Engineering for providing a stimulating academic environment and sharing their knowledge and insights. The resources and facilities provided by the department were crucial in conducting the research and experiments for this project.

We would like to express our sincere thanks to our families for their continuous support, encouragement and understanding throughout our academic journey. Their unwavering belief in our abilities and continuous support have been the driving force behind our achievements.

We are also indebted to our colleagues and friends who have been a source of inspiration and motivation. The spirit of collaboration and exchange of ideas within the academic community has greatly impacted our learning experience.
Finally, we are grateful for the collective efforts of everyone who played a role in shaping our academic and professional development. We truly appreciate the opportunities and experiences that have come our way, and we will continue to strive for excellence in all our future endeavors.

Abstract

This project proposes the design of a Smart Attendance System (SAS) that utilizes facial and fingerprint recognition technologies to record attendance accurately and efficiently. The system aims to streamline attendance tracking in universities and lectures while ensuring data security and reliability.

Current attendance systems, such as **ZKTeco** or **PASS**, rely on single-factor authentication, leading to challenges like:

- Difficulty recognizing faces in poor lighting or with changes in appearance.
- Ineffectiveness when fingerprints are smudged or unreadable.
- Limited scalability and lack of real-time monitoring for academic needs.

This project introduces a dual-authentication mechanism combining facial recognition and fingerprint verification for enhanced reliability. Individuals' facial features and fingerprints are securely stored during registration to prevent tampering or impersonation.

Attendance is first recorded using a high-precision **Hikvision IP camera** for facial recognition. If it fails, fingerprint verification ensures accurate tracking.

The system includes a modern web interface built with **React** and **MongoDB**, offering:

- Real-time attendance visualization.
- Customizable reports for analysis.
- Alerts for anomalies like absences or errors.

By addressing the limitations of traditional systems, this project provides a robust, scalable, and efficient solution tailored to universities, significantly reducing errors and improving attendance management.

الملخص

يقترح هذا المشروع تصميم نظام حضور ذكي (SAS) يعتمد على تقنيات التعرف على الوجه وبصمة الإصبع لتسجيل الحضور بدقة وكفاءة. يهدف النظام إلى تسهيل تتبع الحضور في الجامعات والمحاضرات مع ضمان أمان البيانات ومصداقيتها.

تعتمد أنظمة الحضور الحالية، مثل **ZKTeco**أو **PASS**، على المصادقة بعامل واحد فقط، مما يؤدي إلى تحديات مثل:

- صعوبة التعرف على الوجوه في الإضاءة السيئة أو مع تغييرات في المظهر.
- عدم فعالية النظام عند وجود بصمات غير واضحة أو تالفة.
- محدودية التوسع وعدم وجود مراقبة لحظية تلائم احتياجات الأكاديميات.

يقدم هذا المشروع آلية مصادقة مزدوجة تجمع بين التعرف على الوجه والتحقق من بصمة الإصبع لضمان موثوقية أكبر. يتم تخزين ملامح الوجه وبصمات الأصابع بشكل آمن أثناء التسجيل لمنع التلاعب أو الانتحال.

يتم تسجيل الحضور أولاً باستخدام كاميرا دقيقة من نوع **Hikvision IP**للتعرف على الوجه، وإذا فشلت العملية، يتم استخدام بصمة الإصبع لضمان تسجيل دقيق.

يتضمن النظام واجهة ويب حديثة مبنية باستخدام **React**و**MongoDB**، وتوفر:

- عرض لحظي لبيانات الحضور.
- تقارير قابلة للتخصيص لتحليل الحضور.
- تنبيهات عن أي حالات شاذة مثل الغياب أو الأخطاء.

من خلال معالجة قيود الأنظمة التقليدية، يوفر هذا المشروع حلاً قويًا وقابلًا للتوسع وفعالًا يلبي احتياجات الجامعات، مما يقلل بشكل كبير من الأخطاء ويحسن إدارة الحضور.

Preface

Smart Attendance System is an advanced system that relies on facial and fingerprint recognition technology to record attendance accurately and efficiently. Facial and fingerprint information is stored in a secure database, and when attendance is recorded, the system identifies students by scanning their faces or fingerprints. This system is characterized by ease of expansion, high security, and the ability to monitor instantly.

The smart attendance system is a major step towards developing attendance management methods. By using modern technologies, the system provides a reliable and effective solution that simplifies administrative processes and enhances security and efficiency.

➢ Chapter One – Introduction

1.1 Overview

The Smart Attendance System (SAS) using facial and fingerprint recognition technologies is an advanced solution designed to streamline attendance management processes at the university. This system integrates biometric recognition technology to accurately identify and record the attendance of individuals in real time. By combining facial and fingerprint recognition technologies, this overview section provides a brief introduction to SAS, highlighting its key features, benefits, and applications at the university. **[1]**

One of the key features of SAS is its dual biometric approach, which leverages facial and fingerprint recognition to ensure a high level of accuracy and security. Facial recognition uses advanced algorithms to analyze unique facial features, while fingerprint recognition relies on the distinctive patterns of an individual's fingerprints. The combination of these technologies reduces the risk of misidentification and unauthorized attendance tags. **[1]**

SAS is particularly useful in educational institutions, where accurate attendance records are critical for compliance and performance tracking. The system can be integrated with existing student information systems to further streamline processes.

1.2 Problem Statement

Traditional attendance tracking methods in universities often face significant challenges. Manual attendance records are prone to errors, and it's not uncommon for students to manipulate the system, such as marking attendance for someone who isn't present. Additionally, relying on paper or even electronic inputs can be time-consuming for professors, leading to delays in processing and verifying attendance data.

1.3 Solution

To address these issues, universities can adopt modern solutions like the Smart Attendance System (SAS), which uses advanced technologies such as facial recognition and fingerprint scanning. This system ensures higher accuracy, prevents fraudulent practices, and reduces the time required for attendance tracking. By implementing SAS, universities can streamline the process, provide a more efficient system for students, and ensure reliable attendance records

1.4 Add Value

Our project includes a system similar to what is currently available in companies, where companies currently use fingerprint time recording. Regarding the points that are worth the value in addition to what we offer in our project:

- Integrating the camera with the system:
  A camera in the system takes a picture every 10 minutes.

- Recognition scale:
  Attendance is recorded using the face print captured from the captured images.

- Comparing images with the database:
  The captured face print and fingerprint can be compared to the stored identity database.

- Automatic attendance recording:
  Attendance is recorded accurately and clearly on the face and fingerprint.

With this scale, it provides us with an advanced and accurate attendance system using wide-range identification technology, which increases the efficiency and technology of attendance recording.

1.5 Project Objectives

The main objectives of developing this project are:

- Registering the student's presence or absence through face or fingerprint detection and recognition.
- Making the attendance report given to the doctor in an easier and faster way.
- Completing one of the requirements for obtaining a bachelor's degree in computer engineering.

1.6 Project Benefits

Project Benefits for the Students :

- Enhance knowledge in image preprocessing, image processing and image processing techniques.
- Gain expertise in face detection and face detection algorithms.
- Experience in face recognition and feature extraction.
- Learn a new programming language (Python), which has been widely used in image processing.

Project Benefits for the University:

- Improved accuracy in attendance tracking.
- Automation of attendance processes to save time.
- Real-time monitoring to resolve issues quickly.
- Positive user experience with intuitive interfaces.
- Data-driven insights to make informed decisions.
- Scalability to accommodate organizational growth.
- Increased productivity and efficiency across the university.

1.7 Project Constraints

- Budget: Limited financial resources can prevent the purchase of high-quality hardware or software components.
- Time: Tight deadlines can restrict the time available for development, testing, and implementation phases.
- Technical Limitations: The availability of suitable facial and fingerprint recognition technologies may be limited, affecting system capabilities.
- Hardware compatibility: Compatibility issues can arise between hardware components, affecting system integrity.
- Scalability: Scalability limitations can limit a system's ability to accommodate an increasing number of users or locations.
- Environmental factors: Environmental factors such as light and noise can affect the performance of facial recognition systems.

1.8 Conclusion and Recommendations

Facial detection and recognition technology has emerged as an attractive solution to address many contemporary needs for identification and verification of identity claims. It combines the promise of other biometric systems, which attempt to link identity to individual body characteristics, with the more familiar functions of visual surveillance systems. This project detects and recognizes a face based on previous images in a database.

➢ Chapter Two – System Analysis

2.1 Introduction

This chapter describes the economic feasibility and the requirements of the system. The chapter also defines restrictions on the operation of the system and its implementation. Therefore, this chapter introduces the system requirements by describing the functional requirements and the non-functional requirements.

This chapter also shows the user's perspective in the system by making the Use-Case Diagram, as well as the system specifications and the risks that faces the system and the system developers and the possible solutions.

2.2 Project Implementation Option

The project can be implemented in more than one scope, but as an initial idea, we suggested implementing it at the university in a simple way so that it can be developed in the future.

2.2.1 Hardware Implementation:

Figure 2.1 depicts the hardware architecture of the attendance system, showcasing the integration of multiple components to achieve seamless functionality.

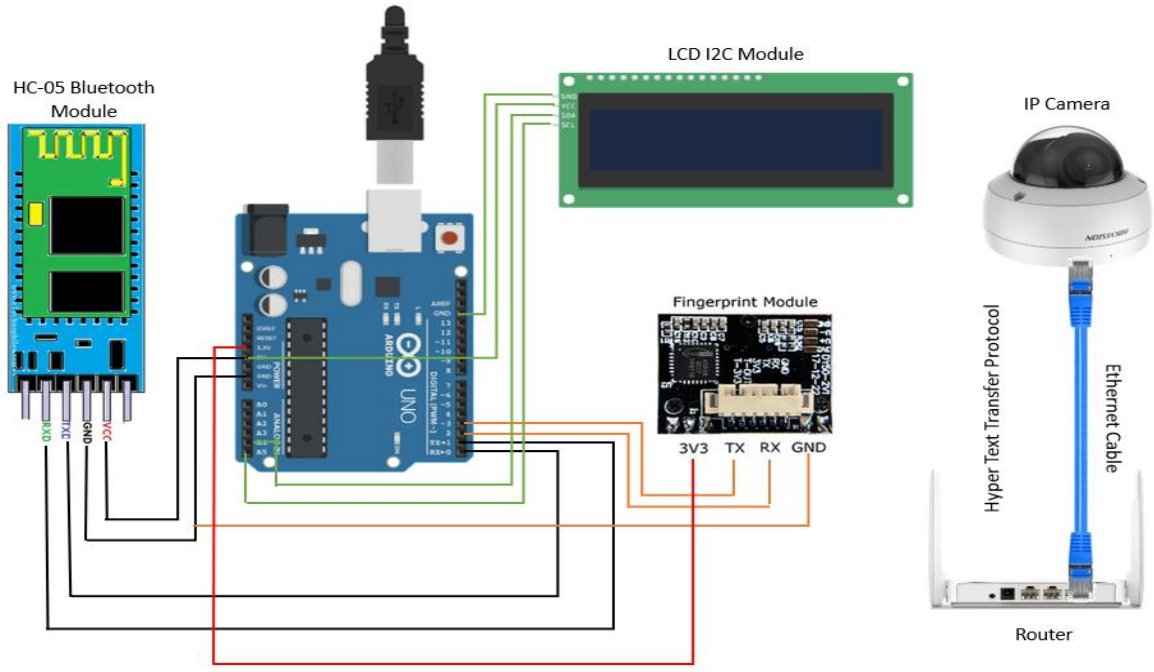

Figure2.1: Hardware Implementation

2.2.2 Software Implementation

Figure 2.2 illustrates the workflow of an attendance system that initially uses a camera to record attendance through facial recognition. If the camera fails to recognize the face for any reason (e.g., poor lighting or unmatched data), the system seamlessly switches to the alternative method, fingerprint verification, to ensure accurate attendance recording. The diagram outlines the steps of data collection, analysis, decision-making, and action execution. Additionally, the system is designed to repeat the process if needed, ensuring reliability and efficiency.

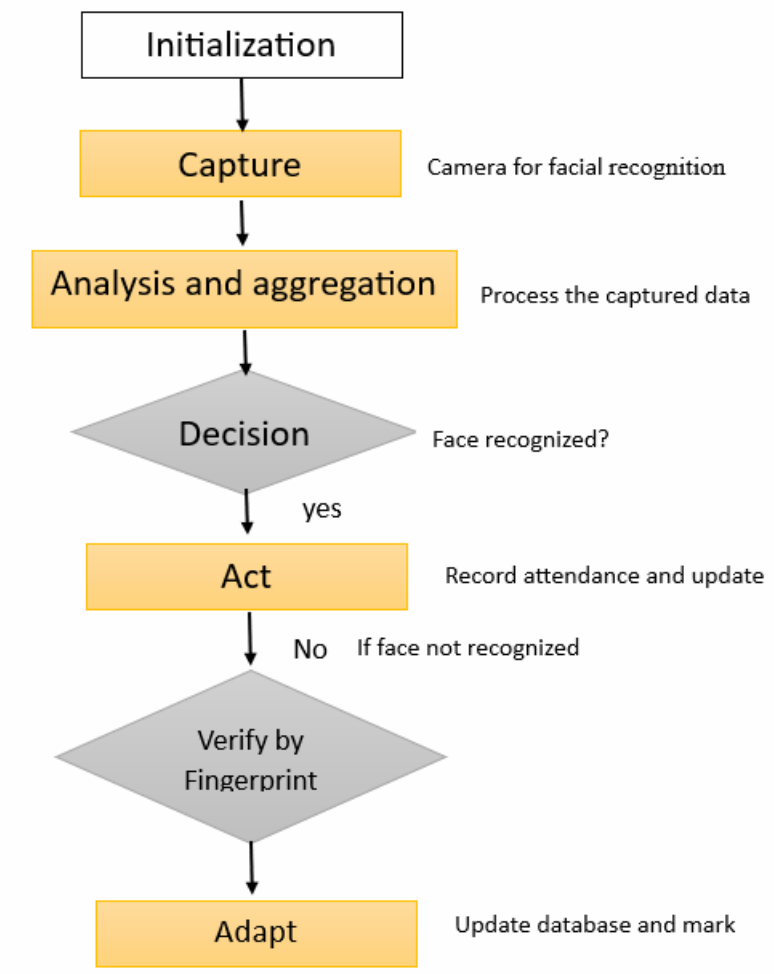

Figure2.2: Software Implementation.

2.3 System Requirements

It is well known that the main purpose of developing any system is to achieve several requirements and services known functions. These requirements can be classified into two types: functional requirements, and non-functional requirements which are explained below:

2.3.1 Functional requirements:

1. **User Management:**

- User Registration: Allow users to register using basic information and register face and/or fingerprint.
- User Management: Allow administrators to add, edit and delete user information.

2. **Registration:**

- Face Recognition: Capture a photo of the face for accurate recognition.
- Fingerprint Recognition: Capture a high-quality photo of the fingerprint more than once for verification.

3. **Attendance Marker**:

- Face Recognition: Capture the user's face through the camera and compare it to the registered database for identification.
- Fingerprint Recognition: Scan the user's fingerprint and compare it to the registered database for identification.
- Time Restriction: Limit the attendance registration window to a specific time frame (e.g. lecture time).

4. Data Management:

- Attendance Registration: Automatically record attendance data (date, time and user ID) upon successful recognition.
- Data Storage: Securely store facial images, fingerprint templates, and attendance records in an encrypted database.
- Data Export: Allow attendance data to be exported in different formats (Excel) for further analysis or reporting.

2.3.2 Non-Functional requirements:

1. **Performance:**
   - Ensure fast biometric matching and attendance recording to reduce user wait times.
   - Ensure the system can handle large numbers of users and attendance events without performance degradation, and is designed to scale effectively for future growth.

2. Credibility:

- Maintain high system availability to track attendance without interruption.
- Implement strong access controls to prevent unauthorized access to sensitive data and system components.

3. Ease of Use:

- Design intuitive user interfaces for registration, verification, and attendance management.
- Provide clear error messages and guidance to users in case of authentication failure or system errors.

4. Compatibility:

- Ensure compatibility with a variety of sensors and biometric devices to capture and verify data. Support common data exchange formats for compatibility with other systems.

2.4 Development Requirements

2.4.1 Hardware

The following **Table2.1** lists the costs of the hardware that needed to develop this project

Item	Number of Units	Unit Cost	Available	Total
Arduino Uno	2	15$	Yes	$30
Camera HIKVISION	1	$180	Yes	$180
Fingerprint sensor	1	$ 15	Yes	$ 15
HC – 05	1	$18	Yes	$ 18
Wires	1	$5	Yes	$5
LCD I2C	1	12$	YES	12$
Total		**$260**		

Table2.1: Development Hardware Cost

2.4.2 Software

The following **Table2.2** lists the costs of the software that needed to develop this project:

Item	Number of Units	Unit Cost	Available	Total
Visual Studio Code	1	Free	Yes	Free
Arduino IDE	1	Free	Yes	Free
Mongo DB	1	Free	Yes	Free
Total				**Free**

Table2.2: Development Software Cost

2.4.3 Humans

The following **Table 2.3** shows the labor hour costs to develop this project:

Member	Number/ Months	Cost $/Month	Available	Total/Months
Eng. Students	4	$1000	Yes	$4000
Supervisor	2	$500	Yes	$1000
Total				**$5000**

Table 2.3: Development Human Costs

2.5 Cost Benefit

This system achieves benefits for the university, as the developer achieved many benefits from building this system, including: saving time and effort for doctors as they are provided with important information through this site, which is a full lecture attendance report.

2.6 Project Management

- Perform the development, testing, and deployment steps.
- Provide training and support during program implementation.
- Monitor system performance and gather feedback.
- Use default metrics to monitor for improvement.
- Document project progress and results.

2.7 Conclusion and Recommendations

In this chapter, we have mentioned a number of options that can be used to develop this system. After identifying the advantages and disadvantages of each option, I chose to implement our system as a standalone system. Then I described our proposed system in detail, and how it can be implemented from an economic, technical and scientific perspective. Next, we talked about the added value of the project. The next chapter will explain the functional and non-functional requirements of the system in detail.

➢ Chapter Three – System Functional Requirements

3.1 Introduction

Chapter has introduced the main functional and non-functional requirements for developing the proposed system. In this chapter, the functional requirements will be explained in more technical terms, so the data will be collected and analyzed to fill the specification of the software requirements. More precisely, this chapter clarifies the context diagram for identifying the system's boundaries by showing other systems that are used or interacted with the system being developed. After that, a Use-Case diagram is used to support requirements elicitation and to show the system functions.

Then, the chapter covers the detailed functional description for the proposed system

3.2 Context Diagram

The system context diagram shows how the project works in general and how it handles images and records them in the database. **[2]**

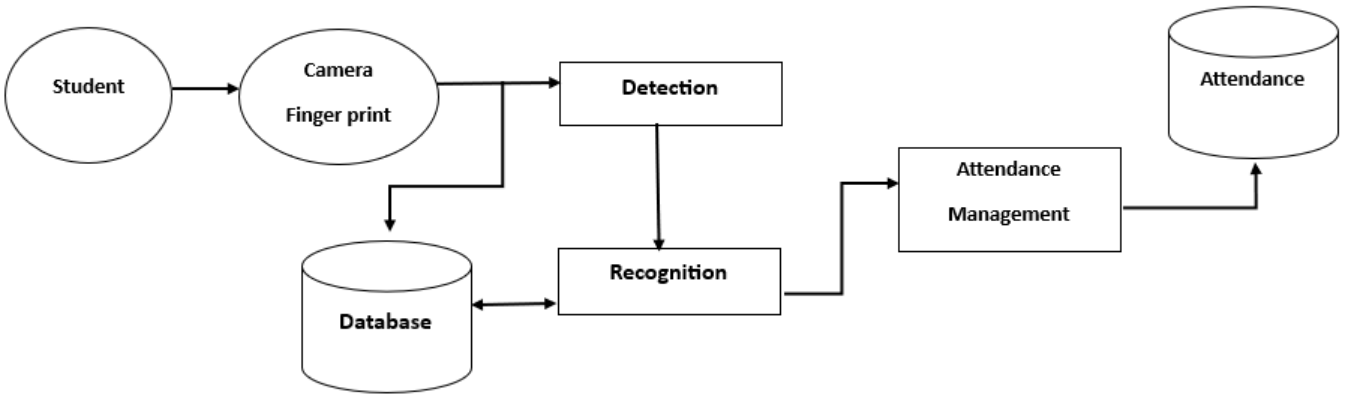

Figure 3.1: System Context Diagram

3.3 Use-Case Diagram

The resource map shows **Figure3.2** the different people who interact with SAS and the different types of information that represent their interaction with the system.

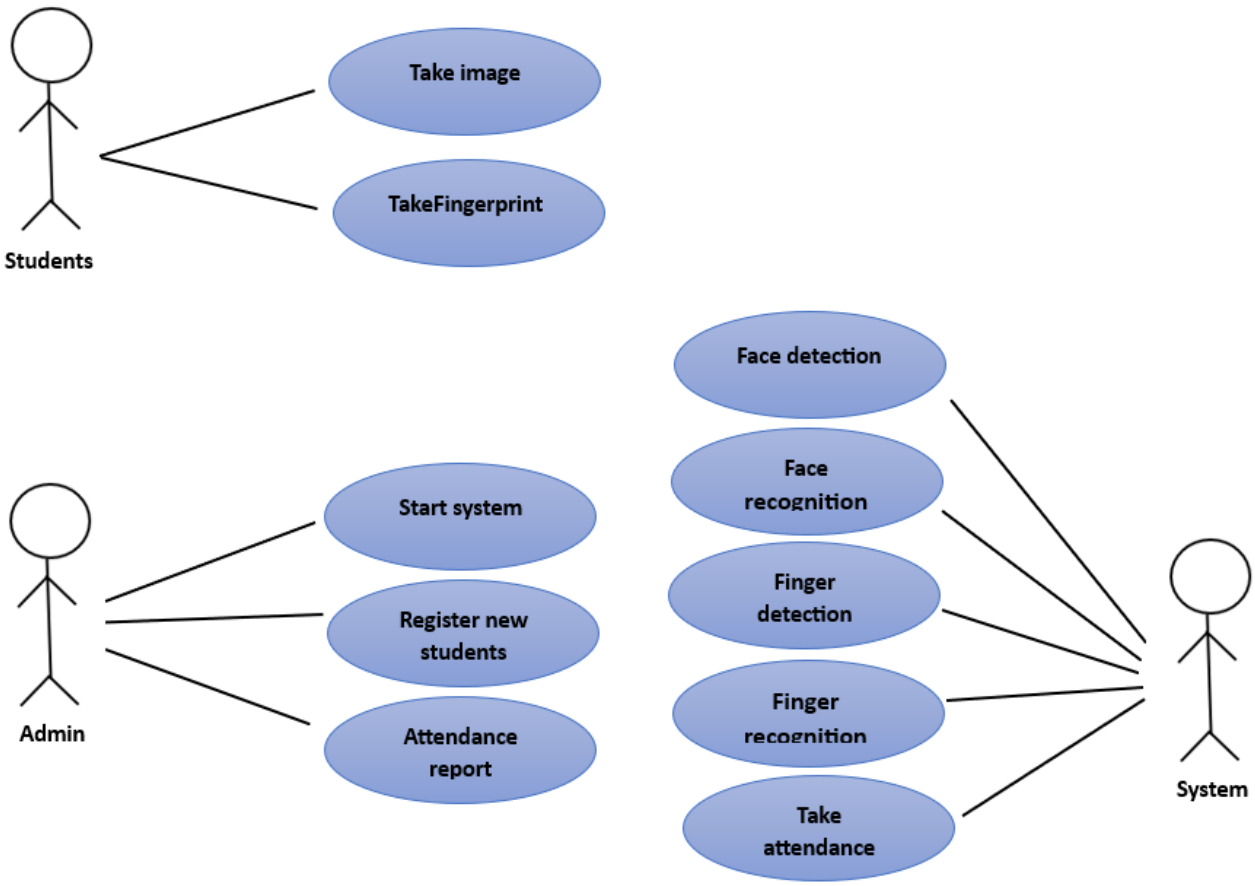

Figure3.2: Use Case Diagram

3.4 Functional Requirement Description

Detailed descriptions of the functional requirements specify the specific features and functionalities of the SAS **Table3.1**, including enrollment, identification, attendance recording, real-time monitoring, reporting, and security measures.

Requirement Name	Information about user
Enrollment	Administrators should be able to register individuals into the system by capturing their facial photos and fingerprint data.
Identification	System users (employees, students, etc.) should verify their identity using facial recognition or fingerprint authentication.
Attendance Recording	System users should record their attendance by verifying their identity through facial or fingerprint recognition.
Real-time Monitoring	Administrators should monitor attendance data in real-time through a dashboard or interface.
Reporting	Administrators should generate comprehensive reports on attendance patterns and trends.

Table3.1: Information About User

3.5 Non-Functional Requirements Description

Description of Non-Functional Requirements

- **Scalability**: The system can handle increasing records over time, using MongoDB which efficiently manages data and allows adding new components like cameras or fingerprint sensors.

- **Availability**: The system is always operational, with the Hikvision camera being reliable and a backup fingerprint sensor. The web interface is hosted on cloud servers for continuous availability.

- **Ease of Use**: The web interface is simple for administrators, and clear instructions are provided for users when using the camera or fingerprint sensor.

- **Efficiency**: Fast processing of images through the Hikvision camera, quick fingerprint verification via Arduino, and rapid data transfer via Bluetooth.

- **Reliability**: The system ensures accurate attendance records with dual verification (face and fingerprint), even in challenging conditions.

3.6 Conclusion and Recommendations

This chapter has explained the functions of the proposed system using context diagram and use case diagram to more precisely define the interaction between the system and its actors. After presenting this, we have explained the functional and non-functional requirements of the system in a more technical way. These functions have been defined according to the system problem explained earlier. We have seen that the main functions are those related to how to detect face and fingerprint. Therefore, the system will be more accurate.

➢ Chapter Four – System Design & Development

4.1 Introduction

In chapter three, we have clarified two system modeling diagrams; the system context diagram, and the system use case diagram. Both diagrams have been used for representing the system from, different perspectives and describe the functions of the system. In this chapter, the next step is, to develop the system. This will be clarified later on. In particular, this chapter will illustrate the system functions design in input/output screens and other system modeling diagrams. More precisely, this chapter will identify the design of input/output screens based on the system functions which have been analyzed in the previous chapter. Then it shows the system class diagram for representing the object classes in the system and the associations between these classes. After that, the system sequence diagrams will show the interactions between the actors of the system and between system objects. Then, the entity relationship diagram (ER-D) which describes the system database. Finally, the functions of the system will be clarified using the activity diagrams which show the activities that are involved in a particular process or in dot processing.

4.2 Class Diagram

This section will show a System Class Diagram (SCD). A System Class Diagram is a type of diagram in the Unified Modeling Language (UML) and is used to show the organization of a system in terms of the classes of objects that make up the system, and the associations that indicate the relationships between these classes. As shown, this **Figure4.1** shows the associations between these classes, and the attributes and methods of each class. The next section shows another diagram, also used to model the system, but from a different perspective. **[3]**

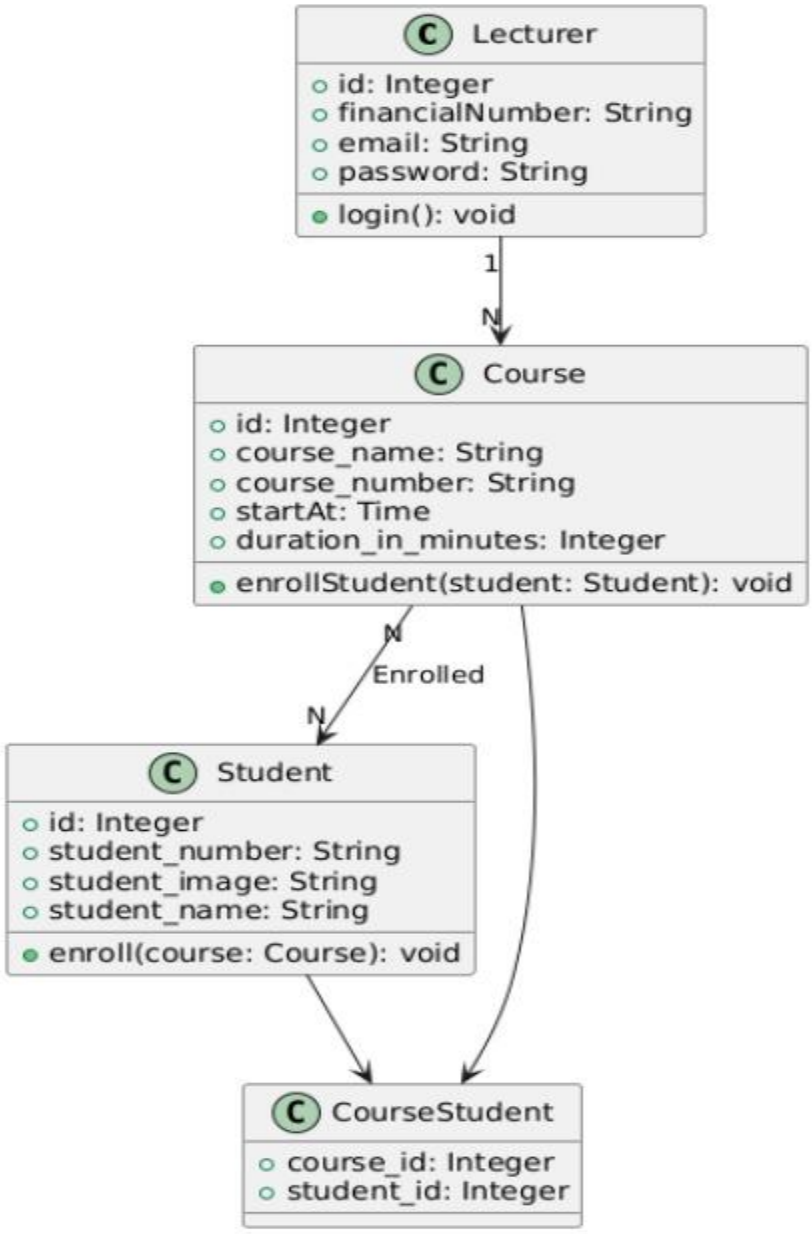

Figure4.1: Class Diagram

4.3 Sequence Diagram

This section will illustrate another type of UML diagram that can be used to show the interactions between system actors and system components, and the interactions.

between these components themselves. This will help us understand whether the proposed system or architecture is capable of delivering the required performance and reliability of the system. The following **Figure4.2** illustrate the sequence diagram for each function of the proposed system which will describe the interactions during a given state. **[3]**

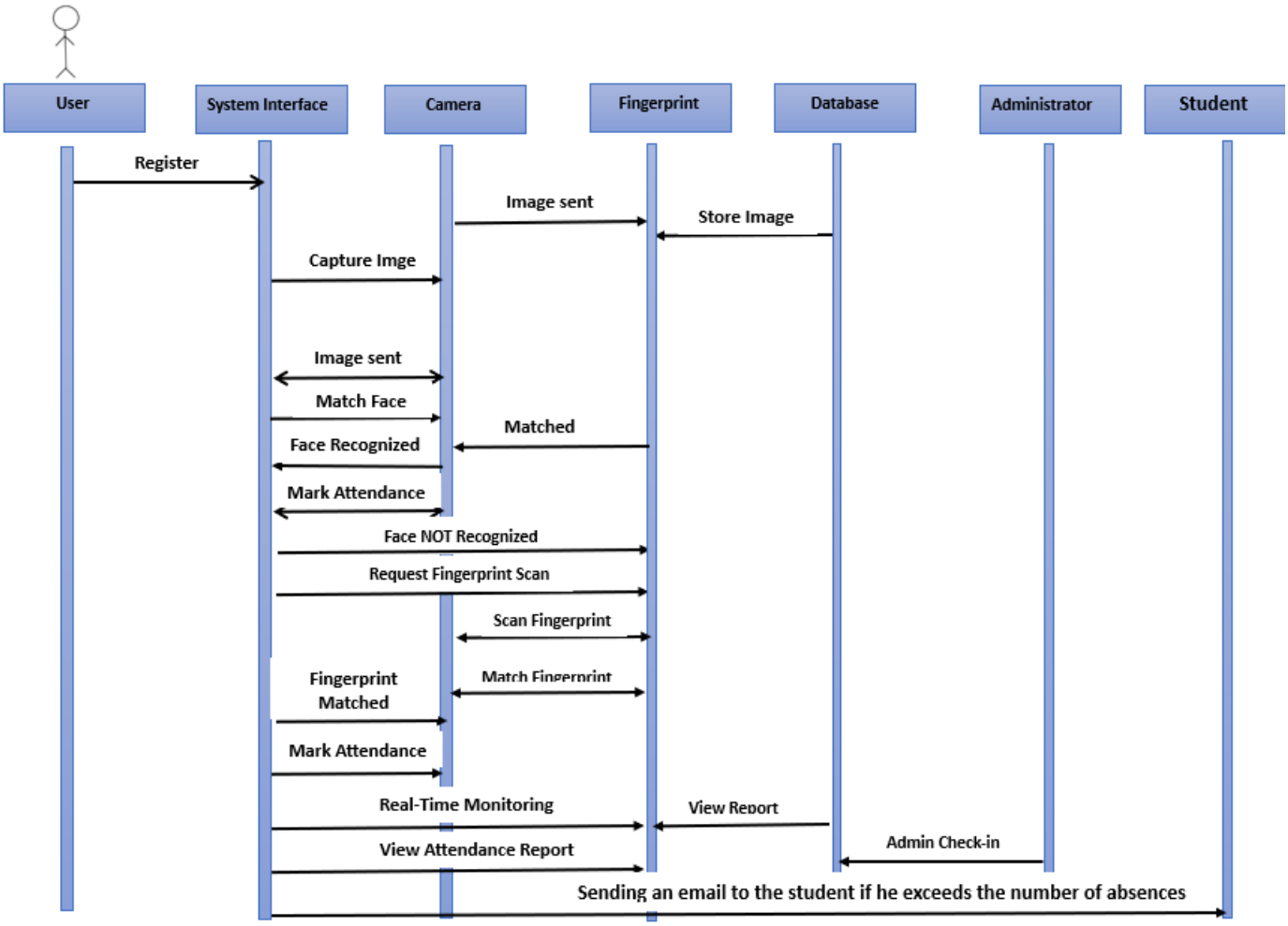

Figure4.2: Sequence Diagram

4.4 Entity Relationship Diagram

Entity relationship diagram (ER-D) **Figure 4.3** is a data model for describing a database in an abstract way. **[4]**

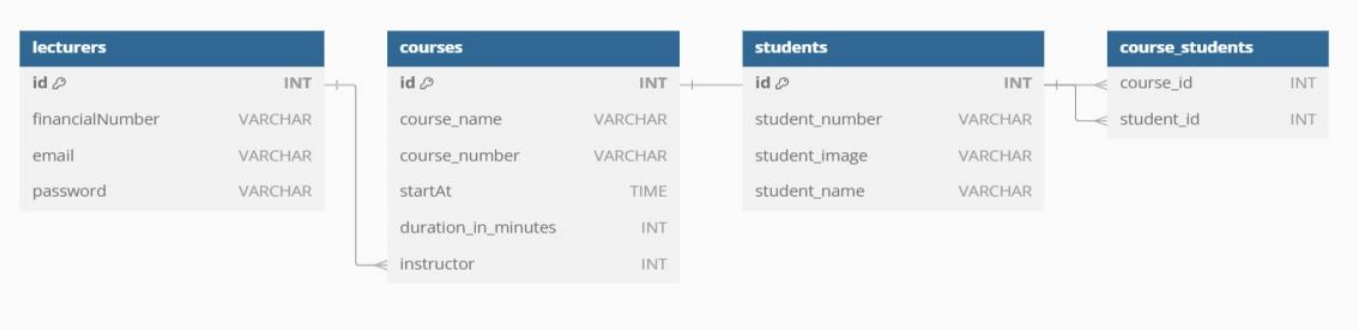

Figure 4.3: Entity Relationship Diagram

4.5 Activity Diagram [2]

The activity diagram **Figure 4.4** illustrates the workflow of the Attendance System (SAS) between the user and the admin. The process starts with the system attempting to recognize the student using the camera. If facial recognition fails, the fingerprint sensor is used as an alternative. The collected data is sent to the database, where an attendance report is generated for the admin.

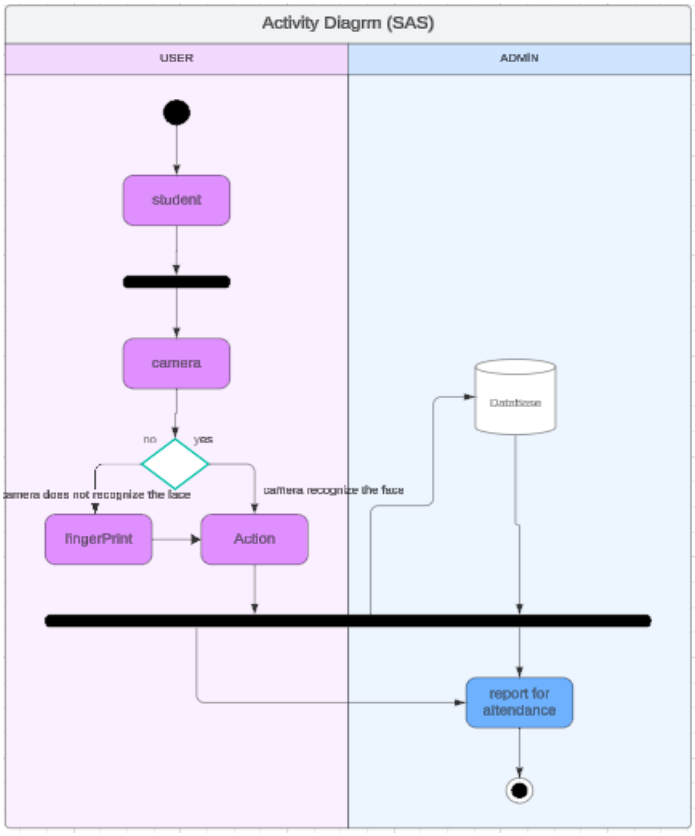

Figure 4.4: Activity Diagram

4.6 System interface

In **Figure 4.5** Create a New Account - The user interface of the new account creation page, featuring fields to enter email, financial number, and password, along with a registration button.

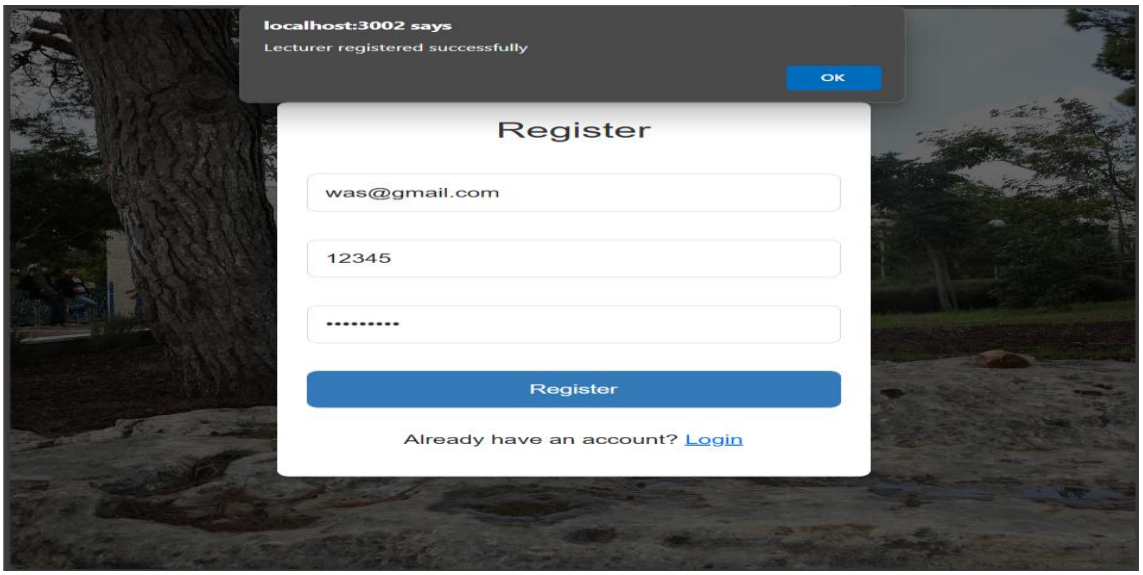

Figure 4.5: Create a New Account

In **Figure 4.6** The image shows a login interface with fields for user ID and password, along with an option to register for a new account.

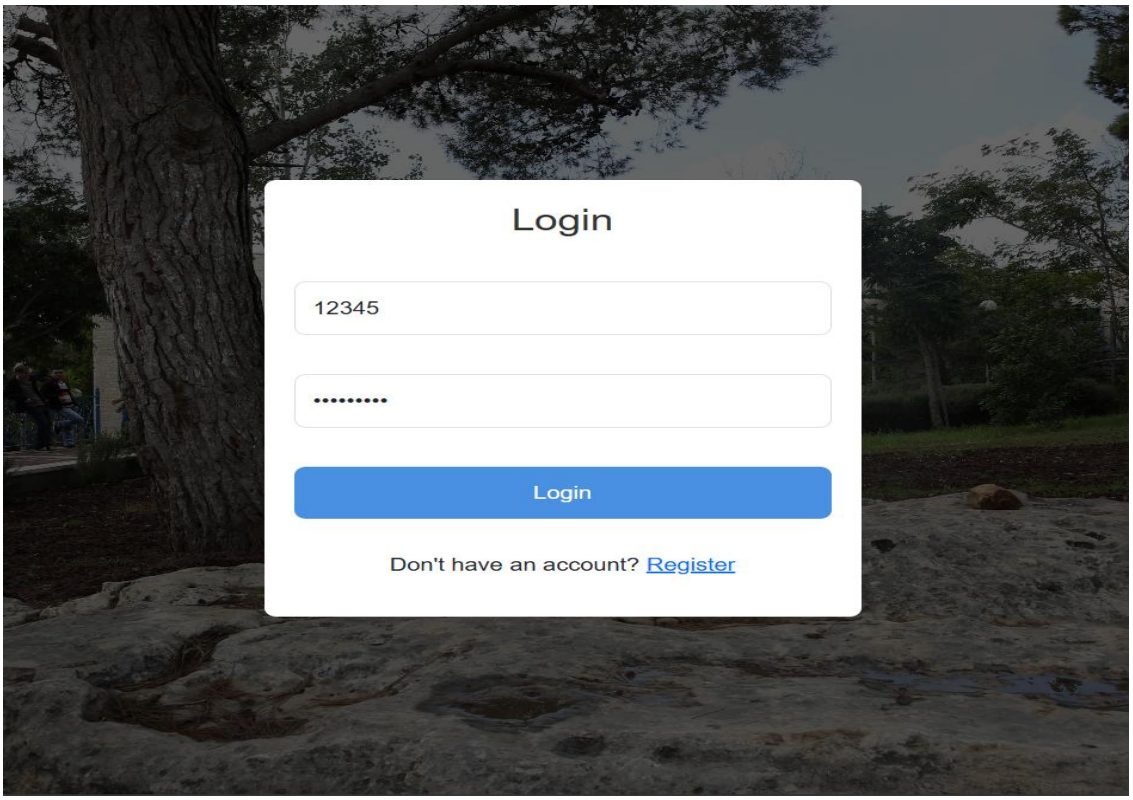

Figure 4.6: Account Login

In **Figure 4.7** When You log in For The First Time" likely illustrates the steps or interface for logging in for the first time to a system.

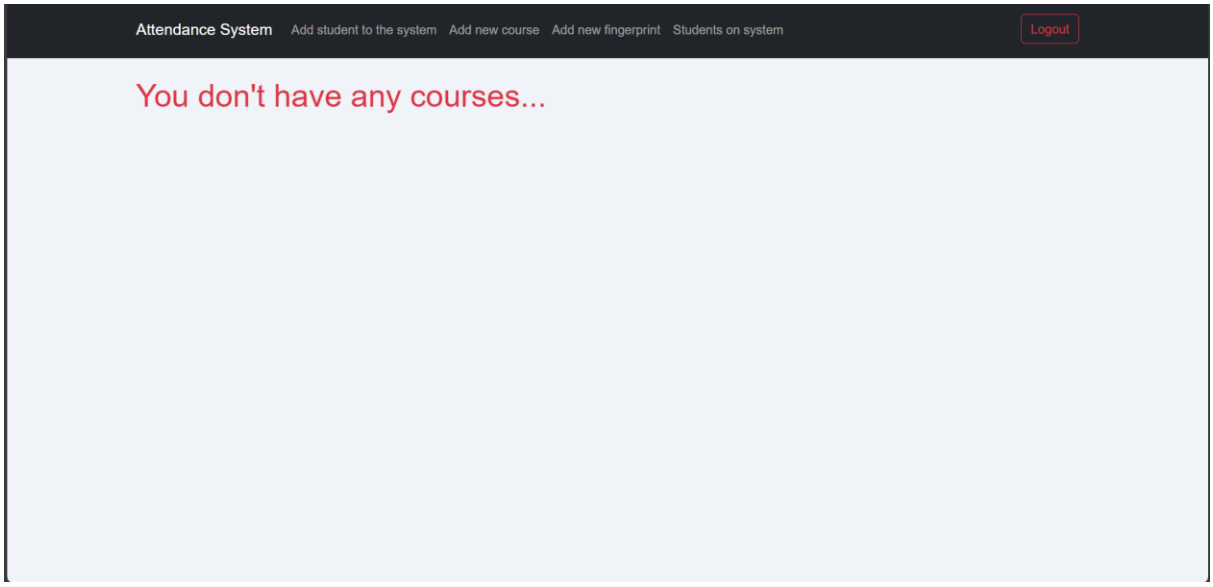

Figure 4.7: When You log in For The First Time

In **Figure 4.8**Add Students interface, featuring fields to enter full name, university ID number, and upload the student's photo, along with an 'Add' button to register the student in the system.

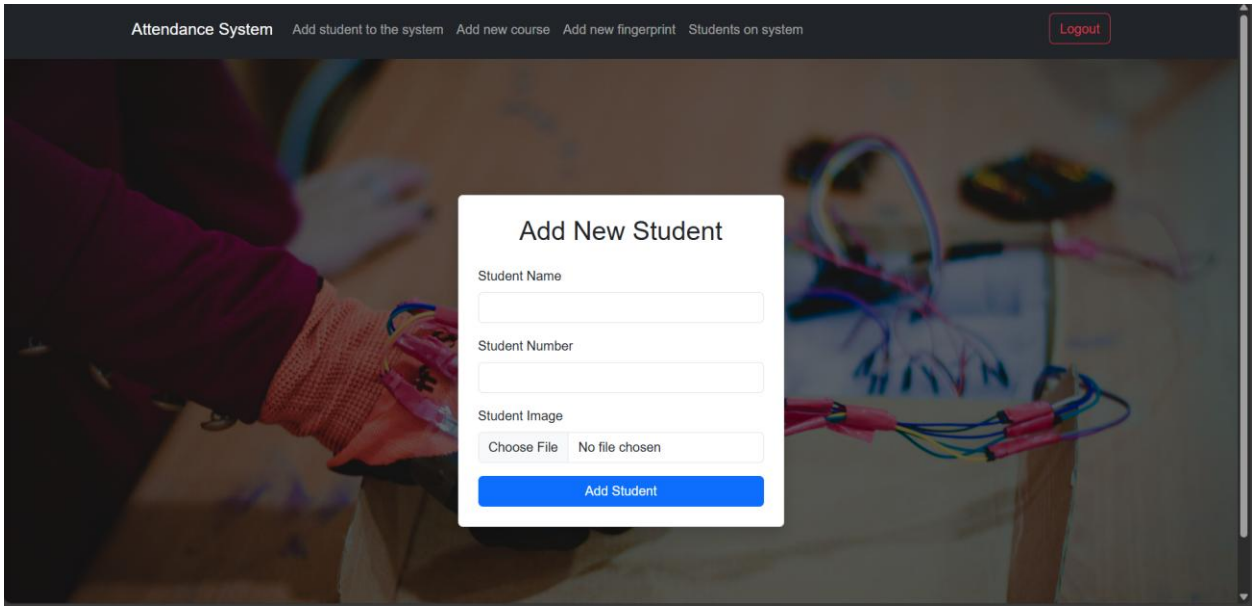

Figure 4.8: Add Students

In **Figure 4.9** The Student Has Been Added Successfully shows a confirmation message or screen indicating that a student has been successfully added to a system or database.

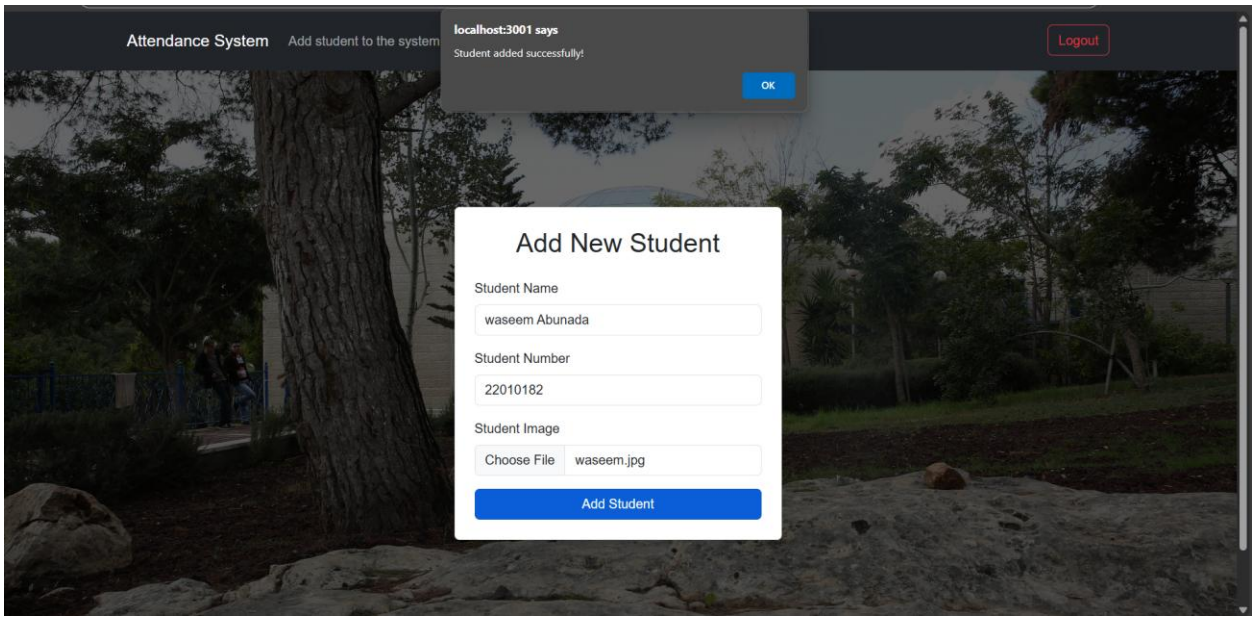

Figure 4.9: Add New Student

In **Figure 4.10** If The Student Has Been Added To The System Before, It Will Not Be Repeated Based On The University Number.

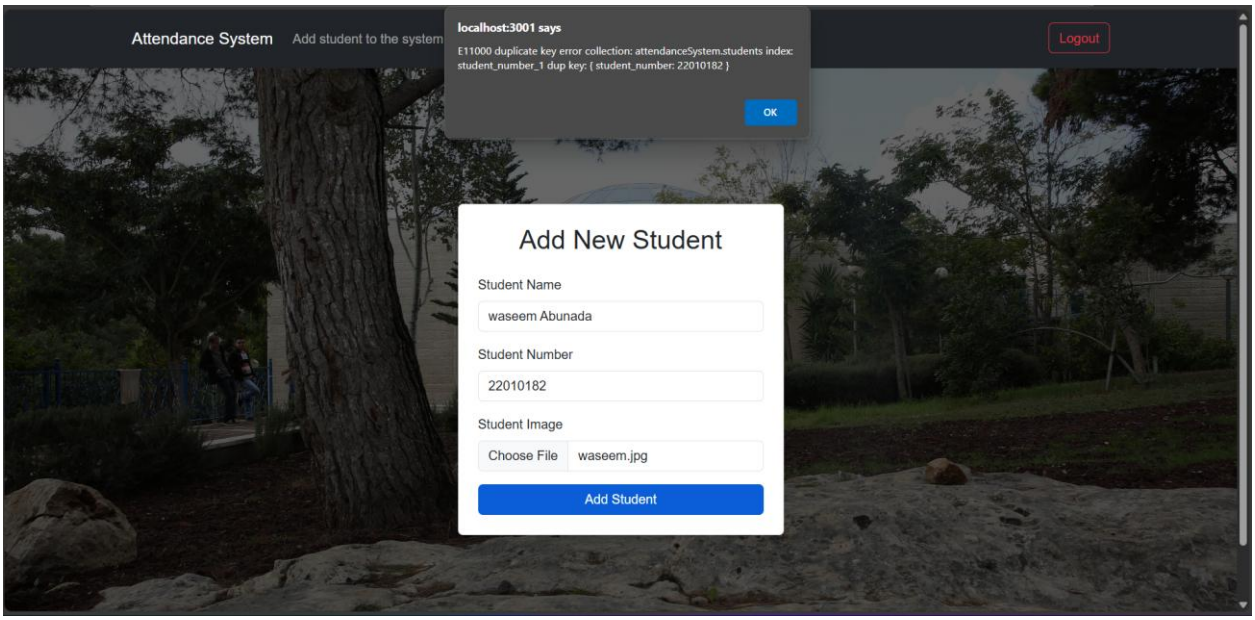

Figure 4.10: Add New Student

In **Figure 4.11** Add a Fingerprint for The Student shows a screen where a fingerprint is being enrolled or added for student identification in a system.

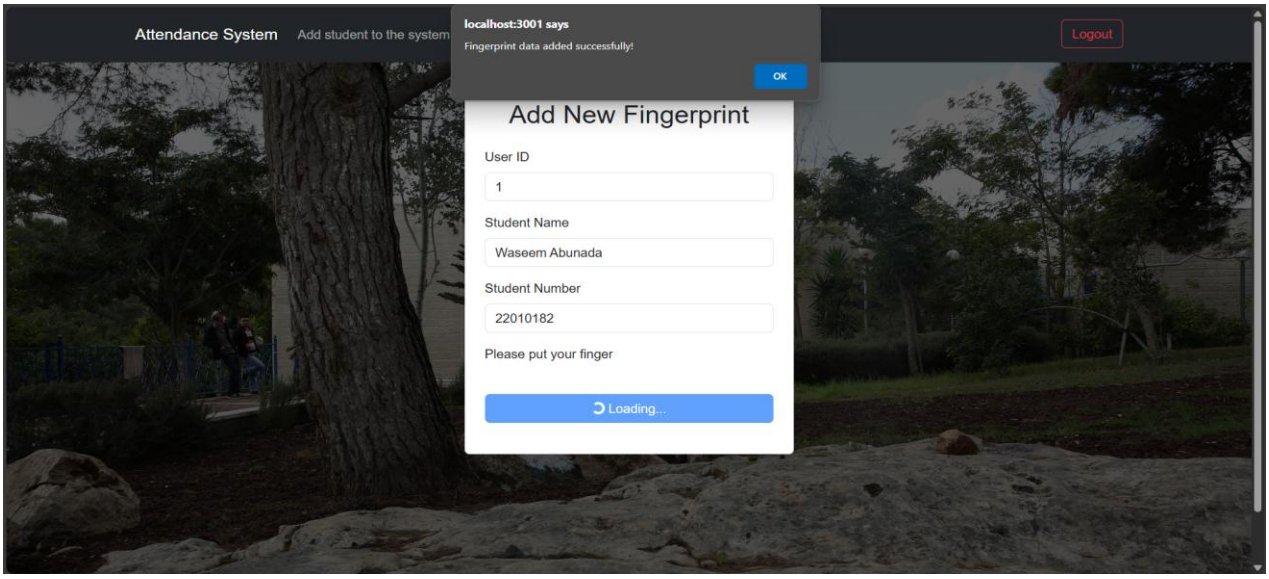

Figure 4.11: Add a Fingerprint

In **Figure 4.12** Add a New Course - Interface for entering course details, including course name, code, and description, with an 'Add Course' button.

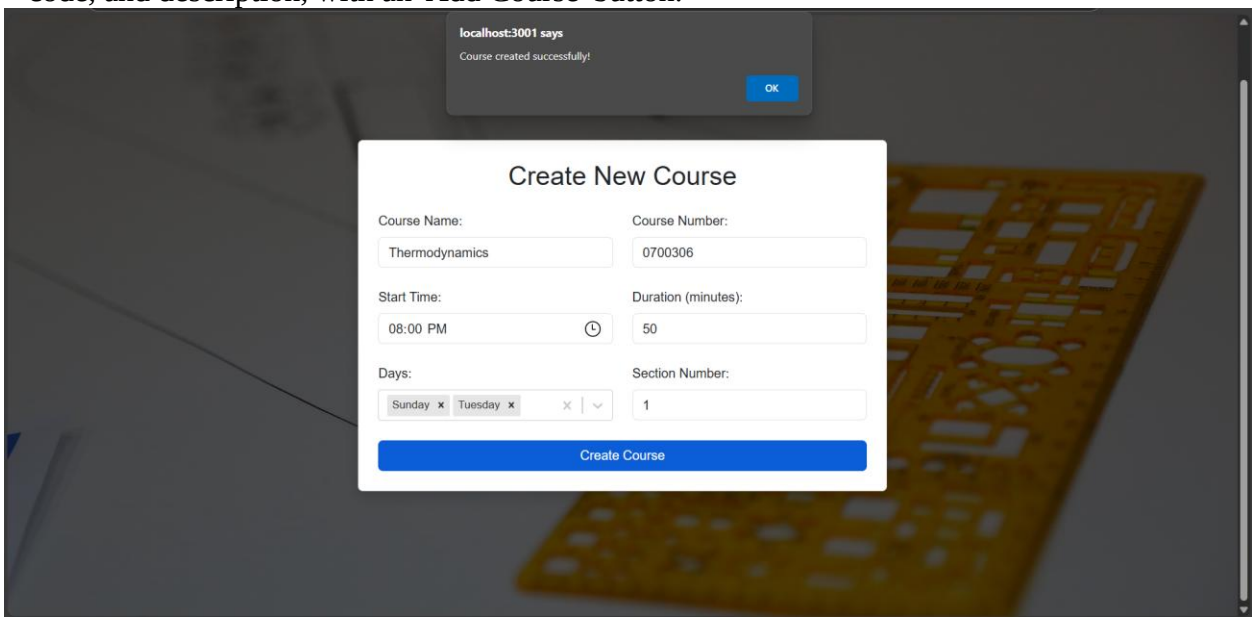

Figure 4.12: Add a New Course

In **Figure 4.1** When The Course Is Added Successfully shows a confirmation message or screen indicating that a course has been successfully added to the system.

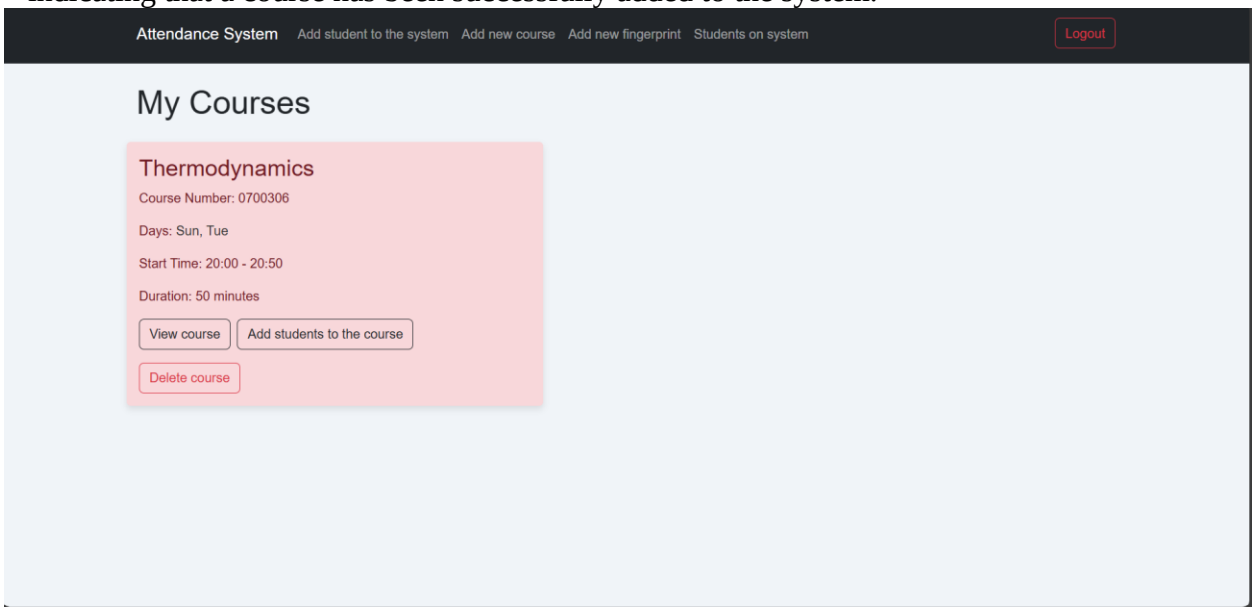

Figure 4.13: When The Course Is Added Successfully

In **Figure 4.14** When Entering The course, The Students Who Will Take This Course Are Added from The Last In Which The Students Are Registered In The System.

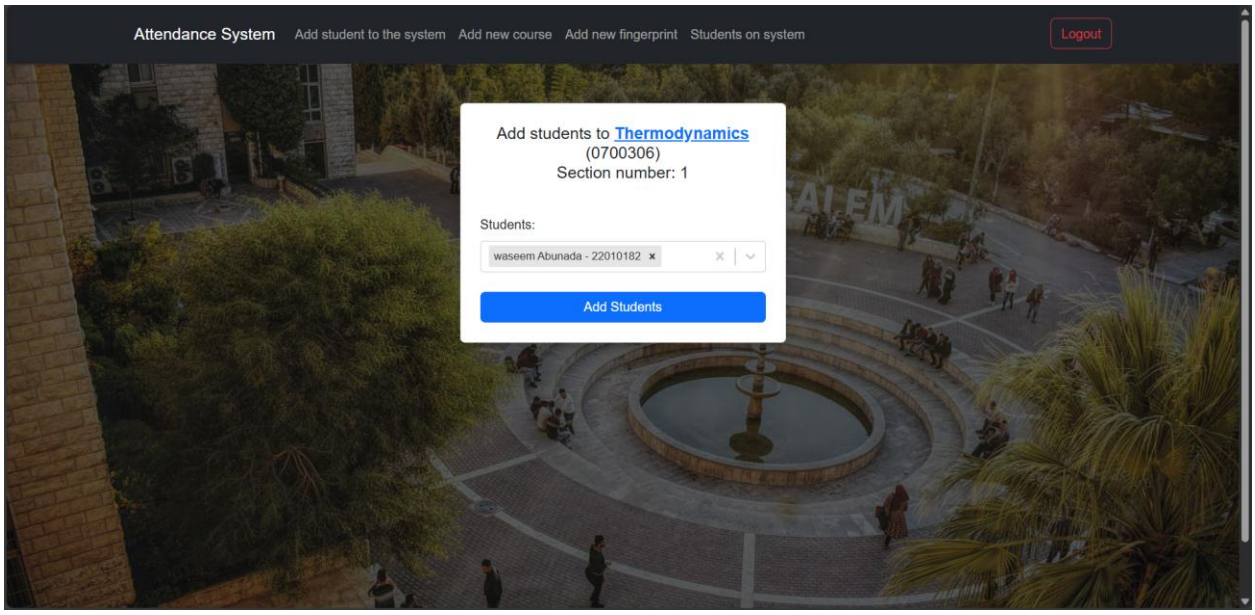

Figure 4.14: Add Registered Students

In **Figure 4.15** Students Who Are Registered Will Appear Inside the Course.

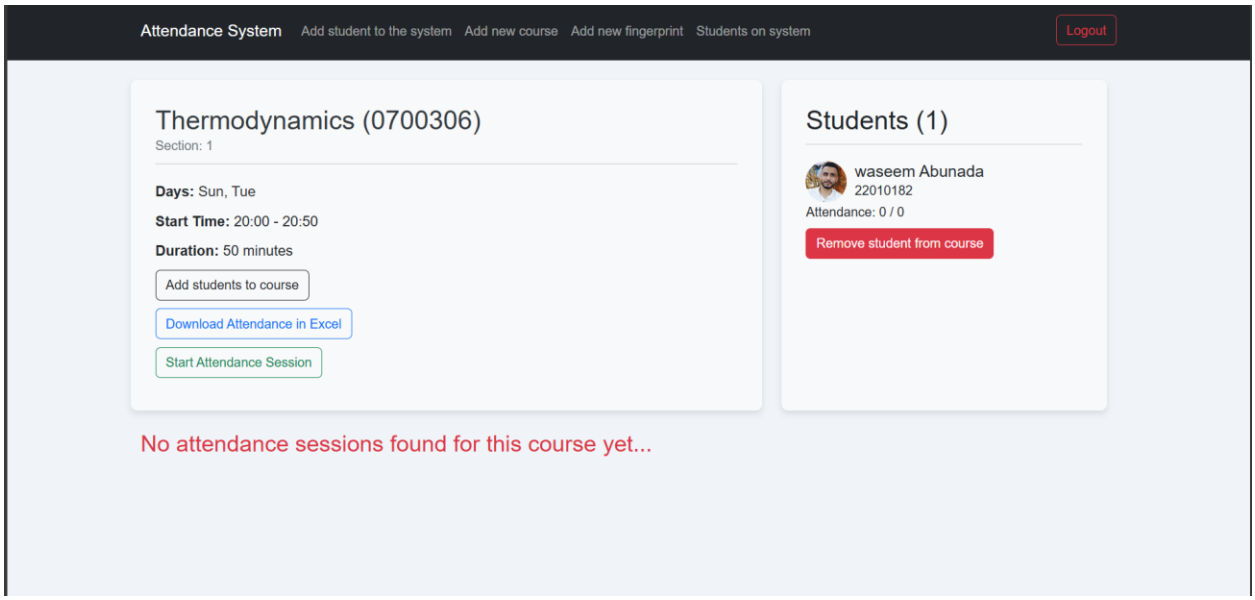

Figure 4.15: Student Interface

In **Figure 4.16** In Case Attendance Is Taken and The Student Is Not Present" can be shortened to Absent Student on Attendance.

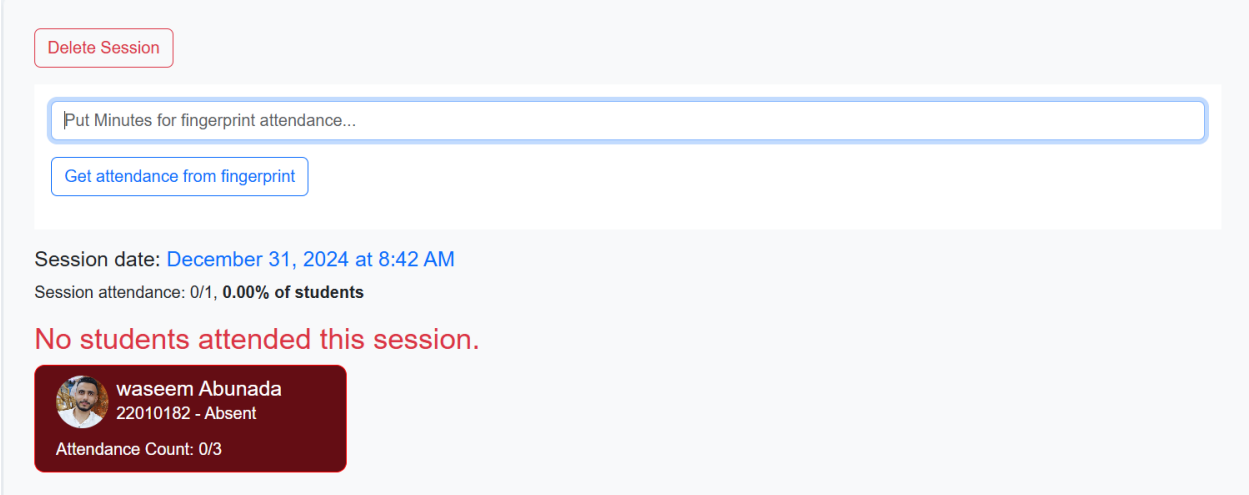

Figure 4.16: In Case Attendance Is Taken and The Student Is Not Present

In **Figure 4.17** The Event That Attendance Is Taken and The Student Is Not Identified, The Fingerprint Will Be Resorted To.

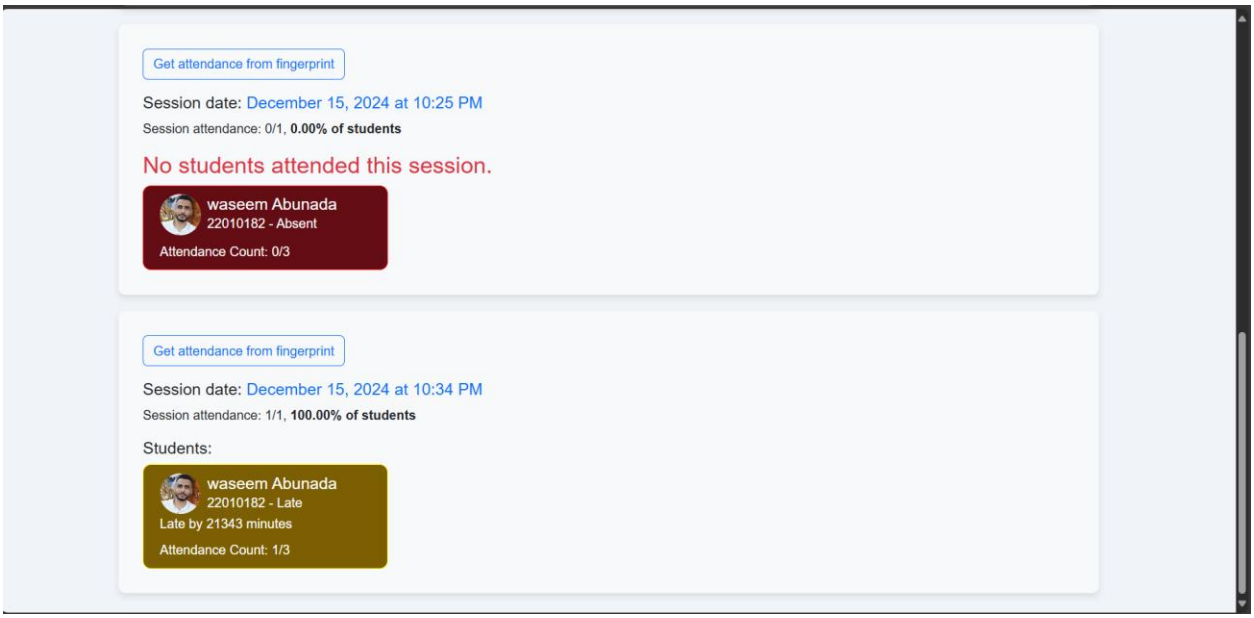

Figure 4.17: Fingerprint Verification

In **Figure 4.18** If The Camera Took Attendance for The Same Student Three Times During The Specified Period.

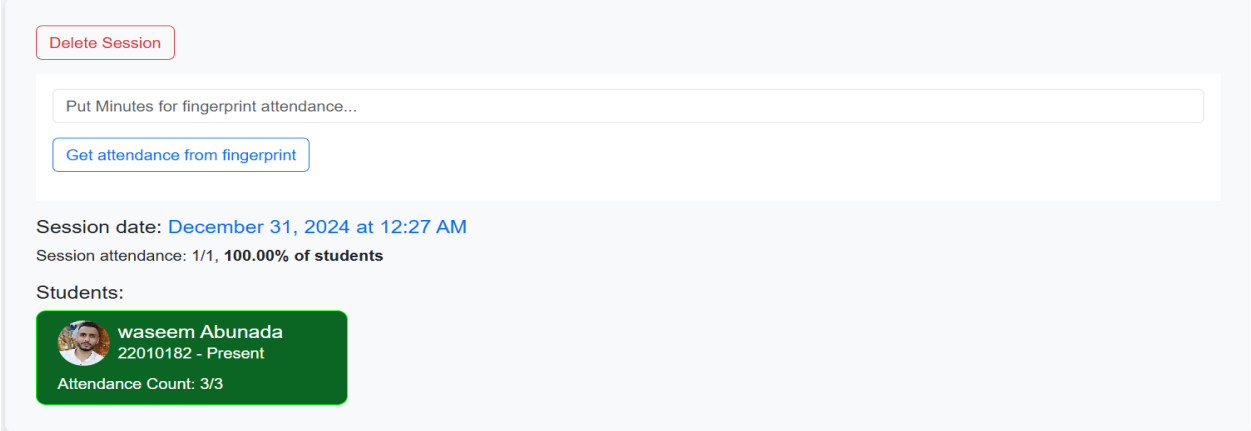

Figure 4.18: Repeated Camera Attendance

In **Figure 4.19** The Event That the Number Is Incomplete, It Means That The Camera Did Not Capture The Student's Face At One Time Or That The Student Left The lecture.

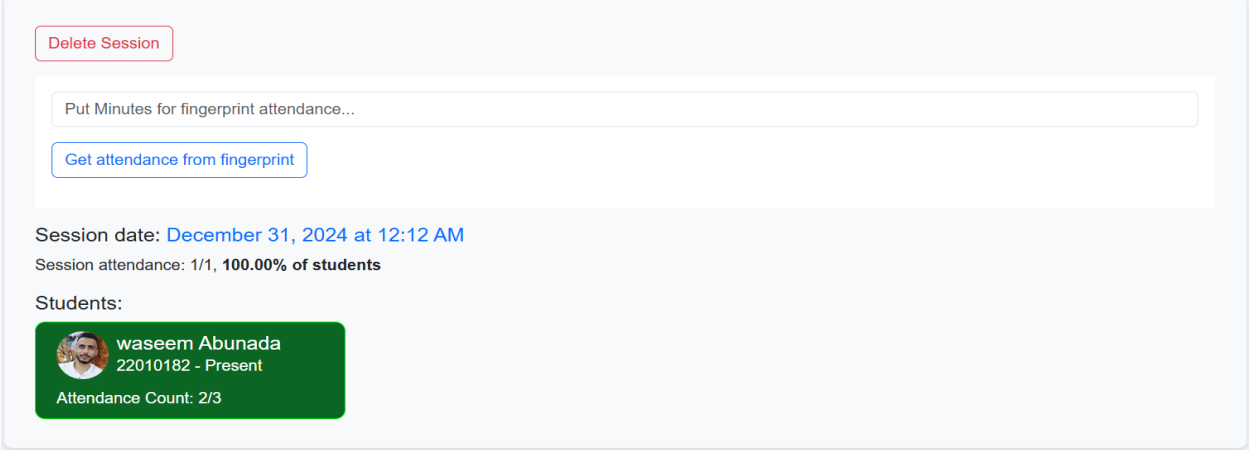

Figure 4.19: Incomplete Attendance Data

In **Figure 4.20** The List of Names, What Was Attended Out of All Lectures Is Counted.

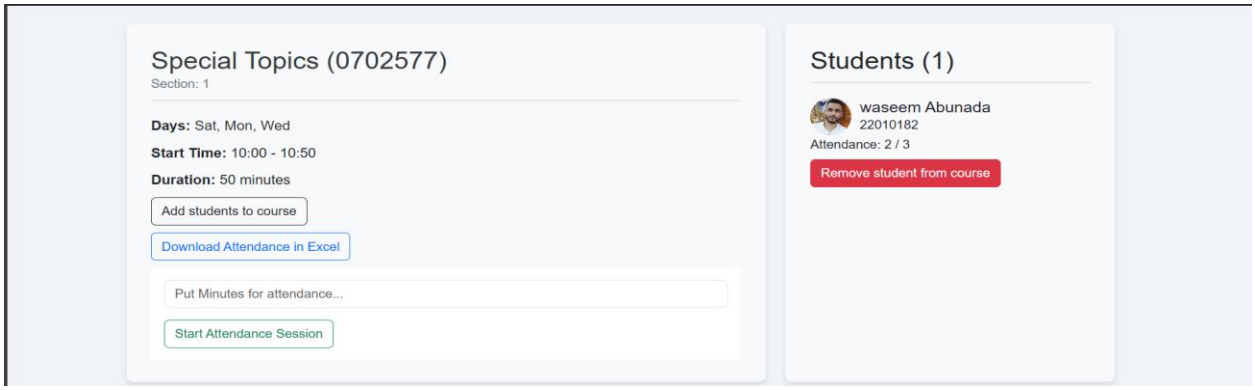

Figure 4.20: Lecture Attendance Count

In **Figure 4.21** The Event That the Student Is Absent Three Lectures According to The Law, A Red Background Will Come Above His Name, Which Will Be a Warning to The Lecturer

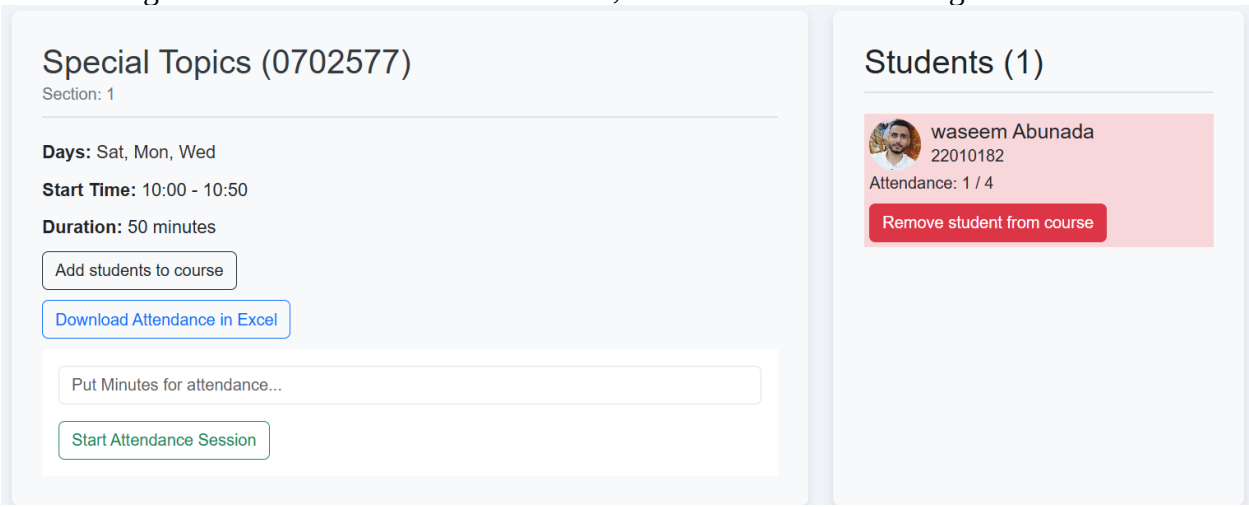

Figure 4.21: Absence Warning Indicator

In **Figure 4.22** Attendance For All Class Is Saved Inside an Excel File

Session Date	Student Name	Student Number	Attendance Stat	Time of Attend	Attendance Count
Monday, 12/09/2024, 11:25 PM					
	waseem Abunada	22010182	Absent	N/A	0
Sunday, 12/15/2024, 10:34 PM					
	waseem Abunada	22010182	Present	08:17 PM	3

Figure 4.22: Attendance Data Export

4.7 Conclusions and recommendations

This chapter has introduced the system design in terms of modeling diagrams that represent the system functions and objects in an abstract way, and in different perspectives. These diagrams include the system class diagram which represents the object that compose the system, the sequence diagrams which show the interactions between the system objects, the entity relationship diagram which describes the system database and the system activity diagrams. These diagrams show the activities involved in a particular process or function. The next chapter will illustrate how we could implement the proposed system using programming languages.

➢ Chapter Five – Troubleshooting & Future Works

5.1 Introduction

In chapter four, we have described several system modeling diagrams for representing the system functions from different perspectives. These diagrams include the system class diagram, activity diagrams and sequence diagrams.

This chapter will identify the actual coding and implementation for the system's functions according to the steps that have been described in the previous chapter using design modeling diagrams. The programming and coding environments that have been used for developing the system functions will also be described with their establishment procedures in this chapter.

5.2 Coding programming language

to develop this system, we have use python programming language and java script and in Arduino we use c language to program an Arduino and fingerprint sensor. **[5]**

Python Programming Language

Python is used for the backend of the system due to its simplicity and extensive library ecosystem. It handles core functionalities like user authentication, database interactions, and logic processing. Its readability and quick development cycle make it ideal for this project. **[6]**

JavaScript Programming Language

JavaScript is employed for the frontend development, enabling dynamic and interactive user interfaces. It facilitates smooth communication between the user and the backend server, ensuring a responsive and engaging user experience. **[7]**

C Programming Language

C is used to program the Arduino and fingerprint sensor, providing low-level control over the hardware. Its efficiency and speed make it perfect for tasks that require direct interaction with the microcontroller and sensor for reliable operation. **[7]**

5.3 Data base system

In this system we used mongo DB

MongoDB is a NoSQL database that offers flexibility and scalability, making it ideal for this project. It stores data in a JSON-like format, which aligns well with modern application architectures. In this system, MongoDB is used to manage user data, including credentials, access logs, and fingerprint templates, ensuring efficient retrieval and storage. Its ability to handle both unstructured and semi-structured data supports the dynamic nature of the application. Additionally, MongoDB's scalability ensures that the system can handle growing data as more users are added over time. **[8]**

5.4 Establishment of Development Environment

Hardware Components:

5.4.1 Arduino:

Integrating Arduino technology in **Figure5.1** with facial and fingerprint recognition technologies provides an innovative solution to create a Smart Attendance System (SAS). By leveraging the flexibility and scalability of the Arduino, along with the capabilities of the facial and fingerprint recognition modules, this system provides an efficient and cost-effective way to automate attendance tracking processes. This section explores the hardware and software components required to build a SAS using an Arduino, including selecting appropriate sensors, data processing techniques, and system integration methods. Additionally, it discusses the advantages of using an Arduino to develop such systems and provides insight into the potential challenges and considerations involved in implementation. **[5]**

Figure5.1: Arduino Uno

5.4.2 HIKVISION DS-2CD2163G2-1

The HIKVISION DS-2CD213G2-l **Figure5.2** is an advanced surveillance camera offering high-resolution imaging and integrated AI features, such as face detection, motion tracking, and object recognition. Its robust, weatherproof, and vandal-resistant design (IP67 and IK10 ratings) ensures durability in both indoor and outdoor environments. The camera supports Power over Ethernet (PoE) for easy installation and seamless integration with existing networks. Ideal for applications in transportation, retail, and public safety, it delivers reliable performance and effective monitoring around the clock.

Figure5.2: HIKVISION DS-2CD213G2-l

5.4.3 Fingerprint sensor

The integration of the fingerprint sensor **Figure5.3** into the Smart Attendance System (SAS) along with facial recognition technology enhances accuracy and security. This section explores the role of fingerprint sensors in the development of SAS, focusing on their integration with facial recognition systems. By leveraging fingerprint sensors, SAS can provide multi-factor authentication, ensuring reliable attendance tracking. This section discusses technical aspects of fingerprint sensors, such as image capture resolution and sensing methods. It also examines how fingerprint data is processed and matched to stored templates within the SAS framework. Furthermore, it addresses considerations for deploying fingerprint sensors in different environments, including durability, compatibility, and user experience. Overall, the integration of fingerprint sensors enriches SAS capabilities, providing a comprehensive attendance management solution. **[9]**

Figure5.3: Fingerprint Sensor

5.4.4 LCD I2C

The LCD I2C module **Figure 5.4** is a convenient interface for connecting Liquid Crystal Displays (LCDs), such as 16x2 or 20x4 screens, to microcontrollers or digital systems using the I2C communication protocol. It reduces the number of pins required for connection to just two lines: SDA (data) and SCL (clock), which makes it ideal for compact and efficient designs. The module attaches directly to the back of the LCD, simplifying wiring and providing a built-in potentiometer for easy adjustment of display contrast. Operating on 5V or 3.3V, it is compatible with low-power applications. With readily available libraries for platforms like Arduino, it is widely used in microcontroller projects, IoT devices, and embedded systems for displaying sensor data, system status, and other outputs efficiently. **[10]**

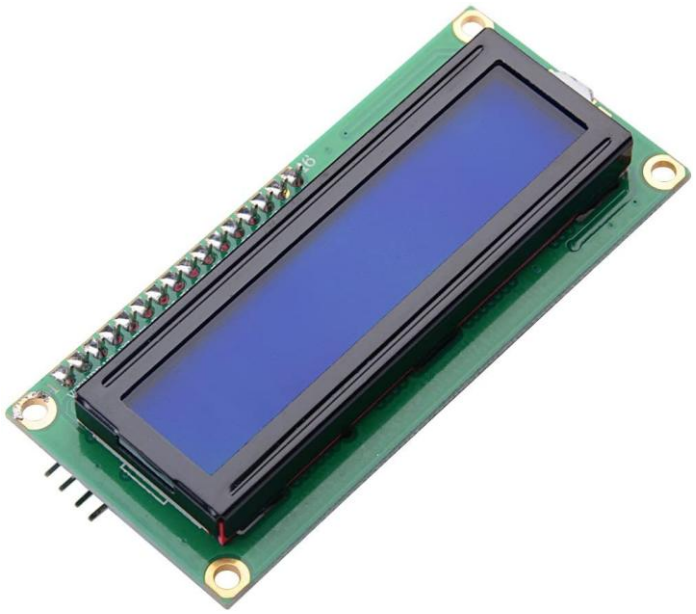

Figure 5.4: LCD I2C

5.4.5 Wires

The "wires" **Figure5.5** smart presence system combines facial fingerprint recognition technology with wired systems to better track location Cameras capture facial images, while fingerprint sensors verify identity. Wired connectivity ensures seamless data communication between the parties, and enables real-time processing and centralized management capabilities. This system provides a secure, scalable and user-friendly solution for attendance management.

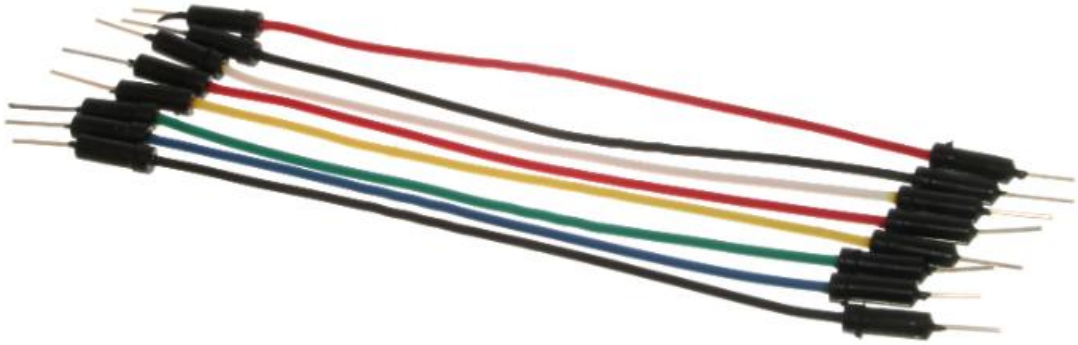

Figure5.5: Wires

5.4.6 HC-05BLUETOOTH MODULE

The **HC-05 Bluetooth module** is a widely used device for enabling wireless communication between microcontrollers and other Bluetooth-enabled devices. It operates on Bluetooth 2.0 technology and supports both master and slave modes, making it versatile for various applications. The module communicates with microcontrollers via UART (Universal Asynchronous Receiver-Transmitter) using AT commands for configuration, such as changing the module's name or setting its mode. It operates on 3.3V for logic but often has onboard voltage regulators, allowing it to be powered by 5V systems. The HC-05 is commonly used in projects like wireless data transfer, home automation, remote control systems, and robotics due to its ease of use, reliable performance, and low cost. **[5]**

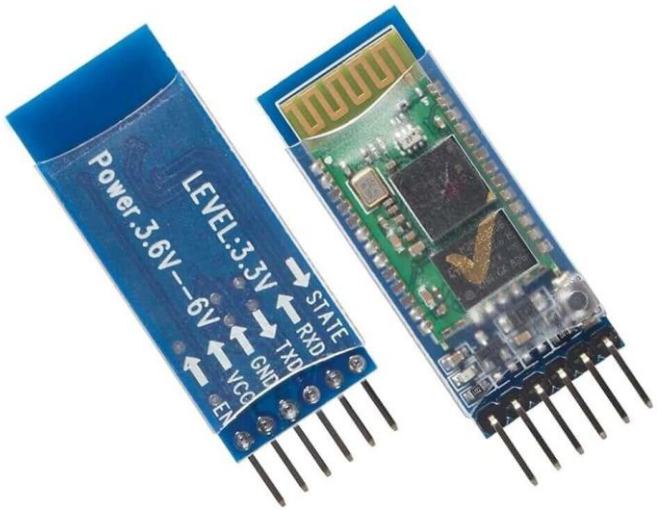

Figure 5.6: HC-05BLUETOOTH MODULE

5.5 Software Components:

We downloaded:

- **Visual Studio**
- **Arduino Uno**
- **MongoDB**

to program the system and the project, which are the electronic parts used and programmed on it.

5.6 Conclusion

In this chapter, we described the programming languages used in this system, as well as how the languages work in the system, and we described the different procedures and functions that were developed using the programming languages used to make the system ready for use.

➢ Chapter six –Coding and Implementation

6.1 Introduction

In Chapter 5, we explained the coding and implementation of system functions using Python and JavaScript programming languages.

This chapter will explain the process of testing the system. First, the system functions must be tested individually to ensure that they perform as expected, then the integration of the system units must be tested to ensure that the system works as a whole, that is, that its operations are identical to each other. After that, the results of the functions tested separately will be displayed, and the results of the system operations after the testing process for the entire system will also be available, so that we can ensure that the system performs as expected, or that there are some errors that need to be corrected. This process can detect whether the system is achieving its goals and functions or not.

6.2 System testing plan

- **Module and Unit Code Testing:** In this type of testing, each function was tested individually to ensure that it operates as expected, so that each function, code, and operation, will be tested by itself to make sure that this is the right implementation of each function.

- **System Integration Testing:** In this type of testing, the integration of all objects was tested to ensure that the whole system performs as expected, so that the system will test as a unit to make sure that all operations are integrated with each other and there is a match between all of them.

6.2.1 Unit code testing

- **Login interface:** After entering a valid login ID and password, the system will allow you to log in by comparing the entered fields with the system fields.

- **Pre-processing stage:** As mentioned earlier, this stage includes several steps, which are comparing the captured images with the images stored in the database and comparing the fingerprints with the stored fingerprints.

- **After applying all these steps, the results are obtained.**

6.3 System integration test

Here the integration of all modules was tested to ensure that the entire system works as expected.
In the previous section, we verified that the system modules work as expected after testing them using Python and JavaScript programming languages. The implemented functions were executed correctly and without errors, and the implemented screens on the website gave the expected results under different test conditions and without errors. This means that the interfaces in the system as a whole work as expected.

6.4 Testing plan results

The functions in the system work as expected when tested separately, work as expected when tested as a unit, each piece works as expected when tested separately, produces correct results when run as an integrated environment, and the entire system works as expected when the application is run as a unit.

6.5 Conclusion and recommendations

After developing this system, the system was tested according to its requirements specifications and ensured that every process works as expected, and the integration between all objects is done correctly.

Conclusions

This proposed system has been presented and described and after developing its functions, we conclude that it meets the requirements and objectives correctly as intended.

Thus, we have some comments regarding the parts we used (Dahua camera), at first, we used the camera after performing all the face detection, pre-processing and face image enhancement and the results were not satisfactory and did not comply with the required specifications. That camera did not meet the requirement we had previously set in this proposed system, and did not provide us with the intended results as required. Therefore, it was necessary to use an alternative advanced camera for face recognition, which is a Hikvision camera. In this case,

We created a database of eight people in order to test our proposed model, where we took a picture of each person. The classification results obtained were interesting.

Finally, this system can be implemented and applied almost anywhere, it is fast and accurate. It can be used in laboratories in schools, universities, hospitals, etc. The proposed system is suitable for those who want it; the cost is not high

Future works

- **Increase the number of cameras**:

  - Add more cameras to enhance coverage and monitoring.
  - Upgrade to higher quality and more accurate cameras for better clarity.

- **Use more fingerprint sensors**:

  - Integrate a larger number of fingerprint sensors to accommodate a higher volume of students.

- **Integrate the system with the university's electronic class system**:

  - Link the attendance system to the university's registration system.
  - Automatically add students to the attendance system as they register for different courses.

Time line

The timeline outlines a project plan for 2024/2025, distributed among three students. The project starts with idea selection and system requirements determination as a group in March and April, followed by task distribution such as drawing models, circuits, and writing the report. Later stages include building the hardware, developing the website, and system testing, with roles divided fairly between individual and group efforts. The final report and presentation are prepared collaboratively in December. This organization ensures efficient project progress and timely achievement of goals.

	2024								
	March	April	May	June	July	September	October	November	December
Choose idea	All together								
Determine system Requirements	All together	All together							
Determine component		All together							
Draw system models		Waseem							
Draw circuit		Mohammad							
Write the first report version		Ashour	All together						
Presentation				All together					
Build project hardware					All together	Waseem	All together		
Build the website					Waseem	Ashour			
System testing						Mohammad	All together	All together	
Re-write report								Mohammad	
Prepare the presentation								Ashour	All together

References

[1] Building a Facial Recognition System with React and OpenCV, Medium, 2024. [Online]. Available: https://medium.com/analytics-vidhya/face-recognition-login-using-reactjs-and-computer-vision-e98a892fd4ef.

[2] Lucidchart, "To create charts," 2024. [Online]. Available: https://lucid.app/users/login#/login?folder_id=recent

[3] PlantUML, "To create UML diagrams," 2024. [Online]. Available: https://planttext.com/.

[4] DB Diagram, "To create database schemas," 2024. [Online]. Available: https://dbdiagram.io/d/6768598bfc29fb2b3b1a9953.

[5] Bluetooth Communication between Arduino and Smartphone, Instructables, 2024. [Online]. Available: https://www.instructables.com/Arduino-and-Mobile-Bluetooth-Communication-messeng/.

[6] RealPython, "Face Recognition with Python and OpenCV," RealPython, 2024. [Online]. Available: https://realpython.com/face-recognition-with-python/.

[7] Developing a MERN Stack Application: MongoDB, Express, React, Node.js, YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=O3BUHwfHf84&t=8092s.

[8] Control your Arduino IoT projects with a MongoDB database, YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=a87dlq7pqB4.

[9] Implementing Fingerprint Authentication with Arduino, Instructables, 2024. [Online]. Available: https://www.instructables.com/Arduino-Fingerprint-Sensor-Tutorial/.

[10] GeeksforGeeks, "I2C Communication Protocol," 2024. [Online]. Available: https://www.geeksforgeeks.org/i2c-communication-protocol/.

[11] React Documentation, "Comprehensive reference on React, including components, state, and data management," React, 2024. [Online]. Available: https://react.dev/.

[12] Developing a MERN Stack Application: MongoDB, Express, React, Node.js, MongoDB, 2024. [Online]. Available: https://www.mongodb.com/resources/languages/mern-stack-tutorial.

Appendices

Appendix A: Hardware Specifications

A.1 Hikvision IP Camera

Feature	Details
Resolution	6 MP (3072x2048).
Night Vision	Up to 30 meters with Smart IR technology.
WDR Technology	Wide Dynamic Range (120dB) for enhanced image details in varied lighting.
Video Compression	H.265+ for optimized storage and streaming efficiency.
Network Support	Ethernet 10/100 Mbps for network connectivity.

Table 12.1

A.2 Arduino Uno

Feature	Description
Operating Voltage	5V
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Communication	UART, SPI, I2C
Flash Memory	32 KB (ATmega328P), 0.5 KB used by bootloader

Table 12.2

A.3 Fingerprint Sensor

Feature	Details
Model	AS608
Fingerprint Capacity	Up to 162 fingerprints
Sensor Type	Optical
Image Resolution	500 DPI
Verification Speed	< 1 second
Communication Interface	UART
Operating Voltage	3.6V - 6.0V
Module Dimensions	20mm x 60mm

Table 12.3

A.4 Bluetooth Module

Feature	Description
Communication	UART (Serial Communication)
Bluetooth Version	2.0 + EDR (Enhanced Data Rate)
Operating Voltage	3.3V to 5V
Operating Range	Up to 10 meters (Line-of-Sight)
Data Transmission Speed	Up to 2.1 Mbps (Basic Rate)

Table 12.4

A.5 LCD Screen

Feature	Description
Display Type	Alphanumeric LCD 16x2
Interface	I2C (Inter-Integrated Circuit)
Operating Voltage	5V DC
Data Lines Required	Only 2 (SDA for data, SCL for clock)

Table 12.5

Appendix B: Attendance Reports

B.1 Daily Report:

Student Name	Student Number	Attendance Time
razi sunbaty	22010489	2025-01-16 12:27:00
Mohammad Shaikh	21811334	2025-01-16 12:27:01
waheeb abu atallah	22011809	2025-01-16 12:27:14
abdalqader shtewi	22011256	2025-01-16 12:27:24
abdallah ewais	22010076	2025-01-16 12:27:25

Figure B.1 : Daily Report

B.2 Finaly Report:

Session Date	Student Name	Student Number	Attendance Status	Time of Attendance	Attendance Count
Thursday, 01/16/2025, 12:26					
	abdalqader shtewi	22011256	Present	02:27 PM	1
	razi sunbaty	22010489	Present	02:27 PM	3
	Mohammad Shaikh	21811334	Present	02:27 PM	3
	waheeb abu atallah	22011809	Present	02:27 PM	1
	abdallah ewais	22010076	Present	02:27 PM	1
	waseem Abunada	22010182	Present	2:30 PM	1

Figure B.2: Finaly Report

B.3 This picture below shows an email sent to the Lecturer that attendance has been taken.

Figure B.3: An Email Message

Appendix C: Code Snippets

C.1 Connecting to MongoDB

```
client =
MongoClient("mongodb+srv://waseemabunada202:Waseem12345@cluster0.rwpecbs.mongodb.net/atten
danceSystem?retryWrites=true&w=majority&appName=Cluster0")
db = client['attendanceSystem']
collection = db['students']

# Step 2: Fetch the course using the course_id
course_data = db['courses'].find_one({'_id': ObjectId(course_id)})
```

C.2 Face Detection and Matching

```
def analyze_image(frame):
                if frame is None:
                    print("No frame available for analysis.")
                    return

                print("Analyzing the image...")
                rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                face_locations = face_recognition.face_locations(rgb_frame)

                if not face_locations:
                    print("No faces detected. Skipping attendance update.")
                    return

                for face_location in face_locations:
                    face_encoding = face_recognition.face_encodings(rgb_frame,
[face_location])[0]
                    matches_students = face_recognition.compare_faces(student_encodings,
face_encoding, tolerance=0.6)
                    if True in matches_students:
                        matched_student_index = matches_students.index(True)
                        matched_image = student_images[matched_student_index]
```

C.3 Recording Attendance

The following code demonstrates how the system updates attendance records in the database:

1. **Increment Attendance Count for an Existing Student:** If a student is already recorded in the session, the system increments the `attendance count` field. **[11]**

```
db['courses'].update_one(
                                    {
                                        '_id': ObjectId(course_id),
                                        'sessions.date': new_session['date'],
                                        'sessions.attendance.student': student_id
                                    },
                                    {
                                        '$inc':
{'sessions.$.attendance.$[elem].attendance_count': 1}
                                    },
                                    array_filters=[{'elem.student': student_id}]
                                )
```

This query locates the session in the course and increments the attendance count for the specified student.

2. **adds a new attendance record.**

Add a New Attendance Record for a Student: If the student is not already recorded, the system

```
db['courses'].update_one(
                                    {'_id': ObjectId(course_id), 'sessions.date':
new_session['date']},
                                    {
                                        '$addToSet': {
                                            'sessions.$.attendance': {
                                                'student': student_id,
                                                'time_of_attendance': attendance_time,
                                                'attendance_count': 1
                                            }
```

This query ensures that a new attendance record is added for the student. The `$addToSet` operator prevents duplicate entries.

C.4 Exporting Attendance to Excel

```
student_attendance_df.to_excel("C:/Users/LENOVO/Desktop/" + attendace_file_path,
index=False)
```

C.5 Sending Data to the Server

```
def send_to_nodejs(attendance_data, courseId, sessionId):
    try:
        url = f'http://localhost:8000/get-attendance-from-
fingerprint/{courseId}/{sessionId}'
        payload = {
            'fingerprint_data': attendance_data
        }
        response = requests.post(url, json=payload)
```

Sending attendance data (which includes the student's ID, name, and timestamp) to the Node.js API for storage or processing.

C.6 Image Upload Using Multer

```
const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, 'students/');
  },
  filename: (req, file, cb) => {
    cb(null, Date.now() + path.extname(file.originalname));
  }
});
const upload = multer({ storage: storage });
```

- **Multer** is used for handling `multipart/form-data`, which is used for uploading files.
- Files are saved in the `uploads/` directory with a unique timestamp to avoid conflicts

C.7 LCD Setup and Initialization

Initialize the LCD screen to display information to the user.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);      // Line8
lcd.init();                             // Line 20
lcd.backlight();                        // Line 21
```

C.8 Fingerprint Sensor Setup

Initialize the fingerprint sensor using **SoftwareSerial** communication.

```
SoftwareSerial mySerial(2, 3);           // Line 9
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);  // Line 10
```

C.9 Bluetooth Module Setup

Initialize the Bluetooth module using **SoftwareSerial** communication.

```
SoftwareSerial bt(8, 9);                 // Line 13
bt.begin(9600);                          // Line 17
```

C.10 Serial Communication Setup

Initialize the serial communication with the computer.

```
Serial.begin(9600);                      // Line 17
```

C.11 Fingerprint Sensor Verification

Verify the connection with the fingerprint sensor using a password.

```
if (finger.verifyPassword()) {           // Line 27
  lcd.print("Fingerprint: OK");          // Line 28
} else {                                 // Line 30
  lcd.print("Fingerprint: Fail");        // Line 31
}
```

C.12 Fingerprint Scanning and Verification

Process and verify the fingerprint.

```
if (finger.getImage() == FINGERPRINT_OK) {             // Line 54
  if (finger.image2Tz() == FINGERPRINT_OK) {           // Line 55
    if (finger.fingerFastSearch() == FINGERPRINT_OK) { // Line 56
```

C.13 Enrolling Fingerprint

Enroll a new fingerprint in the system.

```
void enrollFingerprint(uint8_t id) {                   // Line 118
  finger.deleteModel(id);                              // Line 124
  // Steps for processing and storing the fingerprint
}
```

C.14 Storing Data in EEPROM

Store user data such as name and student ID in **EEPROM**.

```
void storeInEEPROM(uint8_t id, String name, String studentId) {   // Line 173
  EEPROM.write(address, name.length());                           // Line 175
  // Store name and student ID
```

C.15 Reading Data from EEPROM

Read stored data from **EEPROM**, such as name and student ID.

```
String readFromEEPROM(uint8_t id, bool isName) {                  // Line 185
  byte length = EEPROM.read(address);                             // Line 188
  // Read the data and store it in a variable
```

Appendix D: Protocols and Technologies Used

D.1 Overview

This appendix provides a detailed list of protocols and technologies used in the development of the project, explaining their purpose and role within the system architecture. **[12]**

D.2 HTTP (Hypertext Transfer Protocol)

- **Description:** HTTP is used for communication between the frontend (client-side) and the backend (server-side). The requests are made using **Axios**, a library for handling HTTP requests in JavaScript.
- **Usage in Project:**
  - **GET** requests for fetching data (e.g., fetching courses, student attendance records).
  - **POST** requests for sending data (e.g., adding fingerprint data, starting attendance sessions).
  - **DELETE** requests for deleting records.

  Example:

```
Axios.get('http://localhost:8000/my-courses', { ... })
```

D.3 JSON (JavaScript Object Notation)

- **Description:** JSON is used for data exchange between the client and server in a structured format.
- **Usage in Project:** Data like course details, students, and session information are exchanged in **JSON** format between the frontend and backend.

  Example:

```
const formData = { user_id: '', name: '', university_id: '' };
```

D.4 JWT (JSON Web Token)

- **Description:** JWT is used for authenticating users and ensuring secure access to protected resources on the server.
- **Usage in Project: JWT** tokens are included in the headers of HTTP requests to verify the identity of the user.

  Example:

```
Authorization: `Bearer ${localStorage.getItem('token')}`
```

D.5 Client-Side Validation (JavaScript)

- **Description:** JavaScript is used to validate form inputs on the client side before data is sent to the server.
- **Usage in Project:** Ensures that the data entered by users is valid, e.g., checking if the **user_id** is within the expected range.

  Example:

```
if (formData.user_id < 1 || formData.user_id > 100) {
  alert('ID must be between 1 and 100.');
  return;
}
```

D.6 Frontend Technologies (HTML, CSS, JavaScript)

- **Description: HTML**, **CSS**, and **JavaScript** are used to design and implement the user interface.
- **Usage in Project:**
  - **HTML** is used for structure.
  - **CSS** is used for styling and visual presentation.
  - **JavaScript** provides interactivity and handles form validation.