Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Attendo
(Smart Attendance System)

By

Mina Saad Allah Rizk Allah Abu El-Saud [Computer Science]

Mina Emil Fakhry Gaber [Computer Science]

Mina Gergs Nasif Beshay [Computer Science]

Esraa Fouad El-Said [Computer Science]

Hamdy Hassan Mostafa Mohamed [Computer Science]

Under Supervision of

Dr. Mohamed Mabrouk
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Mostafa Mahmoud
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

June 2023

Acknowledgement

First and foremost, we want to acknowledge and thank **God** for His guidance, blessings, and presence throughout our graduation project. His wisdom and support have been the foundation of our journey, and We are deeply grateful for His unwavering love and guidance.

We also want to express our heartfelt appreciation to our project supervisor **Dr. Mohamed Mabrouk** for their invaluable guidance, support, and encouragement. Their expertise and commitment to excellence have been instrumental in shaping the direction and success of our project.

We would like to extend my gratitude to the staff of **Faculty of Computer and Information Sciences, Ain Shams University** for providing a nurturing academic environment and the resources needed for our growth and development. Their dedication to education has greatly contributed to our learning experience.

We are indebted to **our families and friends** for their unwavering support, understanding, and encouragement throughout this project. Their faith in us and their prayers have been a constant source of strength and motivation.

Lastly, we express our appreciation to **all the authors, researchers, and contributors** whose works have been referenced in this documentation. Their contributions have significantly influenced and shaped the theoretical foundations and methodology of our project.

Thank you all.

Abstract

The Smart Attendance System aims to automate and simplify the attendance tracking process in various institutions. Traditional manual attendance systems can be time-consuming, error-prone, and inefficient. With the advent of new technologies like machine learning and computer vision, it has become possible to develop an intelligent system that can accurately and automatically capture attendance data.

The system utilizes a combination of hardware and software components to achieve its objectives. The hardware component consists of biometric devices such as fingerprint scanners or facial recognition cameras, which are used to uniquely identify individuals. The software component involves the development of a robust attendance management system that integrates with the hardware devices.

In operation, the Smart Attendance System captures individuals' biometric data during the enrollment process, creating a unique digital identity for each person. During attendance tracking, the system matches the captured biometric data with the enrolled identities, accurately recording the attendance of each person. This eliminates the need for manual record-keeping and reduces the potential for inaccuracies or fraudulent practices.

Additionally, the system provides real-time attendance monitoring, generating reports and notifications to keep administrators informed. Overall,

this technology enhances the efficiency of attendance management, reducing administrative burdens and facilitating more accurate attendance tracking.

The Smart Attendance System has the potential to streamline attendance management processes in various sectors, including educational institutions, workplaces, and events. By leveraging advanced technologies, it offers a more reliable, convenient, and secure attendance tracking solution, benefiting both administrators and attendees.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

- **QR:** Quick Response
- **URL:** Uniform Resource Locators
- **CNN:** Convolutional Neural Network
- **GUI:** Graphical user interface
- **API:** Application Programming Interface
- **REST:** Representational State Transfer
- **MTCNN:** Multi-task Cascaded Convolutional Networks
- **LFW:** Labeled Faces in the Wild
- **CUDA:** Compute Unified Device Architecture
- **UI:** User Interface
- **CPU:** Central Processing Unit
- **RAM:** Random Access Memory
- **GPU:** Graphics Processing Unit
- **UML:** Unified Modeling Language
- **ERD:** Entity Relationship Diagram

CHAPTER ONE

Introduction

1.1 Preface

A Smart Attendance System is an application or software tool that simplifies and automates the process of monitoring attendance in schools, colleges, companies, and other organizations. This system replaces the old-style paper-based attendance tracking system with a more advanced, digital version.

The Smart Attendance System is designed to keep records of student or employee attendance and simplify the process of maintaining attendance details. It is a time-saving solution that enables organizations to have an accurate and real-time record of attendance, which can be checked at any time for reference.

Typically, the system uses advanced technologies like facial recognition to track attendance. The data is then stored and analyzed to generate reports and attendance analytics, providing insights into attendance behavior and trends. This system not only saves time but also reduces the chances of errors and malpractices.

Overall, a Smart Attendance System project can greatly benefit institutions and organizations in terms of time-management, efficiency, and accuracy in attendance tracking.

While Smart Attendance Systems can be incredibly helpful for modern organizations, there are still some challenges and problems that need to be addressed. One of the main issues is with the accuracy and reliability of the

attendance data collected through the system. For instance, biometric scans or facial recognition might fail to capture attendance due to environmental factors like dim lighting or obstructions. Furthermore, there are concerns surrounding privacy and data protection, which may deter some individuals from using the system.

Another issue is resistance to change, as there are still some organizations and individuals that are hesitant or unwilling to adopt new technologies and systems. Some organizations may not have the budget to invest in this new technology, or they may not have the expertise to implement and manage the system effectively.

Additionally, Smart Attendance Systems are only as effective as the infrastructure that supports them. Poor connectivity, power outages, and lack of access to the internet can also hinder the effectiveness of an attendance system, rendering the data it collects invalid or incomplete.

In summary, the challenges of Smart Attendance Systems project include issues with accuracy and reliability of attendance data collection, concerns regarding privacy and data protection, resistance to change, and infrastructure constraints.

1.2 The Significance and Motivation

- **Increasing Efficiency:** Smart Attendance Systems provide a faster and more efficient alternative to traditional attendance tracking methods. These systems eliminate the need for manual entry and paperwork, which saves time and resources, and can reduce labor costs.

- **Accurate Data Collection:** With the use of biometric recognition and facial recognition, Smart Attendance Systems collect attendance data with a high level of accuracy, which reduces the chance of errors that can occur during a manual process of attendance tracking.

- **Real-time monitoring:** Smart Attendance Systems record attendance in real-time, which means that organizations can monitor their employees or students in a more efficient way. This system provides immediate information about who has arrived and who hasn't, which is particularly important in a time-sensitive environment.

- **Generating Reports:** Smart Attendance Systems can generate reports on attendance data, which can help management make informed decisions and increase accountability. The reports can be used to identify trends and patterns in attendance, alerting managers to critical issues early on.

- **Improved Security:** the introduction of biometric or ID card-based technologies in Smart Attendance Systems can significantly improve the security of an organization, reducing the chances of fraudulent behavior.

1.3 Aims and Objectives

- **Automating attendance tracking process:** The main objective of a Smart Attendance System project is to replace manual methods of attendance tracking with automated, digital systems. This will help to eliminate errors and make the process more efficient.

- **Increasing accuracy and reliability:** Smart Attendance Systems are designed to be more accurate and reliable than traditional methods of attendance tracking. By using technologies like biometric recognition and facial recognition, the system aims to build a more robust and trustworthy database for attendance records.

- **Simplifying attendance record-keeping:** An essential objective of a Smart Attendance System project is to provide an easy-to-use system that simplifies attendance record-keeping. The goal is to have the system generate attendance records automatically and provide real-time reporting.

- **Improving time-management:** Another objective of a Smart Attendance System project is to increase efficiency and reduce time wasted in attendance tracking. This will allow staff and management to use their time more effectively for other tasks.

- **Enhancing security:** Smart Attendance Systems can provide greater security than traditional methods, especially when using biometric identification. This system aims to help to ensure that only authorized individuals have access to certain areas of an institution.

- **Generating reports:** Smart Attendance System project aims to provide reports, which can help management gather insights into attendance behavior and trends. These reports can be used to support decision making and policymaking.

1.4 Methodology:

- **Define User Requirements**: Begin by defining the user requirements of our Smart Attendance System. The requirement definition process should consider factors like organization goals, the number of employees, work environment, and the existing systems.
- **Perform feasibility analysis:** Doing a feasibility study to assess the resources, cost, and technical factors that will affect the implementation of Smart Attendance System. This step will help us identify any technical or financial limitations that may impact the project's successful implementation.
- **Designing our Smart Attendance System:** Based on the user requirements and feasibility analysis, step into the design and development process of Smart attendance system. This stage will typically include designing the system architecture, identifying the hardware and software requirements, and selecting the appropriate technology stack.
- **Developing the Smart Attendance System:** Begin Development and testing of the Smart Attendance System. This stage should include implementing the designs, coding the software according to specifications and testing the system's architecture, functionalities, and security features.
- **System Integration and Deployment:** After successful development and testing, it's time to move to the integration of the whole Smart Attendance System, including database management, integration with the hardware components, and user interface deployment.

- **Training Employees and provide support:** Once the system is deployed, the next step is to train employees and other stakeholders on how to use it correctly. The training will help improve user adoption and ensure that you get maximum value from our Smart Attendance System. Providing post-deployment support may also be necessary.
- **Continuous Monitoring and Improvement:** The final step of methodology is continuous monitoring, system performance review, and further improvement on the Smart Attendance System. It is necessary to ensure the system gets better and function as expected, with improving technology and regular upgrade plans to new infrastructures.

1.5 Time Plan

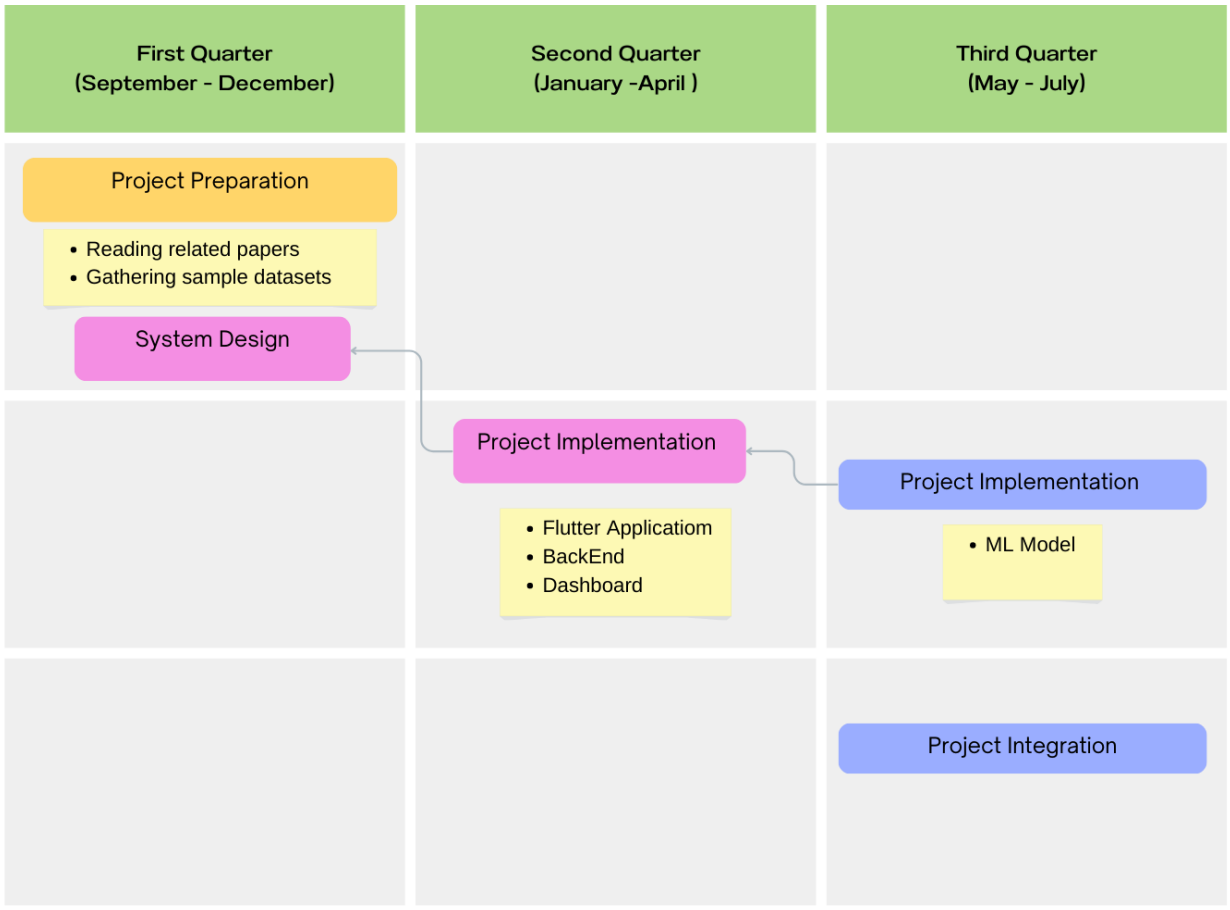

Figure 1.5 Time-plan

CHAPTER TWO

Literature Review

2.1 Introduction

The use of technology in education has become increasingly popular in recent years. One area where technology can be particularly useful is in attendance tracking. Traditional attendance systems can be time-consuming and prone to errors. A smart attendance system, on the other hand, can automate the process, making it more efficient and accurate. The aim of this project is to develop a smart attendance system for educational institutions.

Smart attendance systems are a more accurate, efficient, and transparent solution to traditional attendance tracking methods in educational institutions and workplaces. They offer advanced technologies such as face recognition to accurately track attendance in real-time and generate reports and insights that can help identify trends and patterns. These systems are more accessible and user-friendly and can be customized to meet the specific needs of different organizations. However, there are challenges and limitations that must be carefully considered and managed, such as privacy and data security issues, technical challenges, and user-friendliness.

2.2 Theoretical Background

The smart attendance system will be based on the principles of computer vision and machine learning. Computer vision refers to the ability of computers to interpret and understand visual information from the world around them. Machine learning, on the other hand, involves the use of algorithms and statistical models to enable computers to learn from data without being explicitly programmed.
Smart attendance systems that use face recognition and QR codes are based on two key technologies: computer vision and QR codes.

Computer vision is a field of study that focuses on enabling machines to interpret and understand visual information from the world around them. This involves developing algorithms and techniques that allow computers to process and analyze images and video data, extract relevant features, and make decisions based on that information. Face recognition is a specific use case of computer vision that involves identifying and verifying the identity of an individual based on their facial features.

QR codes, on the other hand, are two-dimensional barcodes that can be scanned using a smartphone or other smart devices to access information quickly and easily or perform an action, such as registering attendance. QR codes can be generated and printed on paper or displayed on a screen, and they can contain a variety of data, including text, URLs, and contact information.

Smart attendance systems that use face recognition and QR codes typically work by capturing an image of an individual's face and comparing it to a database of known faces to determine their identity. This can be done using a variety of techniques, including traditional computer vision algorithms such as Eigenfaces, as

well as more advanced deep learning models such as convolutional neural networks (CNNs).

In addition to face recognition, smart attendance systems may also use QR codes to provide an additional layer of verification and security. For example, an individual may be required to scan a QR code displayed on a screen or printed on a piece of paper to confirm their attendance. This can help prevent fraudulent attendance records and ensure that only authorized individuals are able to register their attendance.

2.2.1 How Does Face Recognition Work?

Face recognition is a technology that uses algorithms to identify and verify the identity of individuals based on their facial features. It involves capturing an image or video of an individual's face, extracting features, and comparing them to a database of known faces. There are different types of face recognition algorithms, including traditional computer vision techniques and advanced deep learning models. Face recognition has a wide range of applications, but there are also concerns about privacy and potential misuse.

2.2.2 How do QR codes work?

QR codes are two-dimensional barcodes that can be scanned using a smartphone or other mobile device equipped with a QR code reader app. They can store information such as text, URLs, and contact information, and are versatile and convenient. QR codes offer several advantages over traditional barcodes, including the ability to store more information and be read more quickly and accurately. They are commonly used in mobile marketing, e-commerce, and a variety of other industries and applications.

2.3 The previous studies and works

Several studies have been conducted in attendance tracking using computer vision and machine learning.

One research study by Singh et al. (2018) proposed a smart attendance system that used both face recognition and QR codes to track attendance in a university setting. The system achieved high accuracy and was able to reduce the time required for attendance-taking compared to traditional manual methods.

A study by Sahu and Bhatt (2018) examined the effectiveness of an RFID-based smart attendance system in a manufacturing company. The study found that the system led to a significant reduction in absenteeism and improved the accuracy and efficiency of attendance tracking. The system also helped the company to identify and address workforce management issues more effectively.

Another study by Khan and Ahmed (2019) examined the impact of a smart attendance system on student attendance and academic performance. The study found that the system led to a significant improvement in attendance rates, particularly among students who had previously been absent or late. The study also found that the system had a positive impact on academic performance, as students who attended more classes achieved higher grades.

Another study by Han et al. (2019) developed a smart attendance system that used a combination of face recognition and deep learning techniques to improve

accuracy and speed. The system was able to recognize faces even in low lighting conditions and achieved high accuracy rates even with large datasets.

Several studies have investigated the effectiveness of smart attendance systems in educational institutions. A study by Alshehri and Drews (2020) found that a biometric-based smart attendance system was more accurate and efficient than traditional attendance tracking methods such as paper-based registers and ID card scanning. The study also found that the system was well-received by students and teachers, who appreciated its user-friendliness and reliability.

In the context of education, smart attendance systems have been proposed to improve student engagement and reduce administrative burdens on teachers. A study by Tsoi et al. (2019) explored the use of a smart attendance system in a primary school setting and found that it improved attendance accuracy and reduced the time required for attendance-taking.

However, the use of smart attendance systems has also raised concerns about privacy, data security, and potential bias. Some studies have highlighted the need for ethical guidelines and regulations to govern the use of these systems and ensure that they are used in a fair and transparent manner.

Overall, previous studies and works have demonstrated the potential benefits of smart attendance systems, particularly in terms of accuracy, efficiency, and usability. However, there is still a need for further research and development to address concerns around privacy, security, and bias, and ensure that these systems are used in a responsible and ethical manner.

CHAPTER THREE

System Architecture and Methods

3.1 System Architecture

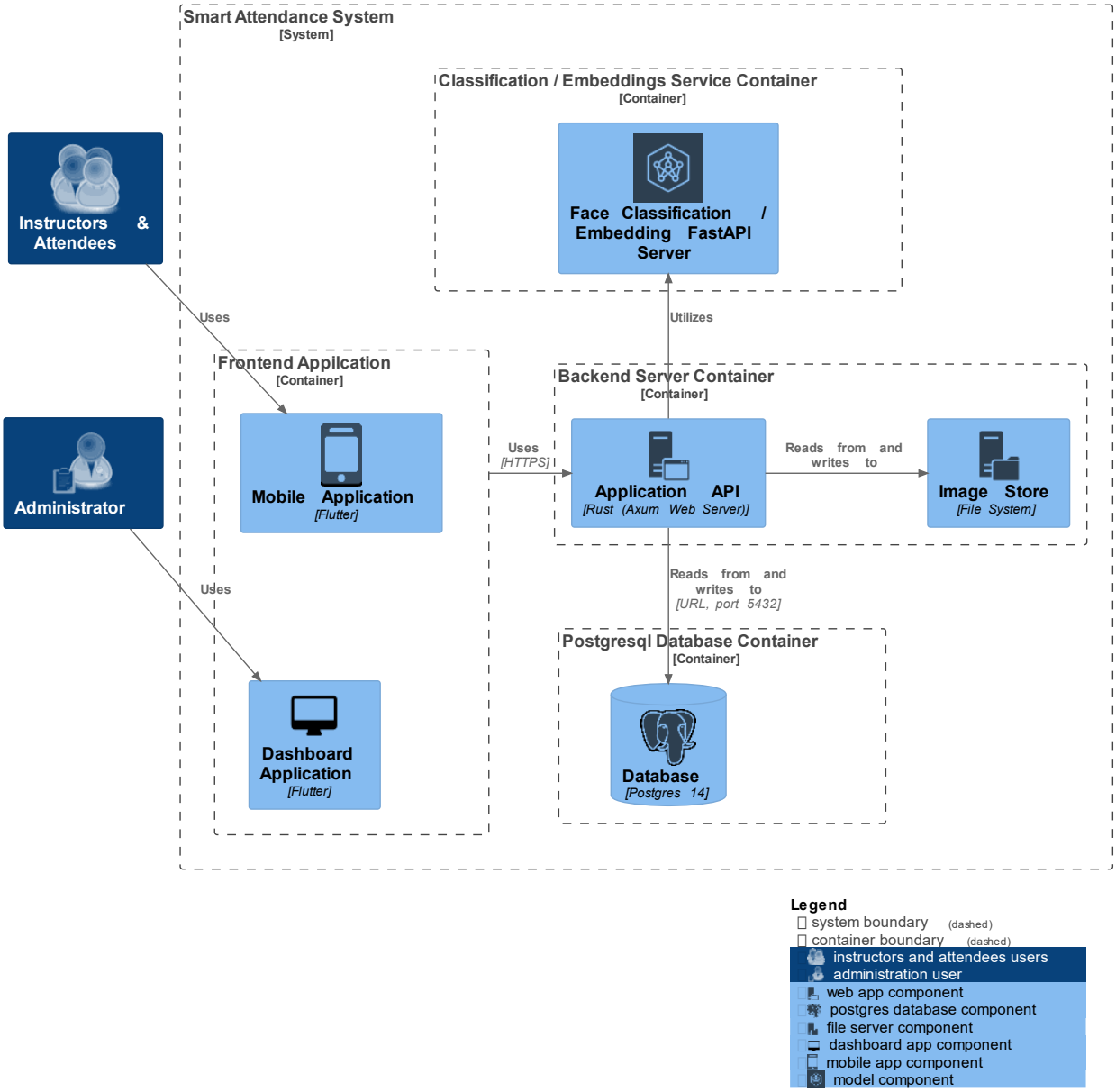

Figure 3.1 System Architecture

- **Face Recognition Module** is a container for faceial recognition that serve a Face Embedding Model and a Face Classification Model.

- **Face Embedding Model** is a deep learning CNN that transforms facial images into high-dimensional numerical representations, known as embeddings. These embeddings capture unique facial features and allow for efficient face recognition and similarity comparisons.

- **Face Classification Model** is a deep learning CNN that discriminate between different individuals, and performs matching or classification tasks to recognize known faces accurately.

- **Web Server Module** is a container that host the backend server.

- **Application API Module** is software component that handles incoming requests from clients (such as web browsers) and serves responses over the internet. It manages communication protocols and processes requests.

- **Image Store Module** is a file system module for manage and serving system's images.

- **PostgreSql Database Module** is a container DBMS stores information about the system resources (ie attendees, instructors, subjects, …) in records in a table format.

- **Front End Application** is software used by the system users (ie attendee, instructors, and admins) that provide an easy to use interface.

- **Mobile Application** is built using Flutter [1] can be designed for taking attendance using face recognition and QR code technologies. The application utilizes the device's camera to capture and analyze facial features or scan QR codes for identification purposes.

- **Dashboard Application** is a GUI that serves as an administrative tool for managing the system resources. It provides functionalities for adding new users, updating existing data, deleting records. The application performs these actions through a user-friendly interface, facilitating efficient

3.2 Methods and Procedurals

3.2.1 Containerization

plays a significant role in the development and deployment of our system. By utilizing containerization platforms like Docker [10] and Podman [14], the app and its dependencies can be packaged as isolated and portable containers. This allows for seamless deployment across different environments, ensuring consistency and eliminating compatibility issues. Containerization enables easy scalability by efficiently managing resources and facilitating the deployment of multiple instances of the app. It also simplifies the management of complex application stacks, making it easier to update or roll back versions. With containerization, apps can be efficiently distributed, deployed, and managed, resulting in improved flexibility, portability, and reliability.

3.2.2 Deep Learning

Deep learning is a powerful technology that has revolutionized the field of face recognition in mobile applications. By leveraging deep neural networks, these apps can accurately identify individuals by analyzing their unique facial features. Deep learning models, such as Convolutional Neural Networks (CNNs), extract intricate facial patterns and encode them into high-dimensional embeddings. These embeddings enable efficient face matching and verification, even in scenarios with variations in pose, lighting, and expression. The integration of deep learning in face recognition apps ensures robust and reliable identification, enhancing security, access control, and personalized user experiences.

3.2.3 RestAPI

plays a crucial role in our system, facilitating seamless communication between the app and the server-side components. By utilizing RESTful principles, the app can send requests to the server API for performing tasks expressed in the use cases. This includes but not limited to retreiving enrolled attendees in subjects, taking attendances by uploading image and retreiving subjects for attendees from the server, which then processes the data using deserialization algorithms. Integrating REST APIs in face recognition apps allows for easy scalability, interoperability, and flexibility, enabling developers to leverage powerful server-side capabilities while providing a smooth and efficient user experience on the client-side.

3.2.4 Database

Provides a structured and efficient way to store, manage, and retrieve of the system. The system utilize the database to store information such as user profiles, facial embeddings, subjects, … etc. Databases offer fast and reliable data access, enabling quick searches for information. They also support data integrity and security measures, ensuring that sensitive user information is stored and accessed appropriately. With a well-designed database, our system can efficiently organize and retrieve relevant.

3.3 System Diagrams

3.3.1 Use Case Diagram

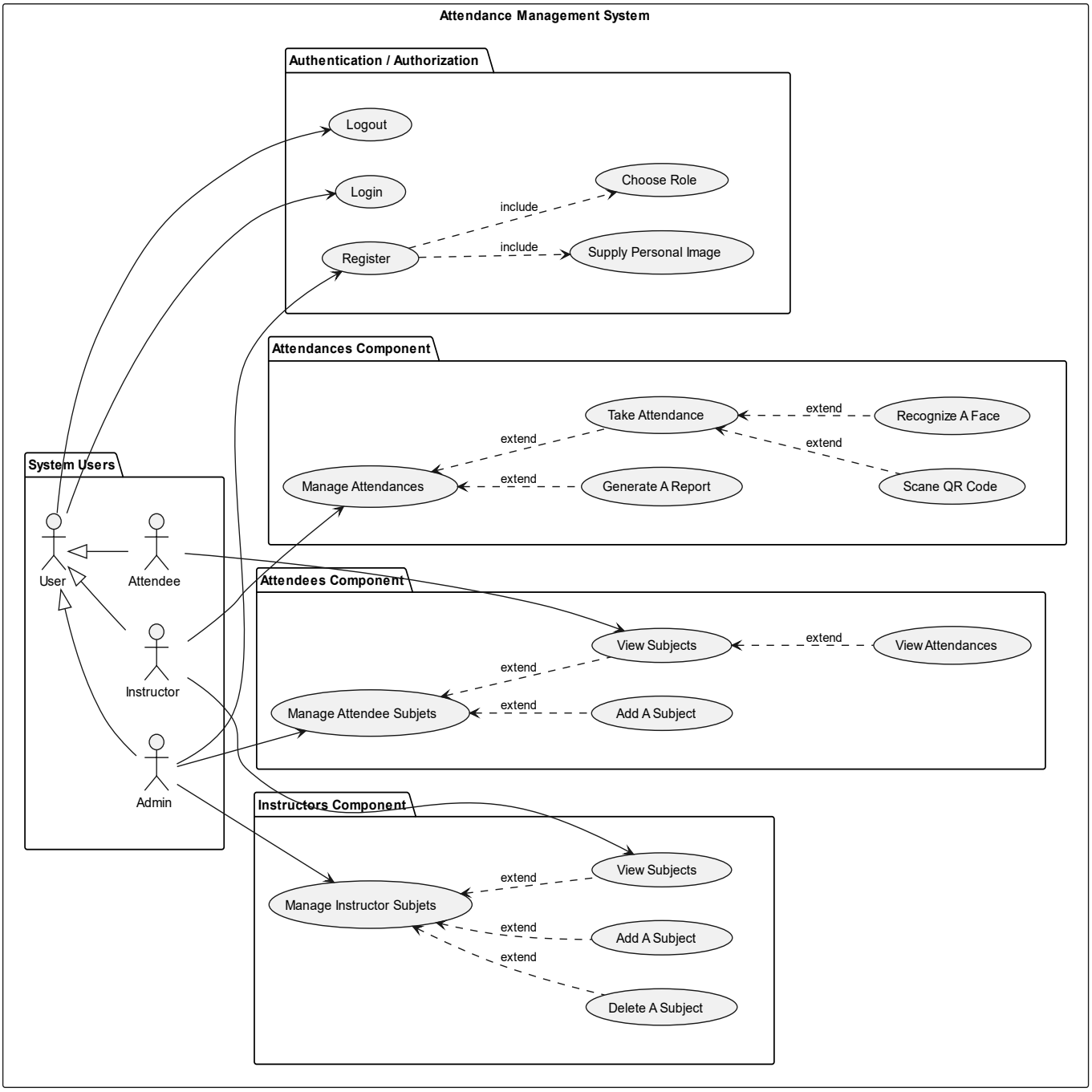

Figure 3.3.1 Use Case Diagram

3.3.2 Sequence Diagram

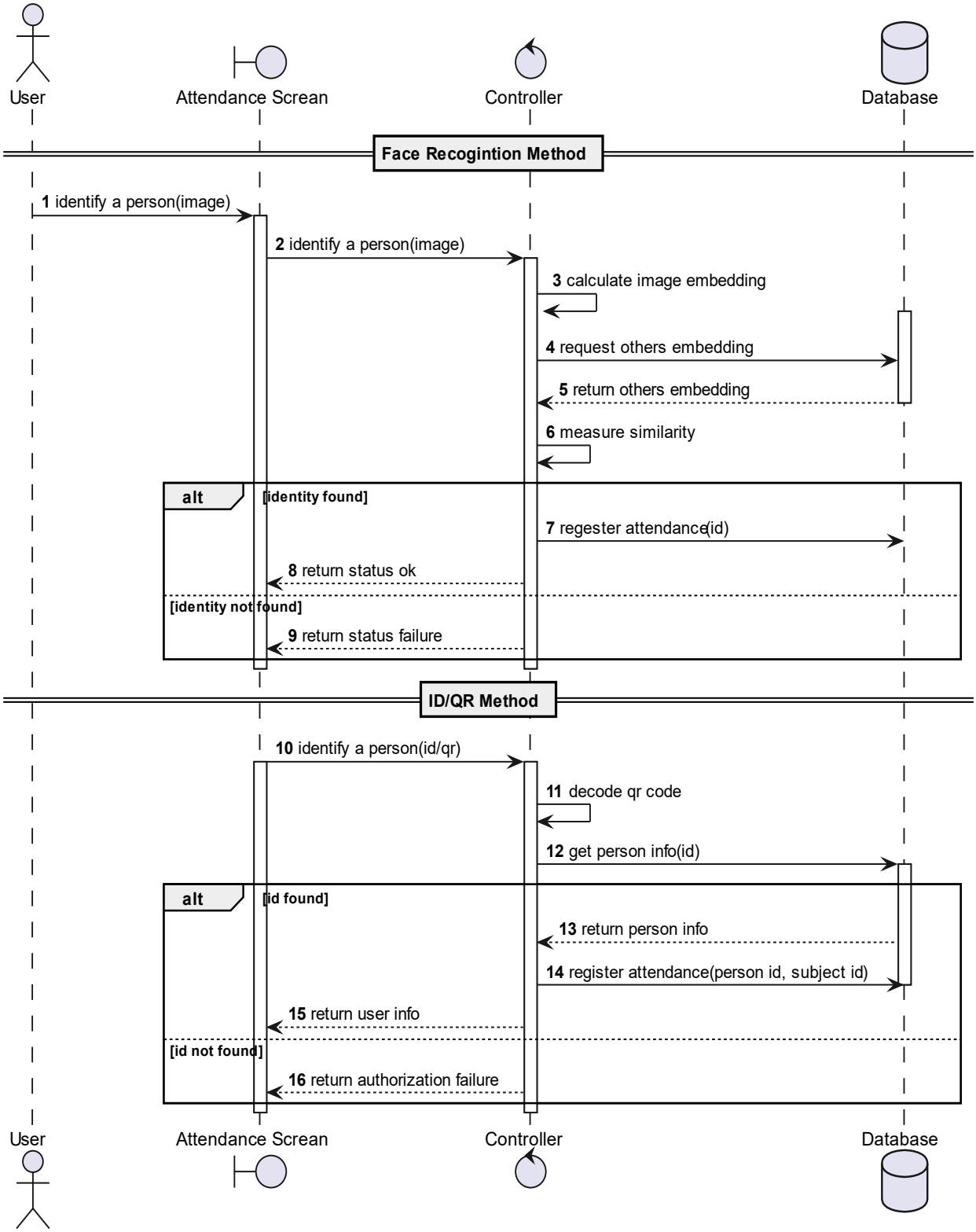

Figure 3.3.2 Sequence Diagram 1

3.3.2 Sequence Diagram (CONT.)

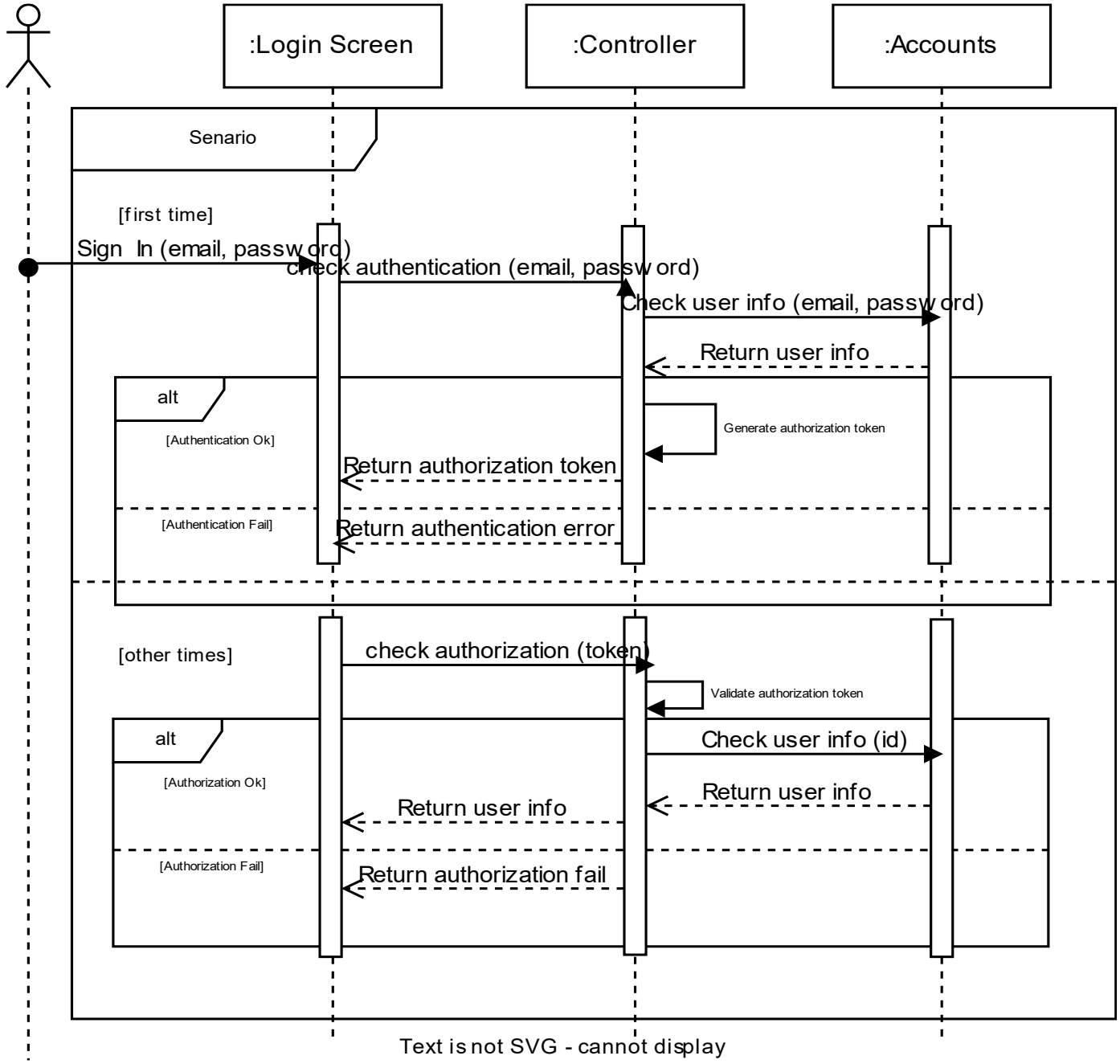

Figure 3.3.3 Sequence Diagram 2

3.3.3 Class Diagram

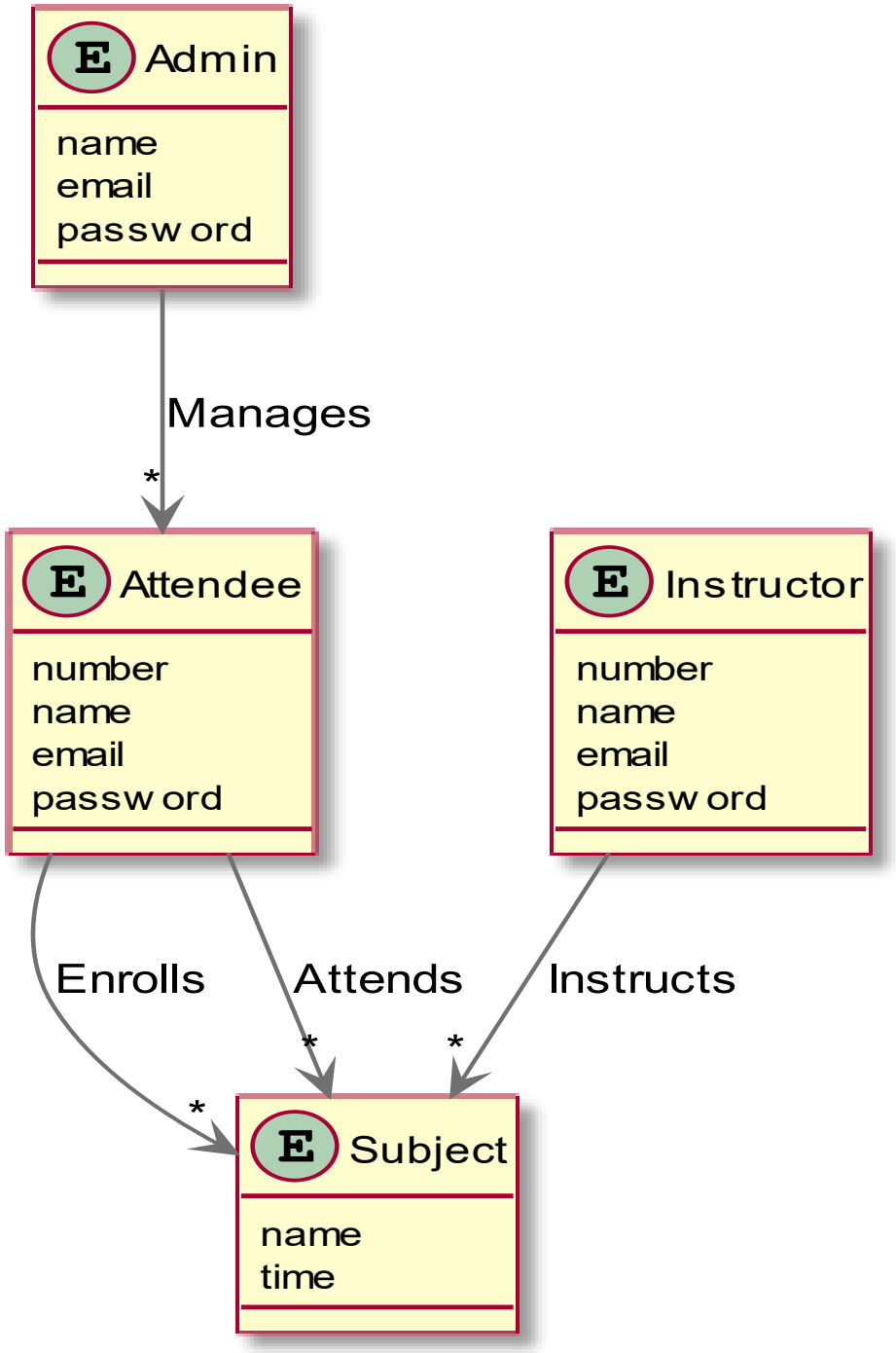

Figure 3.3.4 Class Diagram

3.3.4 Database Scheme

Figure 3.3.5 Database Scheme

CHAPTER FOUR

System Implementation and Results

4.1 Description of Materials and Programs Used

In this section, we provide a detailed description of the materials used in the implementation of our attendance system. These materials include the development frameworks, programming languages, technologies, and model recognition components utilized for building the various components of the system. The following materials were employed:

4.1.1 Recognition Model Components:

The model recognition components in our attendance system utilize the following models and techniques:

a. **Face Detection and Cropping**: We employed the MTCNN face detector for face detection and cropping. MTCNN (Multi-task Cascaded Convolutional Networks) is a widely-used deep learning-based face detection algorithm that accurately identifies and localizes faces in an image. It allowed us to extract facial regions of interest, which were subsequently used for further processing.

b. **Face Classification**: For face classification, we utilized the EfficientNet-B2 model. EfficientNet [4] is a family of convolutional neural network architectures known for their efficiency and high performance. We trained the EfficientNet-B2 model using the Labeled Faces in the Wild (LFW [7]) dataset, along with other relevant datasets, to recognize and classify faces accurately. We compared the performance of different models, including

EfficientNet-B1, and selected EfficientNet-B2 as it exhibited higher accuracy in face classification tasks.

c. **Face Embeddings**: As a secondary option for classification, we implemented a pretrained FaceNet [13] model to generate face embeddings. FaceNet is a deep learning model trained to map facial images to a high-dimensional vector space, where similar faces are closer together. These embeddings were computed and stored in the database, allowing for efficient face recognition by comparing and matching the embeddings during attendance tracking.

The model recognition components formed a crucial part of our attendance system, enabling accurate face detection, classification, and face embedding generation for efficient attendance tracking.

4.1.2 Backend Server:

The backend server of our attendance system was implemented using the Rust [2] programming language. Rust is a modern, compiled systems programming language that prioritizes performance, memory safety, and concurrency. We chose Rust for its unique features, such as the borrow checker, which prevents common memory errors like null pointer dereferences and data races. This helps ensure the stability and reliability of our system by minimizing potential vulnerabilities and crashes.

In addition, we utilized the Axum [3] web framework, a lightweight and asynchronous web server framework built specifically for Rust. Axum allowed us to develop a highly efficient and scalable backend server that could handle concurrent requests with ease. Its asynchronous nature enables our attendance system to handle multiple connections simultaneously, providing improved responsiveness and performance.

By leveraging Rust's compiled nature, memory safety guarantees, and the Axum web framework, we achieved a robust and secure backend server for our attendance system. These choices allowed us to optimize resource utilization, handle high traffic loads, and provide a seamless user experience.

4.1.3 Database:

To store and manage the attendance data, we employed the PostgreSQL [11] database. PostgreSQL is a powerful and open-source relational database management system known for its reliability, scalability, and extensive features. We utilized PostgreSQL to ensure the integrity and efficiency of data storage and retrieval operations in our attendance system

4.1.4 Dashboard Desktop Application and Frontend Mobile Application:

The user interface for our attendance system was developed using the Flutter framework, which is an open-source UI toolkit developed by Google. Flutter allows for the creation of natively compiled applications for various platforms, including mobile, web, and desktop. We utilized Flutter's cross-platform capabilities to develop a unified user interface that caters to both desktop and mobile devices.

4.1.5 Containerization:

To ensure easy deployment and management of our backend server, database, and FastAPI model recognition server, we utilized containerization technology. Specifically, we employed either Podman or Docker to containerize our components. Containerization allows for the encapsulation of an application and its dependencies into a lightweight, isolated container. This approach provides numerous benefits, such as simplified deployment, enhanced scalability, and improved system reliability.

By containerizing our backend server, we isolated its execution environment, ensuring that it can run consistently across different systems and environments. This eliminates potential compatibility issues and reduces the likelihood of dependency conflicts.

Similarly, containerizing our database allowed us to package it with its specific configurations and dependencies, facilitating seamless deployment and ensuring data integrity.

Furthermore, we containerized the FastAPI model recognition server, enabling us to easily manage and scale the server independently from other components. This modular approach allows for efficient resource allocation and facilitates the horizontal scaling of the model recognition capabilities as needed.

4.1.6 Datasets:

In our experiment, we utilized four distinct datasets to train and evaluate our models. Each dataset was carefully selected to provide diverse and representative examples for robust model performance. The four datasets used were as follows:

- Personal Dataset:

The first dataset consisted of a collection of personal photos of the developing team. These images were captured specifically for this project, featuring various individuals from the team. The personal dataset helped us test the model's ability to recognize and generate accurate representations of familiar faces.

- Celebrity Dataset:

The second dataset was created by collecting images of ten well-known celebrities. These images were gathered from different sources, such as public appearances and official photo shoots. The celebrity dataset allowed us to assess the model's capability to handle widely recognized faces and generate realistic celebrity-like representations.

- LFW Dataset:

The third dataset used was the Labeled Faces in the Wild (LFW) dataset. LFW is a popular benchmark dataset in face recognition research. It contains a large number of images captured under unconstrained conditions, encompassing a wide range of subjects. By incorporating the LFW dataset, we aimed to evaluate the model's performance in recognizing and generating faces from diverse sources.

- Microsoft Artificial Dataset:

Lastly, we incorporated the Microsoft Artificial Dataset [8]. This dataset, created by Microsoft, comprises a significant number of artificially generated face images. By including this dataset in our experiments, we intended to assess the model's ability to handle synthetic faces and produce high-quality representations.

By utilizing these four datasets, we aimed to train our models on a diverse range of face images, including personal photographs, celebrity images, real-world unconstrained data, and artificially generated examples. This comprehensive approach allowed us to evaluate the models' performance under various scenarios and test their generalization capabilities.

4.1.7 Diagraming Tools:

- Draw.io: Draw.io [15] is a web-based diagramming tool that allows users to create various types of diagrams, including flowcharts, UML diagrams, network diagrams, and more (used for time plan Gantt chart diagram).
- PlantUML: PlantUML [16] is a text-based diagramming tool that allows users to create diagrams using a simple and intuitive syntax (system architecture C4 diagram, use case diagrams, sequence diagrams, ERD diagram)
- pgAdmin: Schema Generation: pgAdmin [17] is a popular open-source administration and development platform for PostgreSQL databases (database diagram).

4.2 Setup Configuration (Hardware):

The setup configuration of the hardware refers to the specifications and requirements of the computer systems used in the implementation of our attendance system. The hardware setup is crucial for ensuring optimal performance, reliability, and compatibility across different components. Here is a breakdown of the hardware requirements for each aspect of our system

4.2.1 Backend Server and Database:

The backend server and database can be hosted on any machine that supports containerization. However, using a Linux-based system is preferred due to its robustness and compatibility with containerization technologies like Podman or Docker. It is recommended to allocate sufficient resources, including CPU cores, RAM, and storage, based on the expected load and scalability requirements of the attendance system.

4.2.2 FastAPI Model Server:

The FastAPI model recognition server can be run on a CPU-based system. However, utilizing a CUDA-capable device, such as an NVIDIA GPU, can significantly improve the inference speed of the face recognition model. CUDA (Compute Unified Device Architecture) is a parallel computing platform and application programming interface (API) model created by NVIDIA. It allows for efficient execution of parallel tasks on the GPU, resulting in faster computation for deep learning algorithms.

4.2.3 Frontend Applications:

The frontend applications developed using Flutter are cross-platform, meaning they can run on various operating systems including Android, iOS, Linux, Windows, and macOS. The hardware requirements for running the frontend applications are determined by the specific platform on which they are deployed. For instance, running the Flutter mobile app requires a compatible mobile device, while running the Flutter desktop app requires a suitable computer system with the required specifications for the target operating system.

By selecting the appropriate hardware and ensuring compatibility across the different components, we can create a robust and scalable system that meets the needs of our users across various platforms.

4.3 Experimental and Results:

4.3.1 Classification Model:

In our experiment, we explored the performance of four different classification models: EfficientNetB0, EfficientNetB2, VGGFace [5], and ResNet50 [6]. We evaluated these models on three distinct datasets: LFW, Celebrity, and Personal. Each dataset provided a unique set of challenges and variations in facial images.

We systematically combined each model with every dataset to observe the resulting accuracy and loss. By examining the combination of models with different datasets, we aimed to understand the strengths and weaknesses of each model in different contexts.

In the following table we trained the models for 50 epochs and observed:

Table 4.3.1 Observations

Model	Dataset	Time	Test Accuracy (+/- 0.05)
EfficientNet-B0	Personal	16 sec	0.7833
EfficientNet-B0	Celebrity	29 sec	0.6
EfficientNet-B0	LFW	52 sec	0.74
EfficientNet-B2	Personal	25 sec	**0.9**
EfficientNet-B2	Celebrity	48 sec	**0.75**
EfficientNet-B2	LFW	77 sec	**0.88**
VGGFace	Personal	94 sec	1
VGGFace	Celebrity	172 sec	0.7
VGGFace	LFW	226 sec	0.74
ResNet-50	Personal	58 sec	0.7333
ResNet-50	Celebrity	110 sec	0.75
ResNet-50	LFW	145 sec	0.96

Based on our comprehensive evaluation of the four classification models—EfficientNetB0, EfficientNetB2, VGGFace, and ResNet50—in combination with the LFW, Celebrity, and Personal datasets, we have determined that EfficientNetB2 is the most suitable model for our intended application. Through rigorous analysis, we found that EfficientNetB2 consistently achieved high accuracy and demonstrated remarkable performance across multiple datasets.

EfficientNetB2 exhibited superior results on both the Celebrity and Personal datasets, indicating its ability to capture intricate facial features and adapt well to various image variations. Its impressive performance, combined with its efficiency in terms of computational resources, makes EfficientNetB2 an ideal choice for our facial recognition task.

Considering the importance of accuracy, generalization capabilities, and computational efficiency, we have made an informed decision to employ EfficientNetB2 as our primary classification model for the proposed application. This selection ensures a reliable and efficient facial recognition system that can effectively handle diverse real-world scenarios.

Test Accuracy for EfficientNet-B2 on the three datasets:

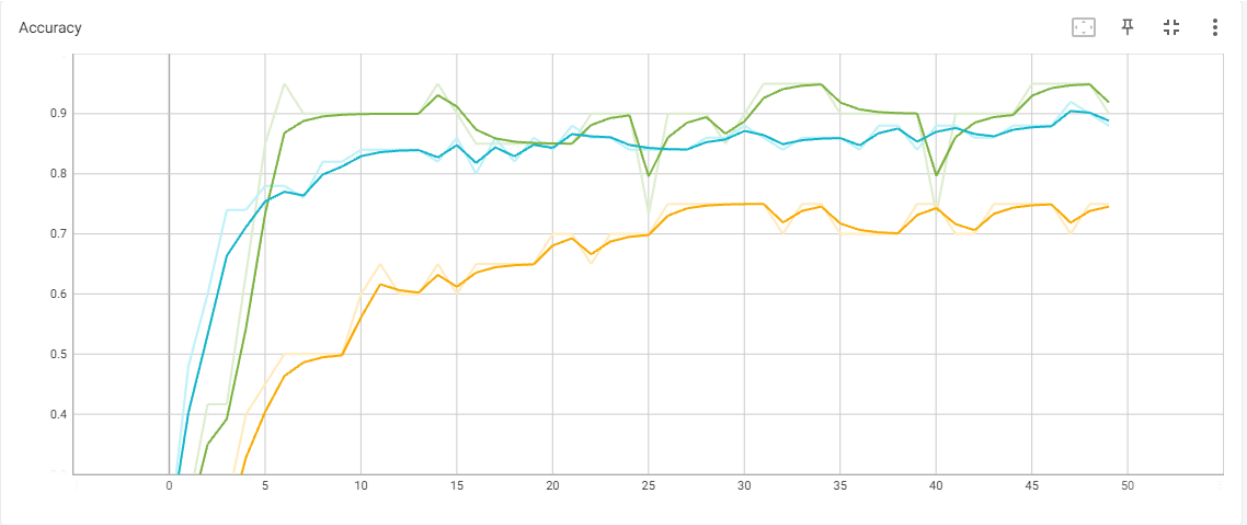

Figure 4.3.1.1 Accuracies

- LFW
- Personal
- Celebrity

Test Loss for EfficientNet-B2 on the three datasets:

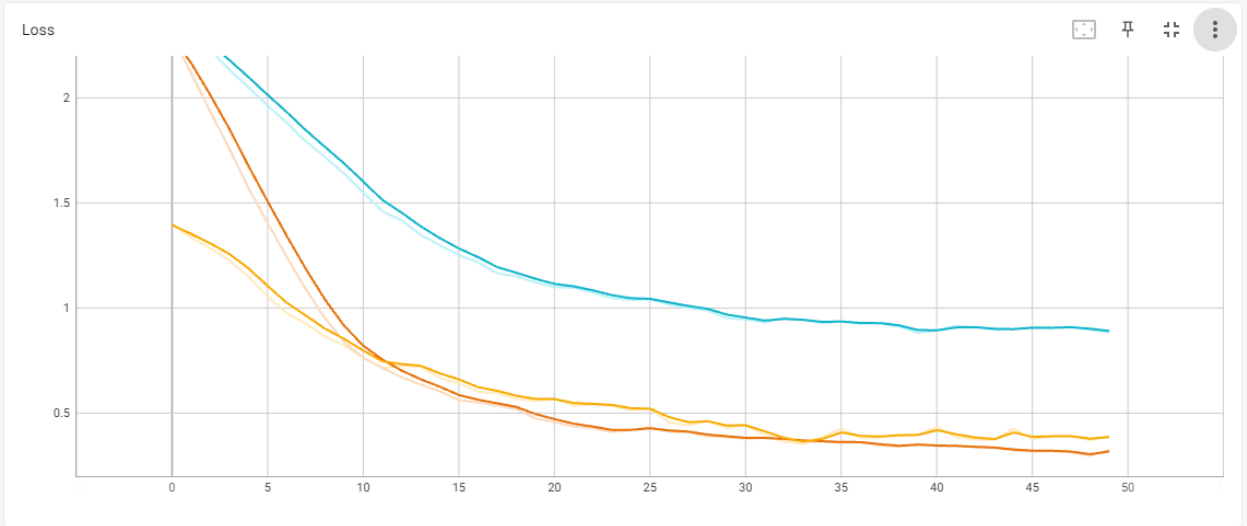

Figure 4.3.1.2 Loss

- Celebrity
- LFW
- Personal

4.3.2 Embedding:

As another technique, we experimented face embedding to transform a face image into a high-dimensional feature representation. This representation, also known as an embedding, captures the unique characteristics of a face in a way that facilitates measuring similarities or distances between faces.

In our experiments, we employed a Siamese network architecture combined with the triplet loss function. The Siamese network consists of three identical subnetworks, sharing the same weights. It takes three face images as input and produces their corresponding embeddings.

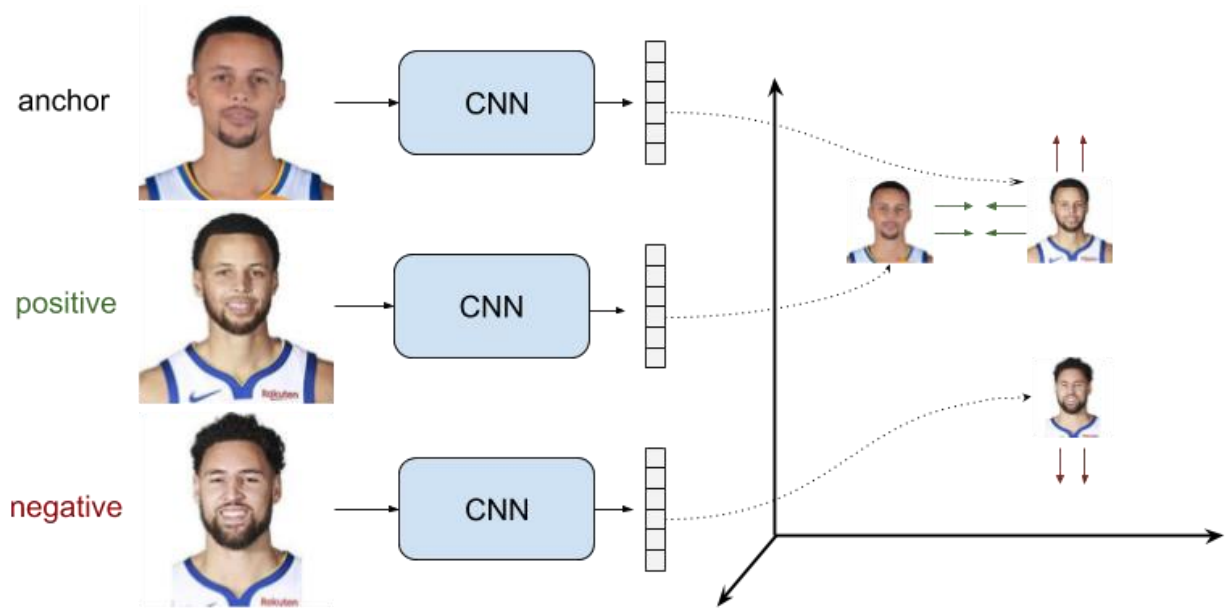

Figure 4.3.2 Embedding

The triplet loss function is then used to train the network by minimizing the distance between an anchor face embedding and a positive (same identity) face embedding, while maximizing the distance between the anchor face embedding and a negative (different identity) face embedding.

$$loss = \max\left(0, \left||A - P|\right|^2 - \left||A - N|\right|^2 + a\right) \quad (4.1)$$

A as the anchor embedding vector,

P as the positive embedding vector,

N as the negative embedding vector,

α as the margin that determines the minimum separation between positive and negative pairs.

The naive approach in face recognition involves randomly selecting triplets for training the embedding model without considering the difficulty of the samples. In this case, the model learns from triplets that may not provide sufficient discriminative information, leading to suboptimal performance.

To overcome this problem, we employed the online triplet [19] approach which dynamically selects hard triplets during training. A hard triplet consists of an anchor, a positive (same identity) sample, and a negative (different identity) sample that are challenging to distinguish. By focusing on hard triplets, the model is pushed to learn more robust and discriminative embedding.

As our experiments progressed, we encountered difficulties in achieving optimal performance solely with our custom-built network architecture. To overcome this, we decided to leverage the power of transfer learning and adopted the FaceNet pretrained model. FaceNet is a widely recognized face recognition model trained on a large-scale dataset and has shown excellent performance in embedding faces into a high-dimensional space. By using the FaceNet model, we were able to benefit from its robust features and achieve higher accuracy in our experiments.

CHAPTER FIVE

Run the Application

5.1 Mobile Application:

5.1.1 Installation Guide:

To use our Smart Attendance System, you need to download the Android Application Package (APK) [20], and then install it.

5.1.2 User Manual:

1. Splash Screen
   This is the first screen that appears in the application.

Figure 5.1.2.1 Splash Screen

2. Onboarding Screen
   This screen provides a guided introduction.

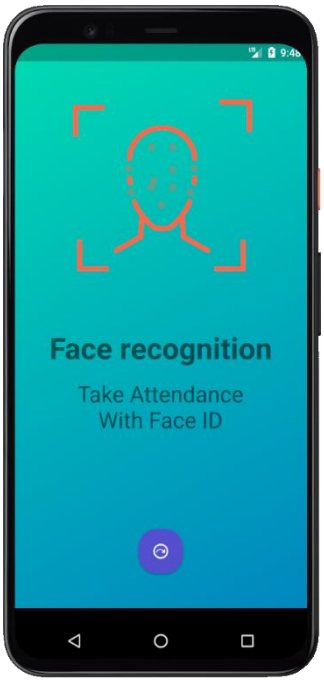

Figure 5.1.2.2 Onboarding Screen

3. Login Screen
   In this screen, you can enter your Email and password to enter.

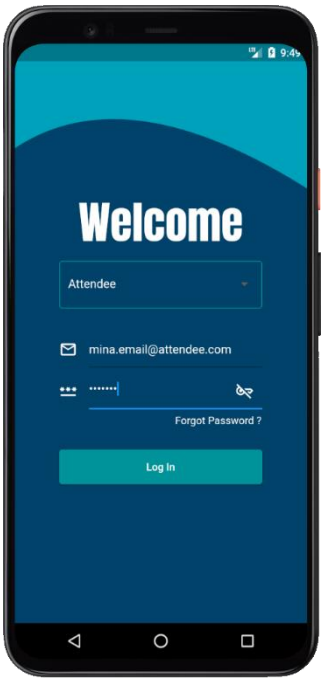

Figure 5.1.2.3 Login Screen

4. Home Screen [Instructor]
   This screen shows a simple info about all the subject that is assigned to the instructor.

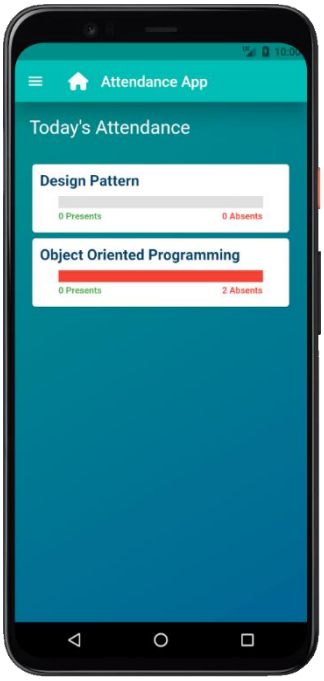

Figure 5.1.2.4 Home Screen [Instructor]

5. Side Navigation Menu
   In this screen, you can navigate between screens.

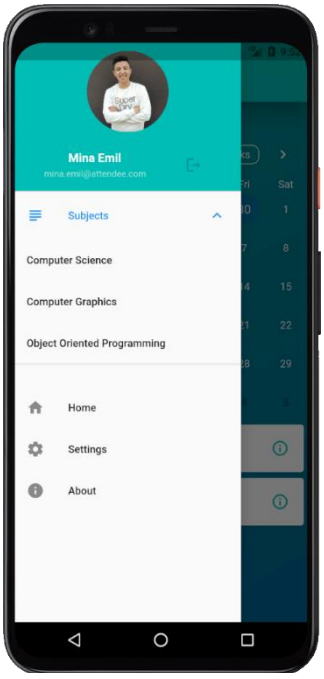

Figure 5.1.2.5 Side Navigation Menu

6. Taking Attendance Screen [Instructor]
   In this screen, you can take attendance with several methods.

Figure 5.1.2.6 Taking Attendance 1

Figure 5.1.2.7 Taking Attendance 2

7. Generating Report Screen [Instructor]
   In this screen, you can generate reports about the taken attendance.

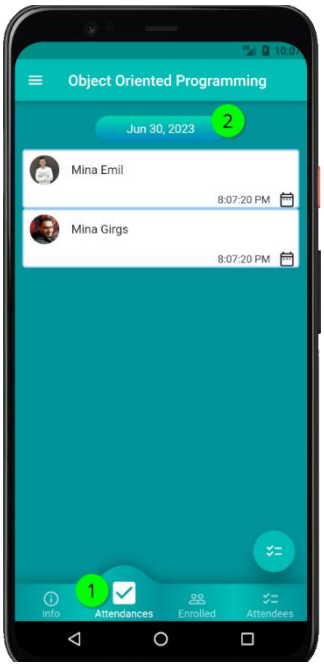

Figure 5.1.2.8 Generating Report 1

Figure 5.1.2.9 Generating Report 2

9. Home Screen [Attendee]
   This screen shows a calendar that shows info about the enrolled subjects.

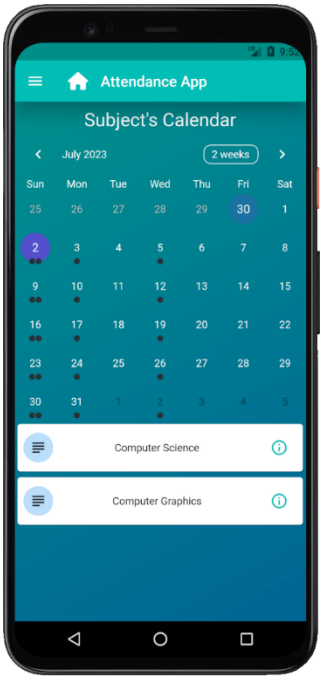

Figure 5.1.2.10 Home Screen [Attendee]

8. Subject Info Screen [Attendee]
   This screen shows info about the selected subject.

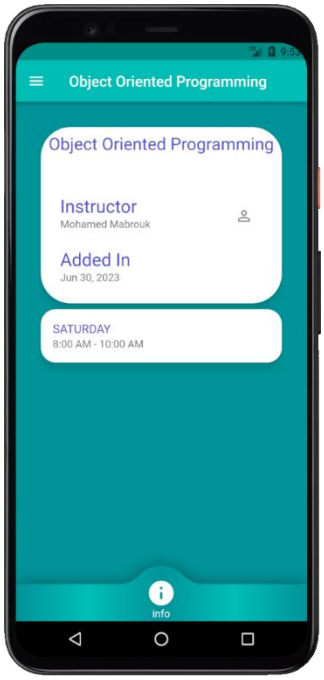

Figure 5.1.2.11 Subject Info Screen [Attendee]

10. Setting Screen
    In this screen, you can the language and other configuration.

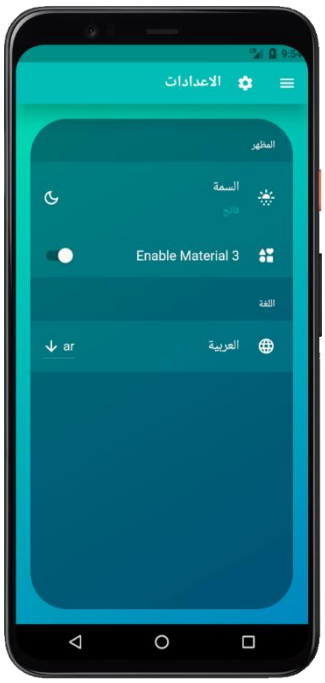

Figure 5.1.2.12 Setting Screen

5.2 Dashboard:

5.2.1 Installation Guide:

To use our Smart Attendance System Dashboard, you need to download the Application [21], and then install it.

5.2.2 User Manual:

1. Create Attendee or Instructor
   This is the steps to create an attendee or an Instructor.

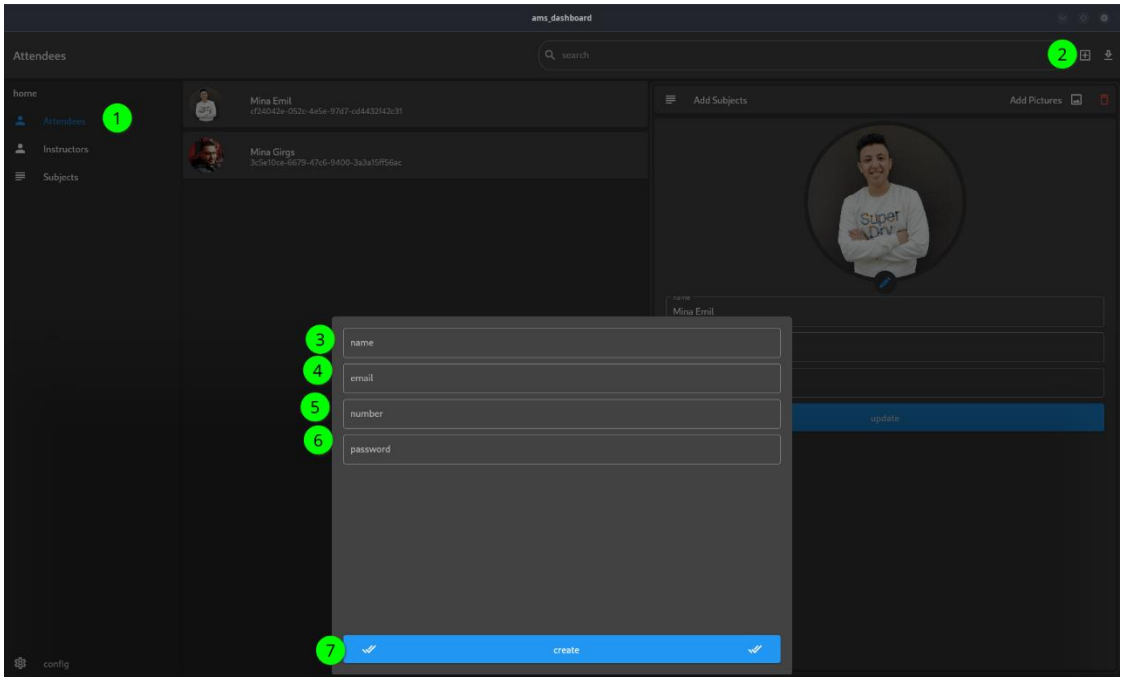

Figure 5.2.2.1 Create Attendee or Instructor

2. Update Attendee or Instructor
   This is the steps to update an attendee or an instructor.

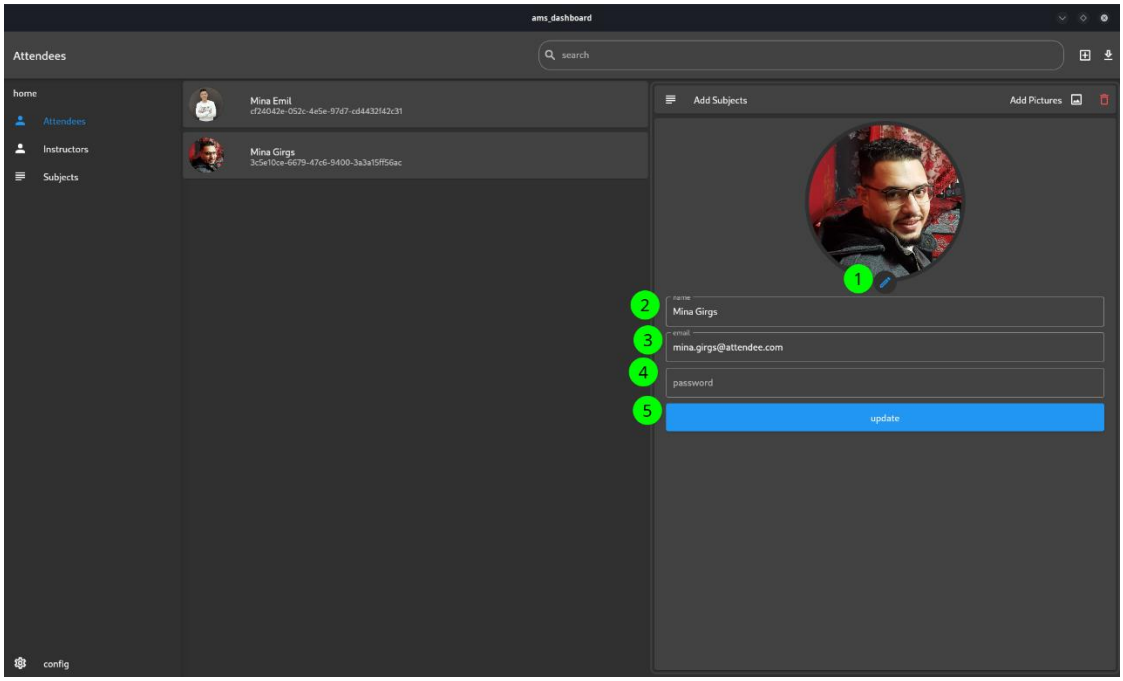

Figure 5.2.2.2 Update Attendee or Instructor

3. Create Subject
   This is the steps to create a subject.

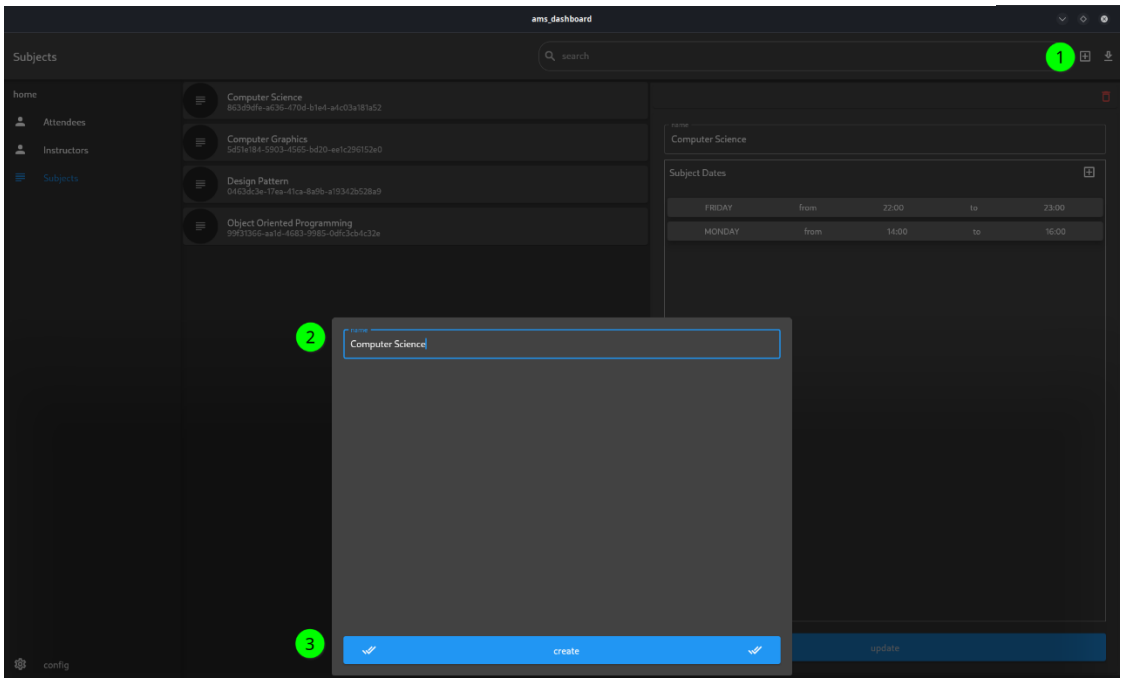

Figure 5.2.2.3 Create Subject

4. Update Subject
   This is the steps to update a subject.

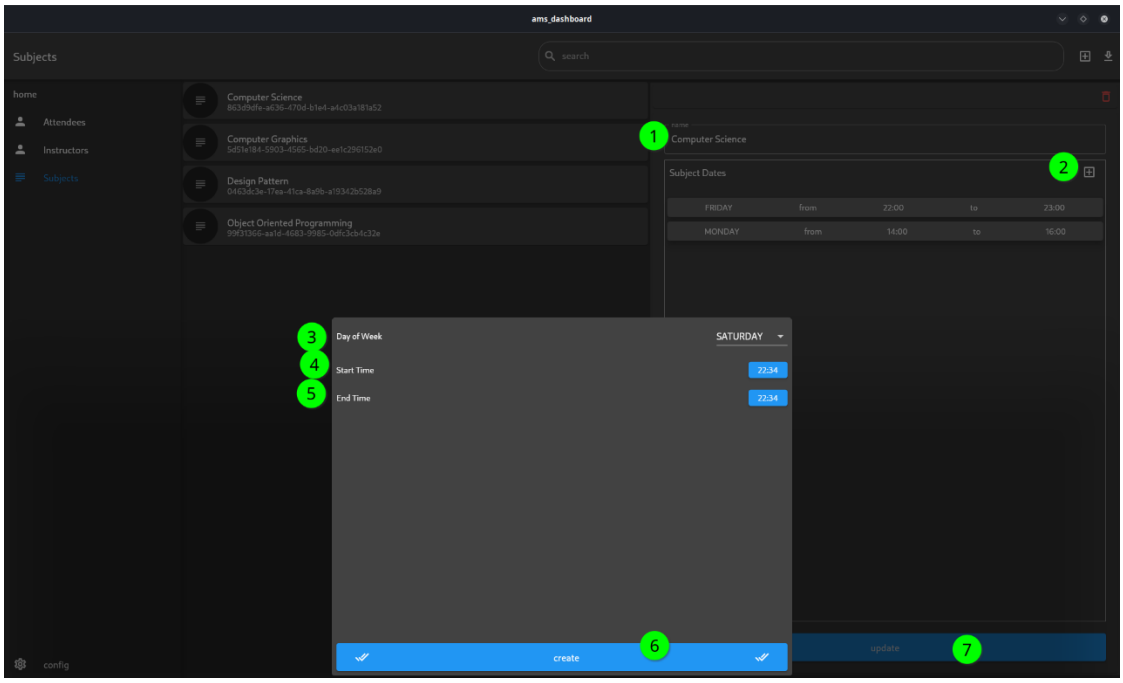

Figure 5.2.2.4 Update Subject

5. Add Subject to Attendee or Instructor
   This is the steps to update a subject.

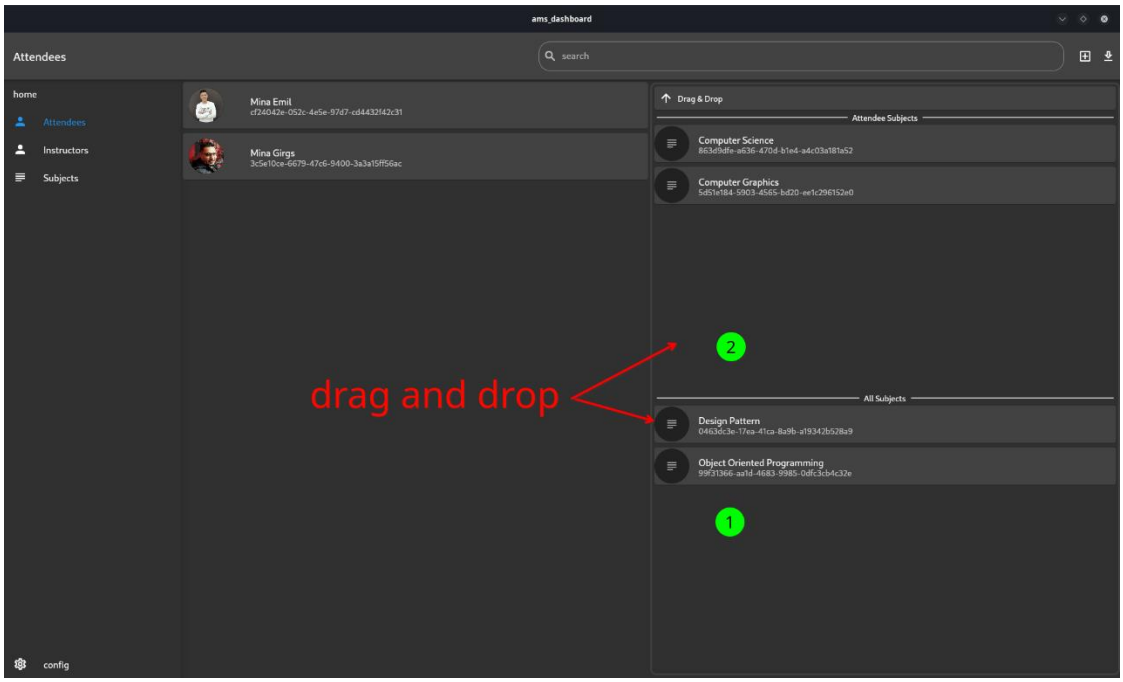

Figure 5.2.2.5 Add Subject to Attendee or Instructor

6. Responsive
   The Dashboard is responsive also.

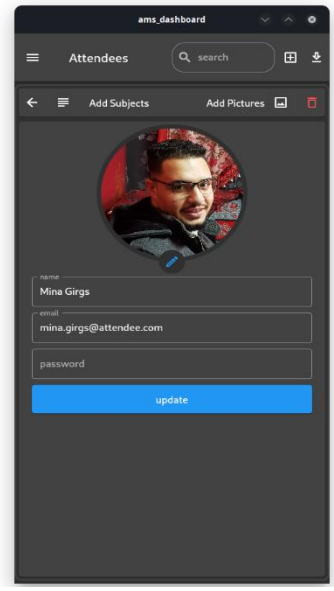

Figure 5.2.2.6 Responsive

5.3 Backend Documentation:

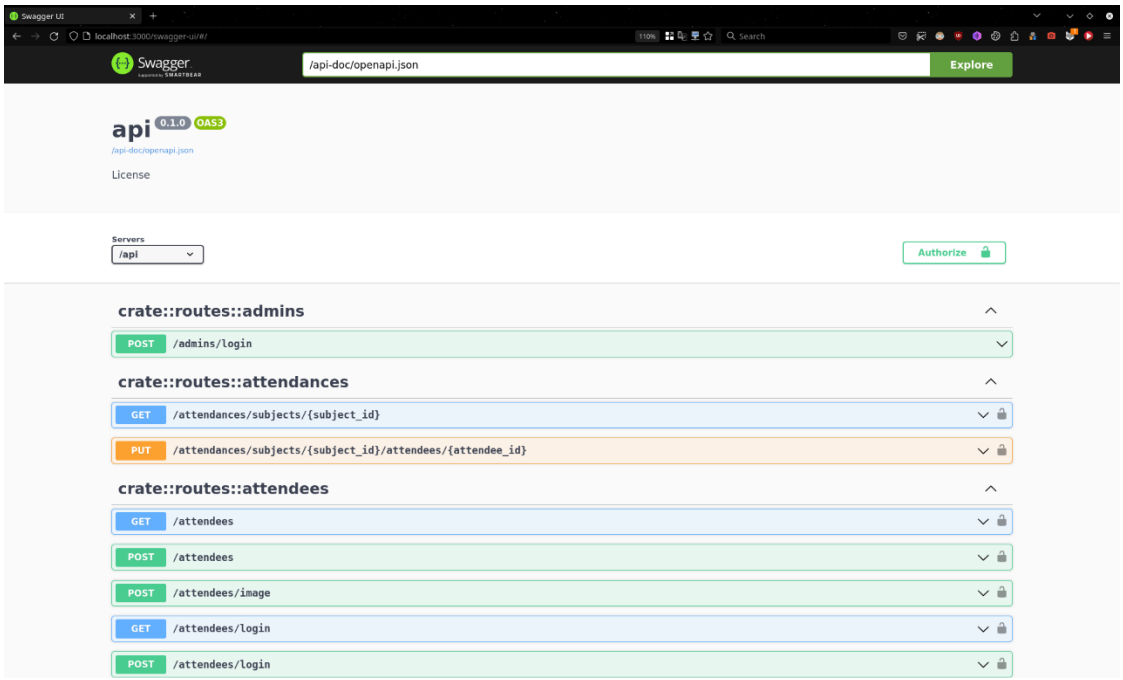

Figure 5.3 Backend Documentation

CHAPTER SIX

Conclusion and Future Work

6.1 Conclusion:

The objective of our graduation project was to create a smart attendance system that leverages face recognition technology and integrates a Rust backend with the Axum framework, accompanied by a Flutter mobile application UI. The system aimed to automate attendance recording, enhance accuracy, and provide an intuitive user experience. The following key outcomes were achieved:

6.1.1 Face Recognition with EfficientNetB2:

The integration of the EfficientNetB2 classification model enabled accurate face recognition within the attendance system. The model demonstrated reliable performance in identifying individuals from a registered database of faces, allowing for efficient attendance tracking and management.

6.1.2 Rust Backend with Axum Framework:

The utilization of Rust as the backend programming language, coupled with the Axum framework, provided a robust and scalable infrastructure for the smart attendance system. Rust's emphasis on safety and performance, combined with Axum's asynchronous capabilities, contributed to the system's efficiency and responsiveness.

6.1.3 Flutter Mobile Application UI:

The integration of the Flutter framework for the mobile application UI resulted in an intuitive and visually appealing user interface. The Flutter UI facilitated seamless interaction with the attendance system, allowing users to register faces, view attendance reports, and manage system settings efficiently.

6.2 Limitations:

While the smart attendance system achieved notable success, it is essential to acknowledge its limitations:

6.2.1 Performance:

The efficiency of the system's face recognition capabilities can be further improved. Future work should focus on optimizing the classification model and backend infrastructure to enhance real-time performance, especially when handling a larger number of users or in complex environmental conditions.

6.2.2 Scalability and Database Management:

Scaling the system to handle a growing number of users and managing the associated database could pose challenges. Future work should address scalability concerns, considering factors such as system architecture, load balancing, and efficient database management techniques.

6.2.3 User Experience Enhancement:

While the Flutter UI provided a user-friendly interface, additional improvements can be made to enhance the overall user experience. Incorporating user feedback and conducting usability studies can identify areas for refinement, ensuring seamless interaction and navigation within the mobile application.

6.3 Future Work:

To further enhance the smart attendance system and address the limitations mentioned, the following areas of future work can be explored:

6.3.1 Performance Optimization:

Continued research and development in the field of face recognition and classification models can lead to the integration of more advanced algorithms, resulting in enhanced accuracy and faster recognition speeds. Optimizing the Rust backend and Axum framework should also be pursued to maximize system efficiency.

6.3.2 Cloud-Based Infrastructure:

Consideration should be given to migrating the system to a cloud-based infrastructure, which can provide scalability, fault-tolerance, and ease of deployment. Utilizing cloud services can alleviate the burden of managing hardware resources and allow for more flexible system expansion.

6.3.3 Enhanced Security Measures:

As the attendance system deals with sensitive personal information, incorporating robust security measures is of utmost importance. Future work should focus on implementing encryption techniques, secure communication protocols, and user authentication mechanisms to safeguard user data and prevent unauthorized access.

6.3.4 Integration with Additional Features:

Expanding the functionality of the smart attendance system by integrating additional features can increase its value and usability. For example, incorporating geolocation tracking or integrating with existing learning management systems can provide a more comprehensive attendance management solution.

6.3.5 Continuous Testing and Evaluation:

Regular testing, evaluation, and user feedback collection are crucial to ensure the system's reliability and usability. Continuous improvement based on feedback and benchmarking against industry standards will contribute to the system's ongoing refinement and success.

References

[1] Flutter documentation: Last Retrieved on 28/6/2023

https://flutter.dev/docs

[2] Rust documentation: Last Retrieved on 28/6/2023

https://doc.rust-lang.org/

[3] Axum Framework: Last Retrieved on 28/6/2023

https://github.com/tokio-rs/axum

[4] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML), 97, 6105-6114

Last Retrieved on 28/6/2023
https://proceedings.mlr.press/v97/tan19a.html

[5] VGGFace: Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. Proceedings of the British Machine Vision Conference (BMVC), 1-12. Last Retrieved on 28/6/2023
http://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf

[6] RESNET-50: He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
Last Retrieved on 28/6/2023
https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[7] Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49 Last Retrieved on 28/6/2023 http://vis-www.cs.umass.edu/lfw/

[8] Microsoft Research. Microsoft Artificial Intelligence: Microsoft AI's artificial faces. Last Retrieved on 28/6/2023 https://github.com/microsoft/DigiFace1M

[9] PyTorch. (2021). PyTorch: An open-source machine learning framework. Last Retrieved on 28/6/2023 https://pytorch.org/

[10] Docker. Docker - Build, Share, and Run Any App, anywhere. Last Retrieved on 28/6/2023 https://www.docker.com/

[11] PostgreSQL Global Development Group. (2021). PostgreSQL: The world's most advanced open-source relational database. Last Retrieved on 28/6/2023 https://www.postgresql.org/

[12] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10), 1499-1503. doi:10.1109/LSP.2016.2603342

[13] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815-823. Last Retrieved on 28/6/2023 https://arxiv.org/abs/1503.03832

Pretrained Models:

- https://github.com/timesler/facenet-pytorch
- https://github.com/davidsandberg/facenet

[14] Podman: Last Retrieved on 28/6/2023

https://podman.io/

[15] Draw.io: Last Retrieved on 28/6/2023

https://www.diagrams.net/

[16] PlantUML: Last Retrieved on 28/6/2023

https://plantuml.com/

[17] The pgAdmin Development Team: Last Retrieved on 28/6/2023

https://www.pgadmin.org/

[18] Swagger. (n.d.). Swagger UI: Last Retrieved on 28/6/2023
https://swagger.io/tools/swagger-ui/

[19] Online Triplet Learning: Last Retrieved on 28/6/2023

- https://omoindrot.github.io/triplet-loss#batch-hard-strategy
- https://github.com/adambielski/siamese-triplet
- https://github.com/KevinMusgrave/pytorch-metric-

[20] APK: https://github.com/MinaSaad47/ams Last Retrieved on 28/6/2023

[21] Dashboard: https://github.com/MinaSaad47/ams Last Retrieved on 28/6/2023

[22] Computer vision: Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

[23] Machine learning: Alpaydin, E. (2010). Introduction to Machine Learning (2nd ed.). MIT Press.

[24] Face recognition: Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face Recognition: A Literature Survey. ACM Computing Surveys, 35(4), 399-458.

[25] QR codes: Attaran, M. (2017). A QR code-based attendance management system. Journal of Education and Practice, 8(7), 6-12.

[26] Deep learning: LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[27] Singh, K., Singh, A., & Singh, A. K. (2018). Smart attendance system using face recognition and QR code. International Journal of Advanced Research in Computer Science, 9(4), 1-4.

[28] Han, H., Kim, J., & Kim, D. (2019). A smart attendance system using face recognition and deep learning. Electronics, 8(9), 1011.

[29] Tsoi, K. H., Wong, J. K., Wong, K. Y., & Law, N. (2019). Developing a smart attendance system for primary schools. International Journal of Emerging Technologies in Learning (iJET), 14(1), 4-17.