2798

e-STUDENTCARD FOR BIOMETRICS ATTENDANCE SYSTEM

1201101703

THAMIZHARAASAN A/L CHANDRAN

Bachelor of Computer Science (Hons) in Software Engineering

MULTIMEDIA UNIVERSITY

July 2024

2798

e-STUDENTCARD FOR BIOMETRICS ATTENDANCE SYSTEM

BY

1201101703

THAMIZHARAASAN A/L CHANDRAN

PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENT FOR THE DEGREE OF

Bachelor of Computer Science (Hons) in Software Engineering

in the

Faculty of Computing and Informatics

MULTIMEDIA UNIVERSITY

MALAYSIA

July 2024

DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institution of learning.

Name of candidate: THAMIZHARAASAN A/L CHANDRAN

Faculty of Computing & Informatics

Multimedia University

Date: 30: 06: 2024

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Dr. Nbhan D. Salih, whose guidance, support, and expertise have been invaluable throughout this journey. Your unwavering commitment to excellence and encouragement have greatly contributed to the successful completion of my final year project.

I am also immensely thankful to my friends and family for their endless support, understanding, and encouragement during this challenging yet rewarding endeavour. Your belief in me has been a constant source of motivation, and I am truly grateful for your presence in my life.

Lastly, I would like to extend my appreciation to all those who have contributed in any way, no matter how small, to the completion of this project. Your assistance and encouragement have played a significant role in its realisation.

Thank you all for being part of this journey.

ABSTRACT

In an era of advancing technology, the conventional methods of attendance tracking in educational institutions are being revolutionised to enhance efficiency and security. This project presents the design and implementation of an innovative system combining biometric authentication with digital student identification for streamlined campus management. The primary objective is to develop a robust and user-friendly solution that leverages facial recognition technology for attendance monitoring and digital student card issuance.

The proposed system offers several advantages over traditional methods. By employing facial recognition, students' attendance can be accurately recorded without the need for physical contact, thus mitigating concerns related to hygiene and privacy. Moreover, the integration of digital student cards into a mobile application provides students with convenient access to their identification credentials, eliminating the need for physical cards and reducing the risk of loss or theft. In addition to that, the system also offers the functionality to verify students' identities in the exam hall, thereby preventing identity theft during examinations and addressing issues such as students forgetting to bring their exam slips and student cards to the exam hall.

The development process encompasses several key stages, including system design, software implementation, and testing. Facial recognition algorithms are utilised to identify and verify students' identities based on unique facial features. A mobile application is developed to enable students to view and display their digital student cards securely. Furthermore, backend infrastructure is established to facilitate data management and system integration with existing campus databases.

Evaluation of the system involves rigorous testing to assess its accuracy, reliability, and user satisfaction. Preliminary results demonstrate promising performance in terms of attendance recording accuracy and user experience. Additionally, feedback from stakeholders, including students and administrative staff, is collected to identify areas for improvement and further refinement.

In conclusion, the developed biometric attendance system integrated with a digital student card presents a novel approach to campus management, offering

enhanced security, convenience, and efficiency. Future work may involve expanding the system's functionalities, optimising performance, and exploring integration with other campus services to meet evolving educational needs and technological advancements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Overview

Most Malaysian university students are familiar with marking their class attendance by scanning Quick Response (QR) codes with their smartphones. It sounds amazing, as this high-tech solution is not time consuming, and the method is pretty simple. Adapting this QR code technology is possible in Malaysia, as smartphones are widely used among students. However, there is a loophole in using this technology. Some students can still mark their attendance even without being present in their class. They could request the attendance link or the QR code snap from their friends who attended the actual physical class. Therefore, the QR code attendance system tends to facilitate attendance cheating among students. As a result, it is devaluing the efforts of other students who join lectures regularly. Besides that, the students complete their semesters without learning anything, resulting in failing their exams. In short, a system with loopholes is considered a failure regardless of how advanced the system is. Multimedia University (MMU) Cyberjaya is one of the universities in Malaysia that utilises QR code technology for recording students' attendance.

MMU should introduce a biometrics attendance system as an alternative effort to put a full stop to attendance cheating. Biometrics technology, such as facial recognition and fingerprint mapping, are commonly used to identify individuals based on their unique physical characteristics. In the biometrics attendance system, students either need to scan their fingerprint or scan their face in the classroom to take their attendance. This method is much easier and simpler than the QR code attendance system. Consequently, students have no other option than attending the lectures physically to mark their attendance. Additionally, students can also track their attendance through this system.

The purpose of this project is to develop a mobile application which allows students to register their biometrics data that is used to mark their attendance for every lecture and tutorial session. On the other hand, this mobile application could

resemble a student card, as it includes features such as digital representation of student identification and recording student attendance that are similar to the traditional student card. It operates as an electronic student card (e-StudentCard) where the students are not required to present their physical student card to enter the library or attend the exam. Instead, they can scan their face or fingerprint, which serves as the credentials for the student verification process.

1.2 Problem Statement

The idea to introduce a biometrics attendance system arose due to some problems observed on the MMU campus.

Physical student card holds multiple purposes. Physical student cards traditionally serve the purpose of identifying students and confirming their affiliation with a specific institution. Student cards play a pivotal role in accessing a plethora of campus services, facilities, and resources throughout the academic year. In MMU, the purposes of student cards are as follows.

1. Students should present their student cards to the security guards at the entrance before entering the campus.
2. Students need to bring their student cards to gain access to the library.
3. Students are only allowed to borrow library materials if they have a student card.
4. Students must bring their student cards to access other facilities in MMU, such as the swimming pool.
5. Students are required to bring their student cards to attend their exams.

However, **students often forget to bring their student cards with them**, despite the fundamental role it plays in the MMU campus. The worst part is when students frequently forget to bring their examination slip to their exam hall. As a result, students could not sit for their examinations and access the campus facilities without a physical student card.

Apart from that, **students may engage in identity theft to attend exams**, especially if invigilators are not thorough in verifying the veracity of student identity. Indeed, it is a serious criminal offence that the campus community should pay attention to. This poor manual security would lead to diminished educational experience when academic achievements are not a genuine reflection of students' efforts and capabilities. After all, the invigilator is just a human being who is prone to make mistakes.

Besides that, **class attendance cheating among students** is also viewed as a serious problem. Some students request attendance links or QR code pictures from their friends who are enrolled in the classes, and use them to mark their lecture and tutorial session attendance, even without actually attending the classes. In short, students learn nothing throughout the semester, stemming from the loophole in the QR code attendance system. Lecturers should find an alternative to combat this issue efficiently.

1.3 Project Objectives

The goal of this project is to implement and introduce a biometrics attendance system within the MMU community, which will also operate as an electronic student card (e-StudentCard).

The objectives of this project are as follows:

- To identify and model the requirements for a Secure Access Control Module within the e-StudentCard mobile application.
- To design the Examination Integrity Assurance Module for the e-StudentCard system. Develop a robust functionality that prevents impersonation during exams by cross-referencing facial or thumbprint data with the registered student information.
- To design a robust and reliable attendance system which marks attendance using students' biometric data.

1.4 Project Scope

The targeted audience of this project comes from the MMU community, especially MMU students and lecturers. It is a mobile based application system which will provide a user-friendly Graphical User-Interface (GUI) experience for all the users. Students can register their biometrics data through this application and can utilise this application as their e-StudentCard while lecturers can mark the students' attendance via facial recognition using the smartphones or camera devices installed in each classroom. This e-StudentCard concept will provide a helping hand to exam invigilators in preventing identity theft among students during exam periods. Exam invigilators could scan students' biometric data to verify their details in case students forget to bring their physical student card and exam slip.

With this e-StudentCard, students can access campus services, facilities, and resources throughout the academic year. This mobile application prioritises the adoption of advanced technology to develop a robust and reliable system with zero loopholes.

1.5 Deliverables

The goal of this project is to develop a mobile application that integrates a Secure Access Control Module, Examination Integrity Assurance Module, and a robust Attendance System utilising biometric data (facial recognition or fingerprint scanning). This application will feature a user-friendly Graphical User Interface (GUI) for MMU students and lecturers. It aims to enhance campus security by preventing identity theft during exams and ensuring accurate attendance records. The project includes comprehensive documentation, rigorous testing, training materials, and a deployment plan to facilitate seamless integration with MMU's existing infrastructure. Post-implementation evaluation will gather user feedback to refine and optimise the application for maximum effectiveness and usability.

1.6 Project Planning

Table 1.1 Project Gantt Chart for FYP 1

Phase 1 - Trimester 1														
Task/Activities	**Week**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Confirmation of proposal														
Introduction														
Literature Review														
Theoretical Framework														
Research Methodology														
Prototype design														
Conclusion														
FYP1 Interim Report Writing														

The project timeline for Phase 1 - Trimester 1 spans 14 weeks, beginning with the confirmation of proposal in weeks 1-2. This is followed by the introduction phase in weeks 3-4, likely establishing the project's scope and objectives. Week 5 is dedicated to literature review, researching existing relevant work. The theoretical framework is developed in weeks 6-7, laying the conceptual groundwork for the project. Research methodology is addressed in weeks 8-9, planning the project's approach and methods. Prototype design occupies weeks 10-11, involving initial designs or mockups. Week 12 is allocated for the conclusion, possibly summarising progress to date. The trimester concludes with FYP1 Interim Report Writing in weeks 13-14, documenting all previous work and preparing a comprehensive report. This schedule demonstrates a logical progression from project initiation through

research, design, and documentation stages, providing a structured approach to the first phase of the project

Table 1.2 Project Gantt Chart for FYP 2

Phase 2 - Trimester 2															
Task/Activities	**Week**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Implementation															
Testing & debugging															
Report documenting															
Submission															
Poster Presentation															

Phase 2 - Trimester 2 of the project spans 15 weeks, focusing on implementation, testing, and finalisation. The implementation phase is the longest, running from week 1 through week 7, involving the core development work based on the designs from Phase 1. Testing and debugging follows immediately after, occupying weeks 8 through 11, ensuring the functionality and reliability of the implemented features. The final stages begin in week 12 with report documenting, which continues through week 14, allowing time to compile all findings and outcomes. Week 14 also sees the submission of the final report. The trimester and project conclude in week 15 with a poster presentation, providing an opportunity to showcase the completed work. This schedule demonstrates a logical progression from development to refinement and presentation, effectively structuring the final phase of the project.

Chapter 2: Literature Review

2.1 Background Study

Machine learning serves as a key pillar for facial and fingerprint recognition. Deep learning, a subset of machine learning, integrates the Convolutional Neural Network (CNN), a renowned method used to build advanced facial recognition models with higher accuracy. CNN architecture offers faster and more efficient recognition models than other methods. CNN operates on a supervised learning basis, where the model can learn from a training set of labelled dataset. It extracts necessary features from training images, providing unique patterns to distinguish one person from another.

Venugopal et al. (2021) conducted research and implemented a custom facial recognition system using the deepface-ensemble algorithm. The algorithm combines various types of neural networks, such as VGG-FACE, FACENET, OPENFACE, and Facebook Deepface. These neural networks that are paired with distance metrics, such as Euclidean Distance and Cosine Similarity, help to increase the performance of the facial recognition system, thus increasing the overall accuracy of the system as well. Figure 2.1 illustrates the working of the deepface-ensemble model.

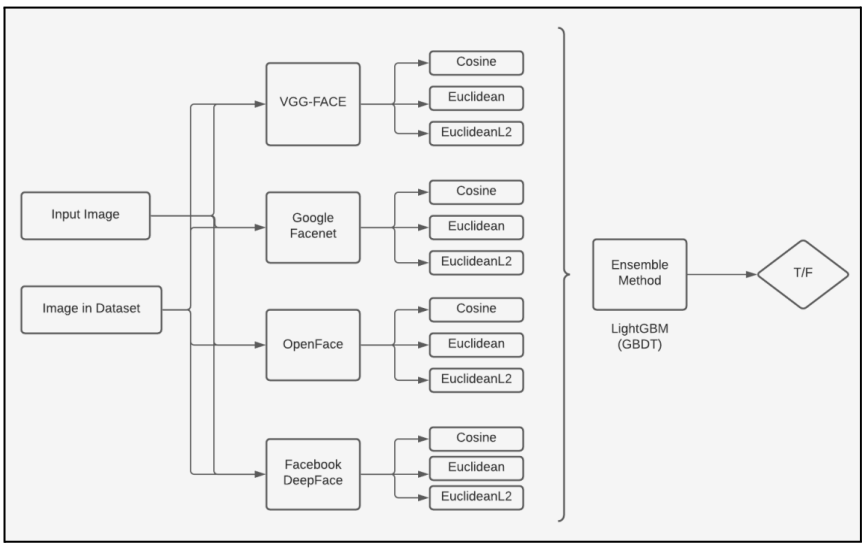


Figure 2.1 DeepFace-Ensemble Model (Venugopal A et al, 2021)

The purpose of distance metrics in this model is to calculate the shortest vector image distance between the input image and the training image dataset. The vector images are obtained using these various neural networks that convert the unique features of the face into vectors. If the value of the input image vector approaches the training image dataset vector, the system can identify a specific face. Since the deepface-ensemble model employs multiple neural networks, the probability of obtaining a matching pair of a specific face feature is higher. Even though it produces promising results, the implementation of the deepface-ensemble model requires more computing time because it processes multiple neural network models simultaneously to achieve higher accuracy.

According to the research carried out by **Krishnan and Manikuttan (2022)**, the algorithm of facial recognition for the attendance system begins with employing Histogram of Oriented Gradient (HOG) technique to detect the face in the picture. It is commonly used as an object detection method in image processing and computer vision. In order to detect the face in a picture, the picture is converted into black and white and pixelated into 16 pixels by 16 pixels. Then, the HOG image is compared with the trained dataset to find the similar part in both the HOG image and dataset to determine the face. After that, 68 face landmark points, which include face shape, eye, nose and lip are obtained in each face. These facial landmark points are used for face encoding, which extracts 128 computer-generated measures from each face.

Using a Python package called Openface makes the work easier to generate these 128 unique measurements for each face as output if we input our images. Last but not least, the SVM classifier is trained to compare the test image's measurements to those in the database. Then, we can use it to find people's names from the encoding. They provide a simple method for facial recognition by stating basic and easy procedures. However, the process of face detection will be hard if we could not obtain the straight profile of a person, thus affecting the accuracy of the system.

The research conducted by **Gómez et al. (2023)** used Haarcascade, which is trained from the huge number of images contained in the OpenCV library for detecting faces. The researchers collected and saved the faces with the corresponding names as labels. Afterward, they transformed the 150 pixels x 150 pixels size of face images into a coded vector of 128 computer-generated unique measurements from each face image and categorised them by the previous name as a label. Finally, they compared the face vector from the real-time video with the database vector to recognise the faces. It clearly illustrated the facial recognition phase in the block diagram below.

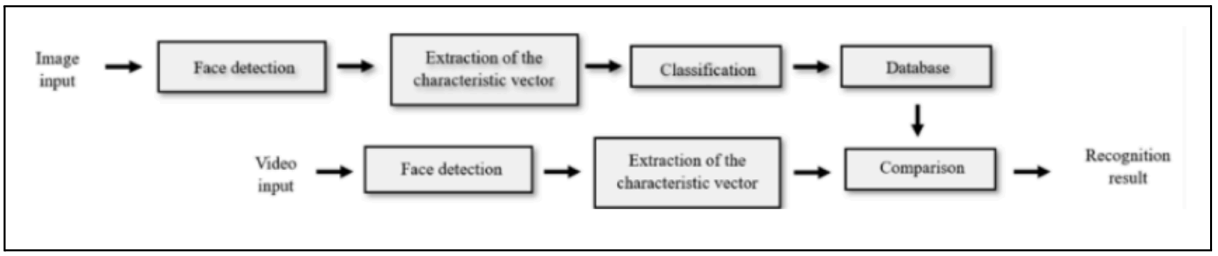


Figure 2.2 Block diagram of facial recognition phases

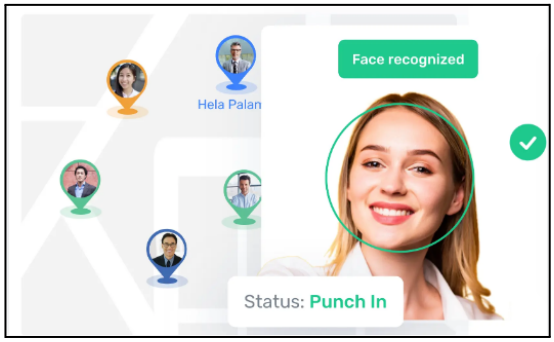
This method is straightforward and easy to implement. The impressive outcome of this method is noteworthy and the straightforward method for facial recognition using the OpenCV library will make our work easier. Unfortunately, there is no mention of a back-up plan if the system fails to recognize the faces.

2.2 Existing Systems

Studies on existing systems playing a crucial role in scrutinising the necessary features required in a biometrics attendance system. Studying existing systems helps to understand the flow of the system and assess how the users could interact with the application's GUI. Moreover, the functionalities of the existing system should be examined thoroughly in order to determine if it meets the requirements or if it missed any functionality that could affect the performance of the system. Also, the pros and cons of the existing system are identified and taken into consideration for developing this project efficiently.

2.2.1 Buddy Punch

Buddy Punch is a mobile application used for tracking employee time and attendance in workplaces. Employees can use their biometric data, such as facial and fingerprint recognition, to clock in and out of the office. The higher officials in the office can serve as administrators of this system to monitor the working hours and extra time contribution of employees in the office. The administrator can view the employees' profile and their position in the company and has the authority to set the employees to either active or inactive. This system can calculate the pay rates based on the employee's time card. The administrator can view time cards and has the capability to submit them for pay rate approval. In addition, the administrator can view the reports, such as hours summary, daily attendance, in/out activity, and employee details and can export them to PDF, CSV or Excel files.



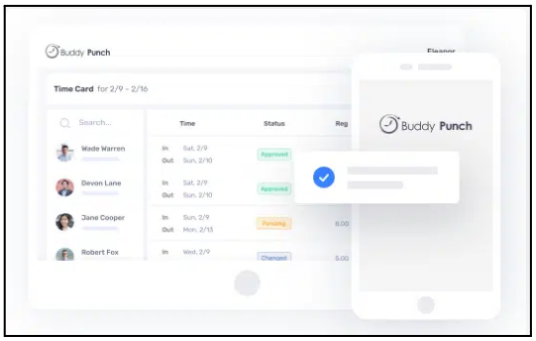


Figure 2.3 Buddy Punch time clock and its updated data

The strength of this system is that it prevents ‘buddy punching’ in the workplace by adding the facial recognition feature so employees can scan their faces to check in or out of work. Also, the system implements a back-up plan in case the system fails to recognise faces. The employees have the option to check in or out of work by using their pin number. Apart from time tracking, the system serves as a multipurpose application in the workplace, such as streamlining the payroll process and function as employee scheduling software.

However, this application prioritises solving workplace scenario problems. It is not suitable for a university environment. To assess the features such as mobile apps, GPS on punches, job tracking, payroll integrations and reporting, the users have to subscribe to the standard monthly subscription.

2.2.2 Timeero

This mobile application shares the same functionality as ‘Buddy Punch’ mobile application. The main purpose of this application is for employees to punch in and punch out of work using facial recognition. Employees have to create their profile in this application with a unique PIN for punch in and punch out. The need to upload employees’ pictures is optional, as the Timeero kiosk application will snap a photo of the employee as a baseline image for facial recognition on the first time they clock in. At the punching zone, the employee chooses their name from a list, enters their PIN, and photographs themselves. The technology compares the picture with the baseline image using an artificial intelligence (AI) algorithm. The software emails the manager or administrator if the person clocking in is not the person they claim to be. It accomplishes so, nevertheless, while allowing the worker to punch in or out.

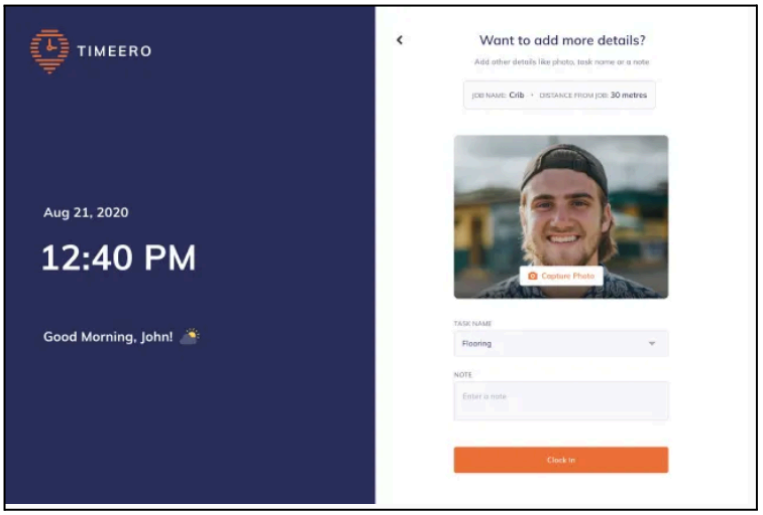


Figure 2.4 Timeero time clock

The strength of this application is that it supports offline mode. Employees still can punch in or out using their smartphone if they lose internet connection. Once the internet connection stabilises, the application will synchronise the time entries to the database.

However, the system only supports the iOS environment and the user should subscribe to basic subscription to enjoy all the benefits from the application.

2.2.3 Jibble

A free time tracking software which supports facial recognition attendance taking in the workplace. This application is an alternative to avoid buddy punching and queues at the attendance kiosk. The system will save the attendance to the cloud, allowing employees to track their attendance seamlessly.

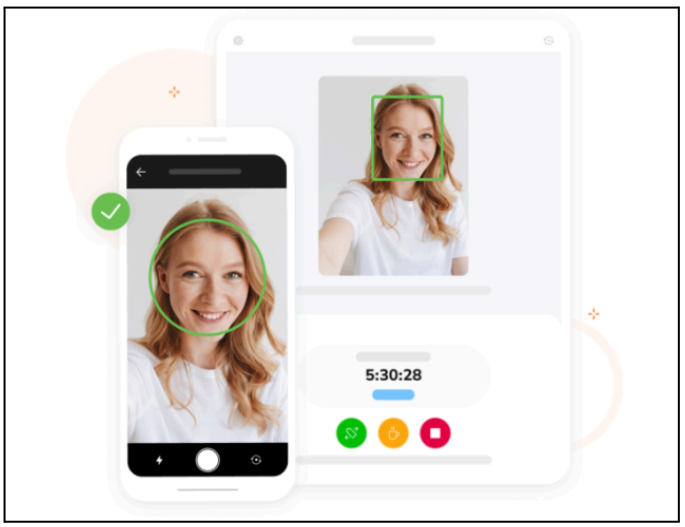


Figure 2.5 Jibble application

The employees can register their biometric data by logging into the Jibble app using a shared device provided by the company and scan their face. Apart from using this kiosk mode, employees can also register their biometric data using their personal devices.

This system supports operation as both a mobile application and a desktop application. It integrates with other leading software products, such as ClickUp and Microsoft Teams, to enhance the functionality of the system and holds a multipurpose benefit.

However, an external device; attendance kiosk is needed to clock in and out of the workplace. Besides, the subscription to utilise the features of this application is too costly for a single user.

2.2.4 Quickbooks

Quickbooks uses biometric facial recognition technology for time clock purposes, so the company can precisely see who is clocking in and out. According to a survey conducted in the U.S among 4906 customers, it is reported that this application could save 3.4 hours in taking the employees' attendance in the workplace. This application helps the organisation to summarise the payroll information more accurately and efficiently every month.

This application could support running on both iOS and Android environments. It takes approximately 20 seconds to clock in and out at the workplace. In addition, this application is used for other purposes such as payroll summarisation and employee collaborative platform. It is known as an excellent inventory management and time tracking application among reviewers.

However, the expensive subscription is one of the cons of this application. There are four versions of QuickBooks Online; they all include a 30-day free trial and are aesthetically and functionally comparable. Simple Start is intended for small enterprises and has a monthly cost of $30. Time monitoring, bill management, and support for three users are added to the $60/month Essentials package. Besides that, it is not suitable for university use because it is giving priority to solve problems in the working environment.

2.2.5 MMU Mobile Application

In MMU, students can mark their class attendance by scanning the QR codes through the MMU Mobile application.

This mobile application is available for free in both the Play Store and Apple Store. Students can access the functionalities of this application using their MMU

student ID and password credentials, making it easy for them to use the application. This application provides the primary benefit of allowing students to check if their attendance is taken after they scan the QR code in the class. In addition, it offers some additional features, such as :

- Display class timetable
- Display course fees
- Display Student Dashboard
- Go to the MMLS page
- Show the programmes in MMU

Figure 2.6 MMU Mobile Application Home Page

However, the disadvantage of this application is that the students still can take their attendance without using this mobile application. The students can also use a QR code scanner mobile application to scan the QR code and retain the attendance link. Then, they can mark their attendance using the link in the browser.

2.3 Proposed Solution

An e-StudentCard with a multipurpose function is to be developed through a mobile application, which will rely on biometric data to verify the students' identities on the MMU campus. MMU students can establish their profiles through this application by using their unique student ID and register their face via the selfie camera in their respective smartphones for facial recognition. Students are permitted to enrol their fingerprints at the MMU Student Service Centre to serve as a backup plan in case facial recognition fails to identify the student. Fingerprint enrollment in the system is an optional feature in this application. The worst-case scenario is the student can use their student ID number to verify their identity. Students could track their attendance and view the accurate attendance percentages for the classes they enrolled.

Lecturers could record attendance in the class by scanning students' faces, preventing the possibility of attendance cheating. Apart from that, the exam invigilators could mitigate impersonation during exams by cross-referencing facial or fingerprint data with the registered student information. Furthermore, lecturers that can play the exam invigilators role could examine students' exam slips to verify whether the students are permitted to sit for that specific exam.

Apart from that, campus staff could scan the students' faces to verify if they belong to the campus community. It could allow students to access campus facilities, such as the swimming pool, library, and meeting rooms.

Overall, this mobile application resembles a traditional student card, encompassing all of its features and functionalities where facial recognition plays an important role in identity verification, access control, examination integrity assurance, and ensuring the students to access campus facilities and services.

2.4 Application Comparison

Table 2.1 Application comparison

Comparison Between Existing Systems and Proposed System						
	Buddy Punch	**Timeero**	**Jibble**	**Quickbooks**	**MMU Mobile Apps**	**Proposed System**
Facial recognition feature for attendance	✔	✔	✔	✔	✖	✔
Facial recognition feature for verification	✖	✖	✖	✖	✖	✔
Free	✖	✖	✖	✖	✔	✔
Mobile Application	✔	✔	✔	✔	✔	✔
Student-Focused	✖	✖	✖	✖	✔	✔
Additional functionalities	✖	✖	✖	✖	✔	✔

2.5 Technological Background

2.5.1 Mobile Application

To develop a mobile application, Flutter has witnessed rapid growth in recent years. Flutter is an open source UI software development kit developed by Google. Flutter can be used to build natively compiled applications for web, mobile, and desktop from a single codebase. It is popular among developers as the code is written only once and it supports deployment across multiple platforms, such as Android, iOS, and web browsers. This approach could reduce the development time and effort,

since the developers can maintain a single codebase for multiple platforms. The fast development process, single codebase, and expressive UI are the main key features of Flutter that improve the productivity, flexibility, and performance of an application, thus leading to its popularity among developers.

Flutter uses the Dart programming language, which was developed by Google as well, for building applications. Dart is an object-oriented programming language that is easy to learn, with features that facilitate the development of scalable and maintainable applications.

In this project, Flutter platform will be used to develop the mobile application, which supports the facial recognition features and other functionalities of the system as it offers a consistent and high-quality application across different platforms and devices.

2.5.2 Database

Since the project will be developed using Flutter, integrating Firebase services will be the optimal choice for working on features such as real-time database, authentication, cloud messaging, and more seamlessly into the application. By integrating Firebase with Flutter, developers can seamlessly add features like user authentication, cloud storage, real-time database, and push notifications to the applications. Firebase is a comprehensive platform for web and mobile application development that was acquired by Google in 2014. It offers various services, such as :

1. Authentication
2. Real-time database
3. Cloud storage

In this project, Flutter integrated with Firebase services will be used to authenticate users before redirecting them to their respective dashboard and to store and fetch facial feature data and other necessary information.

2.5.3 Face Detection

For face detection, the project will utilise Google ML Kit, a machine learning SDK provided by Google for mobile developers. ML Kit includes a face detection API that operates using deep learning models, providing robust detection capabilities directly on mobile devices without requiring an internet connection. This API is advantageous for its speed and accuracy in detecting faces in images or video frames.

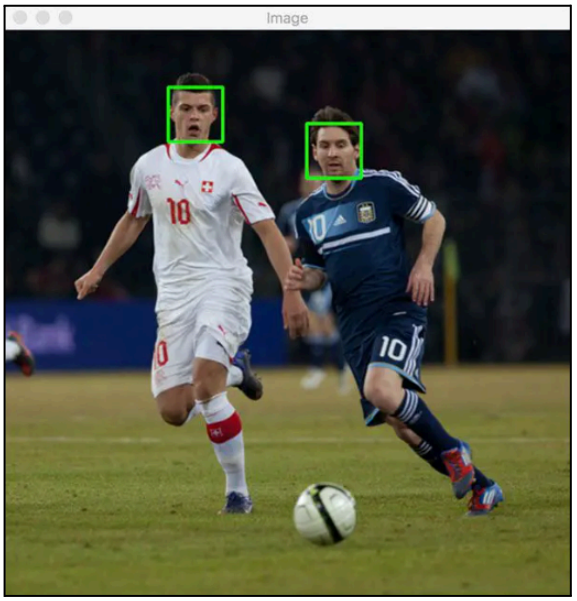


Figure 2.7 Google ML Kit Face Detection

It identifies facial regions and provides coordinates for each detected face, enabling subsequent processing steps such as cropping, resizing, or converting images to grayscale. ML Kit simplifies integration with its ready-to-use APIs, making it suitable for real-time applications where fast and reliable face detection is essential.

2.5.4 Face Recognition

Face recognition is a supervised learning approach to machine learning. To recognize a face, we must extract facial vector feature data from the face, label it, and train the model.

As for the prototype, a built-in face recognizer library from OpenCV Python is used to detect, train, and recognize the faces. During the implementation, the integration of the FaceNet algorithm will be employed to enhance the accuracy and efficiency of face recognition. FaceNet is a deep learning model developed by Google that uses a convolutional neural network (CNN) to directly optimise the embedding of faces into a Euclidean space. This embedding space is designed such that distances directly correspond to face similarity, allowing for precise face recognition across varying conditions. FaceNet's robustness and high-dimensional feature representation make it ideal for accurately identifying faces in images or video frames, surpassing traditional methods like eigenfaces or Histogram of Oriented Gradients (HOG).

Chapter 3: Requirements Analysis

Based on the literature review, the key functionalities of the e-StudentCard for the biometrics attendance system have been identified to ensure practicality and user-friendliness. In addition to these core functionalities, the software will integrate additional features aimed at enhancing convenience, thereby improving overall usability for students and faculty alike.

3.1 Main functionality

1. All users could **log into** the system to perform their respective functions.
2. Lecturers **mark students' class attendance** using biometric data.
3. Students **check their attendance** based on the class they have attended.
4. Students can **view e-Student Cards** through mobile applications.
5. Lecturers **assign examinations to** students.
6. Lecturers or exam invigilators **cross check students'** identity through biometric data.
7. Campus staff **verify students' identity** before students can access the facilities.
8. Administrators can **create and delete accounts** of users.

3.2 Use Case Overview

Table 3.1 Use Case Overview

Actor	Use case
Student	- Register biometric data - Check class attendance - View e-student card - View exam slip
Lecturer	- Log student attendance - Assign students for the examination - Verify student identities in exam
Campus staff	- Verify student identities on campus
Administrator	- Create accounts for users - Delete user account

3.3 Use Case Diagram

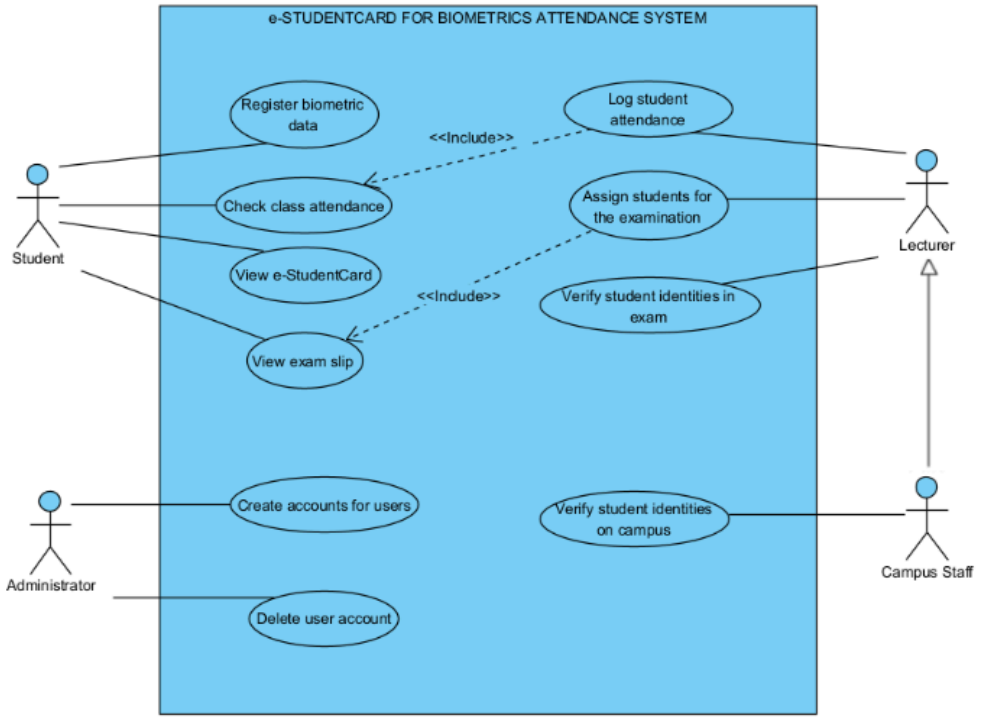


Figure 3.1 Use case diagram

3.4 Functional Requirements

This section provides the use case tables for the common functionalities in the system. The actors involved in this use case are student, lecturer, campus staff and administrator. In the use case table, it will consist of use case ID, use case name, overview, pre-condition, steps, post-condition, and exception flow. The use case table should depict the use of the system, exception flow, and preconditions for each use case.

3.4.1 Student

3.4.1.1 Register biometric data

Table 3.2 Register biometric data (UC001)

Use case ID	UC001
Use case name	Register biometric data
Overview	The students can register and save their biometric data in the system.
Pre-condition	- The student is new to the system and has not registered their biometric data yet.
Steps	1. Students select the 'Register Biometric Data' option. 2. The front camera on the mobile will open. 3. Students have to position their face within a circle on their screen for a few seconds. 4. The system will store the students' biometric data in a database with a label.

Post-condition	Students can go back to the home page after registering the biometric data.
Exception flow	The system might fail to detect the face due to inadequate lighting at the students' side.

3.4.1.2 Check class attendance

Table 3.3 Check class attendance (UC002)

Use case ID	UC002
Use case name	Check class attendance
Overview	The students can double check their attendance in the system after the lecturer marked it.
Pre-condition	- Student attended the class physically - Lecturer has scanned the student's biometric data for marking attendance.
Steps	1. Students select the 'View Attendance' option. 2. The system will display the attendance. 3. Students can view the class code, subject name and attendance marking time.
Post-condition	Students can inform the lecturer if their attendance is not updated in the system.
Exception flow	The attendance record does not reflect in the system.

3.4.1.3 View e-student card

Table 3.4 View e-student card (UC003)

Use case ID	UC003
Use case name	View e-student card
Overview	It resembles a traditional student card, effectively replacing its functionalities.
Pre-condition	- The student is belongs to MMU community - The current program status is Active in Program in MMU
Steps	1. Students select the 'View Student Card' option. 2. The system will display the student card with the information as below : ● Name ● Student ID ● Program Status ● Card validity
Post-condition	Students can go back to the home page after viewing the student's card.
Exception flow	The system could not display the student card if the student has not registered their biometric data and personal information at the Students Affairs Division (STAD)

3.4.1.4 View exam slip

Table 3.5 View exam slip (UC004)

Use case ID	UC004
Use case name	View exam slip
Overview	Students can view the subjects for which they are eligible to sit for their examinations.
Pre-condition	- Students must enroll in the subjects they want to sit for in the examination - Lecturer assigned the student to attend the examination for the specific subject.
Steps	1. Students select the 'View Exam Slip' option. 2. The system will display the exam slip with the information as below : ● Name ● Student ID ● Subject taken ● Exam details ● Exam instructions
Post-condition	The students can save the exam slip as a PDF.
Exception flow	There will be no exam slip if the student does not have any final exam during the semester.

3.4.2 Lecturer

3.4.2.1 Log student attendance

Table 3.6 Log student attendance (UC005)

Use case ID	UC005
Use case name	Log student attendance
Overview	The lecturers can record the students' attendance physically using biometric data during the class period.
Pre-condition	- The student is registered for the subject. - Lecturers must have a smartphone or other camera devices to scan students' faces for attendance marking. - The student must attend the class session physically.
Steps	1. Lecturers select the 'Record Student Attendance' option. 2. Then, they have to select the class session. 3. Click the 'take attendance' option. 4. After that, they can scan students' faces to record the attendance.
Post-condition	- Lecturers can view the students' list of those who attended class physically.
Exception flow	The system might fail to recognise some students' faces. In this case, lecturers are allowed to take manual attendance.

3.4.2.2 Assign students for the examination

Table 3.7 Assign students for the examination (UC006)

Use case ID	UC006
Use case name	Assign students for the examination
Overview	The lecturers can permit the students to sit for the examination
Pre-condition	- The student is registered for the subject.
Steps	1. Lecturers select the ‘Examination’ option. 2. Then, they have to select the class. 3. From the selected class, the lecturers can select the students who can sit for the examination. 4. Finally, they will post the examination update.
Post-condition	Students can view the exam slip
Exception flow	System error may occur while posting the examination update.

3.4.2.3 Verify student identities in exam

Table 3.8 Verify student identities in exam (UC007)

Use case ID	UC007
Use case name	Verify student identities in exam
Overview	The lecturers could prevent impersonation during exams by cross-referencing biometric data with the registered student information.
Pre-condition	- The student is registered for the subject. - The lecturer assigned the student to sit for the examination. - The student is in the exam hall
Steps	1. Lecturers select the 'Student Verification' option. 2. Lecturers are able to scan the students' faces using their smartphones to cross-check students' identity
Post-condition	Lecturers could perform the student identity verification process easily.
Exception flow	System error may occur while scanning students' faces.

3.4.3 Campus Staff

3.4.3.1 Verify student identities on campus

Table 3.9 Verify student identities on campus (UC008)

Use case ID	UC008
Use case name	Verify student identities on campus
Overview	The security guards, librarians, facility managers, and student services personnel play the role of campus staff. They will use this system to verify students' identity so students can access the campus facilities and resources.
Pre-condition	- The student is belongs to MMU community - The current program status of the student is Active in Program in MMU
Steps	1. Campus staff will select the ‘Student Verification’ option. 2. Campus staff can scan the students’ faces using their smartphones to verify students’ identity. 3. Then, they can view the student's status.
Post-condition	The campus staff will permit the students to access the campus facilities and resources depends on the student's current status in MMU
Exception flow	System error may occur while scanning students’ faces.

3.4.4 Administrator

3.4.4.1 Create accounts for users

Table 3.10 Create accounts for users (UC009)

Use case ID	UC009
Use case name	Create accounts for users
Overview	The administrator has the authority to create accounts for students, lecturers, and campus staff.
Pre-condition	- The user does not exist in the system
Steps	1. The administrator selects the ‘Create Account’ option. 2. The administrator fills in the below details : ● Name ● User ID ● Username ● Password ● User type
Post-condition	The users can log into the system using the username and password given by the administrator
Exception flow	Administrator might miss some required information to create an account.

3.4.4.2 Delete user account

Table 3.11 Delete user account (UC010)

Use case ID	UC010
Use case name	Delete user account
Overview	To delete the users' account
Pre-condition	The administrator will delete the user account if they are no longer part of the MMU community.
Steps	1. The administrator selects the 'Delete User Account' option. 2. Then, they can select the specific user to be deleted. 3. The system will delete the selected user account.
Post-condition	Once the administrator deletes the user account, they can no longer access the system.
Exception flow	System error may occur while deleting a particular user account.

3.5 Non-Functional Requirements

Here is a list of non-functional requirements the system should have :

1. Only registered users shall be allowed to log into the system.
2. The system shall properly handle errors or failures that occurred in the system.
3. The system shall maintain an active login session for users until they click the sign out button.
4. The system shall ensure that each screen loads within 5 seconds.
5. The system shall support running on Android mobile devices.

3.6 External Interface Requirements

3.6.1 Software Requirements

This section will list and briefly explain the software development technologies or services required by the e-StudentCard for Biometrics Attendance System mobile application . These resources may include frameworks, data stores, and other necessary components.

- **Flutter**

  Flutter is selected for developing the proposed system, encompassing both the mobile applications for customers and technicians, as well as the web application for the admin portal. Provided by Google as an open-source framework, Flutter enables the creation of cross-platform applications for Android, iOS, desktop, and the web using a single codebase (Flutter (Software), 2023). It enhances developer productivity with improved tools and a robust asset system, ensuring stability throughout development.

Apps in Flutter are written in the Dart language, optimised for frontend development. One of its standout features is hot reload, allowing developers to quickly see changes reflected in the app after code modifications (Thomas, 2019). Flutter also boasts a rich set of customizable widgets that facilitate the creation of visually appealing user interfaces (Thomas, 2019).

- **Firebase**

Firebase is a backend application development platform that supports developers in building, enhancing, and scaling applications. It offers various cloud solutions tailored for integration with Flutter apps. The following solutions are planned for implementation:

1. **Cloud Firestore**: This will serve as the database for the proposed system. Cloud Firestore is a NoSQL database provided by Firebase, offering a flexible and scalable solution for storing, syncing, and querying data. It supports both client-side and server-side development, providing robust capabilities for managing structured data efficiently (Singh, 2019).
2. **Firebase Authentication**: This will handle user authentication in the proposed system, incorporating both phone authentication and email/password authentication methods. Firebase Authentication provides a secure and straightforward way to verify user identities, ensuring reliability and ease of implementation. It offers an end-to-end identity solution that enhances the security and usability of user authentication processes within the application.

- **MySQL**

  In this project, MySQL is employed as a local database to store facial characteristic vectors. This configuration ensures that facial recognition data remains accessible even when offline. By leveraging MySQL, the system can efficiently manage and retrieve necessary biometric information to verify and authenticate users without relying on a constant internet connection. This approach enhances the reliability and performance of the e-Student Card for Biometrics Attendance System, ensuring the application remains functional in environments with limited or no internet connectivity.

3.6.2 Hardware Requirements

To run the Flutter mobile app, you need to set up either an emulator or a real mobile device. The e-StudentCard for Biometrics Attendance System is designed specifically for Android devices. The app must conform to the following system requirements to run:

Table 3.12 Application System Requirements

Mobile Application
Android devices with : ● Version 11 or above ● At least 2GB of RAM ● Minimum 600MB of free space ● Supports front and back camera

3.7 Context Diagram

The following diagram shows a context diagram of this system to illustrate the interactions between the actors and the system.

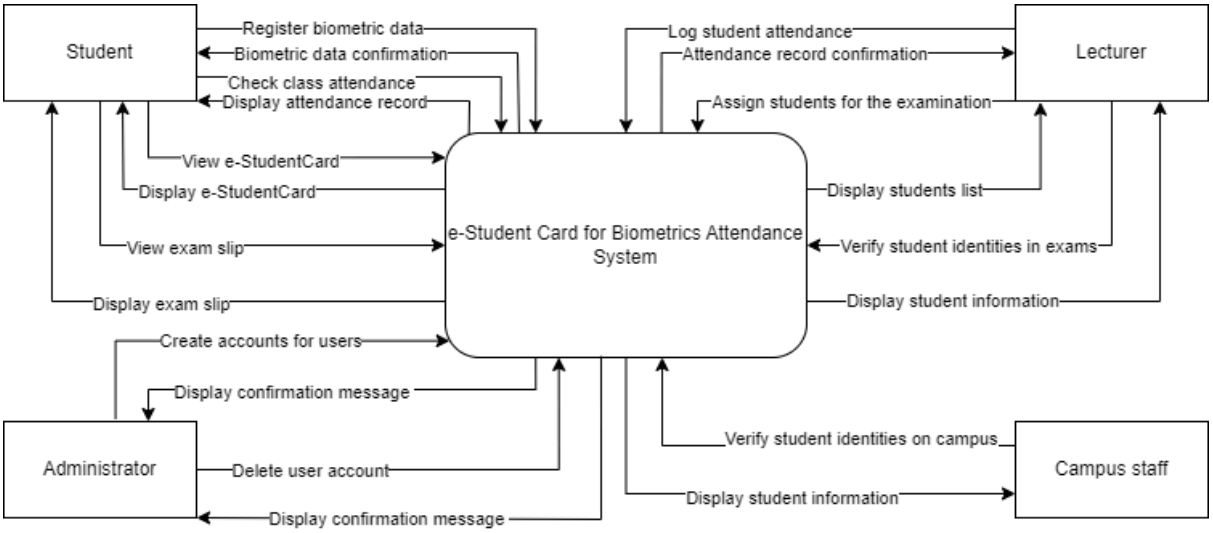


Figure 3.2 Context Diagram

3.8 Data Flow Diagram (Level-0)

The Data Flow Diagram Level-0 as displayed in Figure 3.3 shows the whole system overview and illustrates the vivid information of the system such as the data flows, external entities, data stores, and the processes. The external entities are the student, campus staff, lecturer, and administrator. The diagram also shows how the processes add, fetch, and update the information in the databases. Also, the inputs and outputs of the processes are clearly displayed in the diagram as the entities and the data stores are linked to each other through the processes.

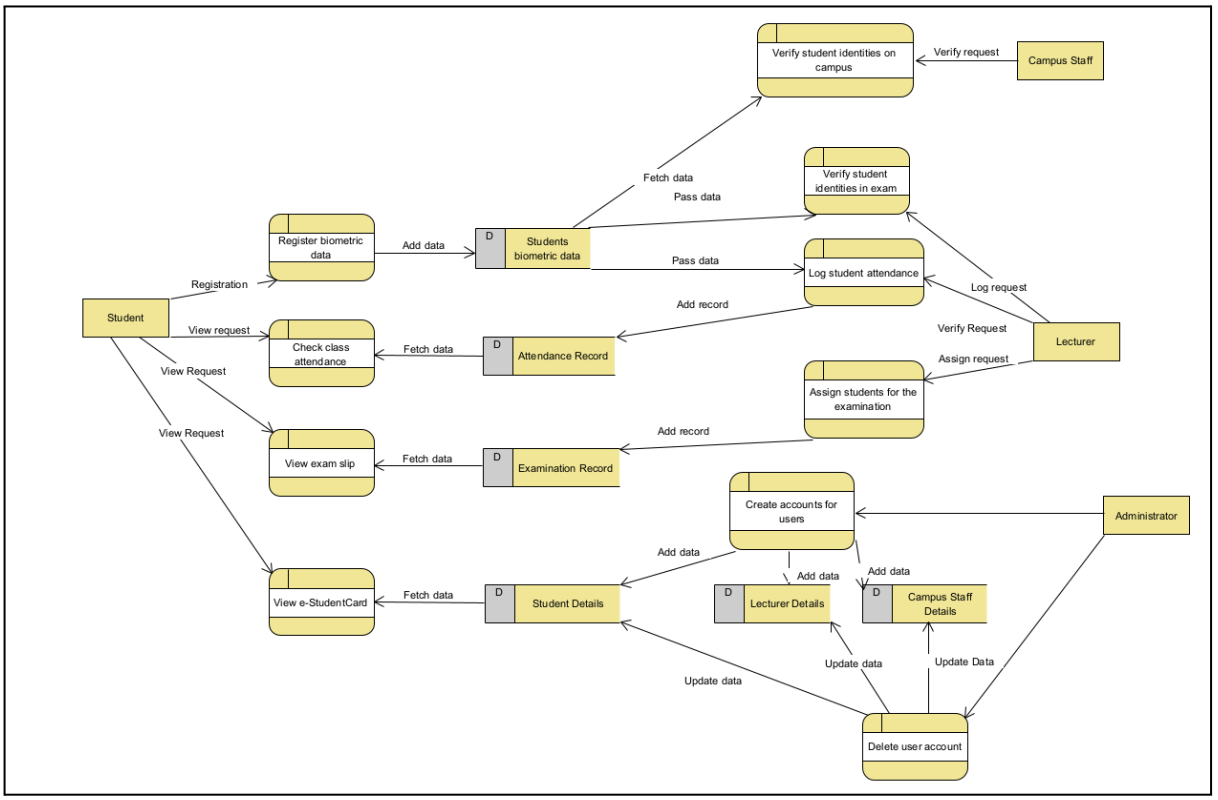


Figure 3.3 Data Flow Diagram

3.9 Entity-Relationship Diagram

The relevant entities, relationships, and cardinalities are illustrated explicitly in the Entity Relationship Diagram as shown in the Figure 3.4 below. The relevant primary and foreign keys are indicated in the diagram for each entity.

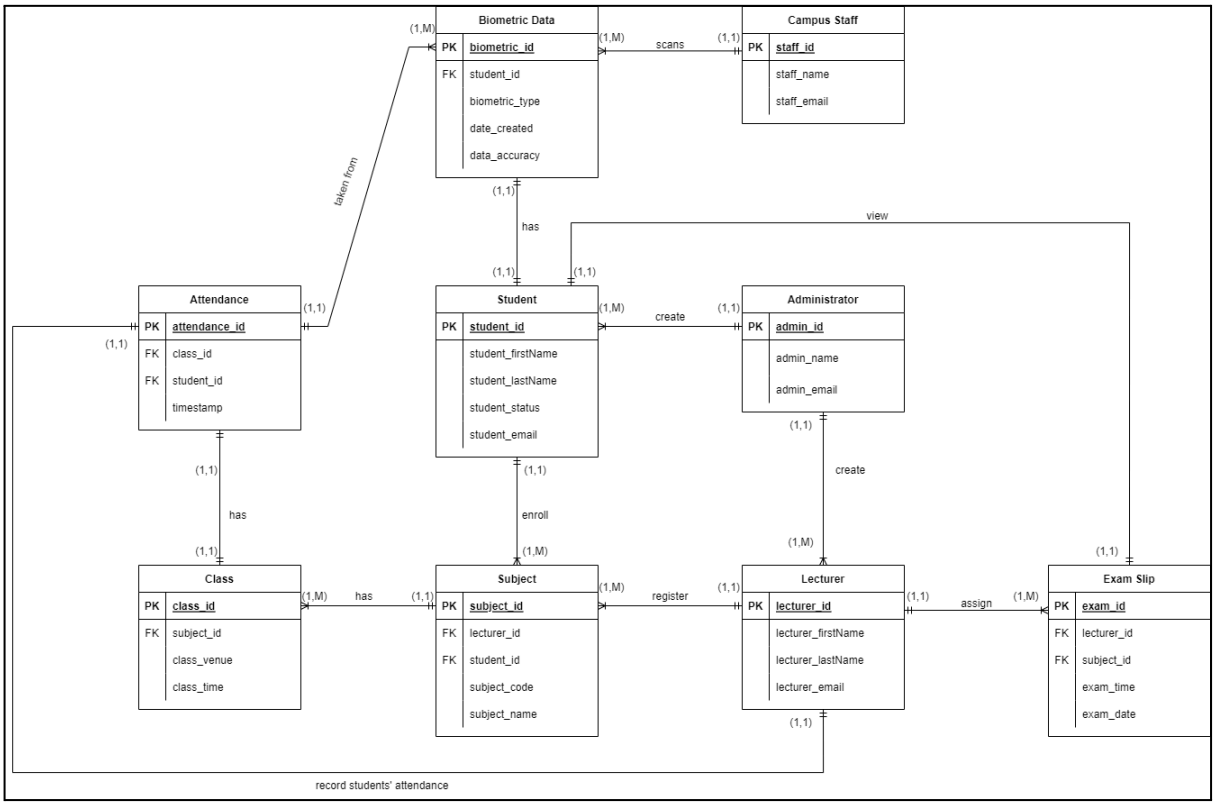


Figure 3.4 Entity-Relationship Diagram

Chapter 4: Design

4.1 System Architecture

Figure 4.1 below illustrates the system architecture of the system. The main users of the are the student, lecturer, campus staff, and administrator. All of them can access the system through the Internet. The cloud server hosts the application and database. Data is transferred between the application and the database via the cloud server.

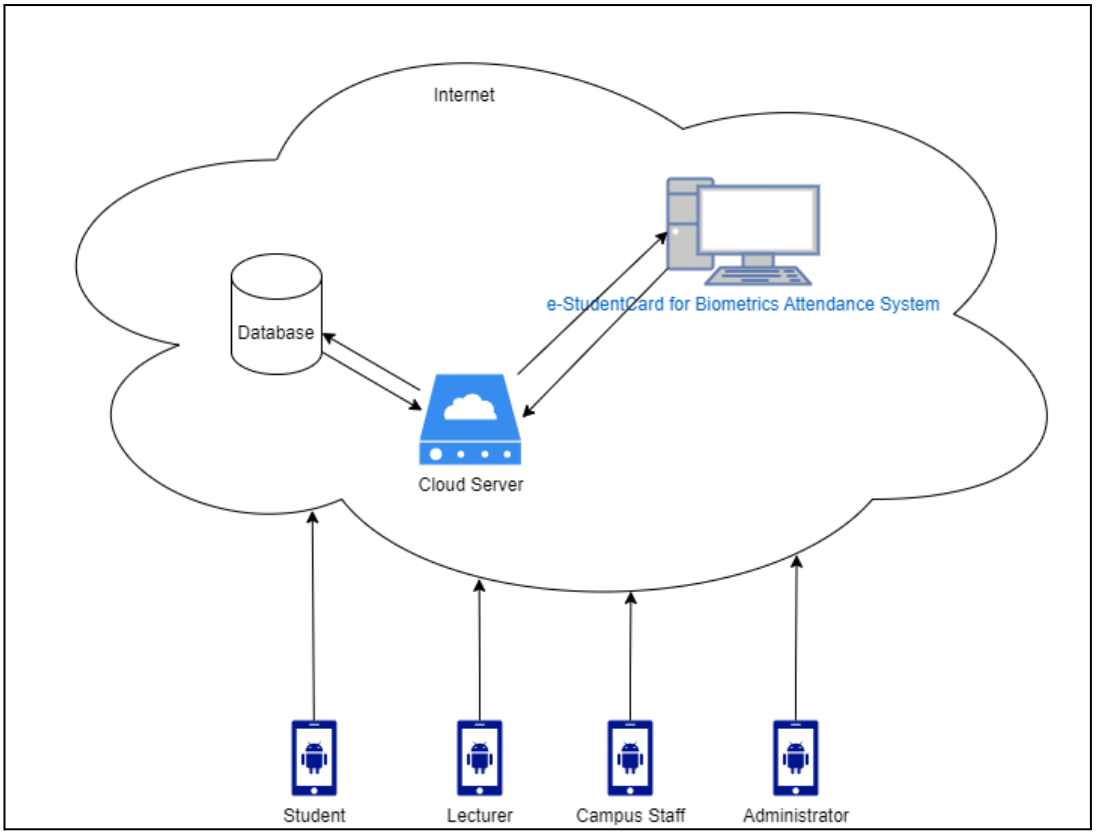


Figure 4.1 Software System Architecture

4.2 Sequence Diagram

This section will focus on the internal design of the system. A vivid understanding of how the program communicates between objects will be explained through this chapter. Sequence diagrams for each functionality in the application will be displayed and described in order to get a clear design understanding of the software. The sequence diagram also will demonstrate all the possible paths a user can perform within the system application.

4.2.1 Student

4.2.1.1 Register biometric data

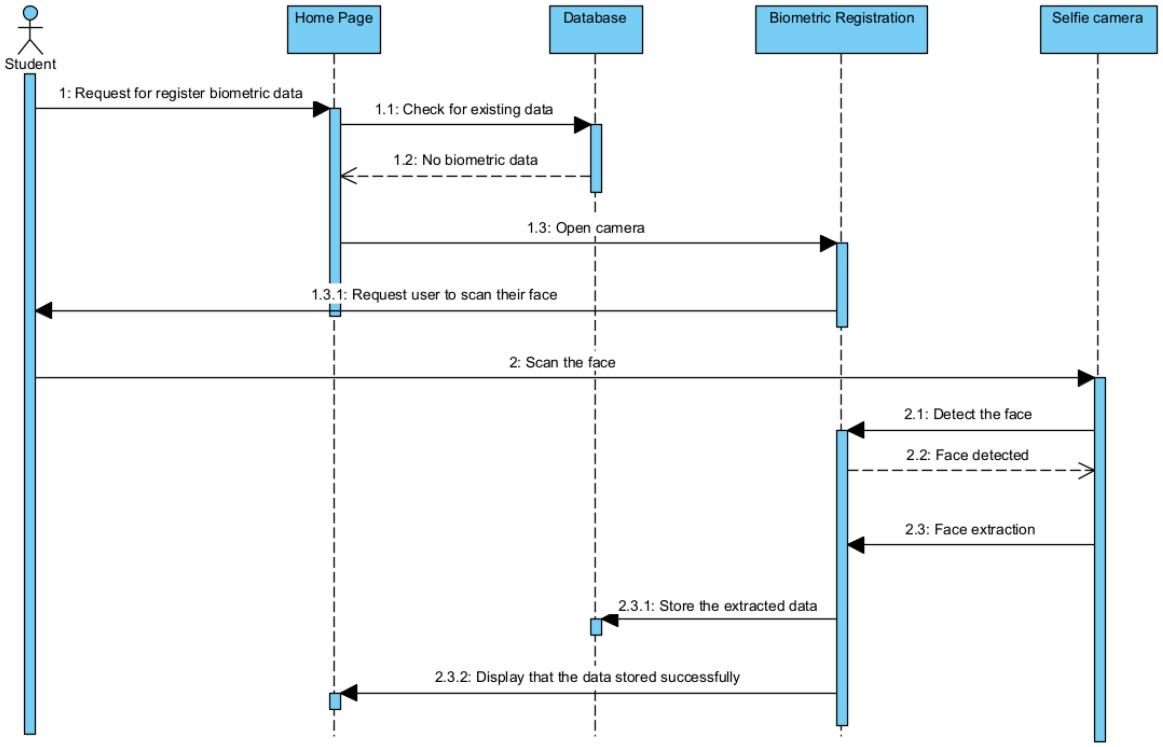


Figure 4.2 Register Biometric Data

The above figure shows how students can register their biometric data through the system. The student can request to register biometric data from the home page. By cross-referencing with existing records in the database, the system verifies

if the students' biometric data has been registered. If the student is a new user and has not registered their biometric data yet, they will be directed to the biometric registration page where the selfie camera on their smartphone will activate automatically, prompting the student to position their face within the designated screen area. In this process, upon detecting the student’s face, the system performs the face extraction procedure, capturing and storing the extracted image of the student’s face in the database. At the end of this function, the system will display that the biometric data was saved successfully.

4.2.1.2 Check class attendance

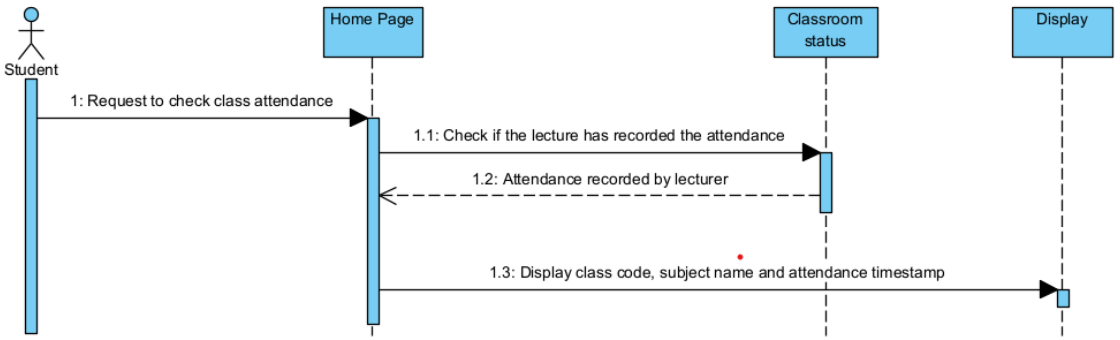


Figure 4.3 Check class attendance

The above figure illustrates how students are able to check their class attendance from their home page. The students have to fulfil the condition that the lecturer has taken the attendance during the class session before viewing the attendance records. The details available for viewing include :

- Class code
- Subject name
- Lecturer name
- Class venue
- Attendance timestamp

4.2.1.3 View e-student card

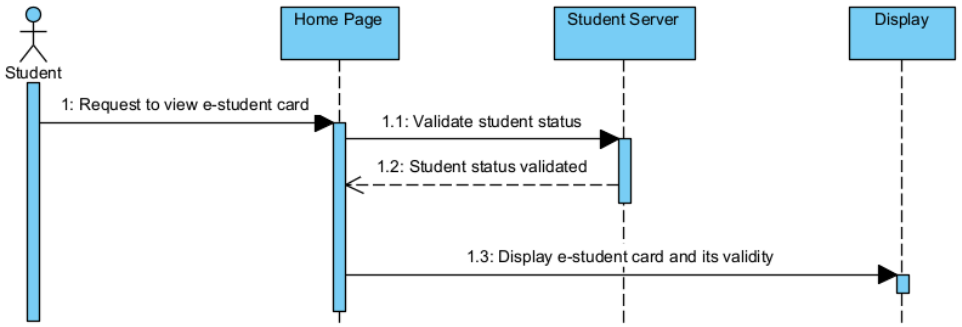


Figure 4.4 View e-student card

The figure above shows the sequence of how a student can view their e-student card. From the home page, students can request to view their e-student card. The system verifies the students' status in MMU before displaying the student card. Once the system verifies that the current student status is active in the program, the student could be able to view their e-student card.

4.2.1.4 View exam slip

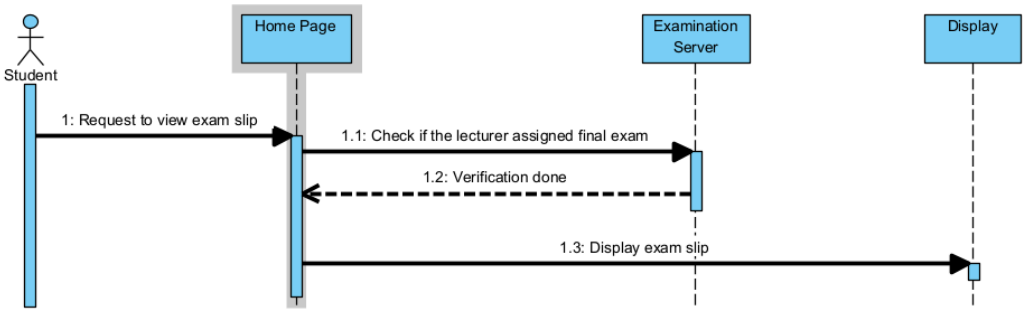


Figure 4.5 View exam slip

Based on the figure above, it shows how students can view their examination slip. The precondition of this function is that the lecturer must assign the final examination for the subject the student is taking in the current semester. Hence, the

system verifies the students' eligibility to take the exam by cross-referencing with the examination server. After the verification process is done, the student can view their exam slip in PDF format.

4.2.2 Lecturer

4.2.2.1 Log student attendance

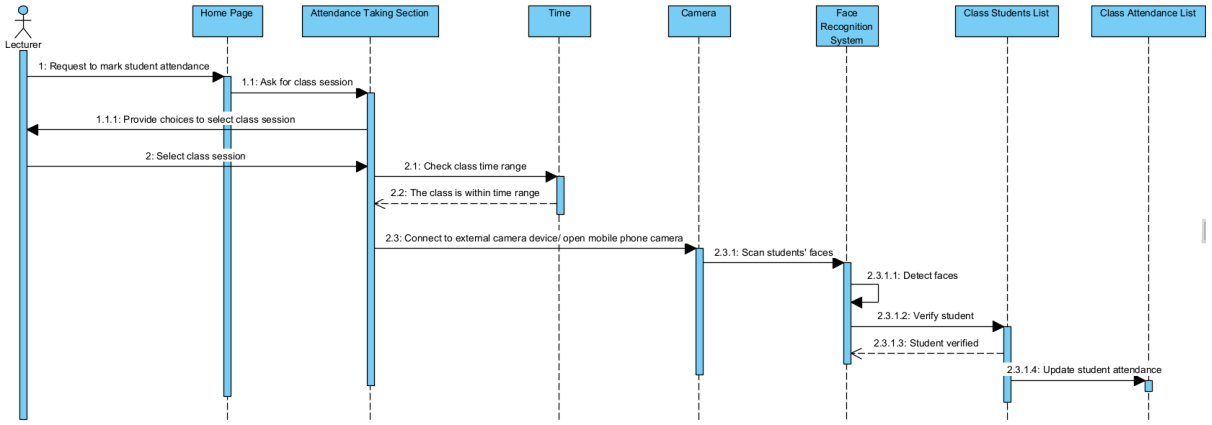


Figure 4.6 Log student attendance

The figure above shows how a lecturer can mark the students' attendance using a facial recognition system in the physical class. When the lecturer requests to mark the attendance from the home page, the system will prompt the lecturer to select the class session. If the selected class session is within the class time range, the system will activate an external camera device or smartphone camera to scan the students' faces. The face recognition system will identify the students' faces and retrieve the corresponding facial data from the class students list. As a final step, the class attendance will be updated and available to view by the students.

4.2.2.2 Assign students for the examination

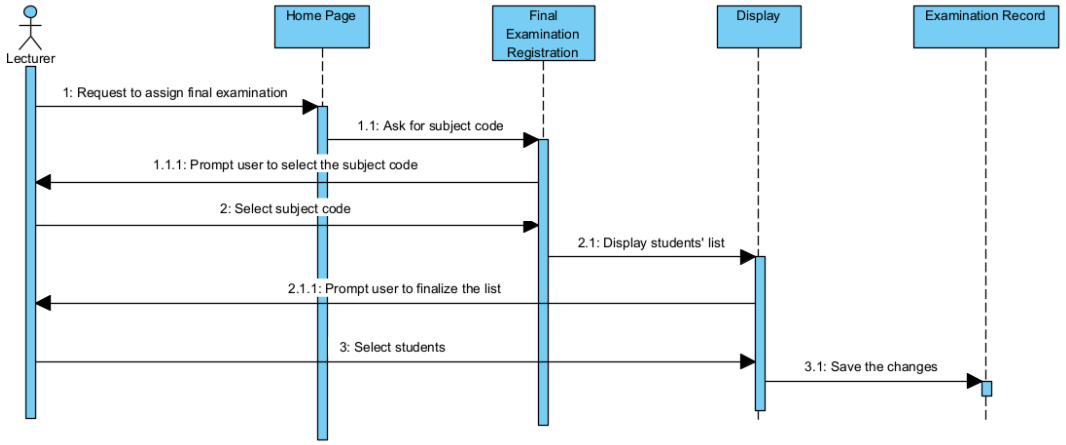


Figure 4.7 Assign students for the examination

The figure above shows a sequence diagram for the lecturer assigning students for the examination. Firstly, the lecturer has to request a final examination from their home page. Then, the lecturer will be directed to the final examination registration page. On this page, they must enter the required credentials to proceed to the next step. Next, the lecturer can view the list of students that are taking that particular course. The lecturer has the authority to toggle the checkbox, thereby granting or revoking permission for students to sit for the examination. After the submission of the form, the data will be saved in the examination record thereby allowing the students to view their exam slip.

4.2.2.3 Verify student identities in exam hall

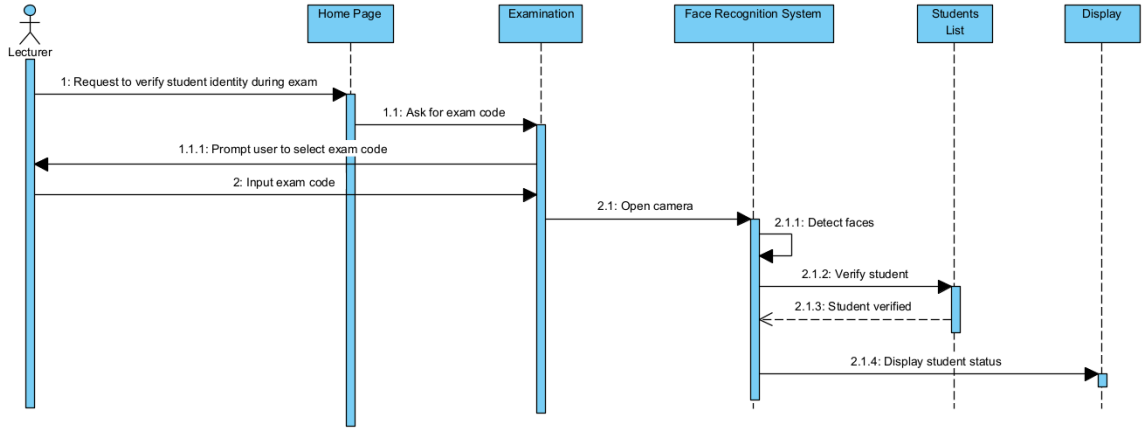


Figure 4.8 Verify student identities in exam hall

The figure above shows how a lecturer can cross-referencing student identities in the exam hall using the face recognition system. During the examination period, the lecturer requests to verify student identity from their home page. Then, the lecturer is required to input the valid exam code to open their smartphone camera automatically and scan the students' faces. Afterward, the face recognition system will identify the students' faces and retrieve the corresponding facial data from students lists stored in the examination records.

4.2.3 Campus Staff

4.2.3.1 Verify student identities on campus

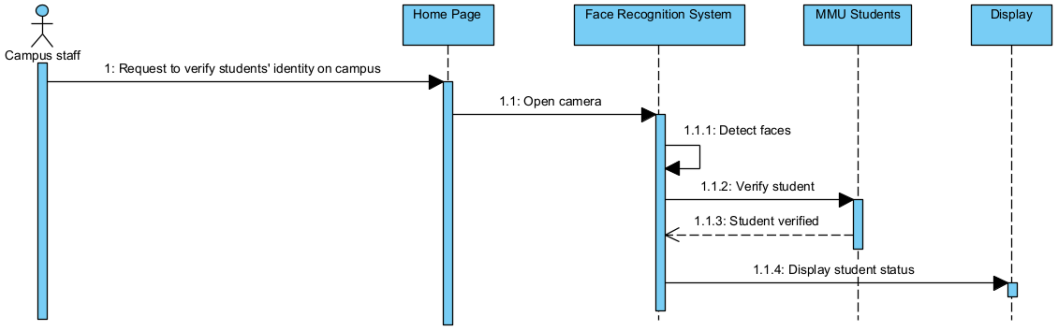


Figure 4.9 Verify student identities on campus

The figure above shows how a campus staff is able to verify students' identity on campus from their home page. Whenever a campus staff grants students access to the campus services, facilities, and resources, they must first verify the students' identities to ensure they belong to the MMU community. Firstly, when campus staff request to verify students' identity on campus, the smartphone camera will be activated automatically to scan students' faces. Afterward, the face recognition system will identify the students' faces and retrieve the corresponding facial data from students' records.

4.2.4 Administrator

4.2.4.1 Create accounts for users

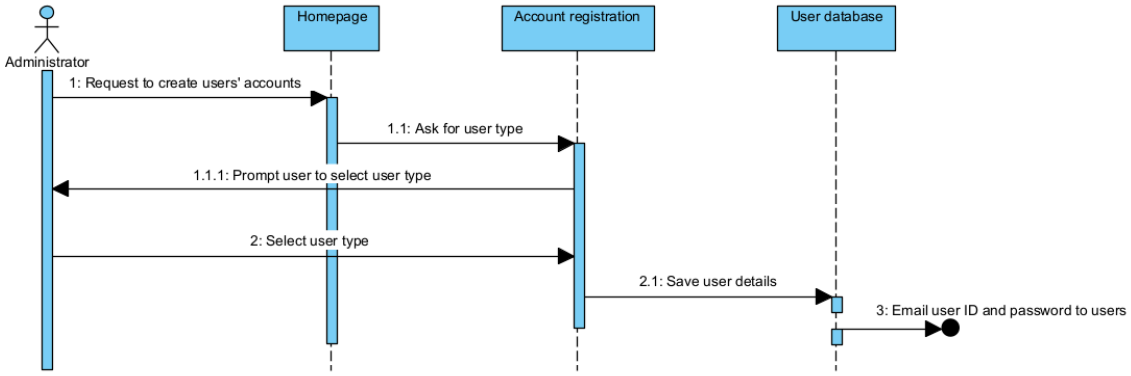


Figure 4.10 Create accounts for users

The figure above shows how an administrator can create an account for users. There are 3 user types which are student, lecturer and campus staff. Firstly, the administrator has to request to create a user account. Then, they will be directed to the account registration page. On this page, they must enter the required credentials such as :

- Username
- User ID
- Password
- User type

Subsequently, the account will be successfully created and saved in the respective users' database, based on the user type. Additionally, the system will send an email to the new users of their user ID and password.

4.2.4.2 Delete user account

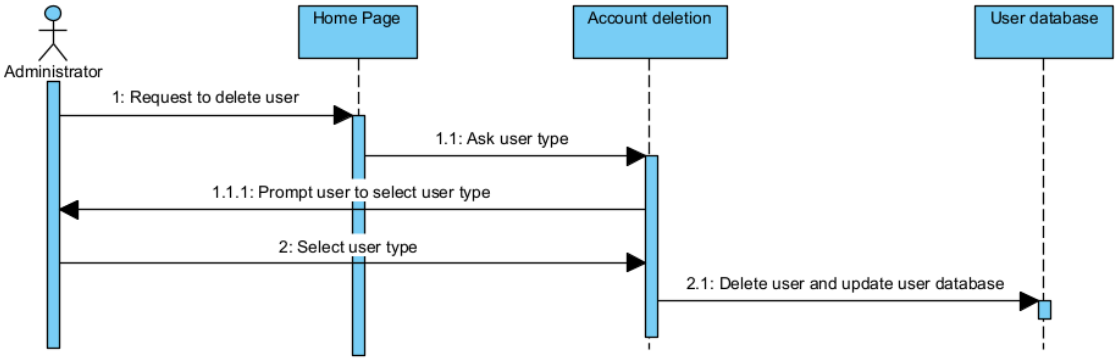


Figure 4.11 Delete user account

The figure above shows how an administrator can delete an user account from the database. Basically, when an administrator requests to delete a user, the system will ask the user type. The administrator has to select the user type and select the specific user from that particular user type to be deleted. As a result, the selected user will be deleted from the database and no longer have access to the system.

4.3 Wireframes

In this section, it shows how the user can interact with the system by displaying the screen designs. This section will show and explain an overall look of the system's UI. These wireframes show how the users can perform their respective tasks using the system effectively. The wireframe shows the necessary functions offered by the system to the users.

4.3.1 Login Page

Figure 4.12 Login Page screen

The figure above shows the login page of the system. All users share the same login page and are required to enter their credentials like User ID and password before being redirected to their respective dashboard. The users cannot proceed to other pages until they provide the correct User ID and password.

4.3.2 Student Screens

4.3.2.1 Student Dashboard

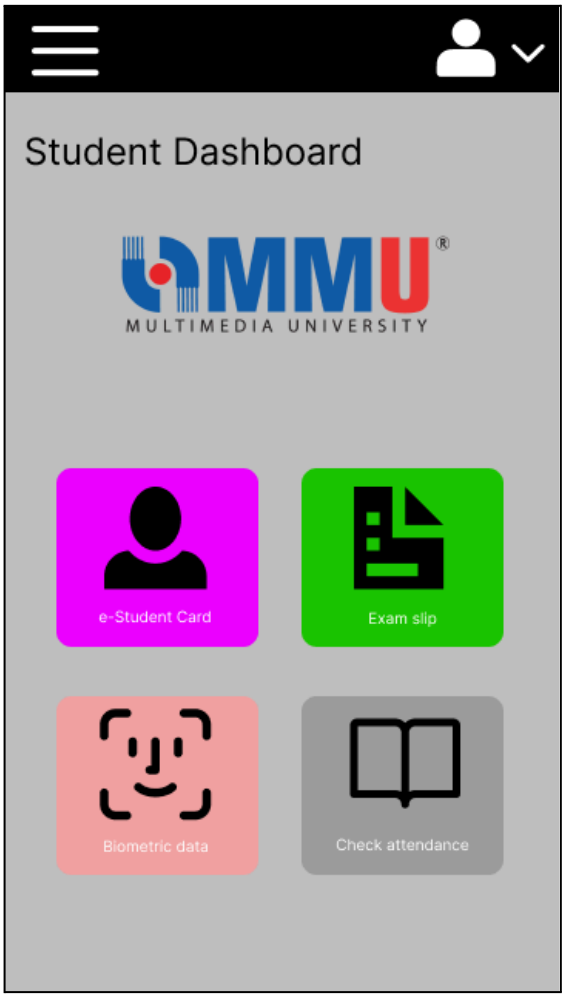


Figure 4.13 Student Dashboard screen

Once the student has successfully logged in, the student can view their dashboard and perform other actions such as :

- Register biometric data
- Check class attendance
- View e-student card

- View exam slip

Other than that, they can also log out from their account and change their password from the dashboard page.

4.3.2.2 Register biometric data

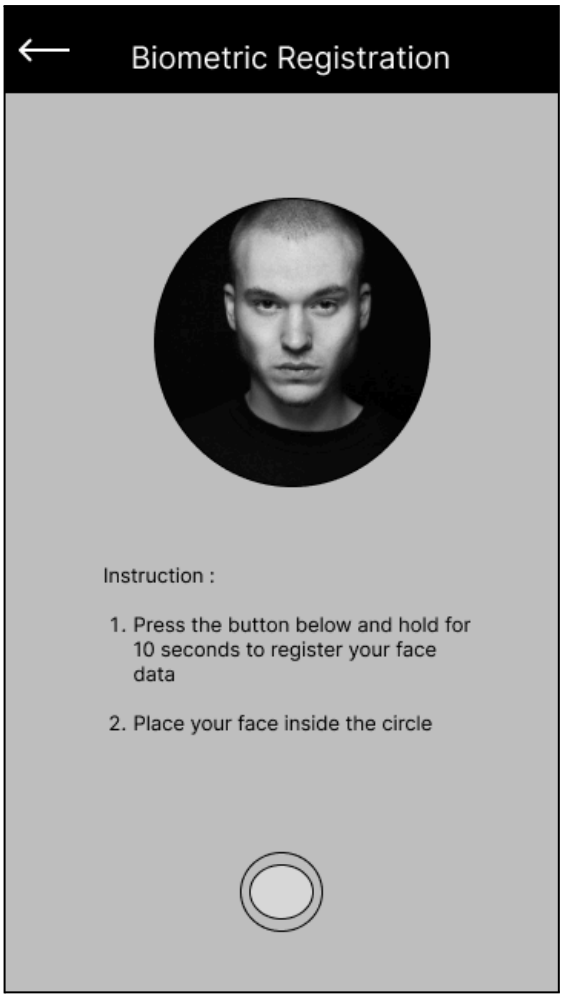


Figure 4.14 Register biometric data screen

From the student dashboard, students can click the ‘biometric data’ button to register their biometric data. In this page, students are required to read the instructions clearly before performing the operation. To register the biometric data, students have to position their face inside the circle frame. Next, they have to press the circle button below and hold it for 10 seconds, enabling the system to capture

their facial features and store it into the database. After completing the operation, they will be redirected to the main page with a message informing the students that the data was saved successfully.

4.3.2.3 Check class attendance

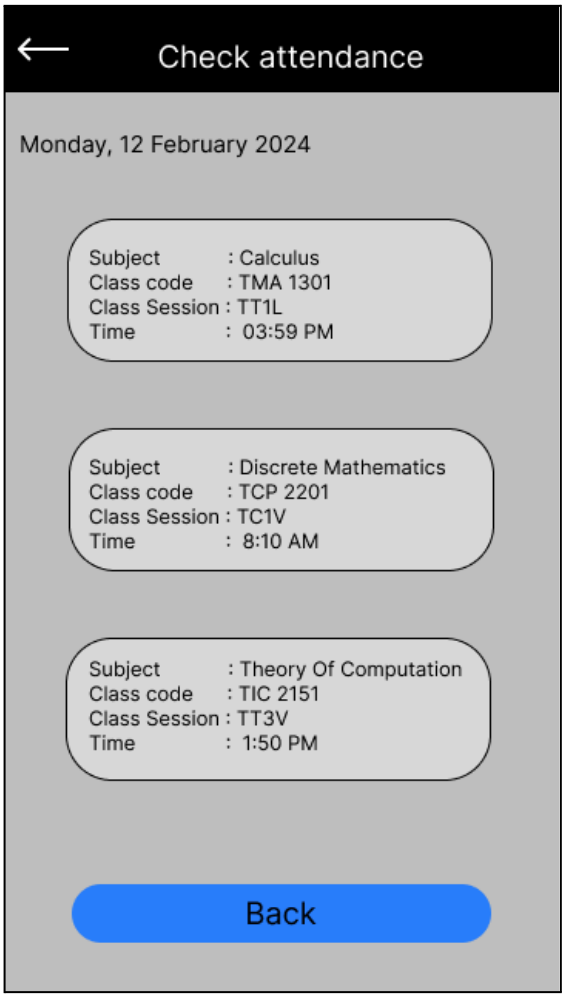


Figure 4.15 Check class attendance screen

To ensure that the class attendance taken by the lecturer through facial recognition is updated in the system, there is an option for students to verify their attendance. Students can view and check their attendance directly from the dashboard. This page displays the date, subject, class code, class session, and timestamp of the attendance taken. If the system does not update the attendance, students can request their lecturer to take the attendance again.

4.3.2.4 View e-student card

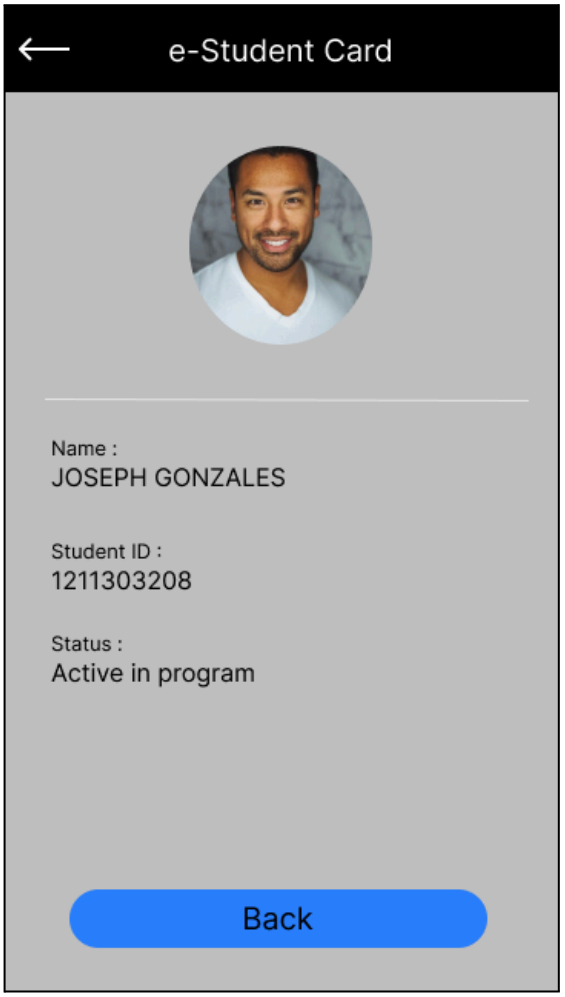


Figure 4.16 View e-student card screen

Student cards play a pivotal role in a student's life, not only on campus but also off-campus.

These are the purposes of the student card both on and off-campus :

Table 4.1 Student card purposes

On campus	**Off-campus**
Identification	Identification for student discounts
Attendance tracking	Retail discounts
Security access	Public transportation discounts
Resource access	Access to events and services
Examination verification	Banking and financial services

Despite the pivotal role played by the student card, students often forget to bring their physical cards with them. To address this issue, the application includes a feature called the e-student card, allowing students to view and display their student card anywhere and anytime, both on and off-campus. From the student dashboard, students can select the option to view the e-student card.

4.3.2.5 View exam slip

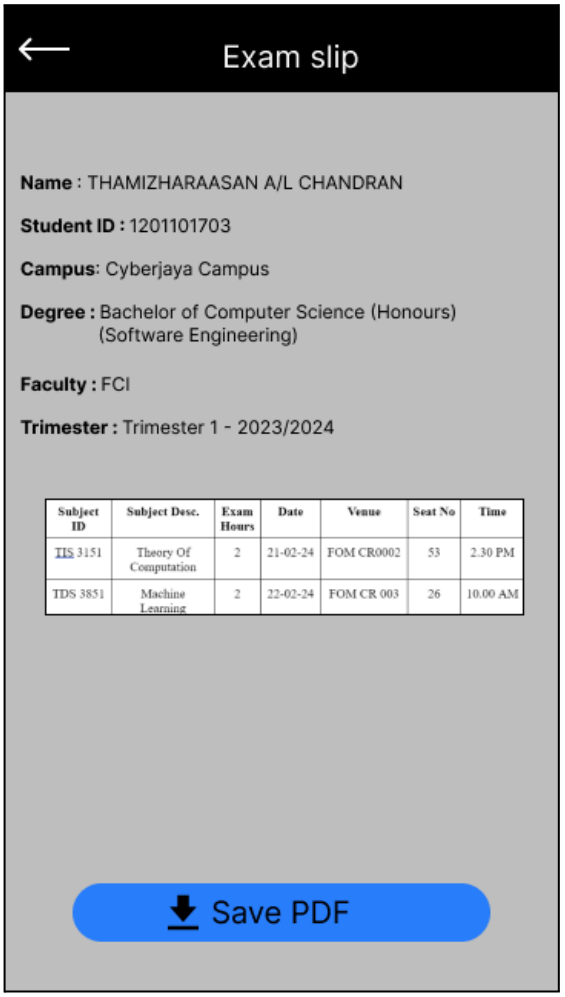


Figure 4.17 View exam slip screen

The figure above shows the screen of a student viewing their exam slip through the application from their student dashboard. Similar to the student card, the exam slip is also commonly forgotten by students when heading to the exam hall. Therefore, students can also view their exam slip through this application. Moreover, they have an option to save the exam slip as PDF.

4.3.3 Lecturer Screens

4.3.3.1 Lecturer Dashboard

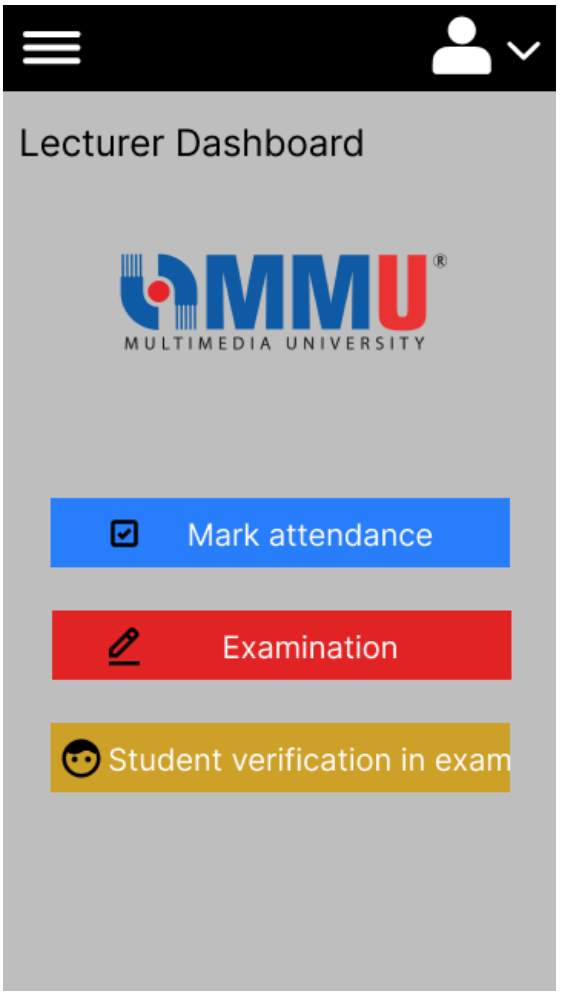


Figure 4.18 Lecturer Dashboard screen

The figure above shows the screen design of the lecturer dashboard where lecturers can perform their main activities after successfully logged into the system. Apart from logging out and changing the password, lecturers can also :

- Mark student attendance
- Assign examination
- Verify student identification in the exam hall

4.3.3.2 Log student attendance

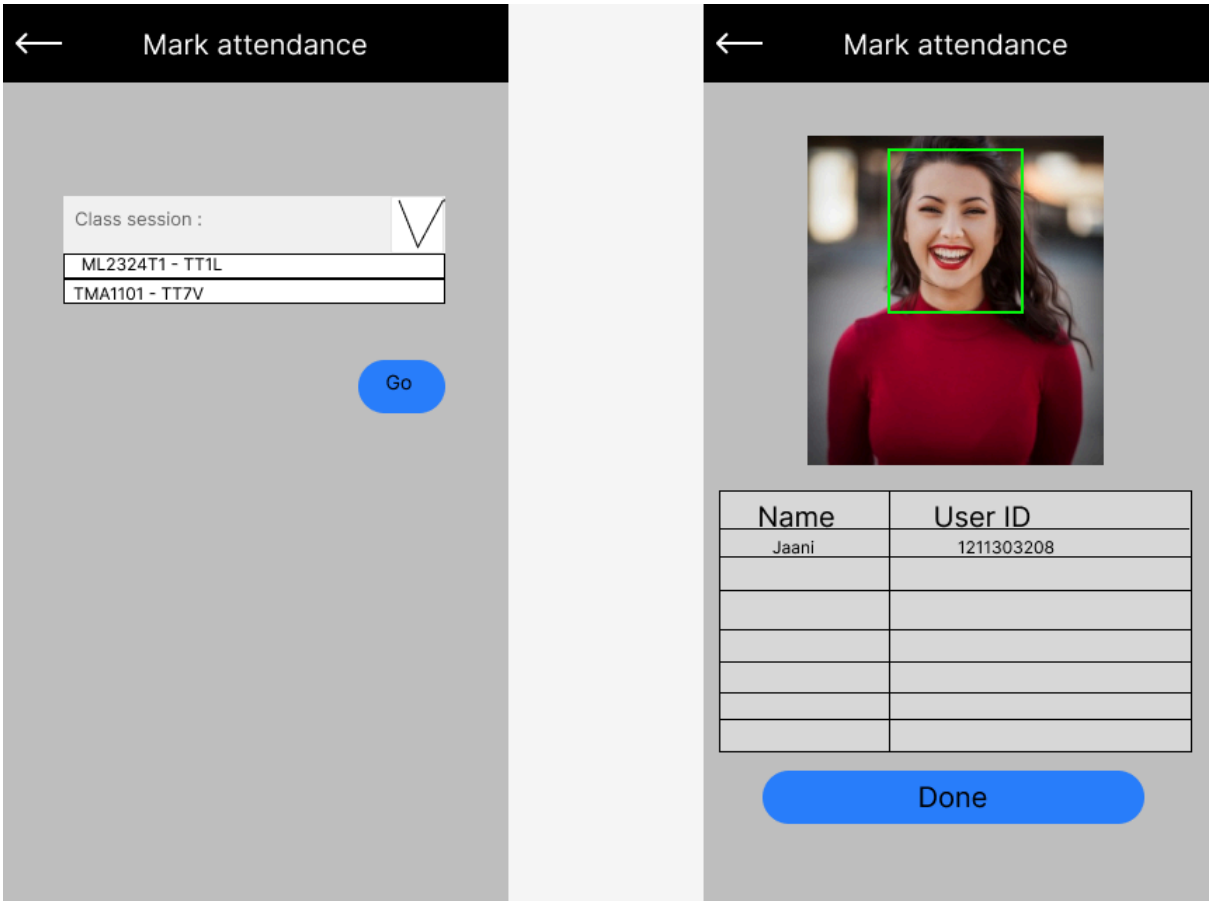


Figure 4.19 Log student attendance screen

The figure above shows the screen design of a lecturer marking student attendance during the class period. When a lecturer selects the 'Mark attendance' option from the dashboard, they will be redirected to a page where they can select the class session. Afterward, the system will bring the lecturer to another page, allowing lecturers to scan students' faces for attendance marking. If a student has registered for a specific class session, the attendance list will be updated when the lecturer scans the student's face. Apart from that, if the system fails to recognise the student's face, the lecturer can mark the attendance manually.

4.3.3.3 Assign students for the examination

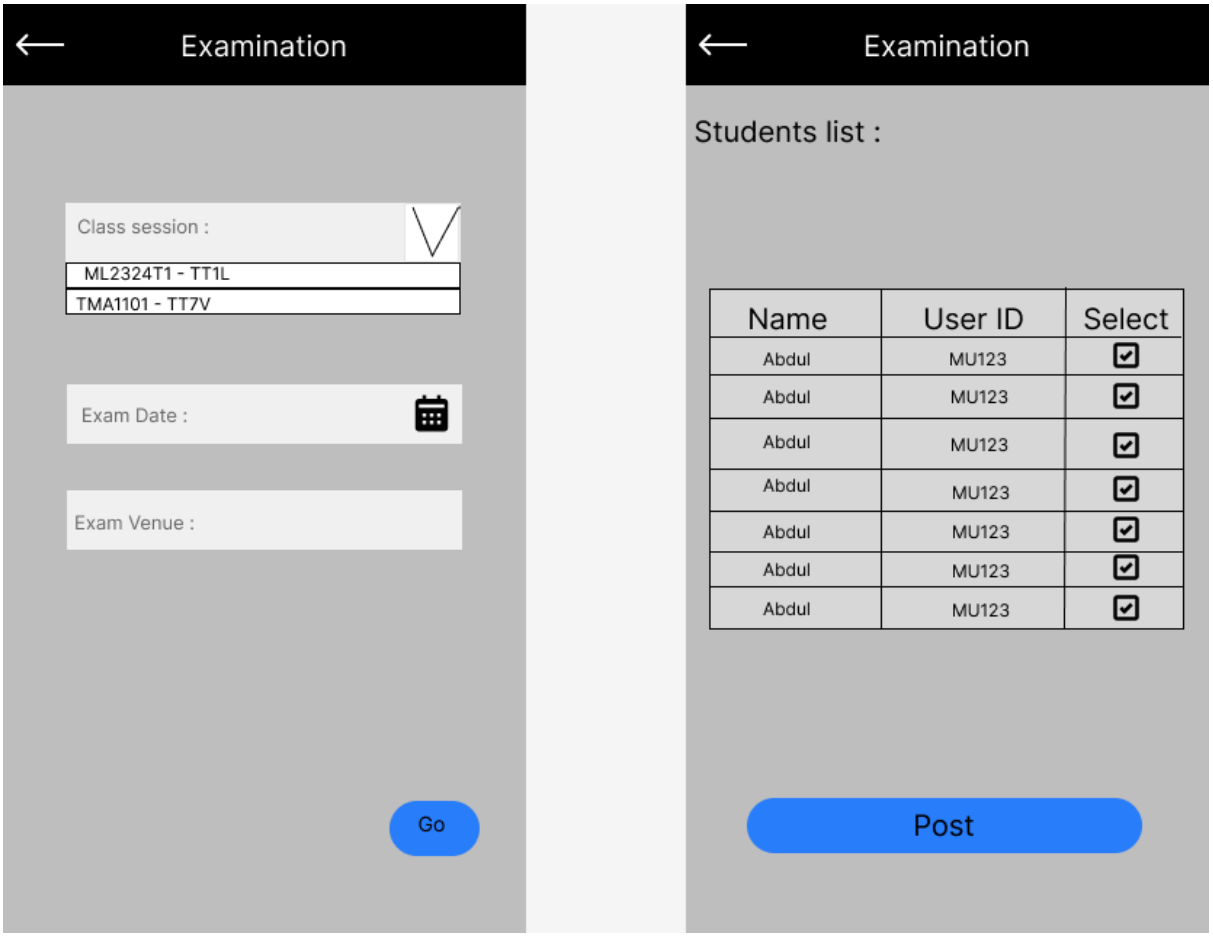


Figure 4.20 Assign students for the examination screen

In MMU, not all subjects will have a final examination. The lecturer will assign the final examinations based on the subjects they are teaching. From the dashboard, lecturers can select the 'Examination' option to assign the final examination for a subject. Firstly, the lecturer has to select the subject code. Then, choose a date and venue for the exam. The lecturer can view the list of students who have registered for that subject. It is up to the lecturer to decide which students can take the examination. Finally, they can publish these examination details to students.

4.3.3.4 Verify student identities in exam hall

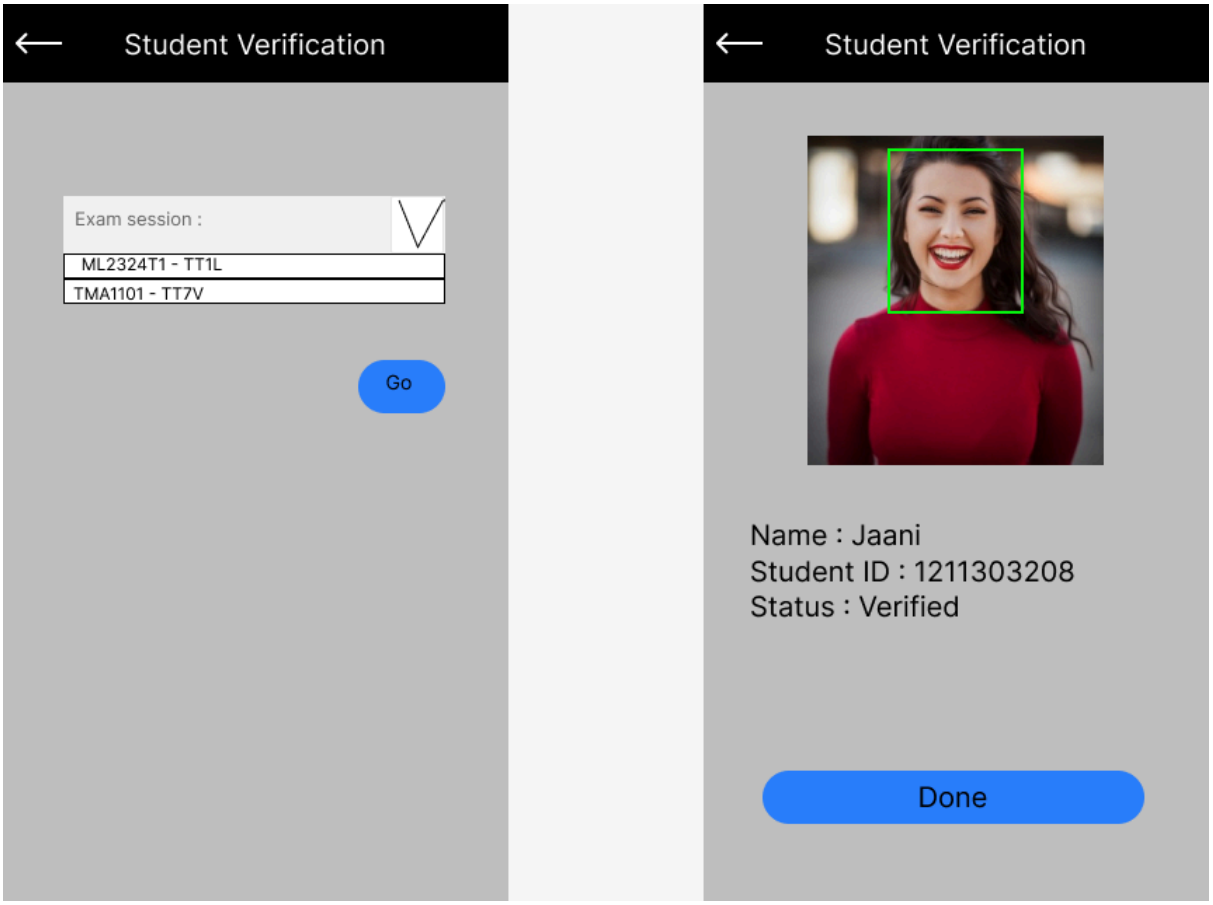


Figure 4.21 Verify student identities in exam hall screen

Verifying student identities in the exam hall is very important to prevent impersonation during exams by cross-referencing facial data with the registered student information. To verify the student, the lecturer can scan the students' faces to access their details and determine if they are permitted to sit for the examination.

4.3.4 Campus Staff Screens

4.3.4.1 Campus Staff Dashboard

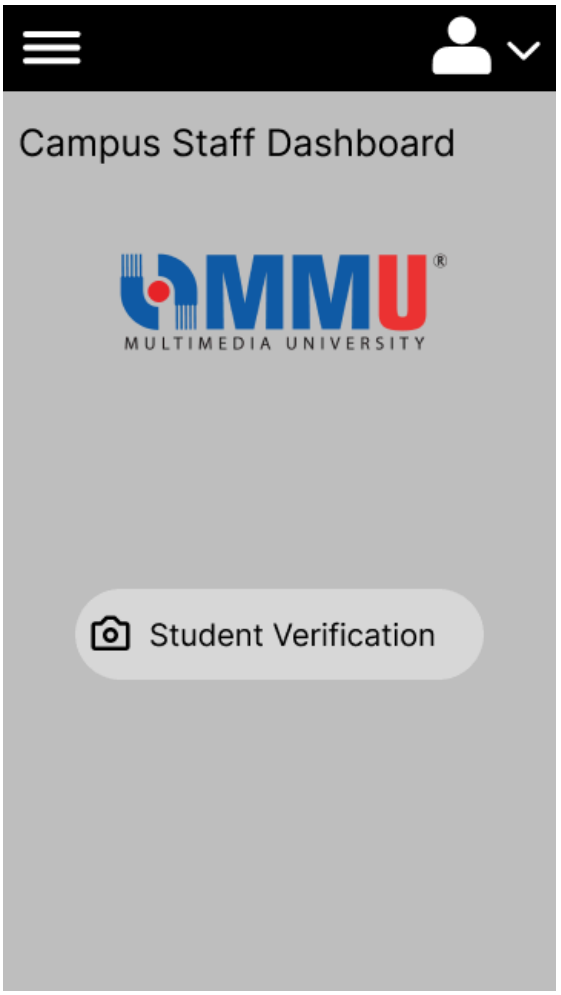


Figure 4.22 Campus Staff Dashboard screen

The figure above shows the campus staff dashboard, where they can assess their main functionality, which is verifying student identities on campus.

4.3.4.2 Verify student identities on campus

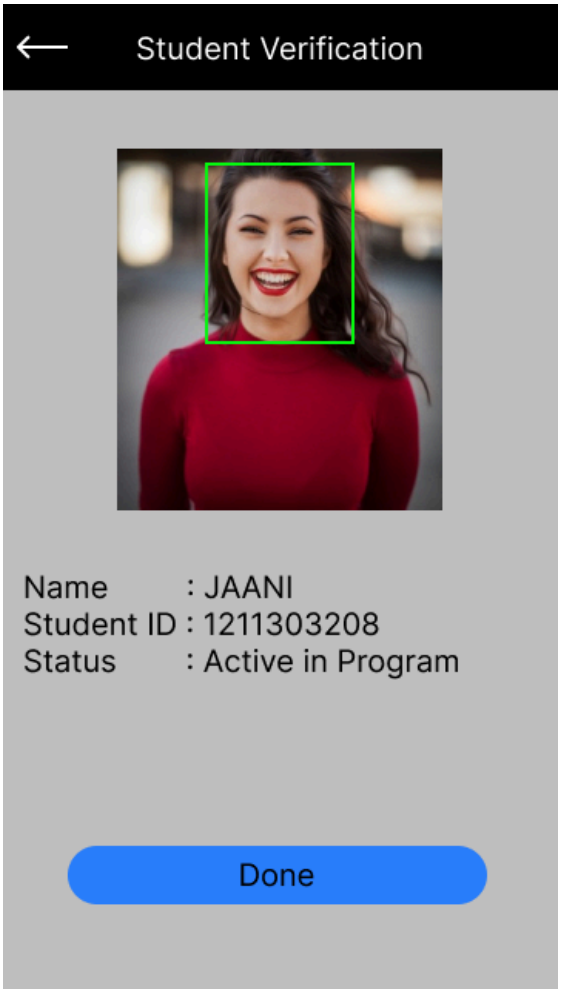


Figure 4.23 Verify student identities on campus screen

There are several parties who can serve as campus staff :

- Faculty members (Lecturers, teaching assistants, research assistants)
- Administrative staff (Registrars, academic advisor, office managers)
- Student Service Staff
- Librarian

These campus staff have to verify the students' identity on campus in order to grant the access to resources and services in MMU. Therefore, the campus staff can

utilise this application to verify the student identities through its facial recognition features.

4.3.5 Administrator Screens

4.3.5.1 Administrator Dashboard

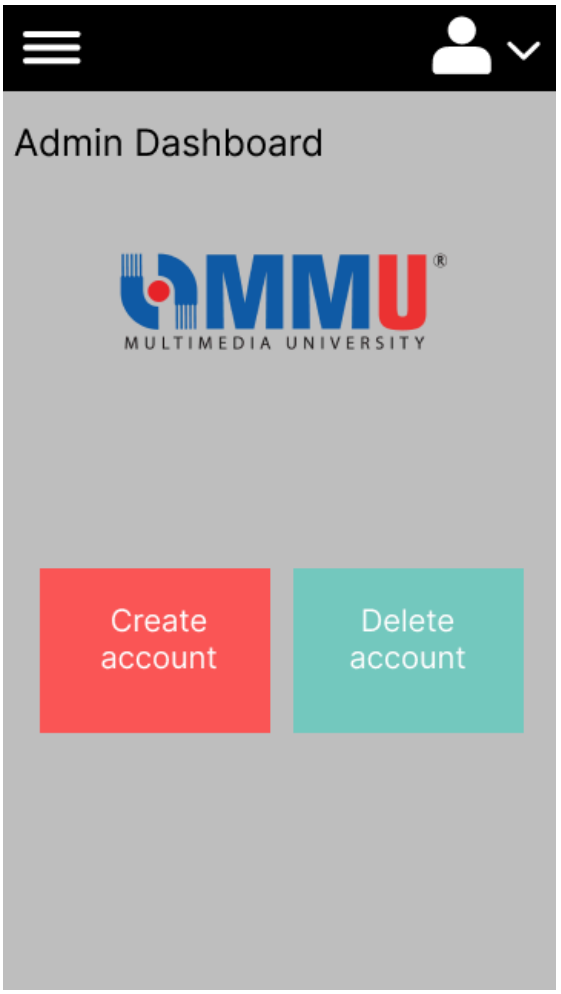


Figure 4.24 Administrator Dashboard screen

The figure above shows the administrator dashboard where administrators can create accounts for other users and delete user accounts.

4.3.5.2 Create account

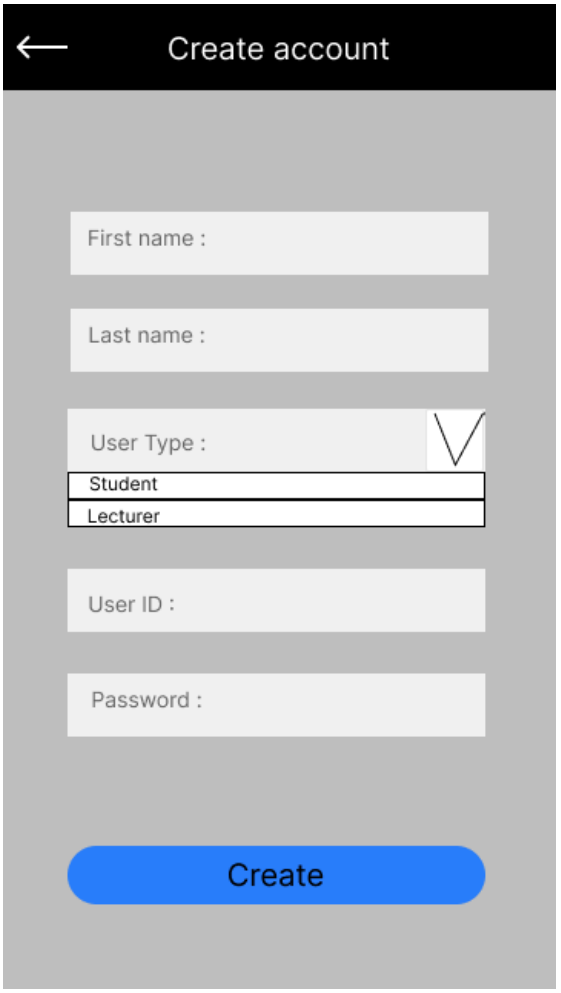


Figure 4.25 Create account screen

For creating an account, the administrator has to key in :

- First Name
- Last Name
- User Type
- User ID
- Password

Once the administrator created an account, the system will send an email to the user of the user ID and password.

4.3.5.3 Delete account

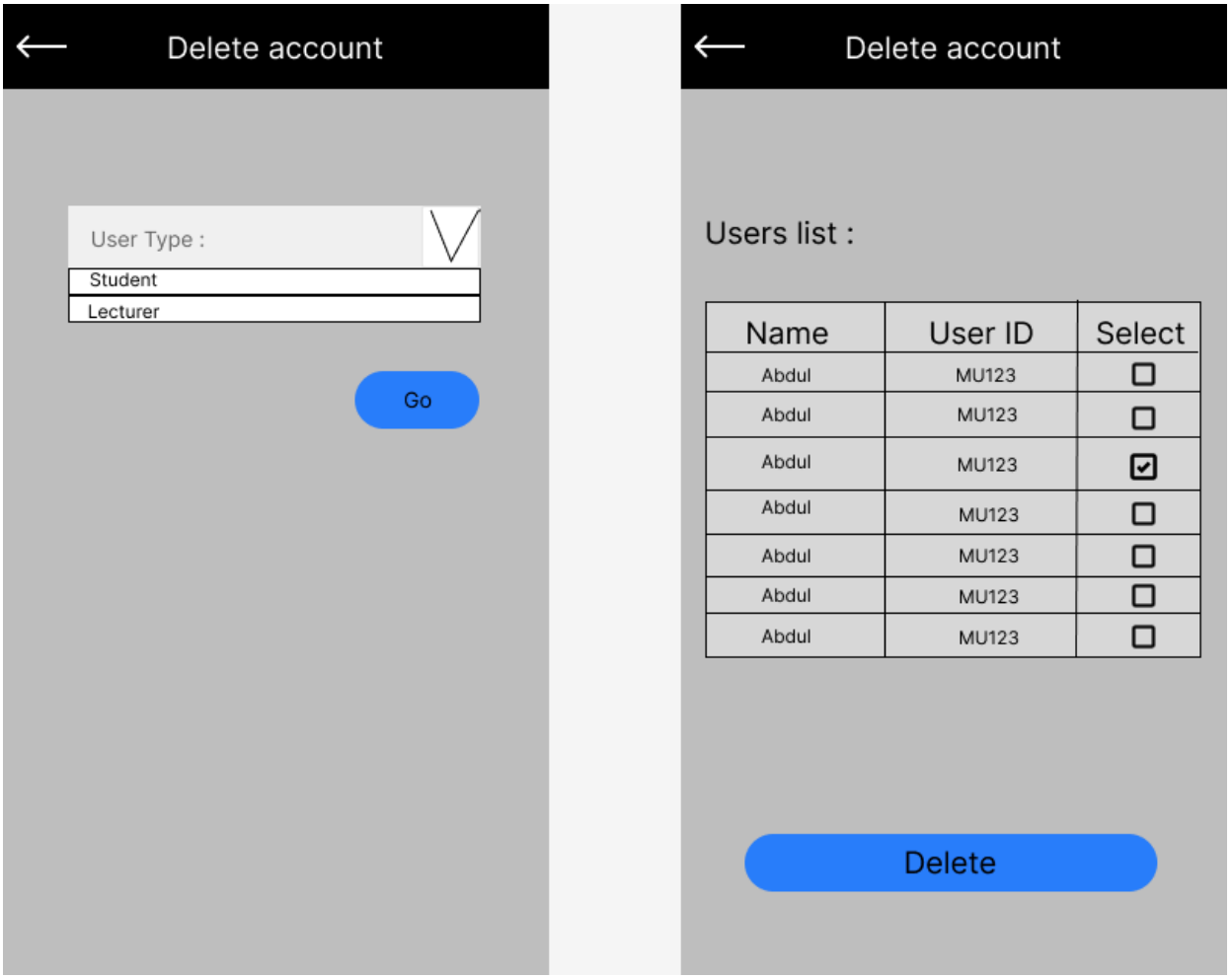


Figure 4.26 Delete account screen

The figure above shows the screen of how an administrator can delete a user from the database. First, the administrator has to select the user type. Then, the next screen will display the list of users based on the selected user type. From here, the administrator can tick the checkbox and click the delete button to remove the user from the database.

4.4 Data Dictionary

Table 4.2 shows the data dictionary for the entities in the database. The table for each entity includes details such as field name, data type, field size, description, and example. This system has a total of 9 entities. Additionally, the table includes the identification and description of the primary key and foreign key for each entity.

Table 4.2 Data Dictionary

Table	Field Name	Data Type	Field Size	Description	Example
Student	student_id	char	10	PK, Unique Student ID	1211303208
	student_firstName	varchar	100	Student First Name	
	student_lastName	varchar	100	Student Last Name	
	student_status	varchar	20	Student Status in MMU	Active,Inactive
	student_email	varchar	40	Student Email	
Administrator	admin_id	char	10	PK, Unique Lecturer ID	
	admin_name	varchar	100	Admin Name	
	admin_email	varchar	40	Admin email	
Lecturer	lecturer_id	char	10	PK, Unique Lecturer ID	
	lecturer_firstName	varchar	100	Lecturer First Name	
	lecturer_lastName	varchar	100	Lecturer Last Name	
	lecturer_email	varchar	40	Lecturer Email	
Campus Staff	staff_id	char	10	PK, Unique Staff ID	

	staff_name	varchar	100	Staff Name	
	staff_email	varchar	40	Staff email	
Biometric Data	biometric_id	char	10	PK, Biometric ID	
	student_id	char	10	FK, Student ID	
	biometric_embeddings	array	-	Represent features extracted from biometric data	
	date_created	date		Date of the biometric data registered	2024-02-14
	data_accuracy	int	100	The accuracy percentage of the biometric data	20
Attendance	attendance_id	char	10	PK, Unique Attendance ID	
	class_id	char	10	FK, Class ID	
	student_id	char	10	FK, Student ID	
	timestamp	datetime		Timestamp of the attendance	2024-02-14 11:58:12
Class	class_id	char	10	PK, Unique Class ID	
	subject_id	char	10	FK, Subject ID	
	class_venue	varchar	100	Class Venue	
	class_time	time		Class start time	
Subject	subject_id	char	10	PK, Unique Subject ID	
	lecturer_id	char	10	FK, Lecturer ID	
	student_id	char	10	FK, Student ID	

	subject_code	char	10	Subject Code	TIS 3151
	subject_name	varchar	100	Subject Name	
Exam Slip	exam_id	char	10	PK, Unique Exam ID	
	lecturer_id	char	10	FK, Lecturer ID	
	subject_id	char	10	FK, Student ID	
	exam_time	time		Exam Start Time	
	exam_date	date		Exam Date	

4.5 Deployment Diagram

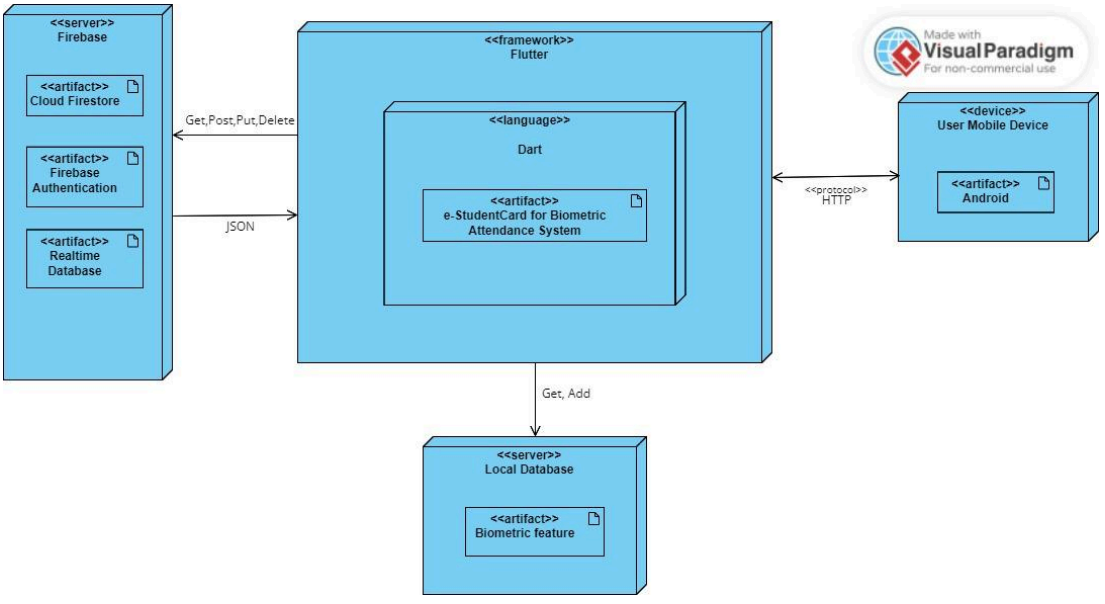


Figure 4.27 Deployment Diagram of the Application

Figure 4.27 illustrates the architecture of an e-StudentCard for Biometric Attendance System. The system is built using the Flutter framework, which uses the Dart programming language. The core of the application is represented by the "e-StudentCard for Biometric Attendance System" artefact within the Dart language component.

The system interacts with Firebase as its primary server, which includes Cloud Firestore for data storage, Firebase Authentication for user authentication, and a Realtime Database. The communication between the Flutter application and Firebase uses GET, POST, and DELETE operations, with data exchanged in JSON format.

On the client side, the system is designed to run on User Mobile Devices, specifically on Android platforms. The mobile app communicates with the server using HTTP protocols.

The system also interacts with a Local Database; MySQL server, which contains a Biometric feature artefact. The Flutter application performs GET and ADD operations with this local database, likely for storing and retrieving biometric data.

This architecture allows for a mobile-based student attendance system that leverages cloud services for data management and authentication, while also incorporating local biometric capabilities. The use of Flutter enables cross-platform development, though the diagram specifically mentions Android for the client devices.

Chapter 5: Implementation

In the preceding chapters, we discussed and detailed the methodology of the e-StudentCard for Biometrics Attendance System. This chapter will now focus on the actual implementation of the project, including the mobile app.

5.1 Overall Description

5.1.1 Mobile Application System Architecture

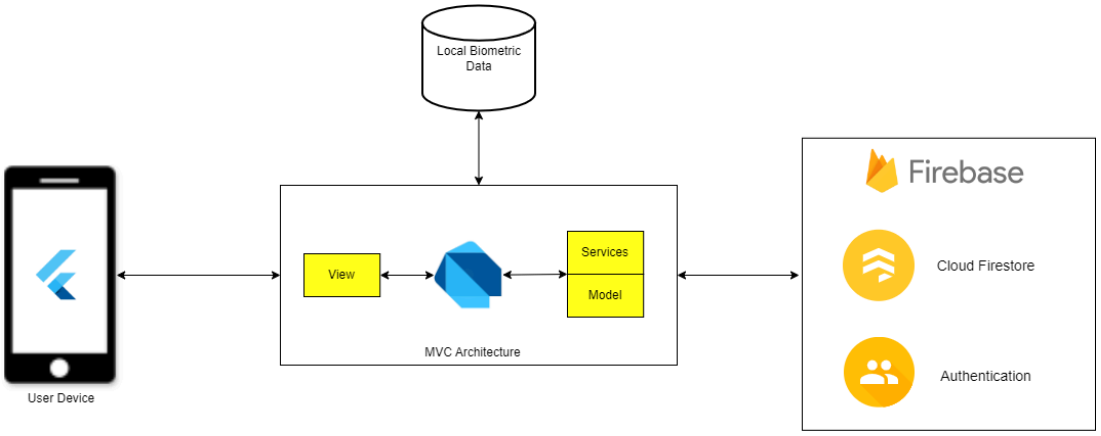


Figure 5.1 Mobile Application System Architecture Diagram

In the project implementation, the Mobile Application System Architecture is designed to efficiently handle biometric attendance tracking. The system is built using the Flutter framework, with Dart as the core programming language. The architecture follows the Model-View-Controller (MVC) pattern, which is obtained via the Dart language, ensuring a clear separation of concerns. The View component manages the user interface, interacting directly with the user's mobile device. The Model and Services components handle data structures, business logic, and external communications. Local biometric data is stored in a dedicated database on the device, enabling quick access and offline functionality. The system integrates with Firebase, utilising Cloud Firestore for cloud-based data storage and synchronisation,

and Firebase Authentication for secure user management. This architecture facilitates seamless interaction between local device operations and cloud services, providing a robust foundation for the e-StudentCard Biometric Attendance System. The design leverages Dart's capabilities within the Flutter framework to ensure scalability, maintainability, and efficient performance, balancing local processing with cloud-based services to deliver a comprehensive attendance tracking solution.

5.1.2 Face Detection and Face Recognition

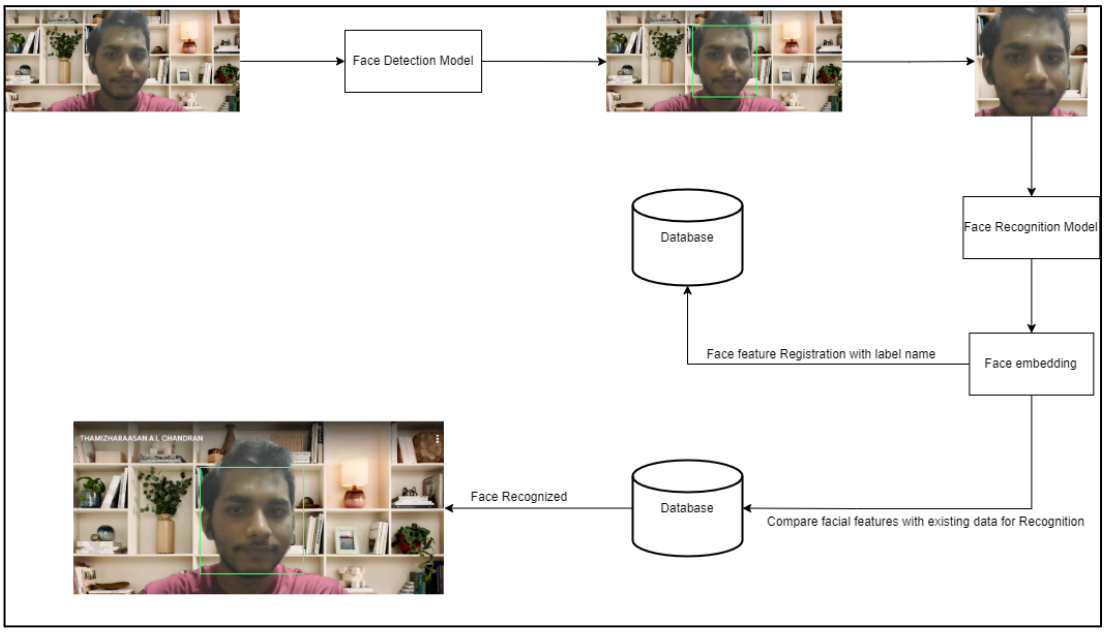


Figure 5.2 Face Detection and Face Recognition Overall Explanation Diagram

Figure 5.2 illustrates the process of face detection and recognition using a combination of Google ML Kit for face detection and FaceNet for face recognition. Here's an elaboration of the workflow:

1. **Face Detection:**
    - The process begins with an input image containing a person's face.
    - This image is passed through a Face Detection Model, implemented using Google ML Kit.
    - ML Kit identifies and locates faces within the image, potentially adjusting for different angles or partial obstructions.
2. **Face Recognition:**
    - Once a face is detected, the isolated face image is passed to the Face Recognition Model, which uses FaceNet.
    - FaceNet processes the face to create a face embedding, which is a numerical representation of the facial features.
3. **Database Interaction:**

- For new users, the face embedding along with a label (person's unique id) is registered in the database.
- For recognition, the system compares the newly generated face embedding with existing data in the database.

4. **Recognition Output:**
    - If a match is found in the database, the system recognizes the face and can label it in the output image.
    - The final image shows the recognized face with a bounding box and the person's name.

Tools Used:

1. Google ML Kit: This is used for the initial face detection step. ML Kit is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package.
2. FaceNet: This is employed for face recognition. FaceNet is a deep learning model developed by Google that learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity.

This combination of tools allows for efficient and accurate face detection and recognition, suitable for mobile applications due to ML Kit's optimization for mobile devices. The system can handle various lighting conditions and angles, as seen in the sample images, making it robust for real-world use in applications like the e-StudentCard Biometric Attendance System.

5.2 Back-End Development

Flutter projects were developed using Android Studio and an Android emulator of Google Pixel 7 Pro along with a physical android device. After successfully setting up the projects, the Dart files were organised using local packages. Additionally, backend configurations, such as integrating Firebase into the project and collaborating with local databases, were completed. The following subsections will provide more details about the implemented solutions.

5.2.1 Firebase Authentication

The email and password authentication is used to authenticate all types of users in this project. Figure 5.3 shows that email and password provider is enabled as one of the sign-in providers in the Firebase console.

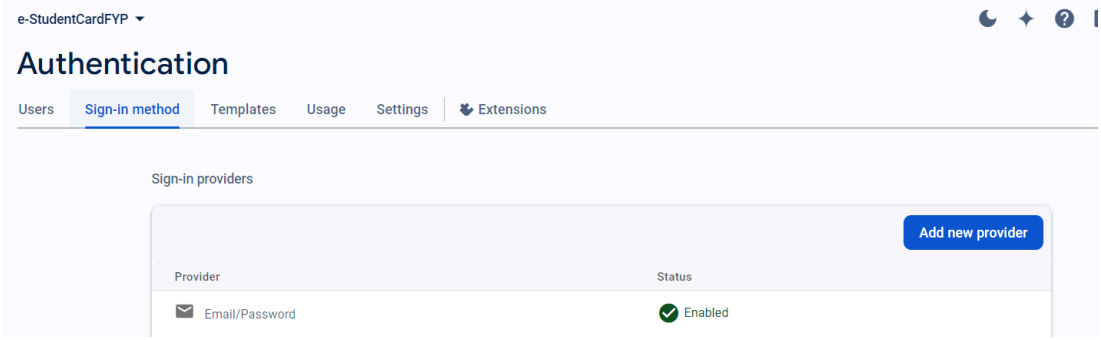


Figure 5.3 Firebase Authentication Sign-In Provider

For email and password authentication, a class named AuthService handles registration, login and logout functions. In this system, only the administrator has the authority to register users, while students, lecturers, and campus staff only can perform login and logout operations only. Figure 5.3 below shows the list of users that have access to the system.

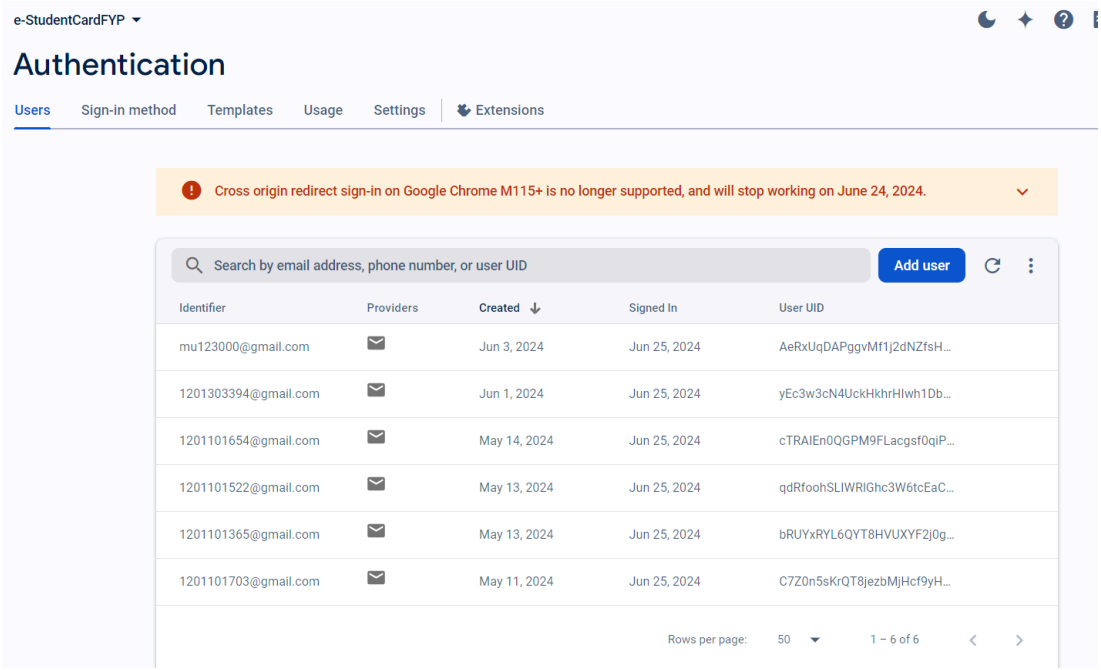


Figure 5.3 Firebase Authentication Users

The login process is simple because it only requires using the built-in signInWithEmailAndPassword() function provided by FirebaseAuth. This function automatically verifies the email and password combination against the registered users in the Firebase project. For user registration, it uses createUserWithEmailAndPassword() function whereas, logout uses signOut() function which is provided by FirebaseAuth as well. In brief, Firebase Authentication empowers this project to deploy secure and robust authentication mechanisms.

5.2.2 Firebase Cloud Firestore

Cloud Firestore from Firebase serves as the project's database, storing data as documents and enabling queries for creating, reading, updating, and deleting data. The organisation of the data store in Cloud Firestore is depicted in Figure 5.4.

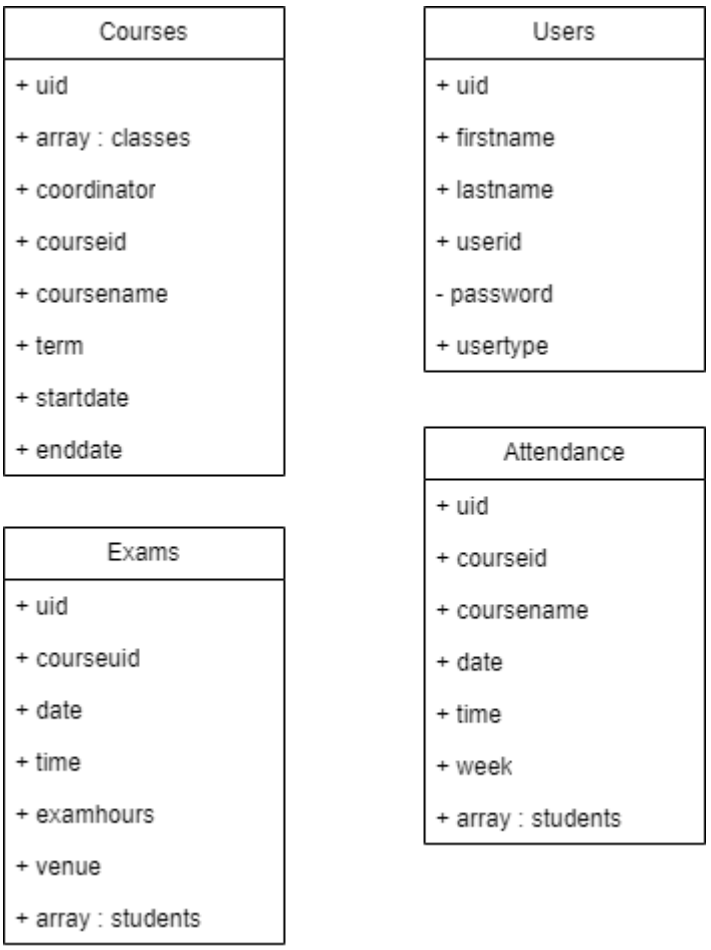


Figure 5.4 Data Modelling Diagram for Cloud Firestore

The database for this project consists of 4 collections, which are users, exams, attendance and courses collections. The courses collection is manually populated in the Firebase console to provide dummy data. It represents the courses offered within the system, including course codes, names, instructors, and other course-related details.

These collections in Cloud Firestore are structured to efficiently manage and query data relevant to user management, academic activities (courses and exams), and attendance tracking within the application.

5.2.3 Local Biometric database

A local database is used to store the facial features extracted using the FaceNet algorithm. Using a local database offers advantages such as enhanced performance, offline access, improved data privacy, scalability, seamless integration, and reduced dependency on network connectivity. These benefits make it particularly suitable for applications requiring efficient and secure data storage and retrieval, such as those utilising facial recognition technology like FaceNet.

For storing facial features locally, MySQL is used. It initialises and manages a local MySQL database within the application, ensuring efficient handling and retrieval of biometric data. The openConnection() method as shown in Figure 5.5 sets up the database, connecting to “biometric_db” database in MySQL to store data.

```
//TODO mySQL initialization
Future<void> openConnection() async {
  _connection = await MySqlConnection.connect(ConnectionSettings(
    host: '192.168.100.49',   // emulator : 10.0.2.2
    port: 3306,
    user: 'root',
    db: 'biometric_db',
  ));
}
```

Figure 5.5 openConnection() method

Additionally, Figure 5.6 displays the SQL code for inserting data like user uid and face embedding in MySQL database.

```
// TODO insertData
Future<void> insertData(String name, String embedding) async {
  await openConnection();
  try {
    await _connection.query(
      'INSERT INTO biometric_table (uid, embedding) VALUES (?, ?)',
      [name, embedding],
    );
  } catch (e) {
    print('Error inserting data: $e');
  }
}
```

Figure 5.6 Sql code to insert data into table

Methods like insert, queryAllRows, update, and delete facilitate operations for adding, retrieving, updating, and deleting records from the database. This setup ensures efficient local storage and retrieval of facial feature data, supporting the application's functionality seamlessly within the device's environment.

5.3 Front-End Development

This subsection will highlight key aspects of the front-end development of this project, accompanied by user interface screenshots.

5.3.1 Log In interface development

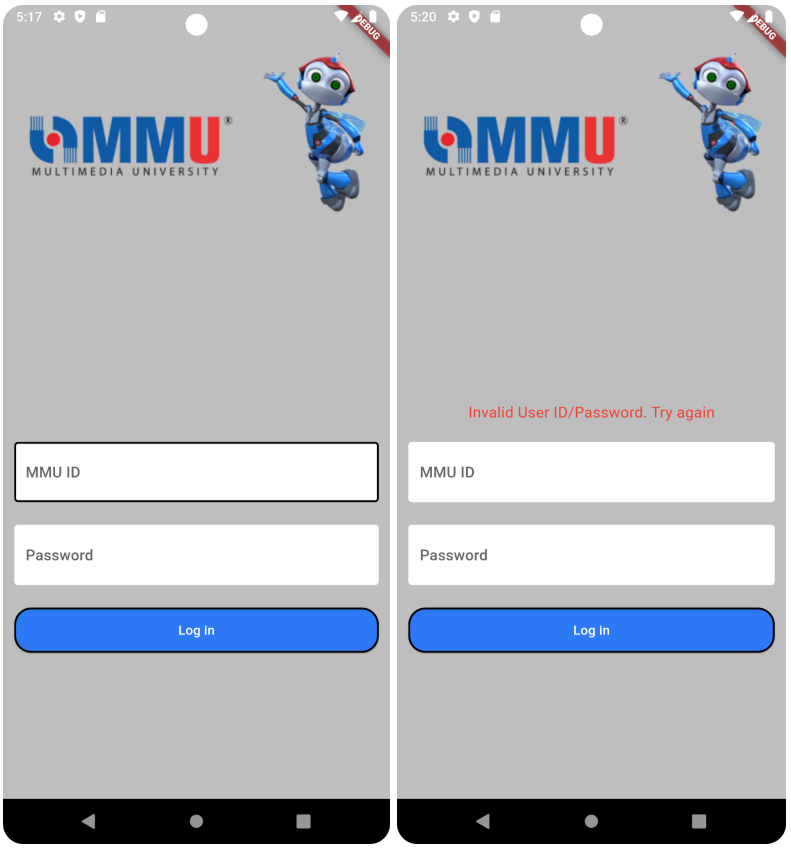


Figure 5.7 User Login Screen

Figure 5.7 shows the login screen of the system, with a sample of invalid login credentials. All types of users share the same login page. Users have to input MMU user ID and password and click the ‘Login' button to access their respective functionalities according to their user type. If the user provided credentials correctly, the system will check the user type and redirect them to their respective dashboard according to their user type.

5.3.2 Student's interface development

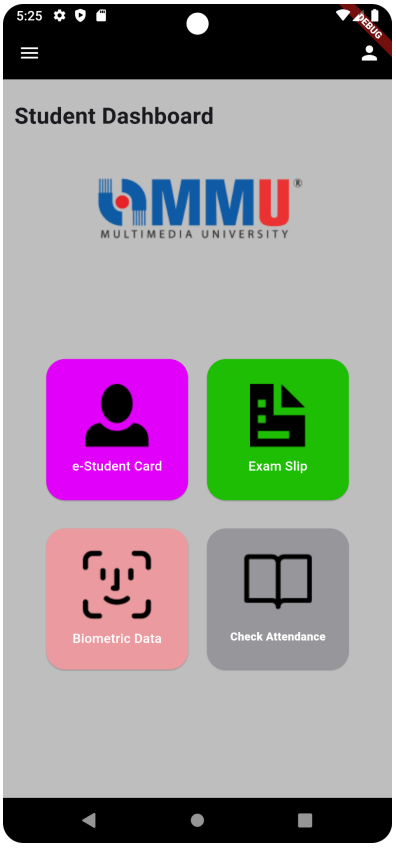


Figure 5.8 Student Dashboard Screen

Figure 5.8 shows the student dashboard screen where students can access this page only if they provide the valid and correct login credentials at the login page. From the student dashboard, students can perform their main functionalities such as view e-student card, view exam slip, register biometric data and check their attendance. If students have nothing to do, they can logout from the system.

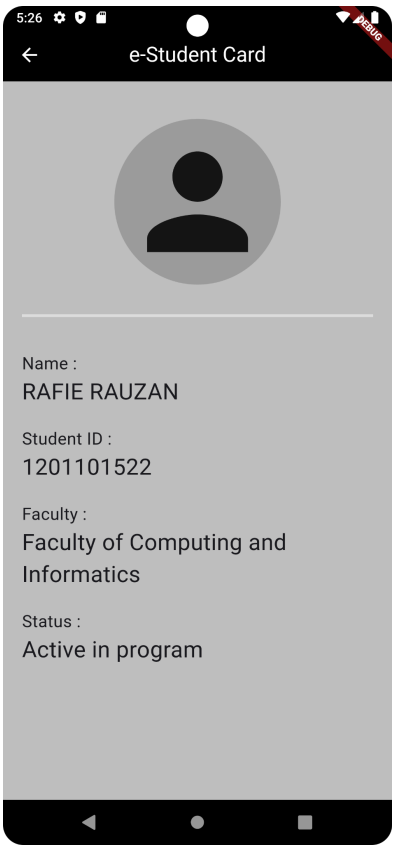


Figure 5.9 Student View e-Student Card Screen

Figure 5.9 shows the screen when students select to view an e-student card from the dashboard. Basically, it resembles a traditional student card and contains exactly the same details as the original student card. The e-student card displays the student name, student id, faculty and their current status as a student. Students can present this e-student card on campus to access university facilities and ensure that the guards permit them to enter the campus. Off-campus, students can still use this e-student card to take advantage of student promotions at cinemas, for transportation, and for other benefits such as discounts at restaurants, retail stores, and recreational facilities.

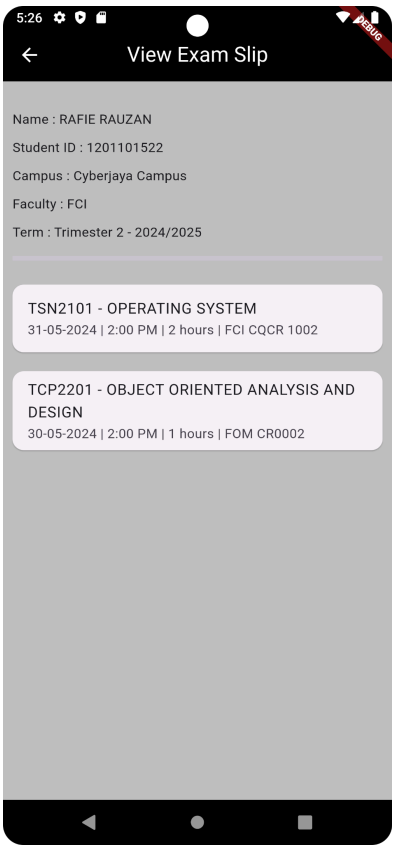


Figure 5.10 Student View Exam Slip Screen

Figure 5.10 shows the exam slip screen from the student side as they have been redirected from the student's dashboard. The condition to view exam slips is the lecturer must assign and post the examination details from their side. Otherwise, the system will display a message that ‘No exam records to display’. For viewing the exam slip, it shows the student's details and the subject details for the exams they need to attend. Here are the details of the subjects to be displayed in this page:

- Subject name
- Subject ID
- Exam Date
- Exam Time
- Exam Venue
- Exam Duration

If students forget to bring their exam slip to the exam hall, they can present it through this system to gain entry.

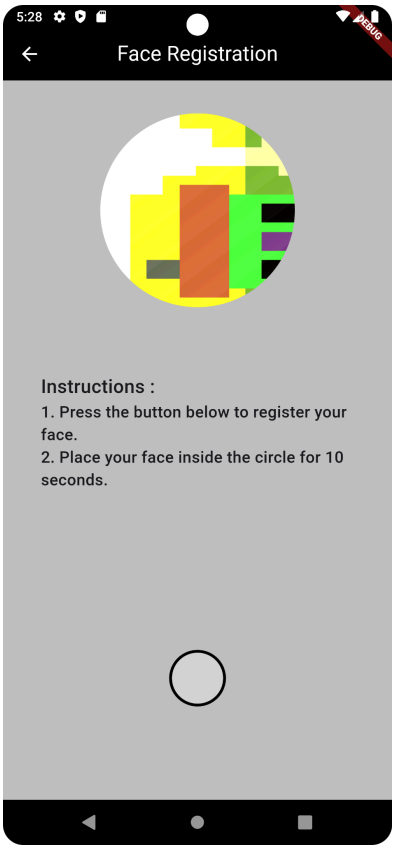


Figure 5.11 Student Face Registration Screen

Figure 5.11 shows how students can perform biometric registration through this mobile application. When students click the biometric icon button from the dashboard, they will be redirected to this screen. In this screen, students have to read the instructions properly before clicking the grey and circle button below. After the system performs face registration operation successfully, students will be redirected to the dashboard screen automatically.

Figure 5.12 Student Check Attendance Screen

Figure 5.12 shows the check attendance screen for students. If there is no attendance taken by the lecturer, it displays a message that “No attendance records”. Otherwise, it displays the attendance details as below for each subjects :

- Subject Name
- Subject Id
- Class Session
- Time

5.3.3 Lecturer's interface development

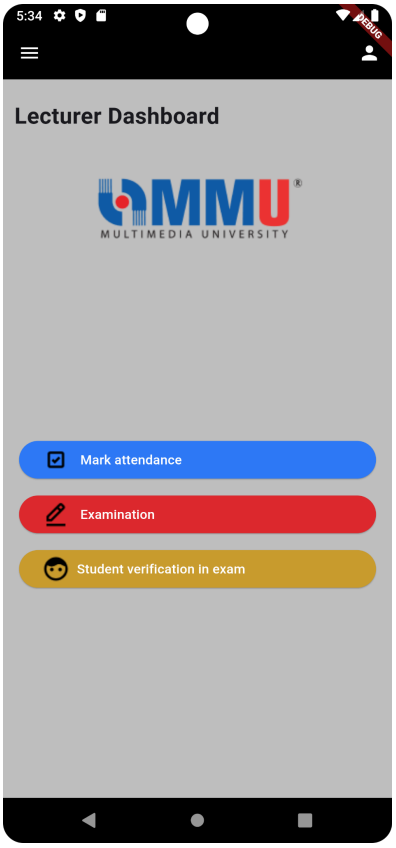


Figure 5.13 Lecturer Dashboard Screen

Figure 5.13 shows the lecturer dashboard screen where lecturers can access this page only if they provide the valid and correct login credentials at the login page. From the lecturer dashboard, lecturers can perform their main functionalities such as mark attendance, assign examinations and verify student identity in the exam hall. If lecturers have nothing to do, they can logout from the system.

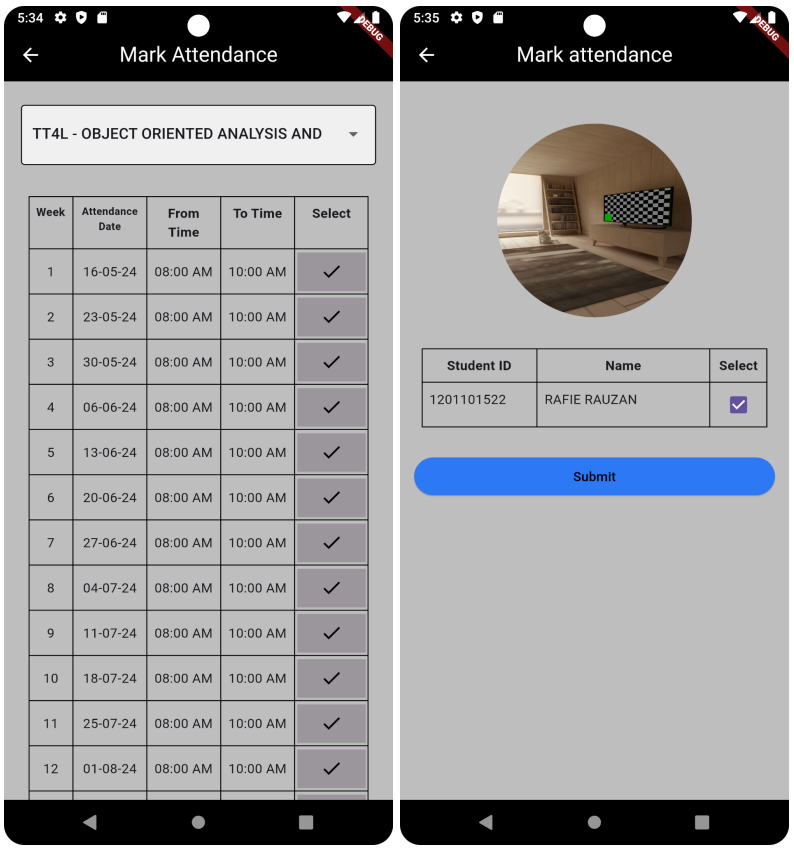


Figure 5.14 Lecturer Mark Class Attendance Screen

Figure 5.14 shows how lecturers can mark students' attendance in class. Initially, lecturers need to choose their class session, and a table displaying the list of class dates will appear. Lecturers can then select a specific week to record attendance. When the select icon button is clicked, the system will redirect to another page displaying a list of students, each with a checkbox in their respective row. On this screen, lecturers can scan students' faces to mark attendance. If a valid student face is scanned, the checkbox will be automatically ticked. Finally, lecturers can submit the attendance records, which students can view from their end.

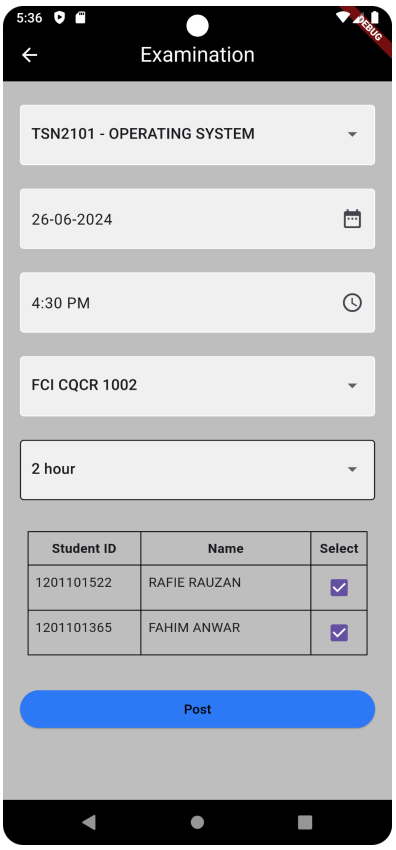


Figure 5.15 Lecturer Assign Examination Screen

Figure 5.15 depicts the examination assignment screen, where lecturers can publish examination details for students. On this screen, lecturers have to select the subject first. When a subject is selected, a list of students, each with a checkbox in their respective row will appear. From the student list, lecturers can deselect specific rows if those particular students are not eligible to sit for the exam. After completing the necessary details in this screen, lecturers can click the ‘Post’ button to publish the examination details.

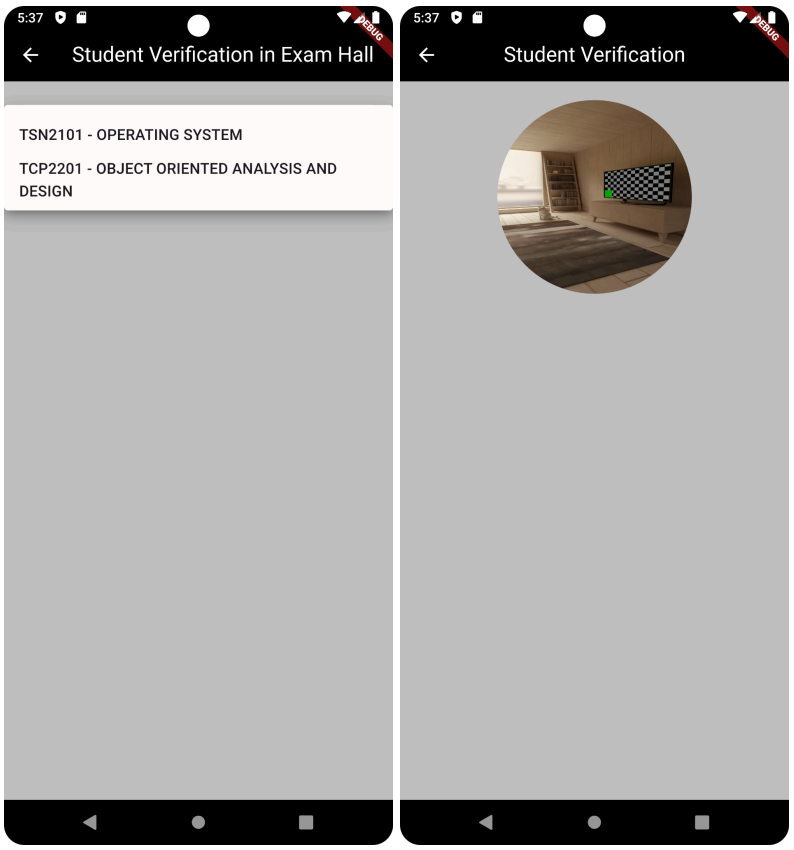


Figure 5.16 Student Verification In Exam Screen

Figure 5.16 illustrates student verification on the exam hall screen. Initially, lecturers must select the subject from the dropdown menu on the first page. Subsequently, the system redirects to another screen for scanning students' faces. Upon successful verification, a message confirms the student's permission to proceed with the exam; otherwise, access is denied.

5.3.4 Campus Staff's interface development

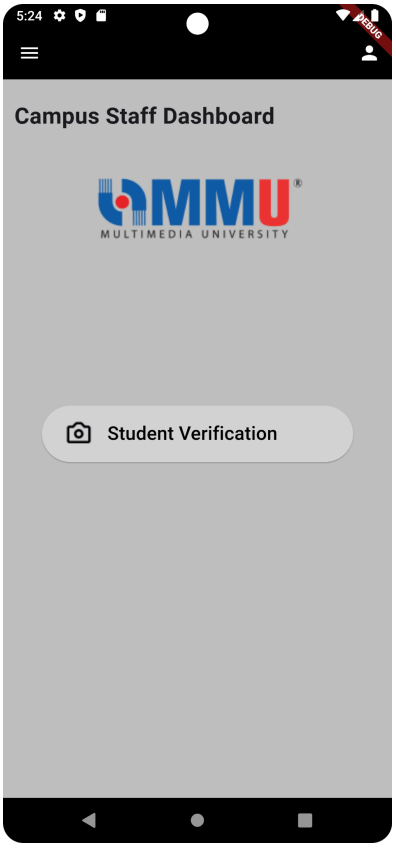


Figure 5.17 Campus Staff Dashboard Screen

Figure 5.16 shows the campus staff dashboard screen where campus staff can access this page only if they provide the valid and correct login credentials at the login page. From the campus staff dashboard, campus staff can perform their main functionalities such as student identity verification on campus. If campus staff have nothing to do, they can logout from the system.

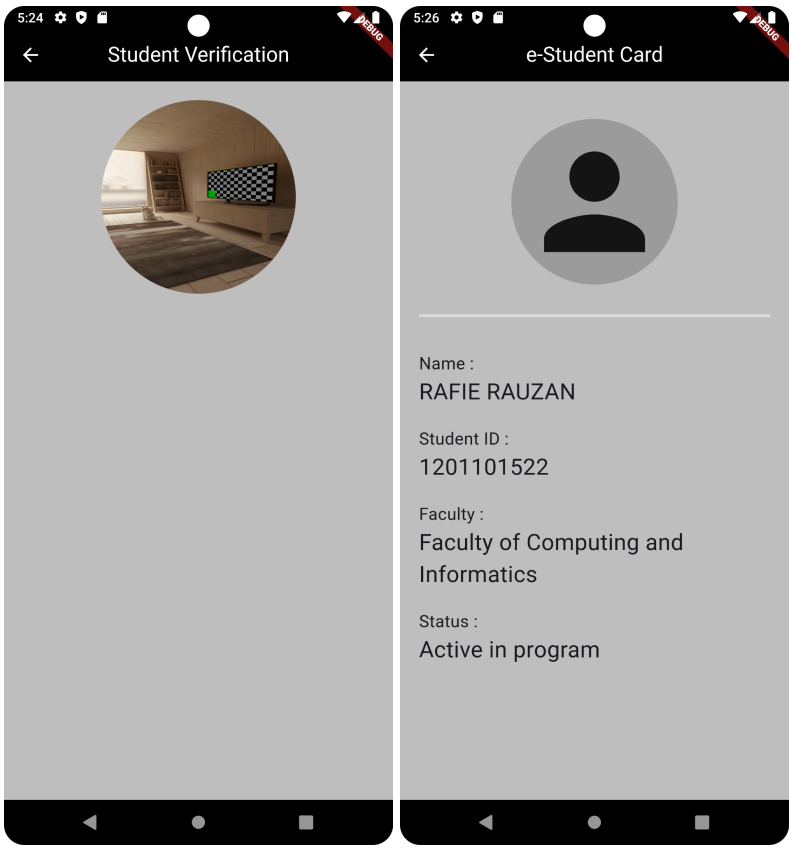


Figure 5.18 Student Verification Screen

Figure 5.17 shows how campus staff can verify student identities on campus. When campus staff select their only functionality from the dashboard, the system redirects them to the student verification screen. On this screen, the camera automatically opens, and the system begins scanning student faces. If the system successfully recognizes the student's face, it will display the student's e-student card on the next screen. Otherwise, it goes back to the dashboard screen.

5.3.5 Administrator ’s interface development

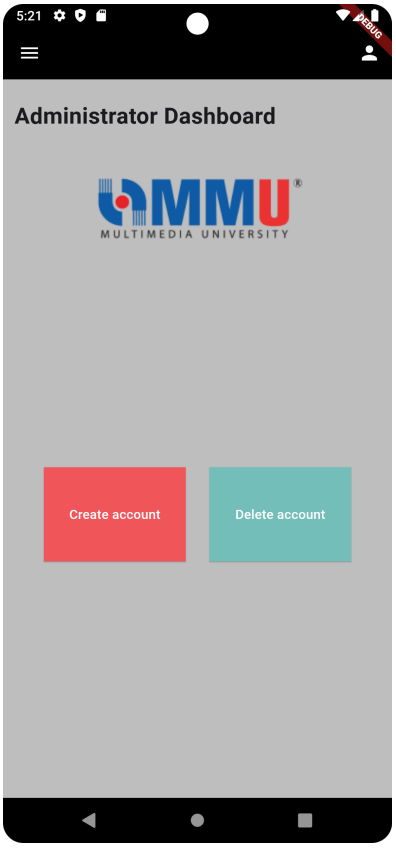


Figure 5.19 Administrator Dashboard Screen

Figure 5.18 shows the administrator dashboard screen where administrators can access this page only if they provide the valid and correct login credentials at the login page. From the administrator dashboard, administrators can perform their main functionalities such as student identity verification on campus. If administrators have nothing to do, they can logout from the system.

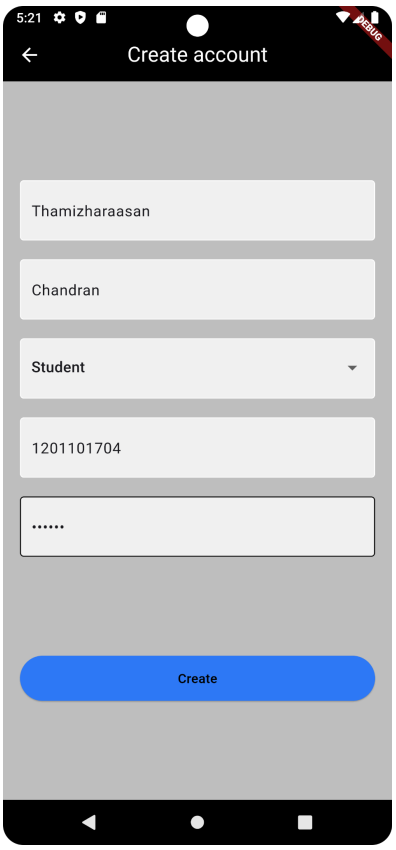


Figure 5.20 Administrator Create Account Screen

Figure 5.19 displays the administrator account creation screen, accessible when the administrator is chosen to create an account from the dashboard. It's important to note that only the administrator can create accounts for other users, such as students, lecturers, and campus staff. Here are the required details to create an user account :

- First Name
- Last Name
- User Type
- MMU User ID
- Password

After completing the create account form, the administrator can click the “Create” button so the system will update the new user data in the firebase authentication section.

Figure 5.21 Administrator Delete Account Screen

Figure 5.20 displays the administrator account deletion screen. In this screen, the administrator has to select the user type from the drop down menu at the top of the screen. When a user type is selected, a list of users based on the selected user type will appear. The administrator can then click the delete icon to remove users.

Chapter 6: Testing

6.1 Functional Testing

The Functional Testing section outlines the rigorous evaluation conducted to ensure that the software performs according to specified requirements and user expectations. It documents the execution of test cases designed to validate functionality, usability, and overall system behaviour, highlighting any identified issues and their resolutions to guarantee the software's reliability and readiness for deployment.

6.1.1 Student

6.1.1.1 Register biometric data

Table 6.1 Register Biometric Data Test Case

Test Case Name	**Register biometric data**
Test Case Steps	1. Select the ‘Register Biometric Data’ option from the dashboard. 2. Read the instructions on the page. 3. Click the round and grey button to open the camera. 4. Position the face within a circle on the screen for a few seconds.
Test Input	-
Expected Results	1. The system will store the students’ biometric data in a database with a label. 2. A message is displayed that the data successfully saved.

Actual Results	1. The system will store the students' biometric data in a database with a label. 2. A message is displayed that the data successfully saved.
Status	Pass
Comments	Lecturers and campus staff can scan students' faces to verify student identities.

6.1.1.2 Check class attendance

Table 6.2 Check Class Attendance Test Case

Test Case Name	**Check class attendance**
Test Case Steps	1. Select the 'Check Attendance' option from the dashboard.
Test Input	"Check Attendance" button
Expected Results	The system will display the attendance. If there is no attendance to show the system will display "No attendance records"
Actual Results	The system will display the attendance. If there is no attendance to show the system will display "No attendance records"
Status	Pass
Comments	Students only can view attendance list if lecturer has marked their attendance in class

6.1.1.3 View e-student card

Table 6.3 View e-Student Card Test Case

Test Case Name	**View e-student card**
Test Case Steps	1. Select the ‘View e-Student Card’ option from the dashboard.
Test Input	‘View e-Student Card’ button
Expected Results	The system must display the e-Student Card of the student.
Actual Results	The system must display the e-Student Card of the student.
Status	Pass
Comments	

6.1.1.4 View exam slip

Table 6.4 View Exam Slip Test Case

Test Case Name	**View exam slip**
Test Case Steps	1. Select the ‘View exam slip’ option from the dashboard.
Test Input	‘View exam slip’ button
Expected Results	The system should display exam slips if the lecturer has assigned examinations for students. Otherwise, it displays nothing
Actual Results	The system should display exam slips if the lecturer has assigned examinations for students. Otherwise, it displays nothing
Status	Pass
Comments	Exam slip is reflected when lecturer assigned the examination for students

6.1.2 Lecturer

6.1.2.1 Log student attendance

Table 6.5 Log Student Attendance Test Case

Test Case Name	**Log student attendance**
Test Case Steps	1. Select the “Mark Attendance” option from the dashboard. 2. Select one class from the dropdown menu item. 3. List of classes for every week is displayed. 4. Click the select icon for a particular week. 5. Camera will be opened in new screen with a list of students in that particular class
Test Input	Class session, week number, “Submit” button
Expected Results	The checkbox in a student's row is automatically marked when that student's face is scanned. Attendance details are then updated accordingly in Firebase.
Actual Results	The checkbox in a student's row is automatically marked when that student's face is scanned. Attendance details are then updated accordingly in Firebase.
Status	Pass
Comments	The attendance record is reflected on the student's page.

6.1.2.2 Assign students for the examination

Table 6.6 Assign Students For The Examination Test Case

Test Case Name	**Assign students for the examination**
Test Case Steps	1. Select the “Examination” option from the dashboard. 2. Choose a subject from the dropdown menu. 3. Verify that a list of students appears below. 4. Enter additional details like exam venue, date, time, and duration. 5. Click the “Post” button to confirm.
Test Input	● Selection of a subject from the dropdown menu. ● Verification of the list of students displayed. ● Inputting exam venue, date, time, and duration details. ● Clicking the "Post" button to submit the examination information.
Expected Results	Upon completion of the test case steps, the expected outcome includes successfully navigating to the "Examination" option, selecting a subject from the dropdown menu, viewing a list of associated students, entering exam venue, date, time, and duration details, and successfully posting the examination information without errors.
Actual Results	Upon completion of the test case steps, the expected outcome includes successfully navigating to the "Examination" option, selecting a subject from the dropdown menu, viewing a list of associated students, entering exam venue, date, time, and duration details, and

	successfully posting the examination information without errors.
Status	Pass
Comments	The examination record is reflected on the student's page.

6.1.2.3 Verify student identities in exam

Table 6.7 Verify Student Identities In Exam Test Case

Test Case Name	Verify student identities in exam
Test Case Steps	1. Select the “Student Verification in exam” option from the dashboard. 2. Choose a subject from the dropdown menu. 3. Proceed to another page where the system prompts for scanning students’ faces.
Test Input	● Navigating to the dashboard and locating the "Student Verification in exam" option. ● Selecting a subject from the dropdown menu. ● Initiating the process that triggers the system to redirect to the page for scanning students' faces.
Expected Results	Upon selecting the "Student Verification in exam" option from the dashboard and choosing a subject from the dropdown menu, the system should redirect to another page where it prompts for scanning students' faces, ensuring smooth transition and functionality without errors.
Actual Results	Upon selecting the "Student Verification in exam" option from the dashboard and choosing a subject from the dropdown menu, the system should redirect to another page

	where it prompts for scanning students' faces, ensuring smooth transition and functionality without errors.
Status	Pass
Comments	Grant access for students to sit for the exam.

6.1.3 Campus Staff

6.1.3.1 Verify students' identity on campus

Table 6.8 Verify Students' Identity On Campus Test Case

Test Case Name	**Verify students' identity on campus**
Test Case Steps	1. Select the "Student Verification" option from the dashboard. 2. The camera opens in a new screen to scan students' faces.
Test Input	"Student Verification" button from dashboard
Expected Results	Upon successful scanning of a face, the system should identify the student and display their e-student card. Otherwise, if the scan fails or is unsuccessful, the system should return to the dashboard.
Actual Results	Upon successful scanning of a face, the system should identify the student and display their e-student card. Otherwise, if the scan fails or is unsuccessful, the system should return to the dashboard.
Status	Pass
Comments	

6.1.4 Administrator

6.1.4.1 Create account for users

Table 6.9 Create Account For Users Test Case

Test Case Name	**Create account for users**
Test Case Steps	1. Select the "Create Account" option from the dashboard. 2. Input the first name, last name, user type, MMU user ID, and password of the new user. 3. Click the "Create" button to create a new user.
Test Input	1. Selecting the "Create Account" option from the dashboard. 2. Entering the following details for the new user: ● First name ● Last name ● User type ● MMU user ID ● Password 3. Clicking the "Create" button to submit the form.
Expected Results	1. Successfully navigate to the "Create Account" page upon selecting the option from the dashboard. 2. The system should accept the inputted first name, last name, user type, MMU user ID, and password without any errors. 3. Upon clicking the "Create" button, the system should create a new user account and add the user data in Firebase

	Authentication and display a confirmation message indicating that the account has been successfully created.
Actual Results	1. Successfully navigate to the "Create Account" page upon selecting the option from the dashboard. 2. The system should accept the inputted first name, last name, user type, MMU user ID, and password without any errors. 3. Upon clicking the "Create" button, the system should create a new user account and add the user data in Firebase Authentication and display a confirmation message indicating that the account has been successfully created.
Status	Pass
Comments	Lecturers, Students and Campus Staff can access the system after this operation

6.1.4.2 Delete user account

Table 6.10 Delete User Account Test Case

Test Case Name	**Delete user account**
Test Case Steps	1. Select the "Create Account" option from the dashboard. 2. Select the user type from the dropdown menu item. 3. Delete any of the users from the user list.
Test Input	User type and delete icon
Expected Results	1. The system should display a list of users based on the selected user type.

	2. When the user is deleted, the user data also will be remove from the Firebase Authentication
Actual Results	1. The system should display a list of users based on the selected user type. 2. When a user is deleted, their user data also will be removed from the Firebase Authentication
Status	Pass
Comments	Lecturers, Students and Campus Staff could no longer access the system after this operation

Chapter 7: Conclusion

In conclusion, the development and implementation of the e-student card with facial recognition system represent a significant advancement in the realm of student attendance management and identity verification. Throughout this project, I have successfully addressed the challenges faced in traditional methods of attendance tracking and student verification, offering a streamlined and efficient solution for educational institutions.

By integrating facial recognition technology into the student card system, we have not only improved the accuracy and reliability of attendance monitoring but also enhanced campus security and examination integrity. This innovative approach not only benefits administrative staff by automating tedious manual processes but also empowers students with convenient access to their exam slips and other essential information.

Furthermore, the successful execution of this project underscores the importance of leveraging emerging technologies to enhance educational processes and institutional efficiency. As we move towards a digital future, solutions like the e-student card pave the way for more seamless and secure interactions within academic environments.

In essence, the e-student card with the facial recognition system stands as a testament to our commitment to innovation and excellence in addressing real-world challenges. Moving forward, we anticipate further advancements and refinements to this system, ensuring its continued relevance and effectiveness in meeting the evolving needs of educational institutions.

References

Venugopal, A., Krishna, R. R., & U, R. V. (2021). Facial recognition system for automatic attendance tracking using an ensemble of Deep-Learning techniques. *Facial Recognition System for Automatic Attendance Tracking Using an Ensemble of Deep-Learning Techniques*. https://doi.org/10.1109/icccnt51525.2021.9580098

Anand, S., Bijlani, K., Suresh, S., & Praphul, P. (2016). Attendance monitoring in classroom using smartphone & Wi-Fi fingerprinting. *Attendance Monitoring in Classroom Using Smartphone &Amp; Wi-Fi Fingerprinting*. https://doi.org/10.1109/t4e.2016.021

Krishnan, P., & Manikuttan, A. (2022). Attendance management system using facial recognition. *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*. https://doi.org/10.1109/ic3sis54991.2022.9885693

Gómez, S., Morales, E., & Peña, F. (2023). Absence-Free Vision: an intelligent classroom attendance system with facial recognition. *Absence-Free Vision: An Intelligent Classroom Attendance System With Facial Recognition*. https://doi.org/10.1109/acirs58671.2023.10240420

Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., & Heinzelman, W. (2012). Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. *Cloud-Vision: Real-time Face Recognition Using a Mobile-cloudlet-cloud Acceleration Architecture*. https://doi.org/10.1109/iscc.2012.6249269

Rohs, M., & Gfeller, B. (2004). USING CAMERA-EQUIPPED MOBILE PHONES FOR INTERACTING WITH REAL-WORLD OBJECTS. *USING CAMERA-EQUIPPED MOBILE PHONES FOR INTERACTING WITH REAL-WORLD OBJECTS*.

Mahesh, G., Jayahari, K. R., & Bijlani, K. (2016). A smart phone integrated smart classroom. *A Smart Phone Integrated Smart Classroom*. https://doi.org/10.1109/ngmast.2016.31

Flutter (software). (2023, January 30). Wikipedia. https://en.wikipedia.org/wiki/Flutter_(software)

Thomas, G. (2019, December 12). What is Flutter and Why You Should Learn it in 2020. freeCodeCamp. https://www.freecodecamp.org/news/what-is-flutterand-why-you-should-learn-it-in-2020/

Singh, V. (2019, January 6). Firebase for Web: Cloud Firestore. Medium. https://medium.com/codingurukul/firebase-for-web-cloud-firestoreb294ed33320b