IN-BUILDING FACIAL RECOGNITION CHECK-IN SYSTEM

YONG LI JONN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : YONG LI JONN

ID No. : 19UEB02914

Date : 19 May 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"In-Building Facial Recognition Check-In System"** was prepared by **Yong Li Jonn** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	
Supervisor	:	Dr. Fatimah Audah binti Md. Zaki
Date	:	19 May 2023

Signature	:	-
Co-Supervisor	:	-
Date	:	-

ABSTRACT

In recent years, facial recognition technology has gained significant attention due to its potential applications in various domains, such as security, access control, and attendance tracking. This project focuses on the development and implementation of an In-Building Facial Recognition Check-in System for the Universiti Tunku Abdul Rahman (UTAR) to enhance the check-in process for students. The system integrates a pre-trained machine learning model for facial recognition, an Android mobile application, and a Firebase back-end database to create a seamless check-in experience. The primary objectives of the project were to develop a functional facial recognition system, create an Android mobile application, and ensure satisfactory performance through user acceptance testing. The system was successfully developed within the set deadline of April 2023 and met the necessary performance and usability requirements. The application allows users to create accounts using their UTAR email addresses, enroll their faces for identity verification, check-in to designated buildings, view their check-in records, and manage their personal information. Despite its achievements, the system has some limitations, including the lack of liveness detection, location detection, and cross-platform compatibility. To address these limitations, future developments are proposed, such as implementing liveness and location detection, integrating the system with other UTAR systems, and expanding platform compatibility. Regular updates to the machine learning model and continuous system improvement will also ensure that the facial recognition system remains accurate and effective. In conclusion, the In-Building Facial Recognition Check-in System has successfully achieved its project objectives, providing UTAR students with a convenient and efficient check-in process. By implementing the recommended improvements and maintaining a focus on continuous development, the system has the potential to significantly enhance the check-in experience at UTAR and serve as a valuable asset to the university and its community.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Service
CAGR	Compound Annual Growth Rate
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
FR	Facial Recognition
FRT	Facial Recognition Technology
IoT	Internet of Things
IT	Information Technology
LFW	Labelled Faces in the Wild
ML	Machine Learning
OTP	One-time Password
RAD	Rapid Application Development
ReLU	Rectified Linear Activation Unit
SDLC	Software Development Life Cycle

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Facial recognition enables identification or recognition of a person's identity through technology. The mapping of face characteristics on a facial recognition system is from a digital image or video frame using biometrics. It analyses the information with a database of recognised faces in order to verify a match (Kaspersky, n.d.).

In 2022, the Nippon Electric Company (NEC) described the dawn of facial recognition. The development of facial recognition took place in the 1960s. It was found that biometric recognition was feasible during those years, despite the recognition technique relying on manual marking of facial locations. In the 1980s, people saw the introduction of linear algebra as a tool for resolving facial recognition difficulties, establishing the groundwork for the technology's further advancement. The Face Recognition Technology (FERET) program, which was started by the National Institute of Standards and Technology (NIST) and the Defence Advanced Research Projects Agency (DARPA) in the early 1990s, involved the collection of a database with images of faces. This was the first time that facial recognition was used commercially. Since its inception, this technology has seen significant development. In addition, FRT is usable to recognize individuals in real-time or real-world scenarios.

The face recognition operation involves any digital photography-capable device that can capture images and collect data required to develop the faces of the people who need to be identified. Instead of using passwords, OTP codes, images, fingerprint identification, biometric face recognition, and email verification is one of the safest and most effective technologies currently accessible because it uses distinctive mathematical and dynamic patterns. Face recognition uses an incoming image as its starting point and aims to find several examples of the identical face in a database of training and testing images. It is extremely challenging to ensure that this operation can go through real-time because not every provider of biometric face recognition system or software have access to this functionality.

Depending on the situation, eID (2021) states that the facial recognition mechanism can execute out one of two variations:

- The first time a face is recognised by a FR, it is associated with an identity and registered in the database. This process is referred to as facial recognition for digital onboarding.
- The second one in which the variation where the user must first be authenticated before they may register. This process compares the data captured from the device with the current data stored in database. If the face matches an identity that has already been registered, the user is granted access to the system using his credentials.

Facial recognition algorithms are becoming increasingly complex, as well as gaining market share. Producers of facial recognition software have updated their algorithms with additional capabilities since the COVID-19 outbreak, notably recognizing faces wearing masks (Canterbury AI, 2021).

Upon a closer look, the ability to verify an identity is crucial at every stage of security. During an entry to schools and universities, a person must first confirm their identity, then only is allowed to proceed, and enters the building. A person's face is constantly present, and it is always convenient. Facial recognition at a university check-in can potentially save time that would otherwise be spent entering a record ID, scanning QR codes, and writing confirmation details on a paper record sheet.

In this project, various machine learning algorithms and deep learning techniques are being explored, with a particular focus on pre-trained models for facial recognition. Research is conducted on the development of a mobile application featuring face recognition capabilities. The main output of this project is a facial recognition check-in system, called FACEIN to verify students' identities and check in at a faster and smoother pace.

1.2 Problem Statement

Face recognition is now widely used. Facial recognition can be used for everything starting from surveillance to marketing. It all started as a feature exclusive to science fiction films. Until today, it is a technology people rely on to unlock our phones, tag pals in Facebook posts, pass-through customs, and especially in the check-in system.

The statistics only serve to highlight how all-over facial recognition is. The global market for facial recognition is anticipated to increase from $4 billion in 2017 to $7.7 billion in 2022; by 2026, it is predicted to reach $11.62 billion, representing a CAGR of almost 21%. This is so because multiple industries are using facial recognition (Shashkina, 2022).

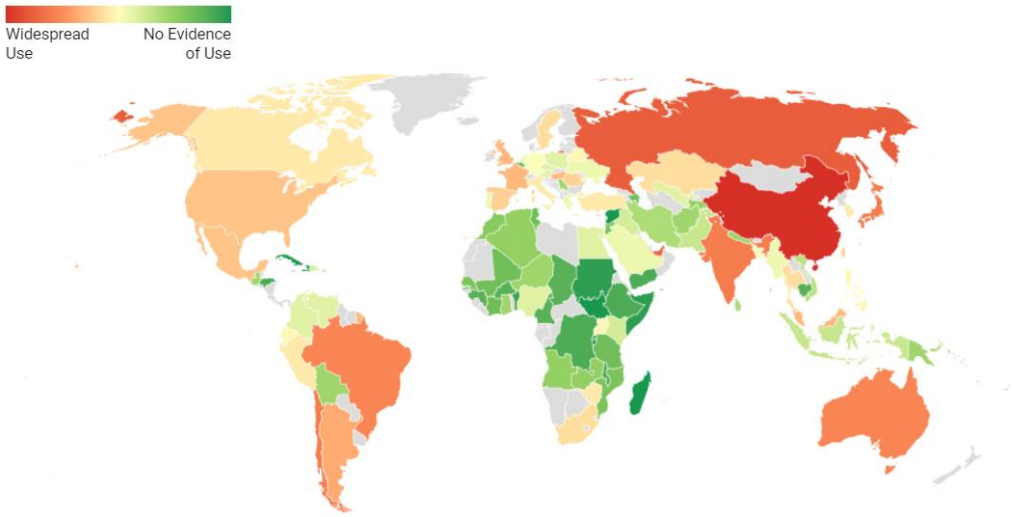


Figure 1.1: The Global Map of Facial Recognition Technologies (Bischoff, 2021 a)

Globally, seven out of ten governments shown in Figure 1. 1 make extensive use of face recognition technology (FRT), including Malaysia, among the 100 countries analyzed by Paul Bischoff and his team in 2021. Among the data analyzed for 99 countries, nearly 80% of nations use FRT in their banking and financial organizations, 70% of police departments have technology available for FR, 60% of countries use facial recognition technology in airports, and yet only about 20% of the world's countries use facial recognition technology in schools. This shows that the FR technology is used and implemented the least in universities and schools, even though facial recognition technology is being used in many facets of public life and aspects.

	Total Score	Total Score (inc. COVID)	Government	Police	Airports	Schools	Banks/Finance
Malaysia	17	19	1	2	1	2	1
Romania	19	21	1	2	1	1	1
Germany	23	25	2	2	1	3	2
Myanmar	24	29	0	1	2	5	1
Guatemala	30	35	5	2	2	5	1
Mali	34	39	2	2	5	5	5
Malawi	37	42	5	5	5	5	2
Somalia	39	44	4	5	5	5	5
Madagascar	40	45	5	5	5	5	5

Figure 1.2: Facial Recognition Use by Countries: Full Score [40] (Bischoff, 2021b)

Figure 1. 2 shows the total score of 9 countries out of 99 countries using facial recognition technology. Each country had a score out of 40; higher values indicated no or minimal use of FRT, while lower scores highlighted more pervasive and intrusive use. The country out of the nine with the lowest ranking in Malaysia. As for why Malaysia only had scored 2 points for the schools' category as compared to the other country was because, it seems like there are only a few of the schools in Malaysia that is using FRT and adapting to this technology. According to The New Straits Times (2020), Rizalman Hammim reported that a primary school in Johor, Malaysia, has become the first in the nation to use facial recognition software to track student attendance. After scanning a student's face for two seconds, the device enters their personal data, such as complete name and class number, into the school database. In Selangor, at Sunway Campus Library, the Sunway University implemented facial recognition software in January 2019. The campus building, which spans three levels and has a floor area of 72,000 square feet, accommodates more than 25,000 students. For them, smart cards will no longer be required to authenticate students' identities for library access, as facial recognition will take their place (Chan, 2019).

Most of the schools in Malaysia do not adapt to this technology because facial recognition raises the risk of data breaches leading to identity theft, cyber

stalking, and abuse. Unlike passwords and credit-card numbers, faces is hard to be replaced. In contrast to this statement, as explained before by eID (2021), instead of using passwords, OTP codes, email verification, selfies or photos, or fingerprint identification, biometric face recognition is one of the safest and most effective technologies currently accessible because it uses distinctive mathematical and dynamic patterns.

In addition, the coronavirus pandemic's lockdown has caused the worldwide country to make facial recognition software a vital tool for identifying people without direct contact. Without in-person contact is considered a protective and essential part of daily routine life against the Covid-19 disease. Therefore, especially after the outbreak of the Coronavirus, it has caused a surge in demand for countless ways of contactless technologies such as contactless payments, keeping people at a safe distance through proximity bracelets, and so do contactless check-in.

Suppose a primary school in Johor can implement and adapt to facial recognition technology without any issues and concerns. Why not University Tunku Abdul Rahman (UTAR) also adopt this facial recognition technology? By using the FRT to check in, the university can provide a safer environment to prevent the spreading of covid-19 through touchless check-in. Besides that, the university can radically simplify and streamline the event check-in procedure. It would be definitely faster than a manual check-in as well as quicker than QR/Barcode check-in.

Moreover, University Tunku Abdul Rahman (UTAR) can free up staff time and the number of staff. Staff such as security guards no longer must wait for students to fill in their information if they forgot to bring their student ID, nor waiting for a lengthy queue of students just to open their Hi-Hive application to scan a QR code. Instead, the security guard can focus more on managing student diversity and enhancing security protocol. This hands-free reality is primarily thanks to QR-code scanning, in which people scan the university’s QR code and check in the building or blocks they are in. Furthermore, generating a new QR code for each check-in can be a waste of resources, both in terms of time and materials. This can lead to long waiting times and delays for students, especially during peak hours. With facial recognition technology,

there is no need to generate a new code for each check-in, saving time and resources while providing a more efficient and streamlined check-in process.

Another problem that arises with the traditional manual check-in system is the issue of record-keeping for attendance. Keeping track of attendance manually can be tedious, time-consuming, and prone to errors. With facial recognition technology, attendance records can be automatically generated and stored in a database, making it easier to keep track of students' attendance and monitor their academic progress. Facial recognition check-in takes the traditional method of identification a step further by requiring only a face, making the process more streamlined and convenient. Facial recognition technology, when used in the right way, has the potential to significantly alter how every one of us lives and encounters the world.

1.3 Project Solution

Since most people do not fully understand how facial recognition works, Facial recognition systems take a two-dimensional or three-dimensional computer-generated filter image and transform it into numerical expressions so that the similarity between them can be assessed. These filters are typically produced using deep learning, which processes data using neural networks (Lewis, 2021). Since they instantly compare the crucial data from the incoming image signal with the digital image stored in a database, these are significantly more stable and reliable than the data contained in a static image. The biometric FR process needs a connection to the internet because the database is maintained on servers rather than being on the capture equipment. This in-person comparison ensures that the biometric information matches the user of the service or the one requesting access to a system, application, a building by statistically analyzing the arriving image with no room for error.

The use of deep learning and machine learning algorithm allows FRT to process with the highest levels of reliability and dependability. Similar to that, by utilising a variety of algorithms and computational approaches, the process can be finished instantly and effectively. To automate the facial recognition check-in system, deep learning in the form of a Convolutional Neural Network (CNN) algorithm has been applied. A Deep Neural Network (DNN) variant known as a CNN is explicitly designed for complex tasks like image processing,

which is necessary for facial recognition. A CNN comprises several layers of interconnected neurons, including an input layer, an output layer, and layers in between (Sightcorp, n.d.).

By observing and applying the CNN deep learning algorithm, the check-in system can analyze the image captured as a collection of pixels. The values included in these pixels are multiplied once scanned as matrices, and the output is passed into the following layer. The network computes an output in the form of an array when the process reaches the output layer after proceeding through each convolutional layer. The array is referred to as a faceprint. A database of faceprints or another faceprint can be used to compare the calculated faceprint (1:1 matching) to validate whether there is a match (1:N matching). According to the selected confidence levels, two or more faceprints will be deemed to match if they are identically similar to one another.

1.3.1 Technologies

This project, FACEIN will be implemented using technologies that are primarily focused on Android development, ensuring scalability and cost-effectiveness. The selection of technologies and project implementation plan are based on thorough research and independent development, without prior discussions with any external entities.

i. Android with React-Native (JavaScript)

- Android is one of the most common operating systems among the worldwide population. React-Native, a popular framework utilizing JavaScript, is utilized for the development of a native-like experience specifically for Android devices in this project.

ii. Firebase

- Firebase, a comprehensive app development platform backed by Google, offers various backend services and tools to build, improve, and grow applications. In this project, Firebase's services such as Firestore, authentication, and cloud storage will be utilized to streamline app development and enhance overall performance.

iii. Firestore

- Firestore is a NoSQL cloud database provided by Firebase to store and sync data for real-time applications. It enables seamless data synchronization, offline support, and efficient querying, enhancing the application's performance and functionality.

iv. Amazon Web Services (AWS)

- AWS is a comprehensive cloud computing platform offering a wide range of services. Leveraging AWS infrastructure ensures scalability, reliability, and security for the application.

v. Amazon Rekognition

- Amazon Rekognition is a deep learning-based image and video analysis service. It will be utilized for facial recognition in the application, allowing for swift and accurate identity verification of students during the check-in process.

vi. Amazon S3 Bucket

- Amazon S3 (Simple Storage Service) is a highly scalable, secure, and durable object storage service provided by AWS. It will be used to store and manage images, videos, and other data generated by the application, ensuring efficient data retrieval, and reducing the load on the application server.

vii. React-Native Camera

- React-Native Camera is a library used to access device cameras for capturing images and videos. It will be employed to enable seamless integration of camera functionality within the application for facial recognition purposes.

1.3.2 Technical Aspects

The technical aspects of the problem in developing the In-Building Facial Recognition Check-in System (FACEIN) can be summarized as follows:

1. Facial Recognition Model: The development and integration of a pre-trained machine learning model for facial recognition is a key technical aspect of the project. The model should be able to **accurately identify and verify individuals based on their facial features**.
2. Mobile Application Development: The creation of an Android mobile application is another important technical aspect. The application should provide a **user-friendly interface** for users to interact with the facial recognition system. It should allow users to create accounts, **enroll their faces**, **check-in to buildings**, view check-in records, and manage personal information.
3. Database Integration: The system requires a back-end database to store user information, enrollment data, and check-in records. Integration with a Firebase database is necessary to ensure secure and efficient data storage and retrieval.
4. System Performance: The system should meet performance requirements to ensure smooth and efficient check-in experiences. This includes fast and accurate facial recognition, minimal latency in processing, and scalability to handle a large number of users.
5. User Acceptance Testing: User acceptance testing is crucial to evaluate the system's usability and gather feedback from users. This aspect involves conducting tests and surveys to assess user satisfaction, identify potential issues, and make improvements accordingly.

1.3.3 Project Implementation Plan

The following plan outlines the steps and structure to achieve the project objectives in section 1.5.2.

To achieve objective 1 in section 1.5.2, applying a pre-trained machine learning model, a thorough investigation of existing pre-trained machine learning models for facial recognition will be conducted. The chosen model should offer accurate and efficient identity verification. This task will involve researching the most suitable model into the application. The duration of this activity is estimated to take 3 weeks, starting from week 1 to week 7 of the project I period.

This task is measured by the completion of at least 2 comparisons of the chosen pre-trained model into the facial recognition system, ensuring accurate and efficient identity verification of students during the check-in process. Experiment will also be conducted to test the accuracy of the facial analysis, and similarity of the face comparison of the pre-trained model.

To achieve objective 2 in section 1.5.2, developing a mobile application for the android platform. An Android application will be developed using React-Native (JavaScript) framework. The application will feature camera functionality for facial detection and a secure connection to the back-end system (UTAR database) with the approval received. This task is expected to be completed from week 4 to week 10 of the project II period.

The completion of this task is measured by the successful development of the Android application, which includes the implementation of facial detection and a secure connection to the back-end system.

To achieve objective 3 in section 1.5.2, conducting User Acceptance Test (UAT). A User Acceptance Test (UAT) will be conducted to evaluate the In-Building Facial Recognition Check-in system. The test will involve selected UTAR students and lecturer who will use the developed application in a real-world environment. Feedback will be collected to identify any issues, improvements, or additional features that may be necessary. This task is expected to be carried out from week 11 to week 14 of the project II period.

The completion of this task is measured by the successful execution of the User Acceptance Test and the collection of valuable feedback from

participants. The gathered feedback will be used to refine the application and ensure it meets the needs of UTAR students and staff.

By following this project implementation plan, all objectives specified in section 1.5.2 can be achieved, resulting in the successful development of the In-Building Facial Recognition Check-in System for UTAR students and community.

1.4 Project Approach

Rapid Application Development, frequently known as RAD, is a technique for creating software that relies on prototyping without any prior design. Less emphasis is placed on planning in the RAD paradigm, and more emphasis is placed on development tasks. It seeks to develop systems and software in a short amount of time (Martin, 2022b). In this project, the RAD approach is adopted for the mobile application development process because this model allows quick iteration and is adaptable to changes. The RAD methodology is suitable to be used as the time required to develop the mobile application along with the facial recognition check-in system is a short span on time in 2 – 3 months. Additionally, because each software component is separated in the RAD approach, it is simple to modify each one independently as the software needs changes. Thus, mobile applications and systems' features, code, and UI can be modified as required in response to customer or user requests.

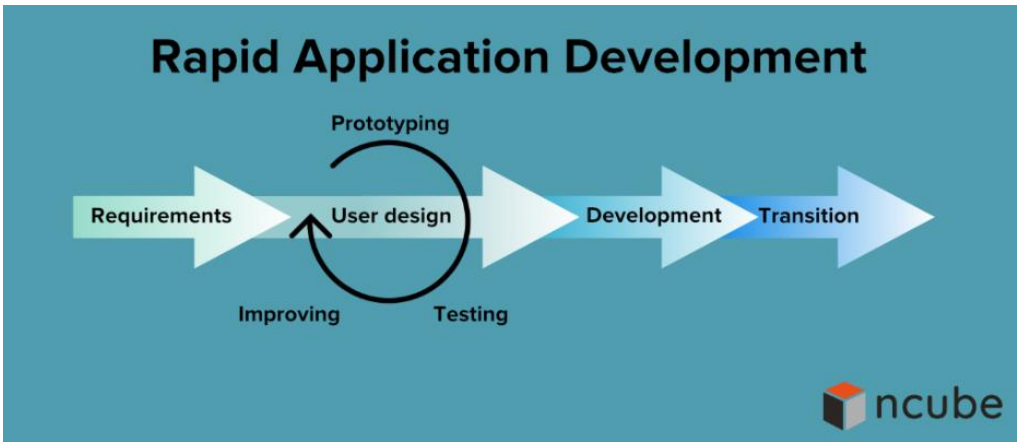


Figure 1.3: Phases in Rapid Application Development (Demchenko, 2020)

Figure 1.3 shows the 4 main phases in the RAD approach. In the first phase, which is the requirement phase, developers, users, and relevant members

plan and agree on the project's requirements, criteria, and limitations. The requirements are less strict and specific in the RAD approach. Upon agreement from all parties on the project's general scope, prototyping will be the next step. The user design phase is the second phase in the RAD process. In this phase, users and developers collaborate to design and build one or more prototypes that satisfy the listed system requirements. Users interact with the prototype throughout this ongoing phase and update feedback up until a real final product is approved. The development phase is the third phase of the RAD approach. Coding and testing are two tasks that are primarily completed during the user design phase. During the construction phase, developers strive to deliver a functional product. Because consumers actively provide feedback during the previous step, the final version is developed faster. Client comments are likewise encouraged during this phase, just like they were during the previous one. The final phase of the RAD approach is the transition phase. The last phase allows the development team the opportunity to bring the components into a real-world production setting where the required thorough testing or user training may take place.

1.5 Aims and Objectives

1.5.1 Project Aims

The project aims to implement the In-Building Facial Recognition Check-in System to help UTAR students to verify their identity and check-in to the university in an easy, quick, and smooth process.

1.5.2 Project Objectives

The objectives that drive the development of this project according to the SMART requirements are shown below:

1. To Develop a **facial recognition system** using a **pre-trained machine learning model** to achieve **at least 95% accuracy** in identifying and verifying individuals based on their facial features.
2. To Create an Android mobile application that allows users to create accounts, **enroll their faces**, **check-in to designated buildings**, view

check-in records, and manage personal information, ensuring a user-friendly and intuitive interface.

3. To Integrate a Firebase back-end database to securely store user information, enrollment data, and check-in records, with the ability to efficiently retrieve and manage data.
4. To Conduct user acceptance testing with a sample group of at least 15 UTAR community that involves (students, lecture, or staff) to gather feedback, assess user satisfaction, and make necessary improvements based on the findings.

1.6 Scope of Project

This project implements a "touchless check-in" or facial recognition check-in system, with two main modules in the mobile-based application. First, the application allows users to sign up and enroll face to complete profile setup. Second, the application requires users to sign in only using the UTAR email account.

Besides this project consists of 4 main modules for identifying UTAR students to check-in successfully. First, users have to open the application and scan their face using their phone’s camera and wait for facial recognition system to process. Second, the system can then use computer vision and deep learning to find a prospective face within its stream. Third, the system will verify whether this detected face matches the face enrolled within the application's database. Fourth, once the face matches following the machine learning algorithm, the system will display the general information such as name, student ID, faculty, and course of that user. Thus, the user's request to check-in and enter the building is granted.

The front end of the project (the application) will be developed using react-native, based on JavaScript, with a focus on Android. On the project's backend, the Firebase, a development platform, is used for user authentication, and to store and synchronize data in real-time between multiple devices and platforms. AWS Rekognition, a cloud-based service provided by Amazon Web Services (AWS) is used for advanced image processing, analysis, feature

extraction and machine learning algorithms to build facial recognition features and connects to the database.

1.6.1 Users of the System

The primary target audience for this application is UTAR students, with UTAR faculty and staff also being potential users Users can use this application to to verify their identity and check-in to the university in an easy, quick, and smooth process.

1.6.2 Scope Covered

While facial recognition technology has a wide range of fully functional features, this project focuses on only a select few that are relevant to the specific needs of the application. As a result, some aspects of the technology will not be covered in this project.

The application of the project contains various functionalities such as:

i. Users can create an account using their UTAR email address through the application's sign-up function, which includes email validation.
ii. Users can update their personal information such as name, student ID, course, and profile picture.
iii. Users can enroll their face to their account for identity verification.
iv. Users can remove the enrolled face from their account.
v. Users can check-in into the building they desired by selecting the building’s name and verify their faces.
vi. Users can view their check-in records and track their attendance history.
vii. Users will receive an overlay popup upon after each successful check-in, displaying their personal information, date, and time.
viii. Users can log out of the application for security and privacy purposes, or to switch to a different account.
ix. Users can delete their account and re-register using the same email address.
x. Users can choose to switch between light and dark themes in the application with a more personalized and customizable experience.

1.6.3 Scope Not Covered

The following aspects are not covered in this project:

i. It is limited to real-time facial recognition using the device's camera. The application does not include the functionality to differentiate between a live face and a face in a still image, which may be an area of future development.

ii. It does not include location detection functionality, which indicates that the application is not able to determine if the user's device is located in the selected building or area during the check-in process. As a result, the check-in will not fail based on the device's location. It also may be an area of future development.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides an in-depth exploration of facial recognition as a biometric identifier, highlighting the advancements and various applications of this emerging technology. In order to develop a comprehensive understanding, a range of related works discussing the trends, security implications, and wide-ranging applications of facial recognition are scrutinized. As a critical component of this discourse, the official websites of leading AI-Powered Facial Recognition and Computer Vision APIs (AWS Rekognition API, Microsoft Azure Face API, and Google Cloud Vision API), are reviewed and compared.

To appreciate the best practices for implementing these APIs, the Rapid Application Development methodology and Waterfall model are examined thoroughly, drawing insights from several research studies. Additionally, a literature matrix summary is developed, summarizing key findings, and offering comparative analysis. Through this detailed review, the chapter seeks to present a nuanced understanding of facial recognition technology, its practical applications, and associated computational interfaces, illuminating the pathway for subsequent integration into the project at hand.

2.2 Facial Recognition as Biometric Identifier

Biometric technology and solutions provide governments, private companies, and society throughout the world to fight crime, prevent fraud, secure national borders, and safeguard identities for a variety of additional uses, including access control and background checks on employees and applicants. Facial recognition is one of the few biometric technologies that has sparked our imagination. In 2022, its arrival has also generated significant concerns and surprising responses.

'Facial Recognition, is a technology that makes it possible for a computer to recognize a digital image of someone's face (Cambridge Dictionary, n.d.)'. Opinions on facial recognition vary widely. If people were to take the news and read it at its value, they would assume that everyone detests and despises facial

recognition technology. There is no way that this is the truth. Facial recognition is not without its critics, much like most technologies. The biggest reason for hostility and mistrust is the worry that facial recognition technology will violate people's privacy. Many people have jumped on the facial recognition bandwagon because of false information in the news, fearing that once people's face has been verified and identified by FRT, hackers will easily break into the database, steal their identity, and they will never be able to retrieve it back. People's faces, fingerprints, and iris are always with us, making them more practical.

Simply said, this is untrue. Recognizing or confirming a person's identification by their face, known as facial recognition, is one of the most secure methods to authenticate one's identity. Of all biometric measurements, facial recognition is currently regarded as being the most natural (NEC, 2021). And with a good reason, identity authentication by looking at faces rather than fingerprints or irises is considerably less likely to be stolen and much safer. Hence, the most popular biometric technology is still facial recognition. This is due to how simple it is to deploy and implement as the end user is not physically interacted with. Additionally, the processes for face detection and face match are swift for identification and verification purposes.

2.2.1 Advancements in Facial Recognition Technology and Applications

According to research and analysis by Thales (2021), There are several projects company's projects competing for first place in the biometric innovation race. Academia, Facebook, Google, Microsoft, and Amazon are significant players in the mix. To advance our understanding as efficiently as possible, theoretical developments in artificial intelligence, machine learning, image processing, computer vision, and facial recognition are now routinely disclosed by all the software web goliaths. A closer examination of the biometric innovation race is as below:

- **Academia**. Compared to the human performance of 97.53%, the GaussianFace algorithm created in 2014 by researchers at The Chinese University of Hong Kong obtained facial identification scores of 98.52%. An excellent rating despite flaws in the amount of RAM needed and calculating speed.

- **Facebook**. Facebook revealed its DeepFace software in 2014, which has a 97.25% accuracy rate for identifying whether two images of a face belong to the same person. Humans perform just 0.28% better than the Facebook computer on the identical test, answering correctly in 97.53% of cases.
- **Google**. With FaceNet, Google improved in June 2015. FaceNet set a new benchmark for accuracy on the broadly used Labelled Faces in the Wild (LFW) dataset, achieving 99.63% (0.9963 0.0009).
- **Microsoft**. In contrast to lighter-skinned men, technologies from Microsoft, IBM, and China's Megvii (FACE++) exhibited higher rates of error in recognizing women with darker cutaneous tones, according to research conducted by Massachusetts Institute of Technology researchers in February 2018. Microsoft published a blog post at the end of June 2018 stating that its biassed facial recognition algorithm had been greatly enhanced.
- **Amazon**. According to a May 2018 article by Ars Technica, Amazon is actively advertising Rekognition, a cloud-based image analysis technology, to law enforcement and the intelligence community. The technology can do face matches across database systems with tens of millions of faces and can identify up to 100 persons in a single image.

Developing countries like Malaysia are currently in rising demand for facial recognition technology despite constant criticism over the morality of using the system. The facial recognition system now is currently being embraced across devices for a variety of use cases since the technology initially became widely used on mobile phones for identification processes. In fact, according to Statista, on statistics facial recognition market size worldwide, it is predicted that by 2025, the industry for facial recognition systems will be worth US$8.5 billion. An increased interest in the surveillance industry, notably by the government and sectors responsible for defense projects, is one of the main drivers fuelling the development of the technology. Despite some systemic shortcomings, in the beginning, technology keeps evolving since improvements in AI algorithms are enhancing the recognition system. Face recognition technology has already been used in several businesses across Southeast Asia to

increase user convenience and security. For instance, people can access many government services using their faces in Singapore. When using Singapore's extensive SingPass digital identity scheme, the nation expressly permits users to connect to their accounts using facial recognition (The Sumsuber, 2021). At the Kuala Lumpur International Airport in Malaysia, facial recognition is gradually taking the place of paper tickets and boarding permits as a speedier and safer method of passenger authentication (Malaysia Airports, 2021). Furthermore, in Malaysia, the government installed a new facial recognition system from an American multinational IT company to help fight crime by identifying criminals' faces or those of those wanted by the police will be controlled from the CCTV command center at the Penang police headquarters in the state of Penang (Alita, 2019). The technology enables governments, sectors, and security services to respond rapidly and confidently and allows corporate corporations access to more outstanding customer data through algorithms.

2.3 Related Work on Facial Recognition Trends and Application

Facial recognition technology is rapidly evolving, and more new ideas and concepts spring out with each subsequent development. Global initiatives at digital transformation and the biometrics business increasingly focus on facial recognition. The rather diverse facial recognition market is predicted to expand in several directions faster than anticipated. The use of facial recognition is growing in various use cases, notably current models, as investments in the field rise and the technology matures. The COVID-19 outbreak has been demonstrated to be a key factor in this. Along with other biometric technologies, facial recognition systems have been developed in response to the COVID-19 pandemic (Norstorm, 2021). FRT will undoubtedly become more prevalent due to the pandemic, as it is anticipated that digitization and digitalization will grow in many industries. The expanding facial recognition market is a sheer economic boon for the technology sector. The technologies that enable face verification from cameras with saved image data in databases and sophisticated FRT are frequently highlighted, which are augmented by many more technologies. These technologies include AI, ML, DL, and computer vision.

In fact, the field of biometrics where numerous technologies converge is facial recognition. It will become even more pervasive than it now is attributable to edge-cutting technologies, 5G mobile networks, and other developments. National security is likely the first crucial component where the 5th generation mobile network and the IoT have a substantial influence. This is because extensive networks of artificial intelligence-supported surveillance cameras can be deployed as God's eye to watch everything. Alternatively, to be more precise, the installation of dense networks of artificial intelligence-enabled surveillance cameras.

2.3.1 Facial Recognition in Security and Beyond

Facial recognition technology (FR), along with the Internet of Things (IoT), 5G, and edge security, plays a significant role in various domains, notably national security, vital infrastructure protection, and the development of smart cities. As highlighted in numerous discussions on building systems integration, these technologies are being increasingly employed across a range of facilities - from smart buildings and airports to data centers. Martin Feder, a specialist in building management systems, predicts a future where audio-visual communication systems will further incorporate camera technology with facial recognition. He also anticipates that the blend of surveillance and facial recognition will augment the security of edge databases emerging under 5G towers, leading to continued growth and new use cases for these technologies (i-Scoop, n.d.).

Upon knowing about facial recognition, many people instantly envision things like airports, border controls, security cameras to monitor, and other related applications; however, FRT are also explored in retail facilities and building not limited to security and is even used for self-check-in and self-checkout. Facial recognition technology is also being used widely in social platform, for online marketing, in health screening, during government elections, to gain access to certain locations, and for criminal investigations, and people believe that it will be crucial for the time ahead of mobile payment and mobile banking. Nowadays, people can download and utilize various facial recognition software and applications from the App Store and Play Store. Facial recognition technology also significantly impacts society on many different levels. The

endless applications range from protecting private data to avoiding fraud, cyberattacks, and unlawful arrests. According to Shah Anas from the TekRevol Team (2022), the Top 5 Facial Recognition Application is as shown below:

- **FaceFirst**. In the face recognition app field, FaceFirst is a well-liked option in police enforcement and the military, helping to match photos of individuals nearly instantaneously in the field against their databases.
- **Blippar**. Blippar is a facial and item detection application for augmented reality. Blippar has the capacity to recognize things like flora, buildings, animals, and food items using its extensive database.
- **Face2Gene**. Face2Gene is a ground-breaking healthcare application that enrolls face recognition technology to assist doctors in clinics and hospitals in making bioinformatic diagnoses of patients with specific genetic illnesses and their variants.
- **Face Phi**. One issue that banks must address to protect their consumers is identity theft and fraud, whether through mobile or internet banking. To address this issue, banks worldwide employ Face Phi's facial recognition technology to confirm the identification of customers who use online banking.
- **LogMe Facial Recognition**. LogMe Facial Recognition is a search engine application for facial recognition. It distinguishes between the faces in a picture using resemblance and distance.

The demand for FR software products and services is the other one that is expanding. However, it is evident that there will be regional differences in both development and suppliers. There is much controversy around the use of face recognition technology for law enforcement, forensic videotape investigations, cross-border checking, public security, and criminal identification and verification. One example is the media attention given to the Clearview AI scandal, which concerns an allegedly enormous AI image collection that was created in questionable methods by scraping social media platforms and open websites and is used extensively by US law enforcement and authorities. Additionally, the business appeared to have agents and private entities in nations other than the United States and Canada. The use of face recognition for mass monitoring in some regions of the world is a topic for other discussions.

Furthermore, there are concerns about prejudice and misuse of facial recognition systems (Klosowski, 2020).

2.4 Rapid Application Development and Waterfall Model

Implementing a facial recognition application/system is time-consuming and more complex than people might imagine. On the surface, users will have to stand in front of a camera and wait for their faces to be scanned and verified. In just a matter of seconds, once an individual's identity has been confirmed, they can pay with their face or check in and out of a building. Deep down, only the developers understand the struggle to build a successful application/system that satisfies the user requirements and system requirements.

Three actions must be carried out for the facial recognition software to operate. It must first find a face. Next, it must nearly instantly recognize the face. Finally, it must perform any additional actions required, like providing access, if necessary, to a verified user. In addition to building facial recognition software that can scan and match faces in a faster process, the software requires a trained model by deep learning (Maksymenko, 2019).

Behind every successful man, there is a supportive woman. It is the same goes for the software. Behind every successful software, there is a good software model. The creation of software benefits significantly from the usage of software models. Through the software model, it is possible to evaluate the software's overall complexity and estimate the software's strategic plan. The sequential Waterfall Model separates software development into pre-established stages. There cannot be any overlap between stages; one must be finished before the next may start. Each stage is created to carry out a certain task throughout the SDLC phase (Martin, 2022a). A prototyping-based software development method without any predetermined planning is known as rapid application development. Less emphasis is placed on planning in the RAD paradigm, and more emphasis is placed on the development tasks. It aims to create software in a brief amount of time (Martin, 2022b).

Considering the waterfall and RAD model, evaluating the project's nature and the team's developer structure is important. According to Kissflow (2022), the first key question to consider is the size or workload of the project. RAD emphasizes small, quickly launched projects with tangible results. Some

projects are too complex and interconnected to be divided successfully. There is no rule of thumb for how small a project must be to use RAD, but the bigger it gets, the harder it is to use. If a small project is to be carried out, the waterfall method can be used to go through rapidly. If the criteria are not too difficult, document them all at once and be done with them. The second key question is the requirement of the prototype in a project. The prototype allows the stakeholders to generate a better set of requirements through an iterative process. Many projects may not need a prototype. When implementing batch processes, a prototype may not be useful. These projects should stick with the waterfall model. Web development and mobile development projects lend themselves to RAD due to their visual design and ability to reuse numerous components. The third key question is the flexibility of the team. When everything needs to be recorded, and all suggested modifications need to go through scope-change management, the waterfall method works well. By its very nature, RAD calls for a high level of adaptability and the capacity to manage change. For instance, an initial prototype can spark the kind of debate that necessitates starting the following iteration from scratch. Teams or organizations who struggle to adapt to change should not employ RAD. The last key question is the expertise of the team. There may be less needed to use the conventional waterfall model as stakeholders, project leaders, and developers gain more expertise. The team can switch to a quicker and more adaptive development method, such as using the RAD model. However, using a traditional development model may be preferable if the team is relatively inexperienced.

Table 2.1: Comparison of SDLC Models: Waterfall vs RAD Model

Key	Waterfall Model	RAD model
Name	Waterfall Model, or Traditional Model	Rapid Application Development Model (RAD), or Iterative Model
Risk	High	Low
Cost	Low	Low
Developer Team Size	Can be small to large team	Requires small team
Changes	Any changes should be made in the earlier phases of development because fixing in later phases would be high cost	Flexible to make changes in any phase.
Product	delivers product after completion of the software development cycle.	Gives earlier deliveries and seeks feedback to update the software as needed.
Waiting Time	long waiting time for running software in waterfall model.	less waiting time for running software in RAD model, as soon as first iteration is complete.

The table 2.1 illustrates the difference between the waterfall model, and RAD model of SDLC model. It allows comparisons of the risk, cost, team size, changes, product, and waiting time among the two models.

2.5 Summary

Table 2.2: Literature Review Study Summaries of Facial Recognition, Check-in System, OpenCV, Deep Learning

No	Citation Author (year)	Purpose, rationale, research question	Major findings, contributions	Research methods, sample, variables	Strengths/ limitations	Significance/ Implication for practice/ research	Make note on how this research is linked to other studies reviewed
Theme/ Area of Studied: Implementation of Facial Recognition as Biometric Authentication							
Study 1	Chen et al. (2020)	To compare the **speed and accuracy of face detection, AI face recognition, and face search** services of Face++ and Baidu	- realized the mobile check-in attendance that can be used for college students or training institutions. - Baidu AI's face detection and recognition performance are better; its speed is more than twice faster	This experiment compares face detection time before system optimization in two groups. Face detection time was measured using 50 images in single and another 50 two-face images	- small sample size (50) is not ample to provide accurate data and reduces power of study -real faces might not be detected upon successful project implementation as images and pictures	Adaboost face detection algorithm based on OpenCV was studied to enables real-time detection of faces	Face recognition focuses on the **deep learning and neural network** technologies that are rapidly developing, such as the genetically optimised GRNN neural network (Rusell, and Fischaber, 2013)

			271.2ms) was much higher than the commercial face detection algorithm (about 25ms)	collected in the mobile phone or pad APP			
Study 2	Petrescu. (2019)	To emphasize on the **use of facial recognition as a Biometric Application** and not an unethical spying tool	-Facial recognition system is described as a Biometric Artificial Intelligence-based application that can uniquely identify a person by analysing patterns based on the person's facial textures and shape -facial recognition is used as a biometric	-Research on recent statistic (2017 – 2019) that the number of people who benefited from the emergency assistance of facial recognition due to armed attacks	-the study only emphasizes on positivity of using facial recognition, but did not reveal the dark side of facial recognition - false detection and identification of a person with wrong identity may result	facial recognition can help protect the public and improve national security on multiple fronts when used correctly and proportionally	Support the use of facial recognition in terms of security as well as **speeding up processes such as check-in** not limited to public places but also private places such as schools, companies, and etc. (Petrescu, 2019)

			passport which allows travellers significantly reduced waiting times for passport control and increase security in, and around airports - Well-designed facial recognition systems installed at airports, multiplexes, and other public places can identify people among fleets and save the faces of suspects, gangs, wanted criminals and those suspected of involvement in serious violent crimes.		in violence to innocents -statistic may not be accurate because it only studied statistic over a short time (few years), new issues may arise for a longer period.		

Study 3	Yang, and Han. (2020)	To design a **face recognition attendance system** based on real-time video processing	- the accuracy rate of the video face recognition system is up to 82%. Compared with the traditional check-in method face recognition attendance system can be reduced by about 60% -The face recognition time and attendance system with real-time video procession are able to 1) quickly complete the task of students in time and	By investigating two colleges (A and B) based on face recognition attendance system accuracy, and selected 200 college students who need to punch cards under time interval of 2 hours	- Data collection is time consuming -Data collected within a day is not ample for research study -current human face and photo input is allowed to check-in. Thus, misuse may still occur.	The face recognition time attendance system is more stable and correctly identify check-ins, and the rate of skipping classes is significantly reduced compared with the control group (traditional fingerprint check-in, punch card, attendance sheet)	The video image recognition system is mainly **composed of four parts**: [login module], [recognition module], [check-in module] and [background management module]. (Best-Rowden, and Jain, 2018)

			attendance check-in system, 2) get rid of the complex naming phenomenon, 3) greatly improve the efficiency of class, 4) play an important role in guiding the development of the time and attendance system				
Study 4	Zhang. (2021)	Reveals how perceived security, privacy, and trust, as well as previous experience, are key to encouraging hotel users to **adopt the facial recognition**	- intelligent check-in can reduce check-in time by two-thirds, massively improving efficiency at hotel receptions -To earn trust, both security and privacy is needed to secure it.	By examining the positive and negative experiences of over 300 real hotel guests with facial recognition systems.	-sample size is large and sufficient to represent a community. - study considers users' prior experiences of technology adoption, expanding	opens a new area of research in the use of facial recognition technology in the hotel sector, providing empirical evidence of user experience. When perceptions of privacy were greater than	Findings in perceived **security and privacy are critical to building users' trust**; positive prior experience increases the chance of adoption of facial recognition technology (Xu et al., 2021)

		technology at the check-in desk.	Thus, the definition of security, privacy, and trust is as follows: security is the protection of data to prevent destruction or unauthorized access; privacy is the way in which a customer's private data is collected, stored, and used; trust is the set of beliefs held by consumers about a service supplier		the understanding of how prior adoption affects current decisions	security, trust in the technology was increased, but when concerns related to privacy were greater than the security benefits, trust was lessened significantly.	
Theme/ Area of Studied: A Model Based on Deep Learning Using Face Recognition							

Study 5	Ríos-Sánchez et al. (2019)	Provide a fair comparison between the two **deep learning models for facial recognition** (FaceNet and OpenFace) but also to measure to what extent a progressive reduction of the model size influences the obtained results	- the influence of the feature extraction model become more evident as the complexity of the image's increases - a big and well-trained model is required to deal with complicated scenarios that present high variability between images used to enrol users into the system and posterior accesses - resource saving is a priority; smaller models are viable if they are trained with a sufficiently big dataset containing	- used well-defined evaluation protocol and a great number of varied databases, public and private. - Organize database by inserting new accesses into the already configured system using the test samples.	- methods were validated and the acceptance threshold was adjusted - calculation of more realistic performance rates	Research ensures fair comparison between two model using evaluation protocol suggested by the ISO/IEC 19795 standard consisting of three parts: i) Dataset Organisation, ii) Computation of Scores, and iii) Metrics Calculation	Evaluated Facial Recognition Technology on Single Sample Per Person (SSSP) model based on **TensorFlow, deep learning (Euclidean distance), and convolutional neural networks (CNNs).**

			enough representativeness.				
Study 6	Salam et al. (2020)	To develop a complete **Facial Recognition system using transfer learning in fog computing and cloud computing**	- system can be trained to recognize a set of people and to learn via an online method - proposed DCNN has superiority over other machine learning algorithms, according to Accuracy parameters. - The cloud server has a greater storage capacity than fog nodes/servers.	Conduct sets of experiment with performance analysis and used publicly available database to evaluate the proposed system	- In the proposed framework, issues are well-handled and considered - Performance is measured under precise measurement that can be used in wide-range comparison of the essential individual classifiers and the proposed system.	Suggest improving proposed algorithm by comparing proposed algorithm with different metaheuristic algorithm and apply proposed algorithm to real-life facial recognition problem in a specific domain in the future	Study is based on the adaptive version of the **most recent DCNN algorithm, called Alex-Net**. The proposed DCNN algorithm is based on a set of steps to process the face images to obtain the distinctive features of the face. (Power et al., 2018)

			therefore, the cloud server can store many training sets and process these sets				
Study 7	Zulfiqar et al. (2019)	Present a **convolutional neural network-based face recognition system** which detects faces in an input image using Viola Jones face detector and automatically extracts facial features from detected	- obtained a promising experimental result, with an overall accuracy of 98.76%, which depict the effectiveness of deep face recognition in automated biometric authentication systems. - proposed system can be used in a wide variety of applications including content-	-Create a large database of facial images (9000) of subjects which is augmented to increase the number of images per subject and to incorporate different illumination and noise conditions for optimal training of the	-comprehensive experimental evaluations are obtained through image transformations and brightness variations are incorporation. -The batch normalization layer scales and adjust the activations	- Automated and quick recognition of authorized persons in a restricted area using the proposed system can ensure hustle-free and secure access	Findings review **Squeezenet was determined as the most suitable network** compared to (Alexnet, VGG16, Resnet18, and Resnet50) for face recognition in terms of computational cost and accuracy. (Iandola et al., 2016)

		faces using a pre-trained CNN for recognition	based data retrieval, web search by image, surveillance, criminal identification, automated attendance systems and auto-enforcement of restricted access to certain areas	convolutional neural network - selected an optimal pretrained CNN model along with a set of hyperparameters experimentally for deep face recognition	-speeds up learning and reduce over-fitting -CNN training is time consuming and exhaustive process		
Theme/ Area of Studied: TensorFlow/OpenCV for Face Detection and Recognition							
Study 8	Khan et al. (2019)	Proposes the **PCA (Principal Component Analysis) facial recognition system**	- reduce the large amount of data storage to the size of the feature space that is required to represent the data economically	build a camera-based real-time face recognition system and set an algorithm by developing programming on	- face pictures with a simple data vector can be described. - positive and negative	-Suggest creating network from data as an output instead of resulting image projection into face space and learn face projection to improve	The computer-**View library for Intel's open-source** makes programming easy to use. This provides advanced capabilities

			- Karhunen-Loeve is the most effective technique for the identification, detection and compression of an image, the principal component analysis (PCA)	OpenCV, Haar Cascade, Eigenface, Fisher Face, LBPH, and Python.	representations are used to construct a cascade function	accuracy of neural network classifier.	such as facial detection, face tracking, facial recognition, and a range of ready-to-use methods for artificial intelligence (AI). It also supports Windows, and MacOS. (O'Reilly Publication, n. d.)
Study 9	Kushal et al. (2020)	Recurring to **deep learning with the use of Convolution Neural Networks** on detecting and recognizing objects concerning the illumination and the viewpoint of the object faced by computer vision.	The research shows that 1)The system was able to detect the presence of an ID card in the image. 2)It was able to detect the presence of faces and provide the co-ordinates. 3) Problem areas were moving object and object of the same	-conduct studies by identifies the presence of a person wearing an ID card using tensor flow object detection API, detects and recognizes the face using Haar Cascade method of OpenCV.	-The system will recognize person with an ID card regardless of the ID card type. -used a large database but does not have a large dataset to conduct experiment	Described a template matching based face recognition for dynamic faces which has either horizontal or vertical movements. An algorithm is used to identify the face by distinguishing the skin region and the non-skin region. It is detected using neutral	Suggested **OpenCV Harr Cascade method** to be used to perform the concept of Cascade of Classifiers. The system is trained with a set of images which are to be detected and with a set of images which are not to be detected and to generate an XML file. (Jalled, and Voronkov, 2016)

			size. 4) Face recognition using LBPH is a tremendous success.			position and rotating position.	
Study 10	Vadlapati, Velan, and Varghese. (2021)	Profound scientific use of **computer technology applied in the fields of AI and Machine Learning** primarily focused on Image Processing and Pattern recognition	- able to train the model to recognize people while wearing masks - can increase the security as well as efficiency whilst making the recognition faster	- feed the model with the reference images, which are of people without mask and images taken at different angles	- restricts or limits capability to use large number of sample pictures to study the features -face detection is based on making an accurate guess and not with scientific and mathematically proven with accurate result.	Python Image Library is immensely powerful library used to support various formats for images like JPG, PPM, PNG and able to detect and recognize faces of people with mask	Support the use of **Python to implement FR** as is a dynamic, object-oriented, high-level programming language. Its high-level built-in data structures, dynamic typing, and dynamic binding make it appealing for Rapid Application Development and scripting. (Vadlapati, Velan, and Varghese, 2021)

2.6 AI-Powered Facial Recognition and Computer Vision APIs

Computer vision APIs are software interfaces that offer image processing and recognition services to other software programs. They are designed to make it easier and faster for developers to integrate computer vision capabilities into their applications without the need for extensive knowledge of deep and machine learning or significant amounts of time. Cloud-based APIs allow developers to access advanced algorithms for processing visual data such as images, photos, and video frames by uploading or linking the data via the internet (Elisha, 2022).

The three most prominent computer vision APIs available on the market are AWS Rekognition API, Microsoft Azure Face API, and Google Cloud Vision API. While these APIs offer developers an accessible way to implement computer vision capabilities, they also raise important issues related to privacy and security since the image data is uploaded and processed remotely. In addition, using cloud-based APIs for real-time applications can be technically limited and expensive. Hence, developers may want to consider on-device computer vision processing for applications that require real-time processing or need to function without an internet connection.

2.6.1 AWS Rekognition API

Amazon Rekognition is a cloud-based platform that offers computer vision technology as a service, providing users with image and video analysis capabilities using advanced deep learning algorithms. With Amazon Rekognition, developers can easily integrate object and scene recognition, text detection, and facial analysis into their applications without requiring any prior machine learning experience. This powerful tool can recognize a wide range of visual data, including objects, people, text, and actions, and can even identify defects in machine parts or sick plants. By providing photos of the items or situations to be identified, the platform handles the rest. Moreover, Amazon Rekognition helps users detect potentially harmful or inappropriate content across both images and videos and offers precise control over what is allowed based on their requirements. The software offers a free trial that allows users to analyze up to 5,000 images per month and save up to 1,000 pieces of face metadata per month (Amazon Web Service, n.d.).

Pros:

- The API supports a wide range of computer vision tasks and can be used for face searches in images and videos.
- The AWS service is fast and reliable, as one would expect from this provider.
- The deep learning networks are robust, offering top-level performance.
- The free tier allows for 12 months of usage, including the analysis of 5,000 images and storage of 1,000 pieces of face metadata per month.

Cons:

- Estimating the cost of API usage with the pay-per-use model is challenging, making it difficult to predict future costs.
- The API may be challenging for beginners to use.

2.6.2 Microsoft Azure Face API

Azure Face API is an AI-based service that allows developers to incorporate facial recognition and analysis into their applications without prior machine learning knowledge. The platform provides a seamless and secure user experience by integrating face detection, identification, and analysis of images and videos. Azure Face API is capable of detecting a range of facial characteristics, including face masks, spectacles, and facial locations, and identifying individuals through a match to a private repository or photo ID. One may utilize this advanced technology in different situations, including but not limited to verifying the identities of novel users, validating authentication for user control access, or expunging visages from pictures. The confidence score provided by Azure Face API assesses the likelihood of two faces belonging to the same individual. Additionally, the platform ensures enterprise-level security and privacy for user data and trained models by avoiding the storage of evaluated images. Azure Face API seamlessly integrates with several other platforms, making it a versatile tool for developers seeking to incorporate facial recognition technology into their applications (Microsoft, n. d. a).

Pros:

- Well-documented guides, tutorials, and samples to learn from are available.
- The API provides good performance with comparably fast response times.
- Integrated to the ecosystem of Microsoft Azure, SQL database, storage, and virtual machines.
- The Microsoft Computer Vision API can be used for free, including 5’000 calls per month.

Cons:

- A high number of API calls beyond the allowed limit per second can result in throttled response times.
- The usage-based pricing is rather expensive for applications that require multiple transactions.

Limited Access Features of the Face API

Microsoft's goal is to enable developers and organizations to utilize AI for positive societal transformation. In order to safeguard individuals' rights and safety, responsible AI practices are encouraged. To prevent the misuse of facial recognition services in accordance with their AI Principles and facial recognition principles, Microsoft has limited access to their facial recognition services. Customers and partners who wish to utilize the Limited Access features of the Face API, such as Face identification and Face verification, must first register for access by submitting a registration form (Microsoft, n. d. b).

2.6.3 Google Cloud Vision API

Incorporating an array of services including storage access and machine learning-based image analysis, Google Cloud APIs serve as a fundamental aspect within the framework that is the Google Cloud Platform. Such incorporation allows developers to seamlessly integrate these indispensable tools into their cloud platform applications with relative ease. These APIs provide developers with detailed insights into their project's usage of the API,

including traffic levels, error rates, and latencies, which can be used to quickly identify and address any issues that arise in applications utilizing Google services. This information is readily accessible through the API Dashboard within the Cloud Platform Console. By leveraging Google Cloud APIs, developers can streamline their application development and enhance the functionality of their applications (Google Cloud, n. d.).

Pros:

- The API can be used for free with a pay-per-use model and free credits, but a credit card is required for registration.
- The API has high-level privacy, security, and compliance standards, including ISO and SOC certificates. This is crucial for computer vision APIs that involve sensitive data transmission.
- Support from Google Image Search is available for object detection.
- Multiple filter parameters can be applied to an individual image.

Cons:

- The payment model is complex and may be difficult for beginners to comprehend. It is not easy to estimate costs.
- Using the API can quickly become expensive, with free processing only for the first 1000 units per month.

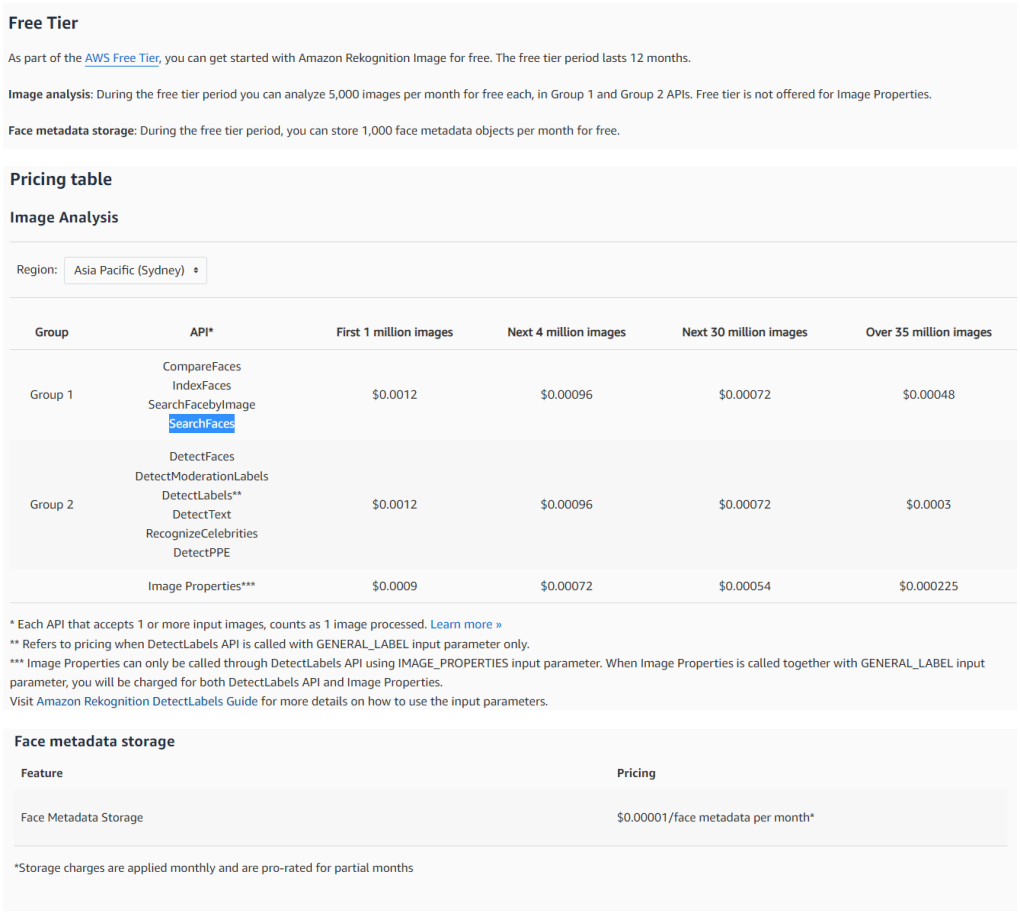
2.6.4 Comparison of Computer Vision APIs

The Table below compares the technical aspects or the pricing, plans, and features between the computer vision APIs: Amazon Rekognition, Microsoft Azure Face API, and Google Cloud Vision API.

APIs / Technical	Amazon Rekognition	Microsoft Azure Face API	Google Cloud Vision API
Pricing Model			
Free Trial	Not Available	Not Available	**Available**
Freemium	**Available**	**Available**	Not Available
Open source	Not Available	Not Available	Not Available
Subscription	**Available**	**Available**	Not Available
Quotation Based	Not Available	Not Available	**Available**
Plans			
	Free Tier	Free – Web/ Container	Google Cloud APIs
Free Custom APIs	Free	Free	Custom
Cost	Figure 2.1	Figure 2.2	Figure 2.3
Features			
Face Detection	**Available**	**Available**	**Available**
Face Verification	**Available**	**Available**	Not Available
Face Identification	**Available**	**Available**	Not Available
Face Grouping	Not Available	**Available**	Not Available

Similar Face Search	**Available**	**Available**	Not Available
Storage	AWS S3	Face Storage	Google Cloud
Celebrity Recognition	**Available**	Not Available	**Available**

Table 2.3: Comparison between Computer Vision APIs



Free Tier

As part of the AWS Free Tier, you can get started with Amazon Rekognition Image for free. The free tier period lasts 12 months.

Image analysis: During the free tier period you can analyze 5,000 images per month for free each, in Group 1 and Group 2 APIs. Free tier is not offered for Image Properties.

Face metadata storage: During the free tier period, you can store 1,000 face metadata objects per month for free.

Pricing table

Image Analysis

Region: Asia Pacific (Sydney)

Group	API*	First 1 million images	Next 4 million images	Next 30 million images	Over 35 million images
Group 1	CompareFaces IndexFaces SearchFacebyImage SearchFaces	$0.0012	$0.00096	$0.00072	$0.00048
Group 2	DetectFaces DetectModerationLabels DetectLabels** DetectText RecognizeCelebrities DetectPPE	$0.0012	$0.00096	$0.00072	$0.0003
	Image Properties***	$0.0009	$0.00072	$0.00054	$0.000225

* Each API that accepts 1 or more input images, counts as 1 image processed. Learn more »
** Refers to pricing when DetectLabels API is called with GENERAL_LABEL input parameter only.
*** Image Properties can only be called through DetectLabels API using IMAGE_PROPERTIES input parameter. When Image Properties is called together with GENERAL_LABEL input parameter, you will be charged for both DetectLabels API and Image Properties.
Visit Amazon Rekognition DetectLabels Guide for more details on how to use the input parameters.

Face metadata storage

Feature	Pricing
Face Metadata Storage	$0.00001/face metadata per month*

*Storage charges are applied monthly and are pro-rated for partial months

Figure 2.1: Pricing Table for AWS Rekognition

Instance	Transactions Per Second (TPS) *	Features	Price
Free - Web/Container	20 transactions per minute	Face Detection Face Verification Face Identification Face Grouping Similar Face Search	30,000 transactions free per month
Standard - Web/Container	10 TPS	Face Detection Face Verification Face Identification Face Grouping Similar Face Search	0-1M transactions - **$1** per 1,000 transactions 1-5M transactions - **$0.80** per 1,000 transactions 5-100M transactions - **$0.60** per 1,000 transactions 100M+ transactions - **$0.40** per 1,000 transactions
		Face Storage	**$0.01** per 1,000 faces per month

* TPS only applies to web endpoint

Figure 2.2: Pricing Table for Microsoft Azure Face API

Prices

Charges are incurred per image. For files with multiple pages, such as PDF files, each page is treated as an individual image.

Each feature applied to an image is a billable *unit*. For example, if you apply Face Detection and Label Detection to the same image, you are billed for one unit of Label Detection and one unit for Face Detection.

The table below shows the price for each feature per 1000 units. Pricing is tiered - the first 1000 units used each month are free, units 1001 to 5,000,000 are priced as marked, etc.

Feature	Price per 1000 units		
	First 1000 units/month	Units 1001 - 5,000,000 / month	Units 5,000,001 and higher / month
Label Detection	Free	$1.50	$1.00
Text Detection	Free	$1.50	$0.60
Document Text Detection	Free	$1.50	$0.60
Safe Search (explicit content) Detection	Free	Free with Label Detection, or $1.50	Free with Label Detection, or $0.60
Facial Detection	Free	$1.50	$0.60
Facial Detection - Celebrity Recognition	Free	$1.50	$0.60
Landmark Detection	Free	$1.50	$0.60
Logo Detection	Free	$1.50	$0.60
Image Properties	Free	$1.50	$0.60
Crop Hints	Free	Free with Image Properties, or $1.50	Free with Image Properties, or $0.60
Web Detection	Free	$3.50	Contact Google for more information
Object Localization	Free	$2.25	$1.50

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Figure 2.3: Pricing Table for Google Cloud Vision API

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

From Table 2.1, it is proved that the Rapid Application Development Model (RAD) becomes a better SDLC model that can be quickly iterated and updated the project numerous times without needing to start from the beginning with a new development strategy each time. Hence, this project chooses the RAD model to develop software applications.

Numerous development tools and software have been recognized, encompassing a wide array of environments, programming languages, as well as development frameworks and libraries. A project plan has been formulated by creating a work-breakdown structure and converting it into a Gantt Chart.

3.2 Software Development Methodology

The software development methodology is a procedure or set of methods used in software development. The rapid Application Development Model (RAD) is the adopted software development process. This methodology has four main phases: the requirements phase, the user design phase, the development phase, and the transition phase. The user design phase becomes the primary emphasis of the development throughout the whole project. More detailed information on each level of this process is covered in the project approach in Chapter 1.4.

3.2.1 User Design Phase

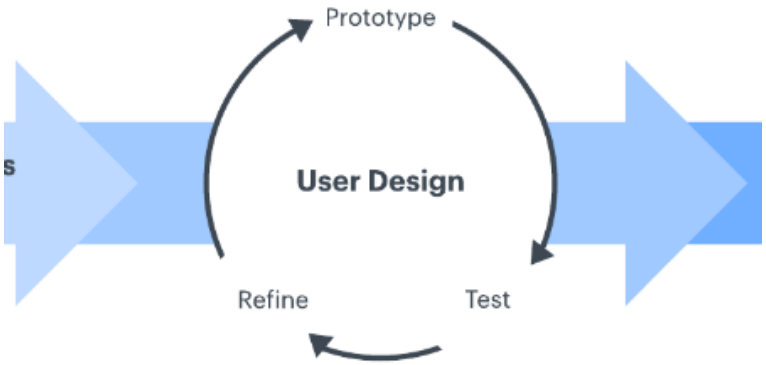


Figure 3.1 : RAD Model, User Design Phase (Lucid Content Team, n.d.)

The three main steps in developing the project at this phase are prototyping, testing, and refining. Clients and developers collaborate closely throughout all stages of the design process to ensure that their requirements are met at each level. Similar to custom software development, users can assess each product prototype at each stage to make sure it meets their expectations. All the faults and kinks are worked out through an iterative process. Users test a prototype the developer has created, and the two parties then discuss what functioned and what did not. With this approach, developers can make changes to the model as they go along until they find a solution.

Table 3. 1: Stages in User Design Phase of RAD Model

Stages	Explanation
Prototype	Create prototypes continuously; begin with low-fidelity models and advance to higher-fidelity models as the project progress.
Test	Test prototypes as they are being developed. Prototype may be improved, get a better understanding of user requirements, and even hone the problem statement through testing.
Refine	Refine by revising over the design phase problem and its success criteria to make sure the solutions adhere to the project's specifications.

3.3 Development Tools

3.3.1 Programming Tools and IDEs

i. Android Studio

- The officially recognised integrated development environment (IDE) for developing applications for the Android operating system is called Android Studio. The build system used by Android Studio is based on Gradle. Building, testing, and deployment are all automated using the Gradle system. Once new code or resources are added to the project, Android Studio enables developers to apply changes to active applications immediately. The application does not need to be restarted for developers to observe changes. To help developers create standard app features, Android Studio offers code templates. Additionally, GitHub integration can be used in Android Studio to add or modify codes.

ii. Android Emulator

- One of the features used in creating this project is the Android emulator in Android Studio. Instead of using mobile devices, the emulator enables developers to run the program and observe changes on their computers. Before the application is made available to the public, developers can test it on an emulator. On the PC, Android emulators flawlessly replicate a mobile device. It offers features found on mobile devices. As a result, developers can evaluate the app's operation without an Android device. All the Android application developers will find this emulator to be convenient.

iii. Visual Studio Code

- The text editor Visual Studio Code, also known as VS Code, is a free, open-source Integrated Development Environment (IDE) developed by Microsoft. It supports various programming languages, including Java, JavaScript, CSS, and Python. Most of the code will be written in this tool.

iv. Figma

- The project prototype is made using the software program Figma. Figma is a collaborative UI and UX design application. Using Figma will have the opportunity to develop sample prototypes with various Android device resolutions. Additionally, by setting up triggers for an occasion that will make the prototype lively, interactive prototypes can be made.

v. Github

- GitHub is a widely used web-based hosting service for software development and an online version control platform employing Git. In this project, we will utilize GitHub for managing various iterations of our source code. We have selected GitHub as our version control system owing to its user-friendly interface for managing and publishing code modifications and monitoring the progress of development.

vi. Amazon Web Services

- Amazon Web Services (AWS) is a comprehensive cloud computing platform provided by Amazon, which offers a wide range of services such as computing power, storage, and databases. In this project, AWS will be utilized to host and manage the backend infrastructure, ensuring scalability, flexibility, and reliability for the application. AWS provides various tools and services, such as AWS Lambda for serverless computing and Amazon S3 for storage, to help developers build and deploy applications efficiently.

vii. AWS CLI

- In order to effectively manage and interact with Amazon Web Services (AWS) through the command line, we rely on a powerful tool known as the AWS Command Line Interface (CLI). Acting in unison, this unified system directly manages our usage of AWS resources. Through the utilization of scripted automation and streamlining access to AWS

services, the AWS CLI facilitates developers' seamless administration of their respective Amazon Web Services provision. With the AWS CLI, developers can access services such as Amazon S3, Amazon EC2, and AWS Lambda, among others. This tool is essential for efficiently deploying, managing, and configuring the project's cloud infrastructure..

viii. Amazon Rekognition

- Amazon Rekognition is a powerful AI-based image and video analysis service provided by AWS, which enables developers to add advanced computer vision capabilities to their applications. In this project, Amazon Rekognition will be used to implement features such as facial recognition, object detection, and scene analysis. By leveraging Amazon Rekognition, the application can analyse and extract meaningful insights from images and videos, enhancing the user experience and enabling unique functionalities.

3.3.2 Programming Languages

i. Java

- Java is a high-level, class-based, object-oriented programming language with the least amount of feasible implementation dependencies. Since the introduction of the Android platform, Java has been the default language for writing Android applications. To test the Android app without using an actual device, an Android emulator called an Android Virtual Device (AVD) is used to perform.

ii. JavaScript (JS)

- JavaScript is a scripting language that can be used locally with NodeJS and browsers. The fundamental client-side JS language enables users to store important parameters inside variables and execute code when specific web page events occur.

3.3.3 Frameworks

i. React Native

- A well-liked JavaScript-based mobile app framework known as React Native is sometimes referred to as RN. While leveraging native-OS views, React Native compiles and renders the app's user interface in JavaScript. It supports code implementation in OS-native languages for more complex functionality, such as Swift and Objective-C for iOS and Java for Android.

3.3.4 Database

a. Firebase

- A Firebase is a comprehensive development platform and Backend-as-a-Service (BaaS) that offers a wide array of services, including hosting, databases, analytics, authentication, and more. It enables real-time data syncing and storage between users. In this project, we have utilized Firebase for user authentication, managing the Firestore database, and handling storage. Firebase Authentication provides a secure and straightforward way to manage user authentication, supporting various methods like email/password and social media logins. Firestore, a flexible and real-time NoSQL database, has been employed for storing and managing application data such as user profiles and app settings. Lastly, Firebase Storage has been used to securely store and manage user-uploaded files, including profile pictures and media content.

b. Amazon S3 Bucket

- AWS provides Amazon S3, a cloud storage service that is both scalable and extensively available. It has been developed to effectively store data of any amount from wherever it may be found on the internet while ensuring durability, low latency as well as ease of access. Our project takes advantage by using Amazon S3 Buckets in order to retain massive files including images or videos which will be quickly accessed for users' usage. The use of Amazon S3 incorporation enhances effectiveness performance optimization; augments our ability to manage

control accessibility and ensures protected backups are readily obtainable when required.

3.4 Project Plan

The Gantt chart attached in Appendix A will be discussed in this section. Each project's activities have a set duration and a planned completion date. Gantt charts are used to keep track of the project progress and prevent delays in work completion. Any task delay could result in overdue time or increased project costs. Gantt charts are the project's guidelines and must be adequately adhered to. The project plan includes a Work Breakdown Structure (WBS) which outlines all the tasks and sub-tasks required to complete the project and provides a detailed breakdown of the project's scope, timeline, and resources required for each task.

3.4.1 Work Breakdown Structure (WBS)

In project management, the Work Breakdown Structure (WBS) refers to breaking down a project into smaller, more manageable components based on deliverables. This facilitates easy and accurate measurement of project activities and progress to estimate completion time. The objective of using WBS in projects is to decompose large projects into smaller activities to make them more manageable. The top-down approach is employed to create the WBS for this project, where major tasks are identified and completed first before breaking them down into smaller tasks. As each smaller task is completed, the parent activity is also completed. Figure 3.2 depicts the WBS that will be utilized for this project.

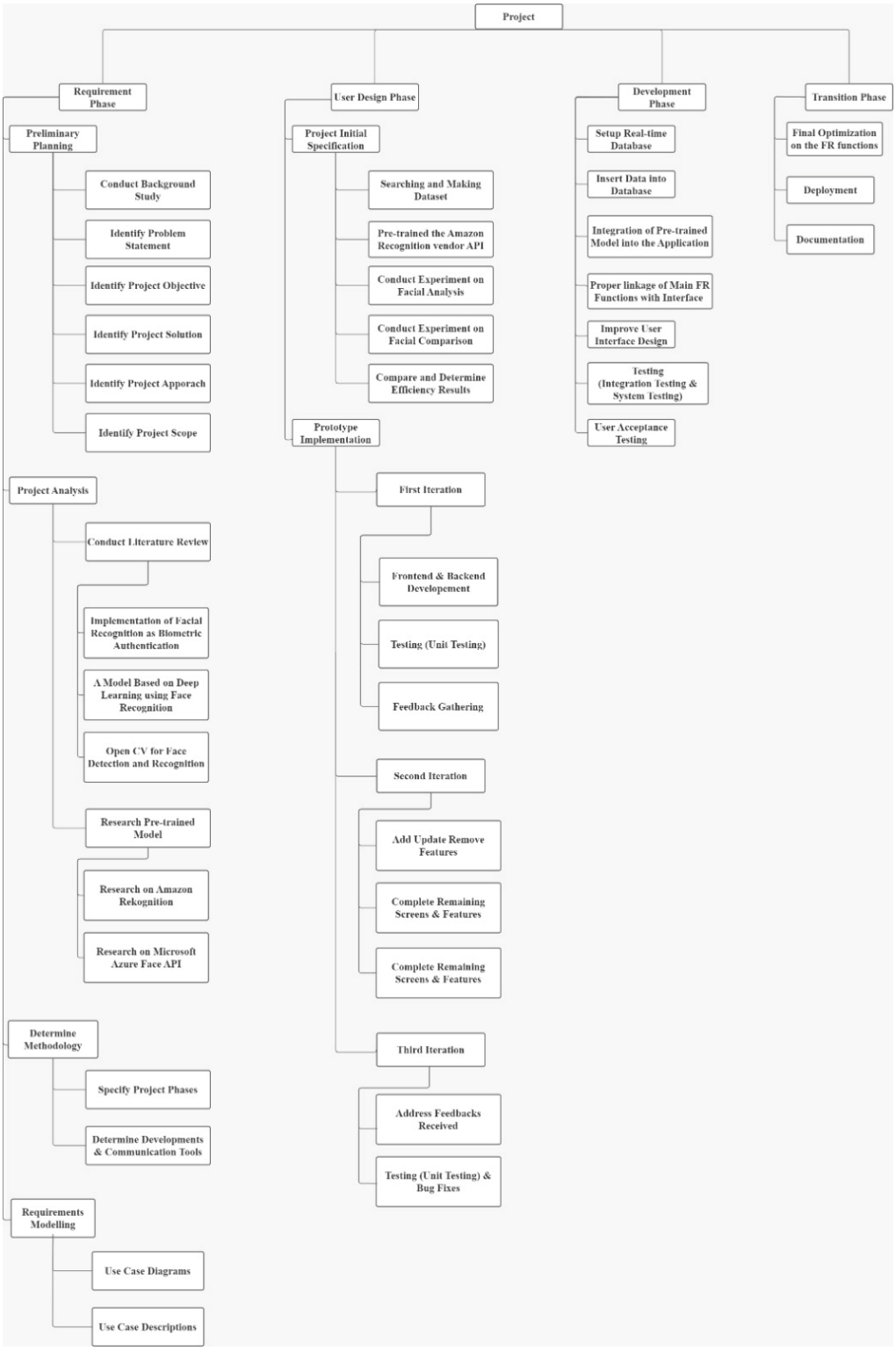


Figure 3.2 : WBS Structure for In-Building Facial Recognition Check-In

3.4.2 Gantt Chart

By using the Gantt Chart, the project's activities are presented in a specific order, along with their estimated start date, end date, and duration. The implementation of the Gantt Chart breaks down the main activities identified in the Work Breakdown Structure (WBS) into smaller tasks, each with an estimated start and end date to determine its duration. The Gantt Chart for this project has been created and is presented in Table 3.2 below.

Tasks	Duration (Days)	Start Date	End Date
REQUIREMENT PHASE	**34**	**13 June 2022**	**16 July 2022**
[Preliminary Planning]	**12**	**13 June 2022**	**24 June 2022**
Conduct Background Study	2	13 June 2022	14 June 2022
Identify Problem Statement	2	15 June 2022	16 June 2022
Identify Project Objective	2	17 June 2022	18 June 2022
Identify Project Solution	2	19 June 2022	20 June 2022
Identify Project Approach	2	21 June 2022	22 June 2022
Identify Project Scope	2	23 June 2022	24 June 2022
[Project Analysis]	**12**	**25 June 2022**	**6 July 2022**
Conduct Literature Review	**6**	**25 June 2022**	**1 July 2022**
Implementation of Facial Recognition as Biometric Authentication	2	25 July 2022	26 July 2022
A Model Based on Deep Learning using Face Recognition	2	27 July 2022	28 July 2022

Open CV for Face Detection and Recognition	2	29 July 2022	30 July 2022
Research Pre-trained Model	**6**	**1 July 2022**	**6 July 2022**
Research on Amazon Rekognition	3	1 July 2022	3 July 2022
Research on Microsoft Azure Face API	3	4 July 2022	6 July 2022
[Determine Methodology]	**4**	**7 July 2022**	**10 July 2022**
Specify Project Phases	2	7 July 2022	8 July 2022
Determine Developments & Communication Tools	2	9 July 2022	10 July 2022
[Requirements Modelling]	**6**	**11 July 2022**	**16 July 2022**
Use Case Diagrams	2	11 July 2022	12 July 2022
Use Case Descriptions	4	13 July 2022	16 July 2022
USER DESIGN PHASE	**32**	**16 Jan 2023**	**27 Feb 2023**
[Project Initial Specification]	**10**	**16 Jan 2023**	**25 Jan 2023**
Searching and Making Dataset	2	16 Jan 2023	17 Jan 2023
Pre-trained the Amazon Rekognition vendor API	2	18 Jan 2023	19 Jan 2023
Conduct Experiment on Facial Analysis	2	20 Jan 2023	21 Jan 2023
Conduct Experiment on Facial Comparison	2	22 Jan 2023	23 Jan 2023
Compare and Determine Efficiency Results	2	24 Jan 2023	25 Jan 2023

[Prototype Implementation]	**32**	**26 Jan 2023**	**27 Feb 2023**
First Iteration	**16**	**26 Jan 2023**	**11 Feb 2023**
Frontend & Backend Development	8	26 Jan 2023	3 Feb 2023
Testing (Unit Testing)	6	4 Feb 2023	9 Feb 2023
Feedback Gathering	2	10 Feb 2023	11 Feb 2023
Second Iteration	**10**	**12 Feb 2023**	**21 Feb 2023**
Add Update Remove Features	4	12 Feb 2023	15 Feb 2023
Complete Remianing Screens & Features	4	16 Feb 2023	19 Feb 2023
Testing (Unit Testing) & Bug Fixes	2	20 Feb 2023	21 Feb 2023
Third Iteration	**6**	**22 Feb 2023**	**27 Feb 2023**
Address Feedbacks Received	4	22 Feb 2023	25 Feb 2023
Testing (Unit Testing) & Bug Fixes	2	26 Feb 2023	27 Feb 2023
DEVELOPMENT PHASE	**40**	**28 Feb 2023**	
[Setup Real-time Database]	6	28 Feb 2023	6 March 2023
[Insert Data into Database]	4	7 March 2023	10 March 2023
[Integration of Pre-trained Model into the Application]	10	11 March 2023	20 March 2023
[Proper linkage of Main FR Functions with Interface]	6	21 March 2023	26 March 2023
[Improve User Interface Design]	4	27 March 2023	30 March 2023

[Testing (Integration Testing & System Testing)]	4	31 March 2023	3 April 2023
[User Acceptance Testing]	4	4 April 2023	7 April 2023
TRANSITION PHASE	**16**	**8 April 2023**	**23 April 2023**
[Final Optimization on the FR functions]	4	8 April 2023	11 April 2023
[Deployment]	2	12 April 2023	13 April 2023
[Documentation]	10	14 April 2023	23 April 2023

Table 3.2: Project Timeline for In-Building Facial Recognition Check-In

CHAPTER 4

PROJECT INITIAL SPECIFICATION

4.1 Introduction

From Table 2.2, it is proved that CNN has become an algorithm that is adopted in the field of facial recognition. Hence, CNN is being chosen as the process for implementing the facial recognition system. This chapter will explain the overview of CNN architecture and how both Amazon Rekognition Face Detection and Microsoft Azure Face API leverage the CNNs to provide facial detection and recognition capabilities. Besides that, several experiments have been conducted in this section to determine whether Amazon Rekognition is suitable for the facial-recognition check-in application.

4.2 Overview of CNN

When processing data with a grid-like topology, CNNs are a form of deep learning method. When processing data with a geographical or temporal link, CNNs are a deep learning method. Despite the fact that CNNs use a number of convolutional layers, they are more complex than other neural networks. Convolutional layers are essential for the operation of convolutional neural networks (CNNs). In addition, CNN used the Rectified Linear Activation Unit (ReLU) because the ReLU activation function layer maintains non-linearity as the data moves through each layer; otherwise, the data would lose the required dimensionality. The three layers of a CNN architecture are typically convolutional, pooling, and fully connected, as shown in Figure 4.1 below.

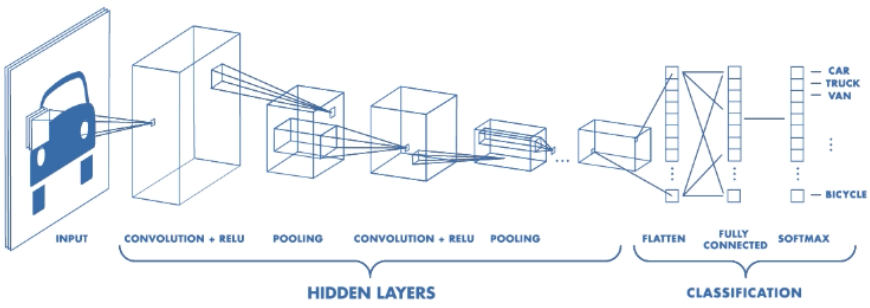


Figure 4.1: CNN Architecture (Pingel, 2017)

4.3 Convolutional Layer

The core element of CNN is the convolution layer, and the majority of the network's computational load is carried by it. The kernel, a collection of learnable parameters, and the limited region of the receptive field are two matrices combined in this layer to form a dot product. Compared to a picture, the kernel is denser but lower in size. Accordingly, the kernel height and width of a picture with three (RGB) channels will be extremely small, but the depth will expand to accommodate all three channels (Kumar, 2022). Convolutional layers can be combined to build more sophisticated models to extract finer details from images.

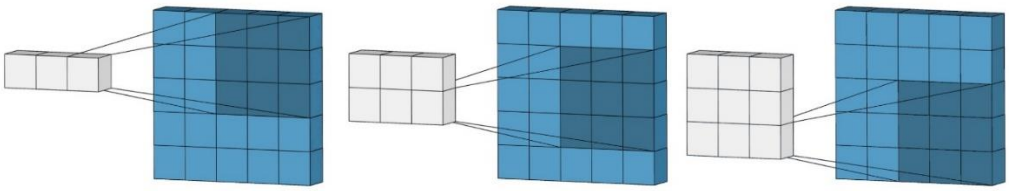


Figure 4.2: Convolutional Operation (Mishra, 2020)

The kernel advances across the picture's height and breadth during the forward pass, generating an image of that receptive region. As a result, an activation map, a two-dimensional image representation, is produced, revealing the kernel's reaction at each spot in the image. The slidable size of the kernel is referred to as a stride.

The output volume may be calculated using the formula below if have an activation map input size of W x W x D

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

where,

F = number of kernels with a spatial size

S = Stride

P = Amount of Padding

An output volume of size Wout x Wout x Dout will result from this.

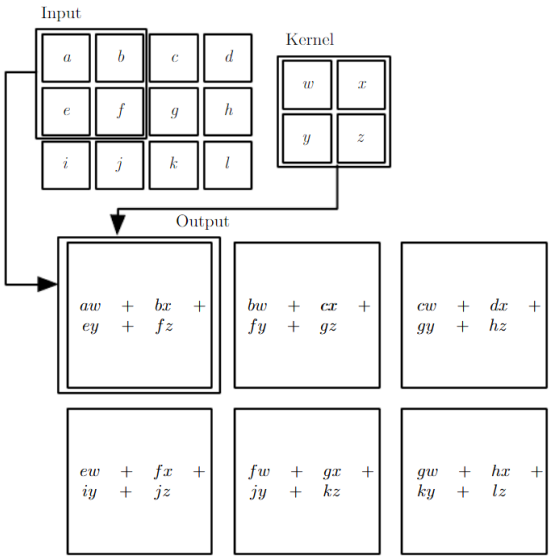


Figure 4.3: 2-D convolution Without Kernel Flipping (GoodFellow et al., 2018)

4.3.1 Convolution

Convolution is a linear process applied to the feature extraction process. Convolution involves applying a tiny array of numbers called the kernel to a more extensive array of numbers called the tensor as the input. The element product of each input tensor element and kernel at each place is calculated for the feature map illustrated in Figure 3.5. The output value for each position of the output tensor is summed. The process involves repetitively applying several kernels to create any number of feature maps that reflect different features of the input tensor. Therefore, various kernels can be thought of as extractors of distinct properties. The two primary hyperparameters that define the convolution process are size and the number of kernels. The depth of the output feature maps is determined by the former, which is usually 3x3 but occasionally 5x5 or 7x7. In contrast, the latter is random (Yamashita et al., 2018).

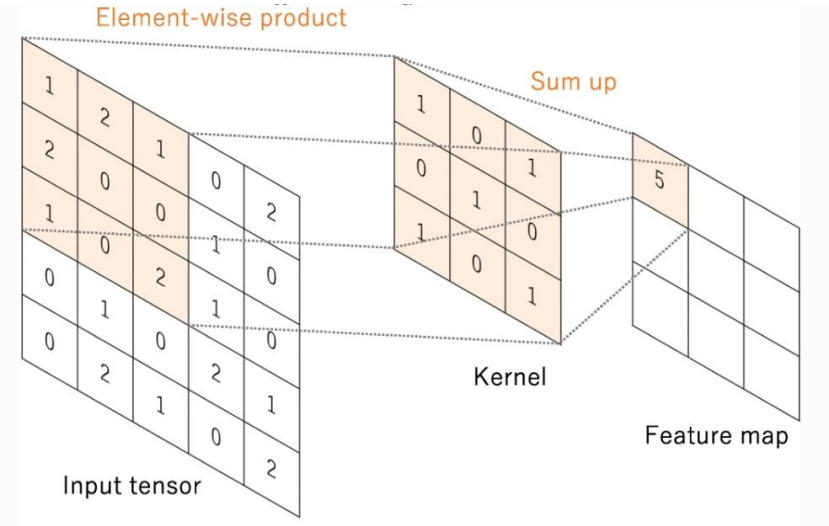


Figure 3. 5:

Figure 4.4: 2-D Convolution Operation with Kernel Size 3x3, without padding, and stride of 1 (Yamashita et al., 2018)

The output feature map's height and width will be less than that of the input tensor because the center of each kernel prevents it from overlapping the outermost element of the input tensor in the process outlined above. Therefore, padding, especially zero padding, is crucial in helping to resolve this issue. The convolution process with zero padding is used to add rows and columns of zeros on either of the input tensor to fit the kernel in the middle of the outermost element and maintain the same plane dimension (Yamashita et al., 2018).

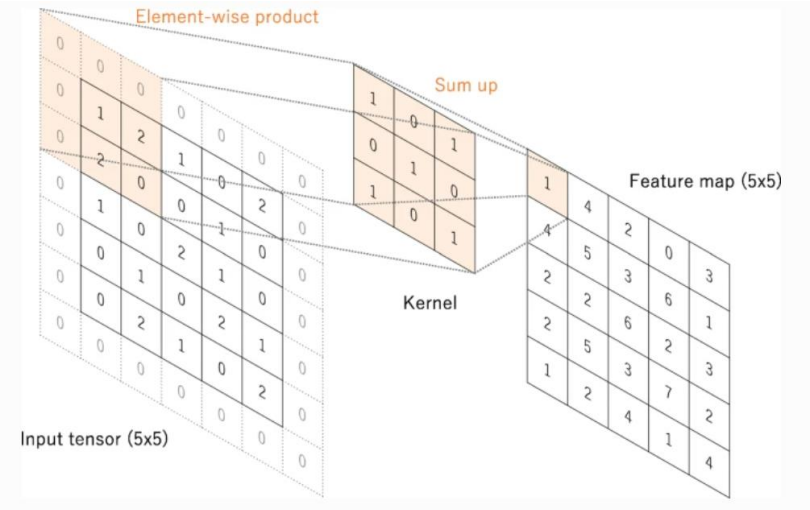


Figure 4.5: 2-D Convolution Operation with Kernel Size 3x3, without padding, stride of 1, and added 5x5 input dimension (Yamashita et al., 2018)

4.4 Non-Linearity Layer

Non-linearity layers are directly included right after the convolutional layer to add it to the activation layer because convolution is a linear process and images are non-linear.

4.4.1 ReLU

A non-linear activation function is then applied to the convolution operation's output. The most common non-linear activation function in CNN is the Rectified Liner Unit (ReLU) (Sharma, 2017). This results in $f(x) = \max(0, x)$. ReLU can train more quickly and effectively by converting negative values to zero while maintaining positive values. Because only activated characteristics will be transferred to the following layer, this is frequently referred to as activation. Other non-linear functions that have been ultilised include Sigmoid and Tanh activation function (Mishra, 2020).

4.5 Pooling Layer

A pooled feature map is the result of a layer that pools data. A pooling layer decreases the feature map's sample size and speeds up processing by lowering the overall amount of parameters a network must consider. Max pooling and average pooling are the two main types used to create a pooled feature map (C, 2020).

As shown in Figure 4.6, when using max pooling, each filter's maximum value is taken, and a new output with the dimensions 2x2 pixels is assembled. In comparison, the average value of the filter size is used for pooling. The flattening, hidden layer and activation functions make up the classification layer.

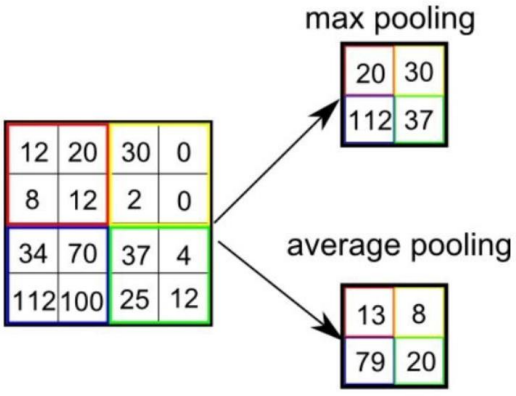


Figure 4.6: Max Pooling and Average Pooling (C, 2020)

4.6 Fully Connected Layer

One of the most fundamental varieties of layers in a convolutional neural network is the fully connected layer (CNN). Each neuron in a layer connected to the layer below is completely connected to every other neuron in that layer. Fully linked layers are typically used in the last stages of a CNN when it is desirable to use the features uncovered by the final levels to create predictions. (Kumar 2020). The output of the final convolutional or pooling layer is passed into the fully connected layer where it is flattened before being used (Arc, 2018).

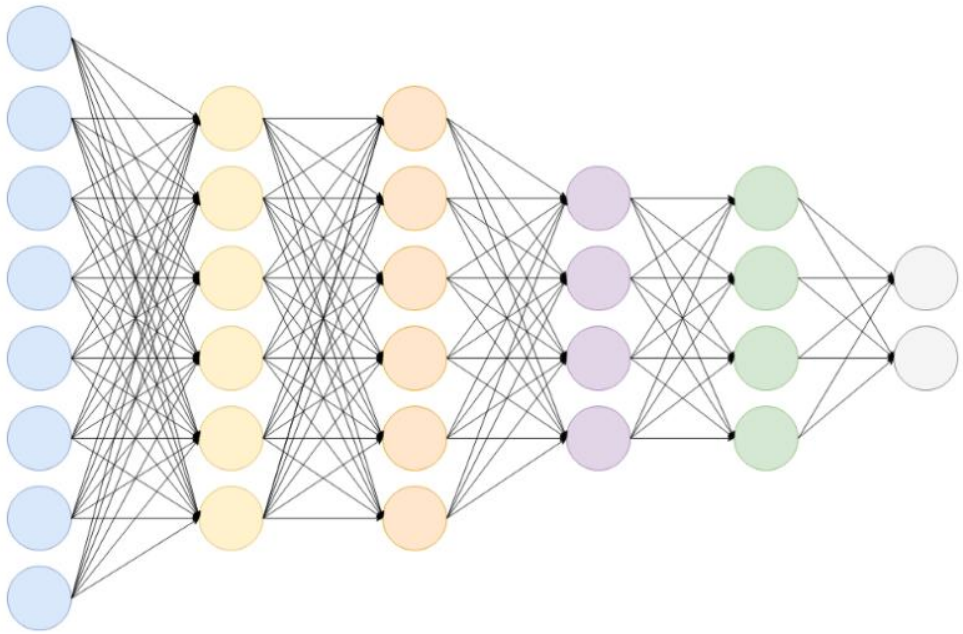


Figure 4.7: Fully Connected Layer (Arc, 2018)

4.7 Pre-trained Model

Convolutional Neural Networks (CNNs) form the backbone of Amazon Rekognition and Azure Face API by providing a robust architecture for detecting and recognizing faces in images. These services employ pre-trained models that utilize large datasets of labelled images to train their CNNs, allowing them to recognize patterns and features associated with human faces. When an image is processed, the CNN extracts essential features such as facial landmarks, face orientation, and facial attributes, which both Amazon Rekognition and Azure Face API use to deliver information about detected faces. These pre-trained models empower the services to offer a range of functionalities, including face detection, face recognition, and face attribute analysis. Additionally, the efficiency of CNNs enables real-time face detection using video input, further enhancing the capabilities of Amazon Rekognition Face Detection and Microsoft Azure Face API.

4.8 Experiments

The experiment involved modifying and degrading the images from a carefully taken picture with a mobile phone. The degraded images were then assessed for resilience using Amazon Rekognition API. The properties altered in the images included blur, brightness, contrast, rotation, and, the subject not facing the camera, and transparency. The data obtained from the tests was analysed and evaluated to identify the quality and accuracy of Amazon Rekognition in the facial recognition software industry. This analysis aimed to determine whether Amazon Rekognition is suitable for the facial-recognition check-in application, FACEIN.

The experiment is only conducted with Amazon Rekognition because Microsoft Azure Face API has made the Access to Face service is limited to Microsoft-managed customers and partners who adhere to Responsible AI principles recently while Google Cloud Vision API do not have the Face Identification function.

4.8.1 Dataset

The dataset for the study was carefully constructed to ensure authenticity, using images captured with a mobile phone that had individuals of the same gender and ethnicity, as the aim was to evaluate the performance of APIs based on the gender and age attributes of degraded images, which comprised males with different facial features and expressions, such as facial hair, glasses, and black and white as well as coloured photos.

The images were degraded using 'PhotoScape X' to mimic the conditions of a typical selfie, with conditions ranging from blurring (50%), brightness (100%), contrast (100%), not looking at the camera, to rotation, measured in specified percentage in order to capture precise results.

The performance of the APIs was measured based on their accuracy acquired from the facial analysis and similarity in face comparison.

Figure 4.8: Original and Degraded Photo in the Dataset

4.8.2 Results

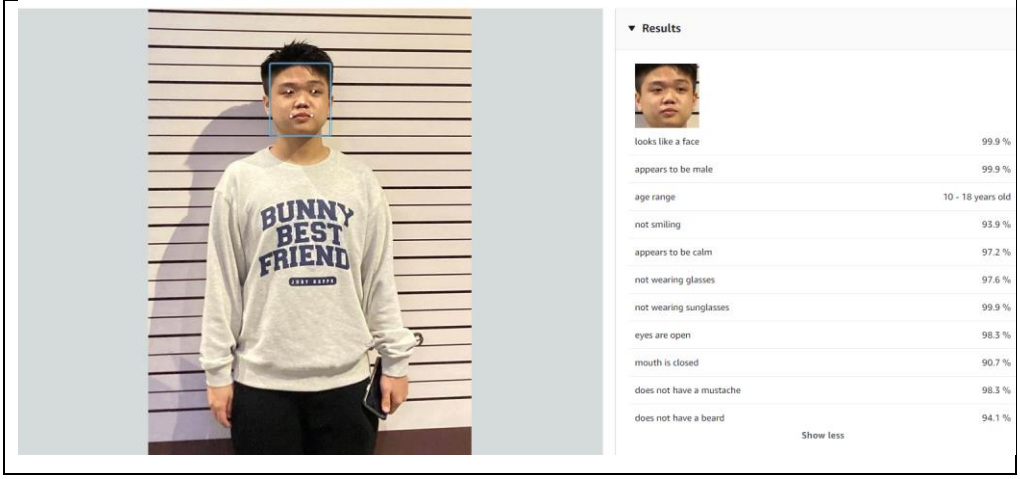
The API Vendor software was tested by subjecting it to degraded images under different test conditions such as blurring, brightness, contrast, not looking at the camera, and rotation.

To generate the results, Amazon Rekognition operations such as detecting faces in an image and searching for faces in a collection were used through AWS CLI. The images were first uploaded to an Amazon S3 bucket, and the ones to be compared were indexed into the collection before performing the search face from collection operation.

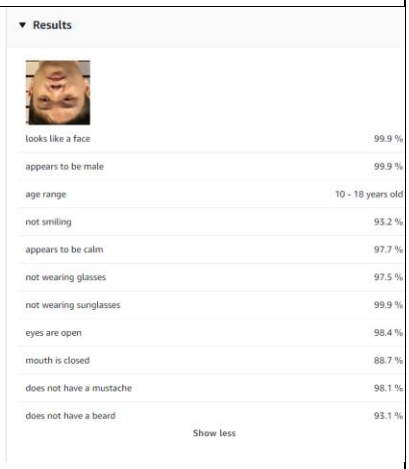
To call the Amazon Rekognition operations, the input image is passed as a reference to an image in an Amazon S3 bucket, and the AWS CLI is used under image in JPG, JPEG, or PNG formatted file.

For each detected face, the operation provided face details such as the bounding box of the face, a confidence value that the bounding box contains a face, and a fixed set of attributes such as facial landmarks, the presence of beard, sunglasses, and so on. Face identification was carried out by searching for faces in a collection that matched the largest face in a supplied image using the SearchFacesByImage operation.

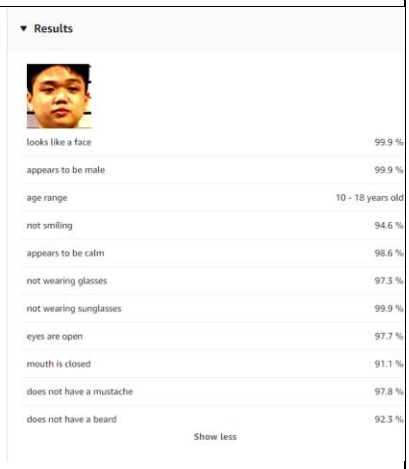
The screenshots result from AWS Rekognition demo were used instead of the AWS CLI for better result display and view. Table 4.1 shows the results from facial analysis operation, and Table 4.2 shows the results from face comparison operation.

Original Photo


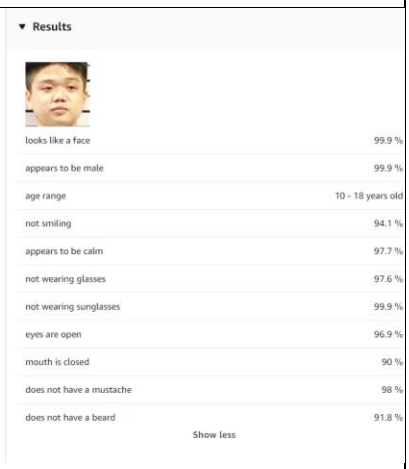
Rotated Photo



Contrast 100% Photo



Brightness 100% Photo



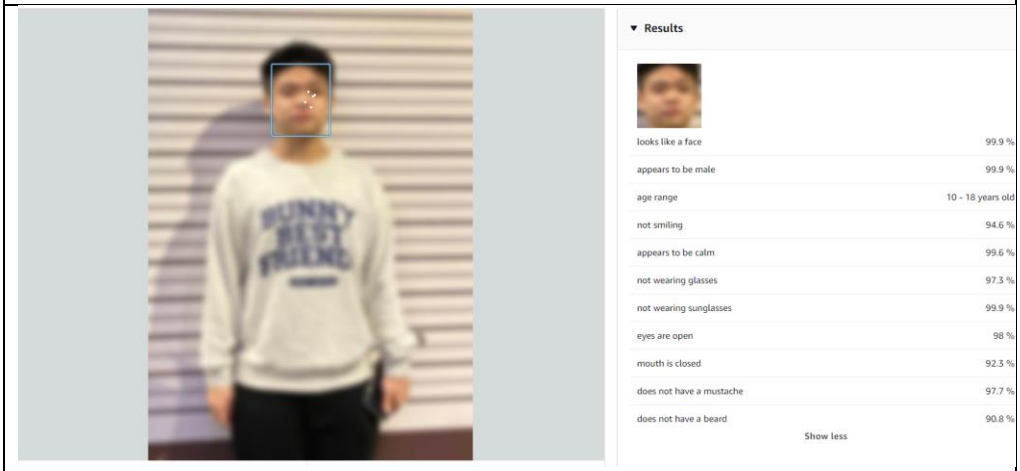
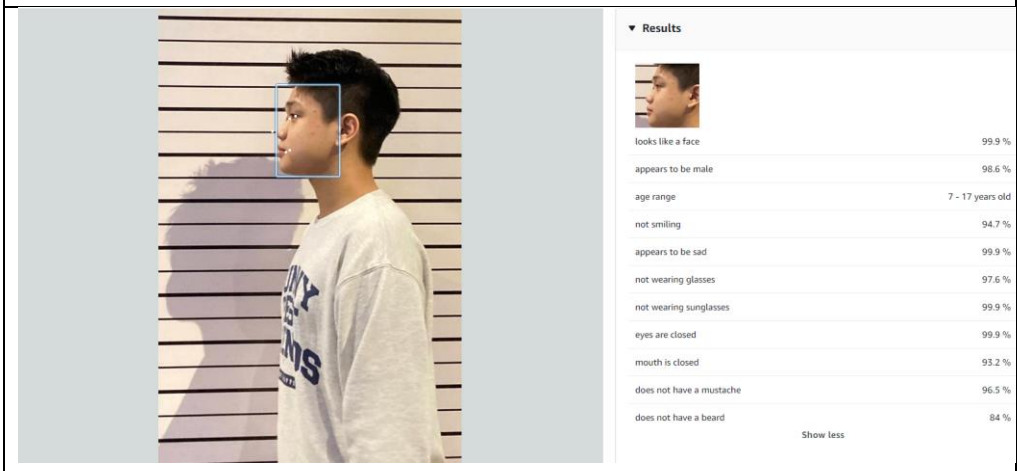
Blurring 50% Photo

Not Looking at The Camera Photo


Table 4.1 : Results Obtained from Facial Analysis Operation

Compare Original Photo with Original Photo



```
{
    "SourceImageFace": {
        "BoundingBox": {
            "Width": 0.1877046674489975,
            "Height": 0.15948854386806488,
            "Left": 0.3742721676826477,
            "Top": 0.1210950240492820 7
        },
        "Confidence": 99.9763412475586
    },
    "FaceMatches": [
        {
            "Similarity": 100,
            "Face": {
                "BoundingBox": {
                    "Width": 0.1877046674489975,
                    "Height": 0.15948854386806488,
                    "Left": 0.3742721676826477,
                    "Top": 0.1210950240492820 7
                },
                "Confidence": 99.9763412475586,
                "Pose": {
                    "Roll": 0.005972708109766245,
                    "Yaw": 3.3107640743255615,
                    "Pitch": 14.469149589538574
                },
                "Quality": {
                    "Brightness": 89.21708679199219,
                    "Sharpness": 20.927310943603516
                }
            }
        }
    ],
    "UnmatchedFaces": []
}
```

Compare Rotated Photo with Original Photo


```
{
    "SourceImageFace": {
        "BoundingBox": {
            "Width": 0.1877046674489975,
            "Height": 0.15948854386806488,
            "Left": 0.3742721676826477,
            "Top": 0.12109502404928207
        },
        "Confidence": 99.9763412475586
    },
    "FaceMatches": [
        {
            "Similarity": 100,
            "Face": {
                "BoundingBox": {
                    "Width": 0.18188300728797913,
                    "Height": 0.16052763164043427,
                    "Left": 0.4384101331233978,
                    "Top": 0.7165928483009338
                },
                "Confidence": 99.98954772949219,
                "Pose": {
                    "Roll": 178.54783630371094,
                    "Yaw": 3.365051507949829,
                    "Pitch": 12.85616397857666
                },
                "Quality": {
                    "Brightness": 89.28597259521484,
                    "Sharpness": 26.1773681640625
                }
            }
        }
    ],
    "UnmatchedFaces": []
}
```

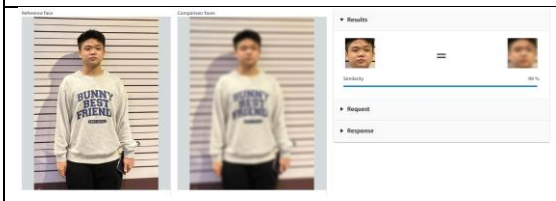

Compare Contrast 100% Photo with Original Photo


```
{
    "SourceImageFace": {
        "BoundingBox": {
            "Width": 0.1877046674489975,
            "Height": 0.15948854386806488,
            "Left": 0.3742721676826477,
            "Top": 0.12109502404928207
        },
        "Confidence": 99.9763412475586
    },
    "FaceMatches": [
        {
            "Similarity": 100,
            "Face": {
                "BoundingBox": {
                    "Width": 0.18401741981506348,
                    "Height": 0.15838535130023956,
                    "Left": 0.3764055669307709,
                    "Top": 0.1235356479883194
                },
                "Confidence": 99.97039794921875,
                "Pose": {
                    "Roll": 0.23225060105323792,
                    "Yaw": 3.7445764541625977,
                    "Pitch": 14.800933837890625
                },
                "Quality": {
                    "Brightness": 92.48522186279297,
                    "Sharpness": 32.20803451538086
                }
            }
        }
    ],
    "UnmatchedFaces": []
}
```


Compare Brightness 100% Photo with Original Photo


```
{
    "SourceImageFace": {
        "BoundingBox": {
            "Width": 0.1877046674489975,
            "Height": 0.15948854386806488,
            "Left": 0.3742721676826477,
            "Top": 0.12109502404928207
        },
        "Confidence": 99.9763412475586
    },
    "FaceMatches": [
        {
            "Similarity": 100,
            "Face": {
                "BoundingBox": {
                    "Width": 0.18766584992408752,
                    "Height": 0.16054239869117737,
                    "Left": 0.3748299181461334,
                    "Top": 0.12154800444841385
                },
                "Confidence": 99.97468566894531,
                "Pose": {
                    "Roll": 0.11108127981424332,
                    "Yaw": 3.4335663318634033,
                    "Pitch": 13.54172134399414
                },
                "Quality": {
                    "Brightness": 97.78656768798828,
                    "Sharpness": 32.20803451538086
                }
            }
        }
    ],
    "UnmatchedFaces": []
}
```

Compare Blurring 50% Photo with Original Photo	

	{
	"SourceImageFace": {
	"BoundingBox": {
	"Width": 0.1877046674489975,
	"Height": 0.15948854386806488,
	"Left": 0.3742721676826477,
	"Top": 0.12109502404928207
	},
	"Confidence": 99.9763412475586
	},
	"FaceMatches": [
	{
	"Similarity": 99.08954620361328,
	"Face": {
	"BoundingBox": {
	"Width": 0.17864276468753815,
	"Height": 0.1582484394311905,
	"Left": 0.37852707505226135,
	"Top": 0.12220723181962967
	},
	"Confidence": 99.98399353027344,
	"Pose": {
	"Roll": 2.326028823852539,
	"Yaw": 2.3225202560424805,
	"Pitch": 12.209784507751465
	},
	"Quality": {
	"Brightness": 88.77447509765625,
	"Sharpness": 1.0484901666641235
	}
	}
	}
	],
	"UnmatchedFaces": []
	}

Compare Not Looking at The Camera Photo with Original Photo	

	{
	"SourceImageFace": {
	"BoundingBox": {
	"Width": 0.1877046674489975,
	"Height": 0.15948854386806488,
	"Left": 0.3742721676826477,
	"Top": 0.12109502404928207
	},
	"Confidence": 99.9763412475586
	},
	"FaceMatches": [
	{
	"Similarity": 99.99653625488281,
	"Face": {
	"BoundingBox": {
	"Width": 0.204428032040596,
	"Height": 0.1875591278076172,
	"Left": 0.3867680430412925,
	"Top": 0.15587970614433289
	},
	"Confidence": 99.96664428710938,
	"Pose": {
	"Roll": -14.316142082214355,
	"Yaw": -75.81501770019531,
	"Pitch": 18.572364807128906
	},
	"Quality": {
	"Brightness": 88.22943878173828,
	"Sharpness": 38.89601135253906
	}
	}
	}
	],
	"UnmatchedFaces": []
	}

Table 4.2 : Results & Response Obtained from Facial Comparison Operation

4.8.3 Discussions

The results of the experiment demonstrate the resilience and accuracy of Amazon Rekognition API under various image degradation conditions. Across different test cases, including rotation, contrast, brightness, blurring, and the subject not looking at the camera, the API maintained impressive performance. In the rotation test, Amazon Rekognition accurately detected and analyzed facial attributes, with only minimal differences in the reported confidence values when compared to the original photo. When the image contrast was increased to 100%, the API maintained a high level of accuracy in detecting faces and analyzing facial attributes, with confidence values comparable to those of the original photo. For the increased brightness test, Amazon Rekognition effectively detected faces and analyzed facial attributes, albeit with a slight decrease in confidence values for certain attributes. Even with 50% blurring applied to the image, the API maintained a high level of accuracy, although some confidence values showed a decline when compared to the original photo. Lastly, when the subject was not facing the camera, Amazon Rekognition accurately detected the face and provided facial attribute analysis, despite lower confidence values for some attributes compared to the original photo.

Based on the results from the face comparison, we can discuss the following findings. When comparing the original photo with itself, the similarity is 100%, indicating a perfect match. Similarly, when comparing the rotated photo, the contrast-enhanced photo, and the brightness-enhanced photo with the original photo, the similarity remains at 100%. This suggests that the face recognition algorithm can accurately detect and match faces even when the images are rotated or have varying brightness and contrast levels.

However, when comparing the 50% blurred photo with the original photo, the similarity drops slightly to 99.09%. Although the difference is minimal, it indicates that the face recognition algorithm may be slightly less accurate in matching faces when the image quality is reduced by blurring. Finally, when comparing the photo where the subject is not looking at the camera with the original photo, the similarity is 99.9965%. This demonstrates

that the algorithm can still accurately detect and match faces even when the subject's gaze is not directed at the camera.

4.8.4 Summary of Results

In summary, the experiment's results indicate that Amazon Rekognition performs well under various image degradation conditions, maintaining a high level of accuracy in detecting faces and analyzing facial attributes. This suggests that the API could be suitable for use in the FACEIN application, as it can effectively handle a range of image quality levels typically encountered in real-world scenarios. However, it is important to consider potential limitations in extreme cases of image degradation, which may require additional optimization or the incorporation of alternative APIs to improve overall system performance.

Furthermore, the face recognition algorithm effectively detects and matches faces across various conditions, including rotation, brightness, and contrast adjustments, maintaining a 100% similarity. While the similarity drops marginally to 99.09% for the 50% blurred photo, it still performs well. The algorithm also remains highly accurate at 99.9965% similarity when the subject is not looking directly at the camera. Overall, the face recognition algorithm demonstrates strong performance and robustness in various scenarios.

CHAPTER 5

PREMINARY RESULTS

5.1 Introduction

This chapter discusses system requirements specification, use case, and prototype. To better comprehend the system's functionality, a use case diagram and use case descriptions are made. Prototypes are designed using Figma to provide a better vision and a clearer picture of the actual system's design.

5.2 Software Requirements Specification

The functional requirements and non-functional requirements are the two structures in part of the software requirements specification. Functional requirements are more concerned with the features that will be added to the application and used by users. Performance, reliability, security, scalability, usability, adaptability, and product requirement comprise the non-functional criteria. Instead of focusing on functionality, these non-functional requirements are primarily concerned with the requirements for the system's functioning.

5.2.1 Functional Requirements

1. The application shall allow users to create an account using an email and password.
2. The application shall allow users to login into the account using existing email and matching password.
3. The application shall be able to display users' name, email, student id, course, and profile picture in the user profile screen.
4. The applications shall allow users to edit their surname, last name, student id, and course.
5. The applications shall allow users to pick from library or take and upload an image as their profile picture.
6. The application shall allow users to delete their existing account.
7. The application shall allow users to sign out of the application.

8. The application shall allow users to enroll their face for facial recognition check-in verification purpose.
9. The application shall allow users to delete the registered facial data.
10. The application shall allow users to view the checked-in history such as location, date, and time.
11. The application shall allow users to check-in by face recognition.
    a) The application shall be able to display users' name, email, student id, course, location, current date, and current time upon successful facial recognition verification.

5.2.2 Non-Functional Requirements

i. Performance

- Any changes made by the users should be updated within 3 seconds in real-time.
- The application shall not take more than 3 seconds to start up the loading screen.
- The application shall not hinder user input.

ii. Reliability

- The application shall be able to handle data faster by streamlining the process of storing and accessing data.

iii. Security

- Users shall receive a warning message when performing critical actions such as deleting the account.

iv. Usability

- The application's user interface (UI) shall make it simple to reverse most of the functionalities.
- The application's user interface (UI) needs to be simple enough for users to navigate without expert guidance.

v. Adaptability

- The application should be adapted for all devices of Android.

vi. Product Requirement

- The application must be available at all times 24/7.

5.3 Use Case

5.3.1 Use Case Diagram

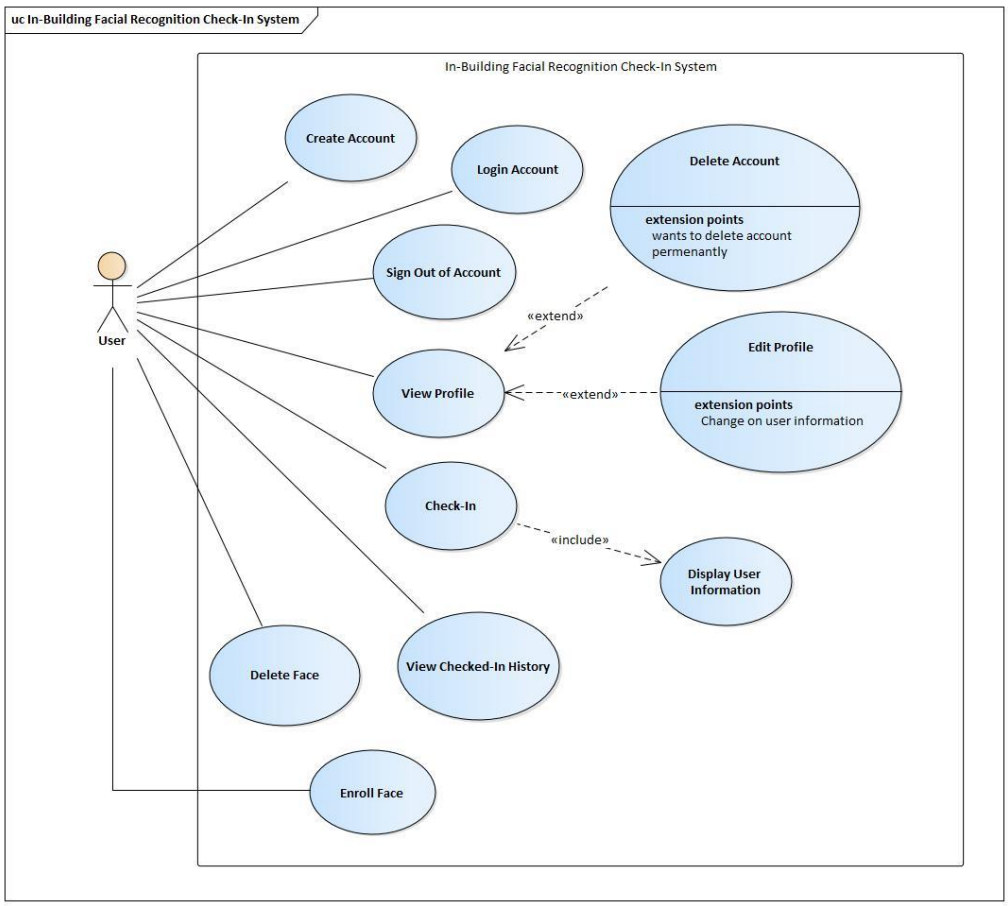


Figure 5.1: In-Building Facial Recognition Check-In System Use Case Diagram

5.3.2 Use Case Description

Table 5.1: Use Case Description of Create Account

<table>
<tr><td>Use case name: Create Account</td><td>ID: 1</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:

User: can check-in into building using the application</td></tr>
<tr><td colspan="3">Brief Description:

This use case describes how the user creates a new account and input their personal details throughout the application.</td></tr>
<tr><td colspan="3">Trigger:
User(s) who wishes to check-in into the building using the application.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: -
Extend: -
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

1. User wants to check-in into the building using the application.
2. User creates a new account using a UTAR email and password.
- <u>Continue to S2: Format of password- sub-flow 2.1</u>

3. User enters surname, last name, email, password, and confirm password.
4. The system saves and store information of user into the database.
5. The application then proceed to the “Login Account” use case (ID 2).</td></tr>
<tr><td colspan="3">Sub-flows:
<u>S2: Format of password</u>
2.1 Users’ password must include at least one lowercase character, one uppercase character, one special case character, and a minimum length of 8 characters.</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
N/A</td></tr>
</table>

Table 5.2: Use Case Description of Login Account

<table>
<tr><td>Use case name: Login Account</td><td>ID: 2</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
User: can login to the check-in application to perform application functions.</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how the user login into the account to perform application functions.</td></tr>
<tr><td colspan="3">Trigger:

When User who wants to login to the application to perform application functions.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: -
Extend: -
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

1. User wants to login into the application to perform their specific functions.
2. User enter email and password.
If: the email and password entered correctly, the application will login successfully and redirect to the profile screen.
Else: the application will prompt “Invalid login, please enter a correct email and password”.

3. Successful login to the application enables user to “Sign Out of Account” use case (ID 3), “View Profile” use case (ID 4), “Edit Profile” use case (ID 5), “Delete Account” use case (ID 6), and “Check-In” use case (ID 7), “Enroll Face” use case (ID 9), “Delete Face” use case (ID 10), “View Checked-In History” use case (ID 11).</td></tr>
<tr><td colspan="3">Sub-flows:
-</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
N/A</td></tr>
</table>

Table 5.3: Use Case Description of Sign Out of Account

Use case name: Sign out of Account	ID: 3	Importance level: High
Primary actor: User	Use case type: Detailed, Real	
Stakeholders and Interests: User: Wants to sign out of their account		
Brief Description: This use case describes how the user can sign out of their account using the application.		
Trigger: When User wants to sign out of their account		
Relationship: Association: User Include: - Extend: - Generalization: -		
Normal flows of event: 1. User logins into the account. 2. User wants to sign out of the application. 3. User proceeds to the **"Log out"** functions. 4. The application logs user out of the application and back to the .		
Sub-flows: -		
Alternate/ Exceptional Flows: N/A		

Table 5.4: Use Case Description of View Profile

<table>
<tr><td>Use case name: View Profile</td><td>ID: 4</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
User: can view their profile.</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how the user can view their profile detail.</td></tr>
<tr><td colspan="3">Trigger:

When User wants to view their profile detail.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: -
Extend: Edit Profile (ID 5), Delete Account (ID 6)
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

5. User logins into the account.
6. User proceeds to profile screen.
7. The application then display the name, email, student id, course, and profile picture of user.
8. User has an option to either edit their account or delete their account then proceeds to “Edit Profile” use case (ID 5), “Delete Account” use case (ID 6) respectively.</td></tr>
<tr><td colspan="3">Sub-flows:
-</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
1. If there are no profile editing made by user
- The Application will only display name, and email by default</td></tr>
</table>

Table 5.5: Use Case Description of Edit Profile

<table>
<tr><td>Use case name: Edit Profile</td><td>ID: 5</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
User: can edit their profile</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how the user can edit their profile detail.</td></tr>
<tr><td colspan="3">Trigger:

When user wants to edit their profile detail.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: -
Extend: “View Profile” use case (ID 4)
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

1. User logins into the account.
2. “View Profile” use case (ID 4) is performed, and user profile is displayed.
3. User proceeds to the “Edit Profile” functions.
4. User selects the information they would like to edit.
- <u>Continue to S5: Information to be Edited- sub-flow 5.1, 5.2</u>

5. System update user saved information into the database.
6. “View Profile” use case (ID 4) is performed, and user profile is displayed with updated information.</td></tr>
<tr><td colspan="3">Sub-flows:
<u>S5: Information to be Edited</u>
5.1 User can choose to edit their surname, last name, student id, and course
5.2 User can choose to take or upload their profile picture</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
N/A</td></tr>
</table>

Table 5.6: Use Case Description of Delete Account

<table>
<tr><td>Use case name: Delete Account</td><td>ID: 6</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
User: can delete their account</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how the user wants to remove their account permanently.</td></tr>
<tr><td colspan="3">Trigger:

When user wants to delete their account.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: -
Extend: “View Profile” use case (ID 4)
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

1. User logins into the account.
2. “View Profile” use case (ID 4) is performed, and user profile is displayed.
3. User proceeds to the “Edit Profile” functions.
4. User proceeds to the “Delete Account” functions.
5. User wants to delete the account.
6. The application prompts user to confirm delete account.
7. The system remove user account from the database.</td></tr>
<tr><td colspan="3">Sub-flows:
N/A</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
1. Once user decide and confirm to delete the account, their account cannot be restored.</td></tr>
</table>

Table 5.7: Use Case Description of Check-In

<table>
<tr><td>Use case name: Check-In</td><td>ID: 7</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
User: can check-in building using facial recognition system.</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how the user can check-in into the building using face recognition.</td></tr>
<tr><td colspan="3">Trigger:

When user wants to check-in into a building.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: “Display User Information” use case (ID 8).
Extend: -
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

1. User logins into the account.
2. User proceeds to the “Check-in” functions.
3. User verify their identity using facial recognition.
4. System match the face with the profile picture image data in database.
5. User identity is verified, and “Display User Information” use case (ID 7) is performed, and user information along with verified button is displayed.</td></tr>
<tr><td colspan="3">Sub-flows:
N/A</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
1. If the there is no face detected, it will display the popup message “No Face Detected”.
2. If the user’s face is detected but does not match the registered Face, it will display the popup message “Failed to Check In, Please Try Again”.</td></tr>
</table>

Table 5.8: Use Case Description of Display User Information

<table>
<tr><td>Use case name: Display User Information</td><td>ID: 8</td><td>Importance level: High</td></tr>
<tr><td>Primary actor: User</td><td colspan="2">Use case type: Detailed, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
User: can check-in building using facial recognition system and display their verified identity.</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how the user can check-in into the building using face recognition and display their verified identity.</td></tr>
<tr><td colspan="3">Trigger:

When user wants to check-in into a building and display their verified identity.</td></tr>
<tr><td colspan="3">Relationship:
Association: User
Include: “Display User Information” use case (ID 8).
Extend: -
Generalization: -</td></tr>
<tr><td colspan="3">Normal flows of event:

1. User logins into the account.
2. “Check-in” use case (ID 7) is performed.
3. User proceeds to the “Display User Information” functions.
4. The application display user’s name, student id, course along with verified button.
5. User enters the building.</td></tr>
<tr><td colspan="3">Sub-flows:
N/A</td></tr>
<tr><td colspan="3">Alternate/ Exceptional Flows:
N/A</td></tr>
</table>

Table 5.9: Use Case Description of Enroll Face

Use case name: Enroll Face	ID: 9	Importance level: High
Primary actor: User	Use case type: Detailed, Real	
Stakeholders and Interests: User: register Face ID for facial-recognition check-in purpose		
Brief Description: This use case describes how the user can enroll their face by scanning their facial features for facial recognition verification.		
Trigger: When user wants to enroll face by registering their Face ID.		
Relationship: Association: User Include: - Extend: - Generalization: -		
Normal flows of event: 1. User logins into the account. 2. User proceeds to the **"Enroll Face"** functions. 3. User scan their face to register the facial feature. 4. User's face is detected and successfully stored into the database.		
Sub-flows: N/A		
Alternate/ Exceptional Flows: 1. If the there is no face detected, it will display the popup message "No Face Detected". 2. Only one Face ID is allowed to be registered per account. 3. User can only register the face again only if the current is removed.		

Table 5.10: Use Case Description of Delete Face

Use case name: Delete Face	ID: 10	Importance level: High
Primary actor: User	Use case type: Detailed, Real	
Stakeholders and Interests: User: can delete their enrolled face		
Brief Description: This use case describes how the user wants to remove their enrolled face.		
Trigger: When user wants to remove their face or wants to register a new one.		
Relationship: Association: User Include: - Extend: - Generalization: -		
Normal flows of event: 1. User logins into the account. 2. User proceeds to the **"Delete Face"** functions. 3. User wants to delete the enrolled face. 4. The application prompts user to confirm delete the enrolled face. 5. The system remove user's enrolled face from the database.		
Sub-flows: N/A		
Alternate/ Exceptional Flows: 1. Once user decide and confirm to delete the enrolled face, it cannot be restored.		

Table 5.11: Use Case Description of View Checked-In History

Use case name: View Checked-In History	ID: 11	Importance level: High
Primary actor: User	Use case type: Detailed, Real	
Stakeholders and Interests: User: can view their checked-in record in history screen.		
Brief Description: This use case describes how the user can view their successfully checked-in history.		
Trigger: When user wants to view their checked-in history.		
Relationship: Association: User Include: - Extend: - Generalization: -		
Normal flows of event: 1. User logins into the account. 2. User proceeds to the **"History"** functions. 3. User wants to view the checked-in record. 4. A List of checked-in record such as location, date, and time is displayed.		
Sub-flows: N/A		
Alternate/ Exceptional Flows: 1. "No Record Found" will be displayed if there is no check-in history. 2. For record older than a month will be automatically deleted.		

5.4 Prototype

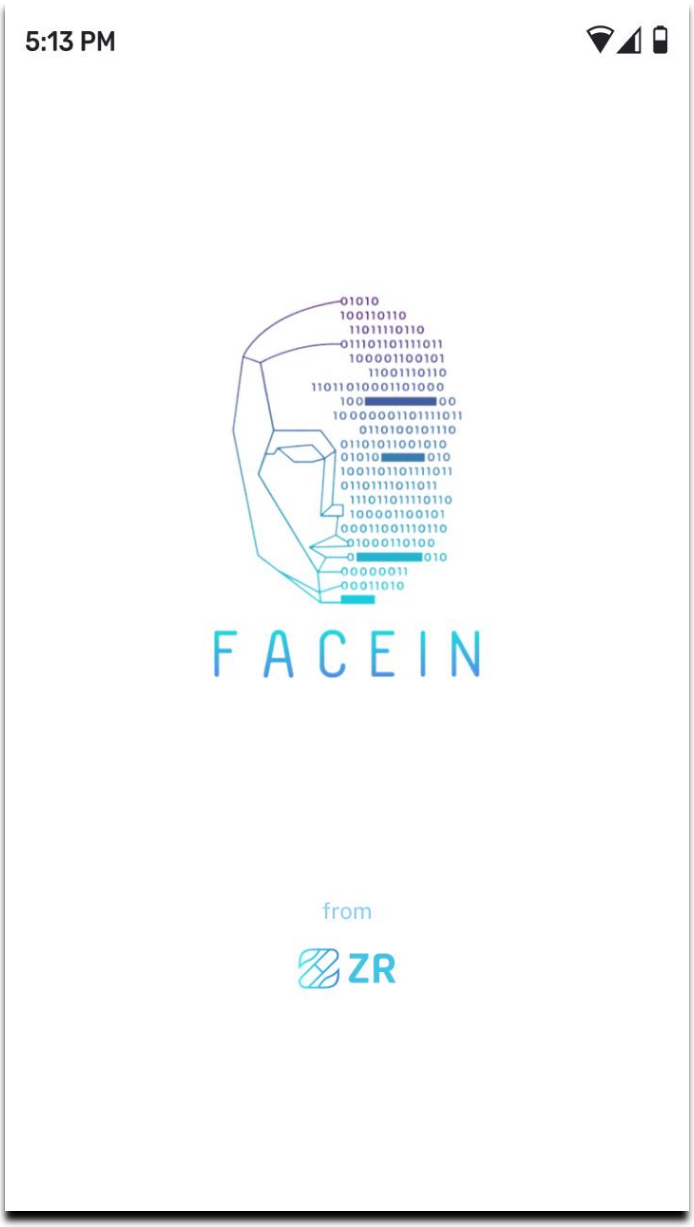


Figure 5.2: Loading Screen

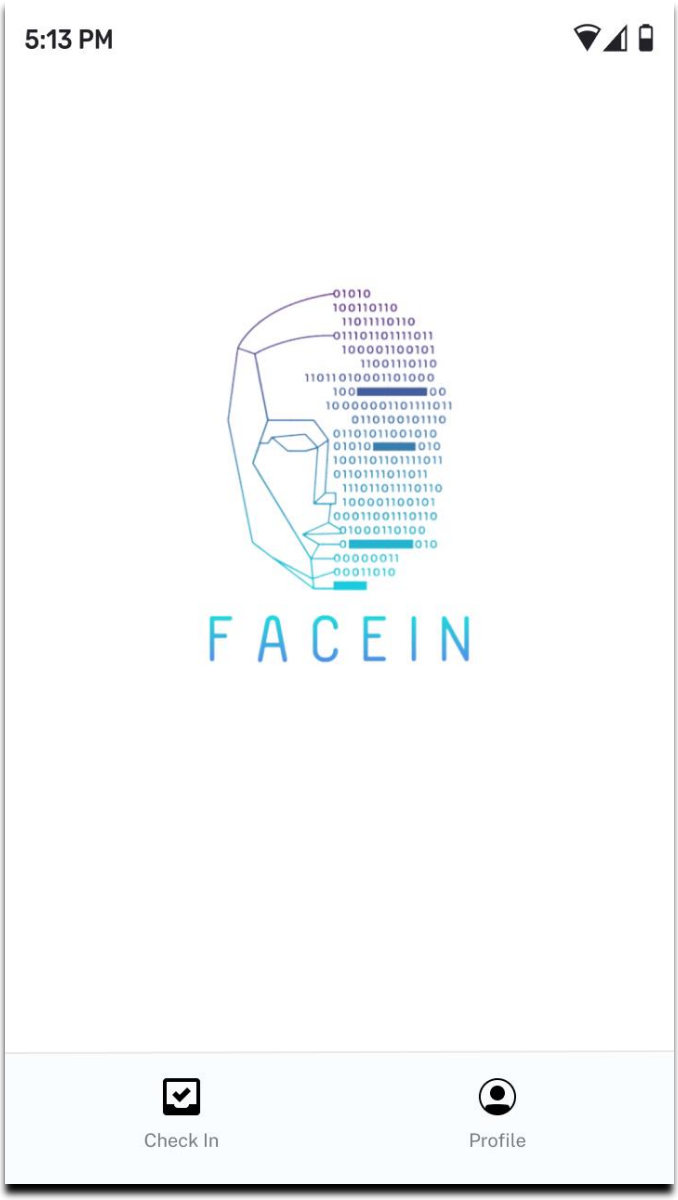


Figure 5.3: Home Screen

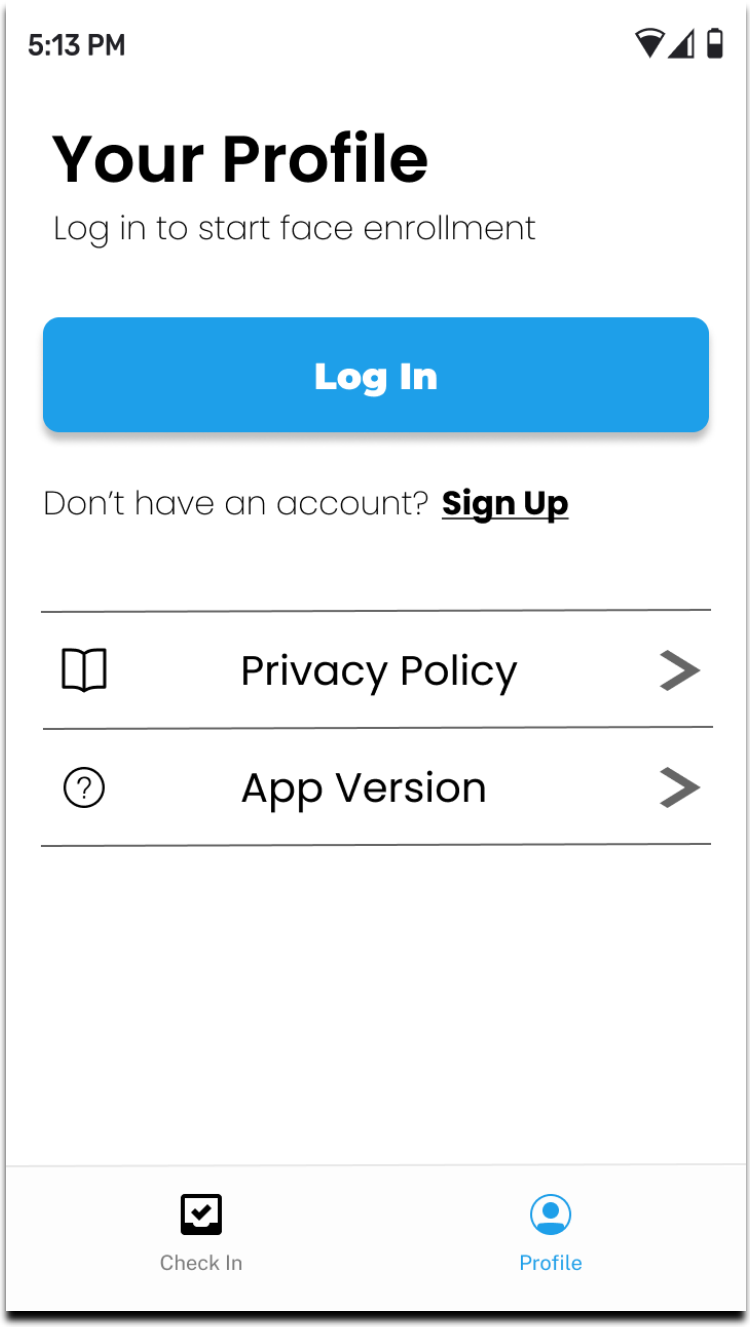


Figure 5.4: Profile Screen

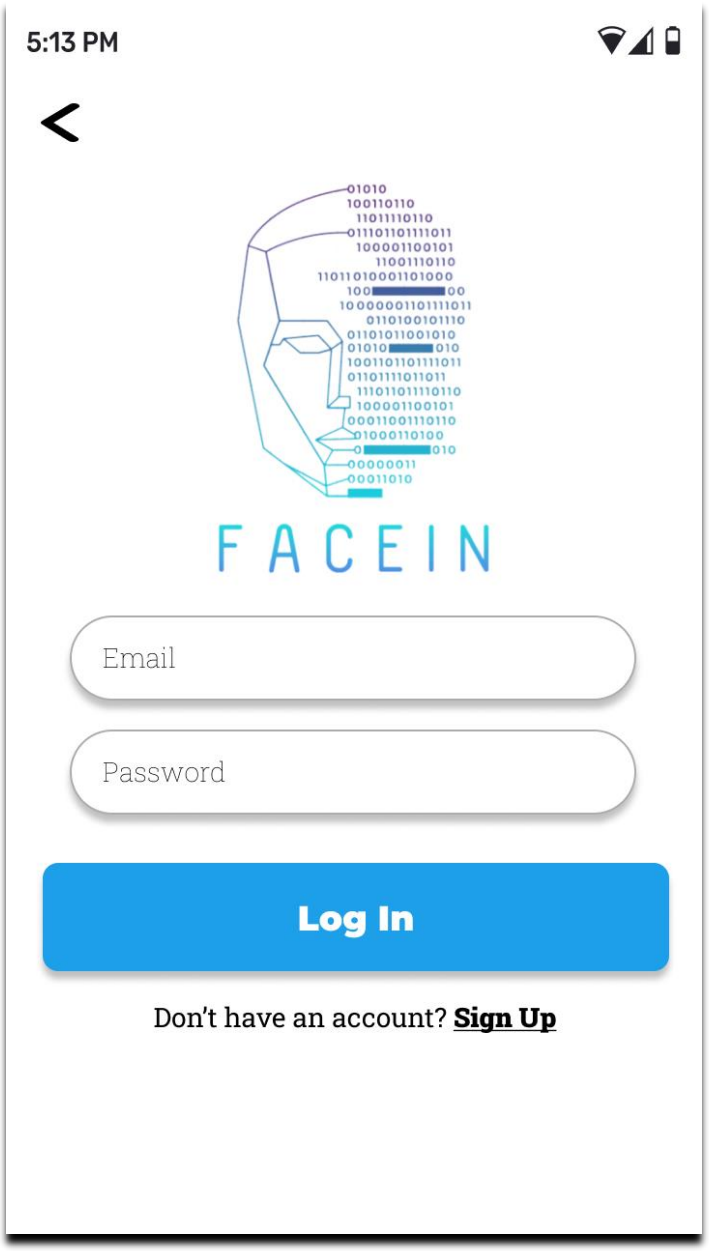


Figure 5.5: Login Screen

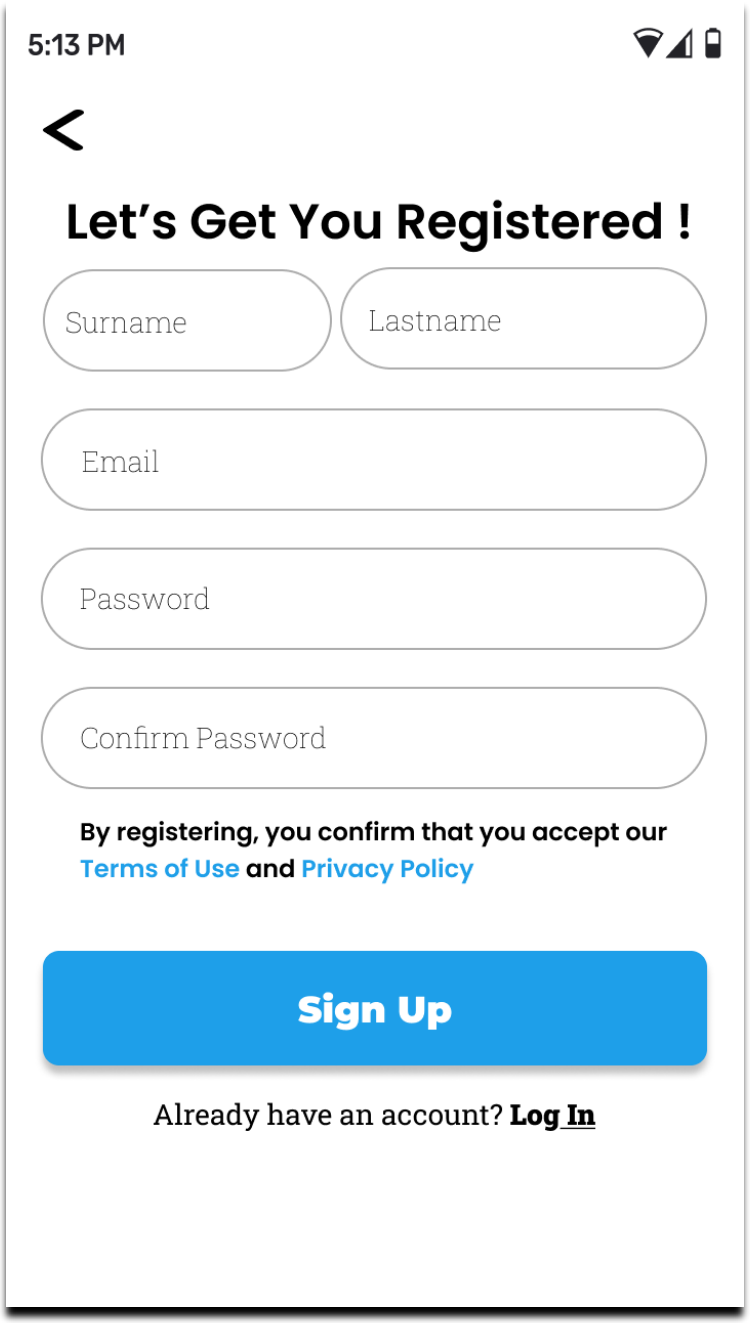


Figure 5.6: Sign Up Screen

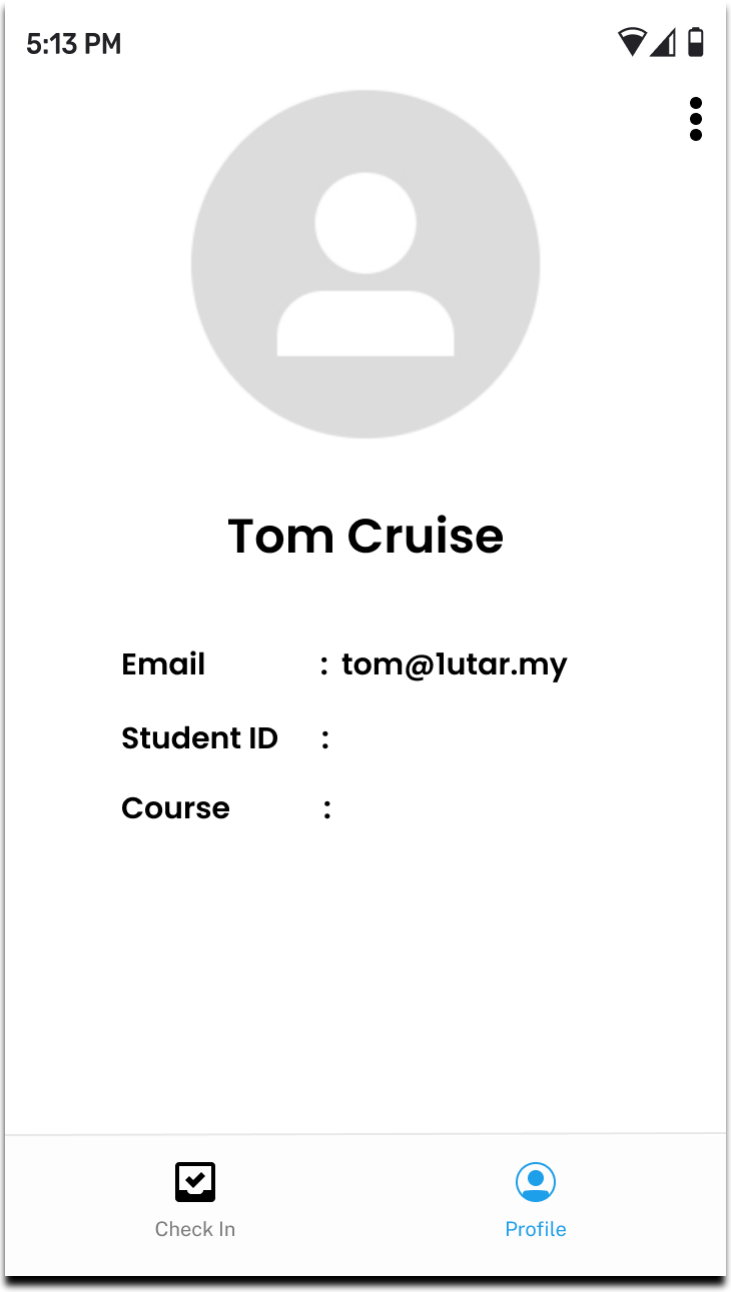


Figure 5.7: After Login Screen

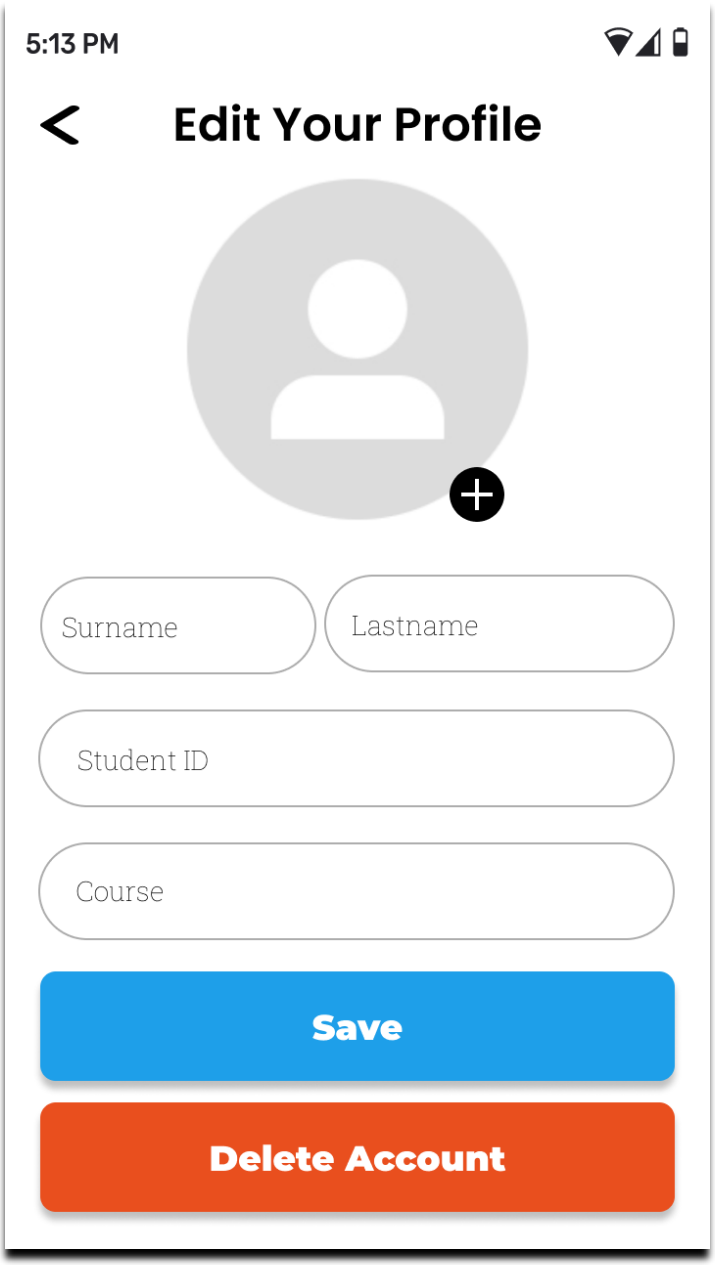


Figure 5.8: Edit Screen

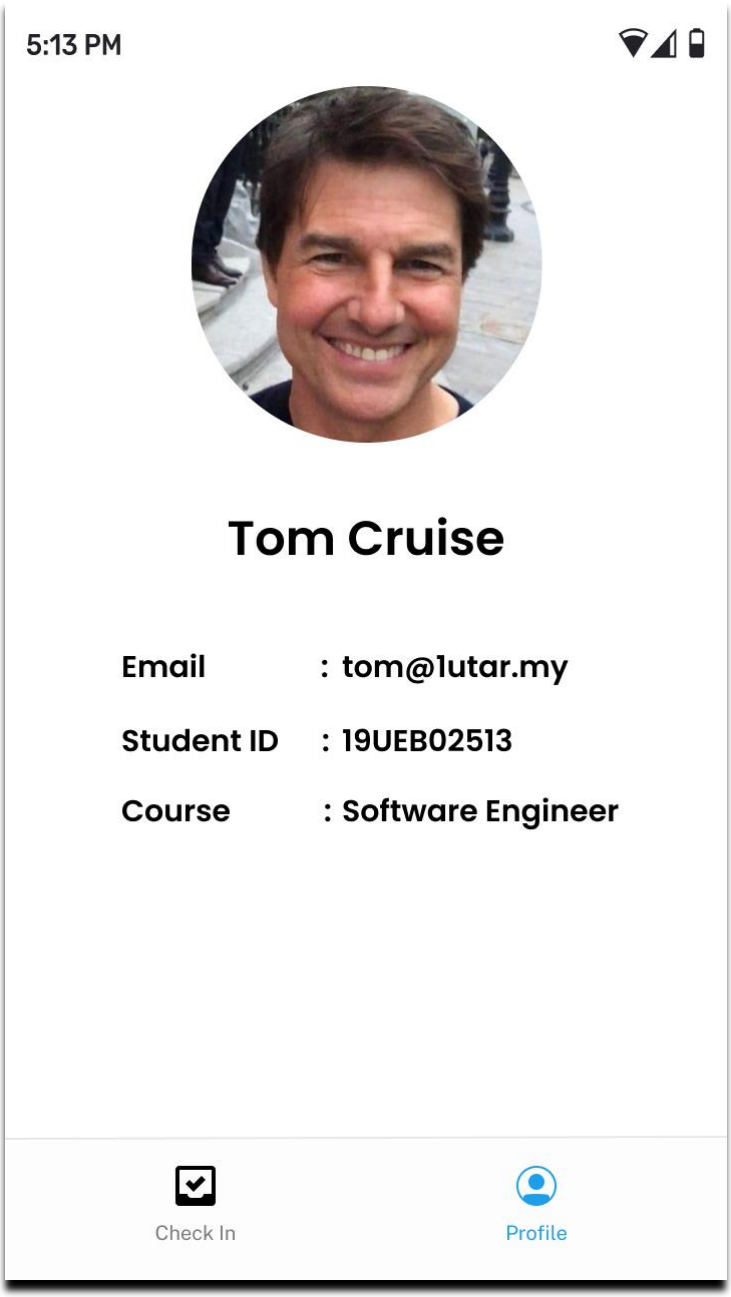


Figure 5.9: After Editing Profile Screen

Figure 5.10: Check-In Screen

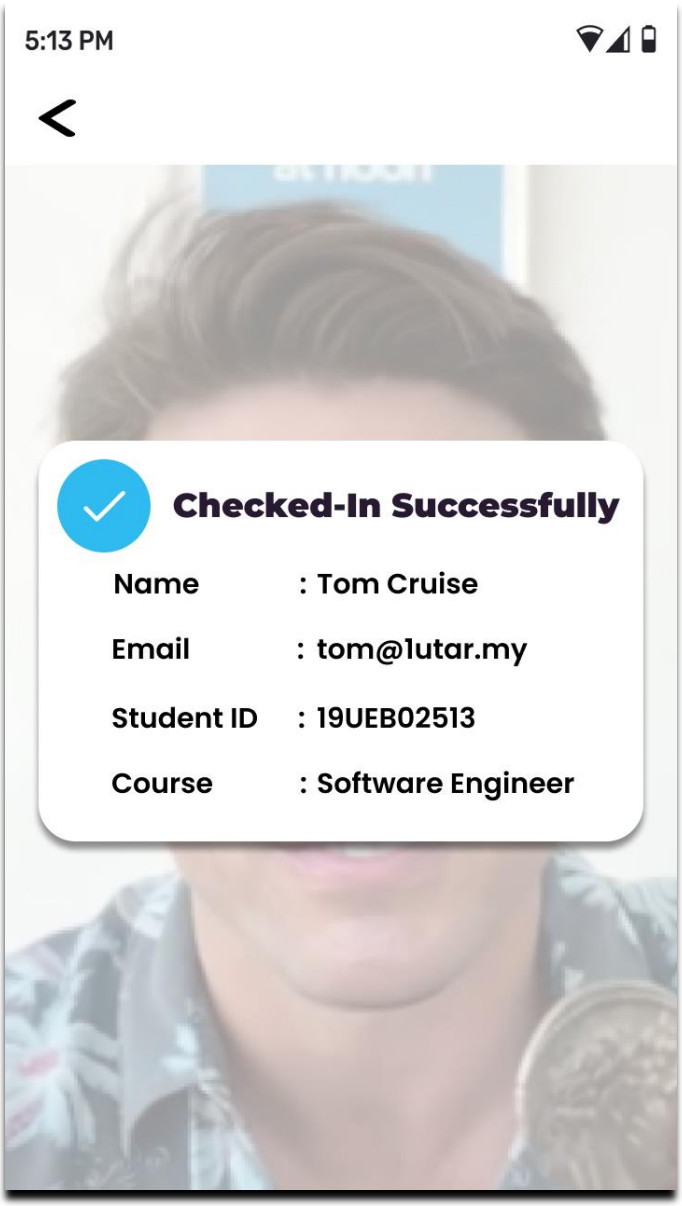


Figure 5.11: Successfully Checked-In Screen

CHAPTER 6

SYSTEM DESIGN

6.1 Introduction

This section provides an overview of the system design for the In-building Facial Recognition Check-in Application. The design encompasses various modules and components that work together to deliver the desired functionality. This chapter will outline the system architecture, database design, user interface design, and various functional components that constitute the system.

6.2 System Architecture

This section outlines the overall system architecture, including the main components and their interactions.

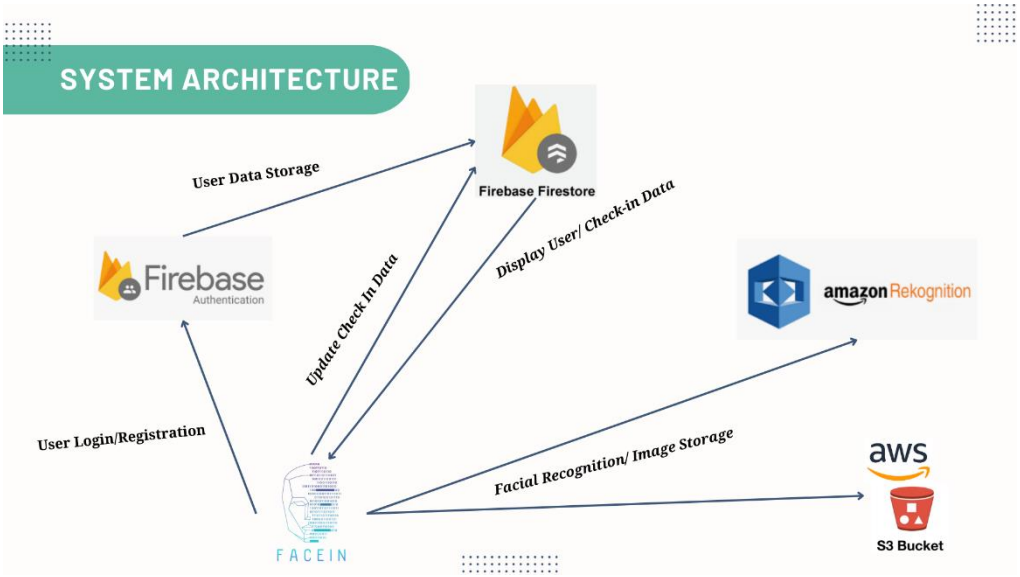


Figure 6.1 : Main System Components and Interactions

In Figure 6.1, the main components of the system interact with each other to provide a seamless experience for users. The User Interface serves as the point of interaction for users, allowing them to navigate through the application's various features. Firebase Authentication is responsible for managing user registration and login processes, verifying user credentials, and providing secure access to the system. Firebase Firestore stores and retrieves user data and check-in information, enabling real-time synchronization of data across the application.

AWS Rekognition and S3 Bucket work together to facilitate facial recognition and store user images securely.

6.2.1 System Components and Database Design

User Interface

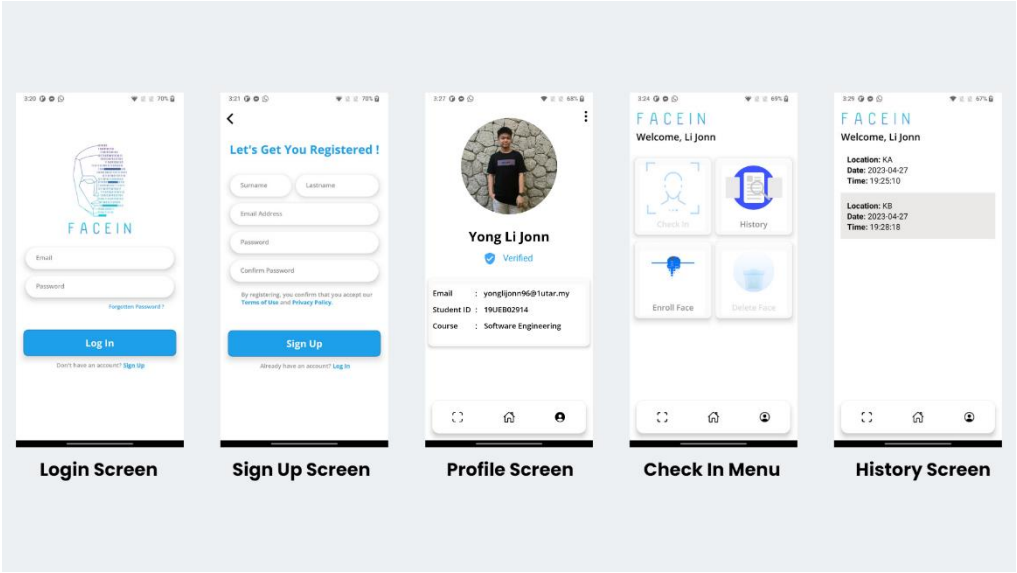


Figure 6.2 : User Interface Component

In Figure 6.2, the User Interface component consists of multiple screens that provide users with access to the application's features. These screens include login, registration, profile, check-in menu, and history screens, allowing users to interact with the system, manage their account, perform check-ins, and view their check-in history.

Firebase Authentication

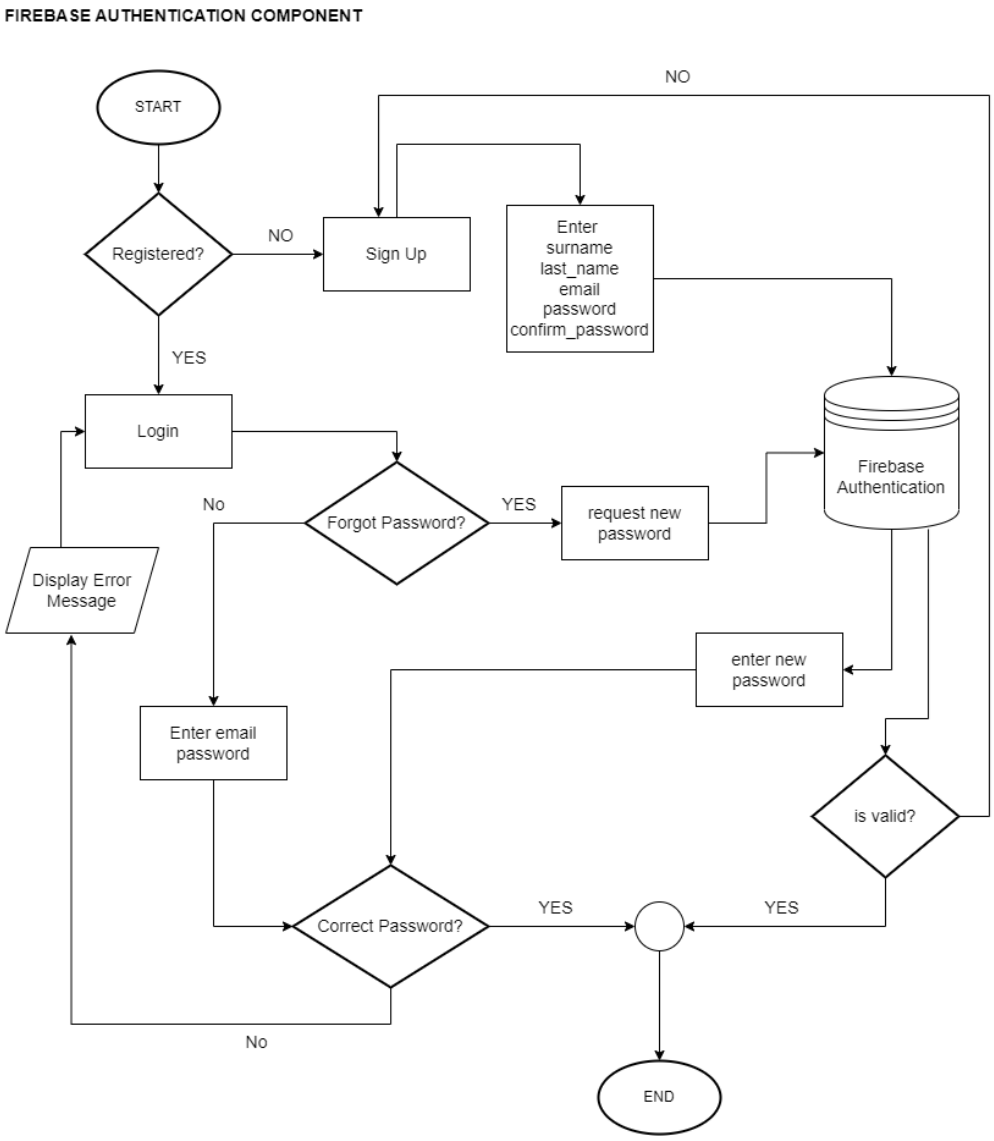


Figure 6.3 : Firebase Authentication Component

As shown in Figure 6.3, Firebase Authentication component manages the user registration and login processes. When a user attempts to register or log in, their credentials are passed to Firebase Authentication, which validates the provided information. If the credentials are valid, the user is granted access to the application and navigate to the Home Screen otherwise, an error message is displayed.

Firebase Firestore

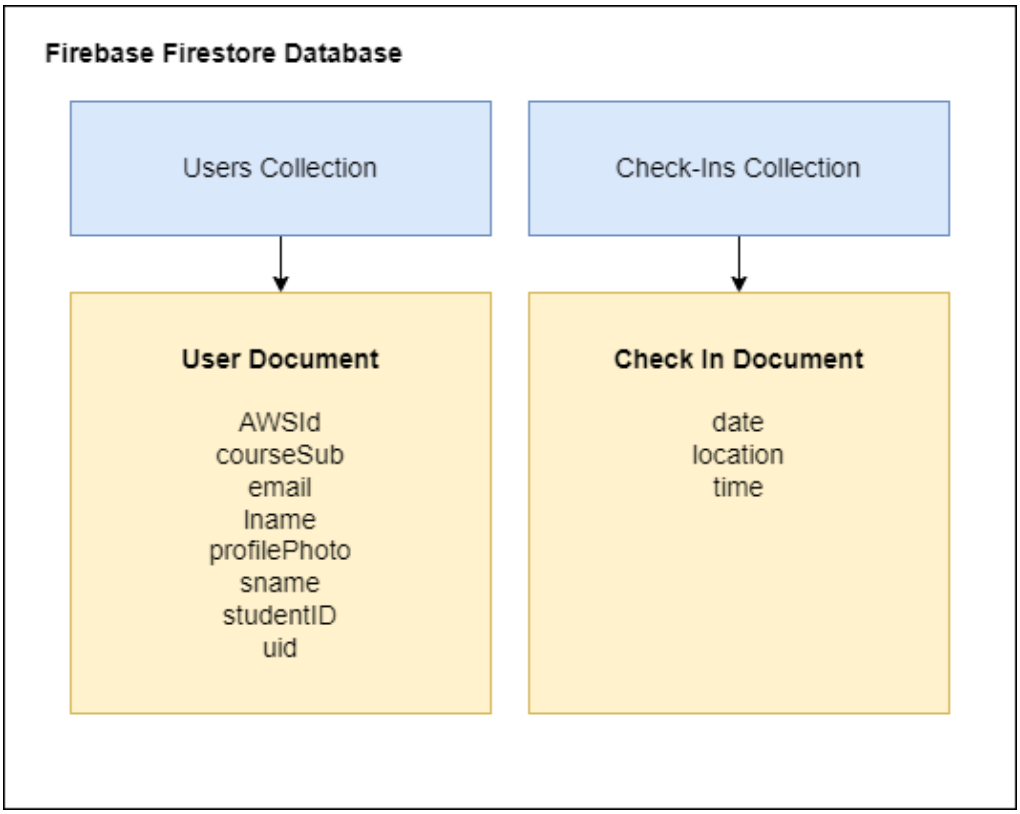


Figure 6.4 : Firebase Firestore Component and Design

In Figure 6.4, Firebase Firestore is depicted as the primary database for storing and managing user data and check-in information. The data is organized into collections and documents, allowing for efficient querying and real-time synchronization across the application.

Amazon Web Services (AWS) - Rekognition and S3 Bucket

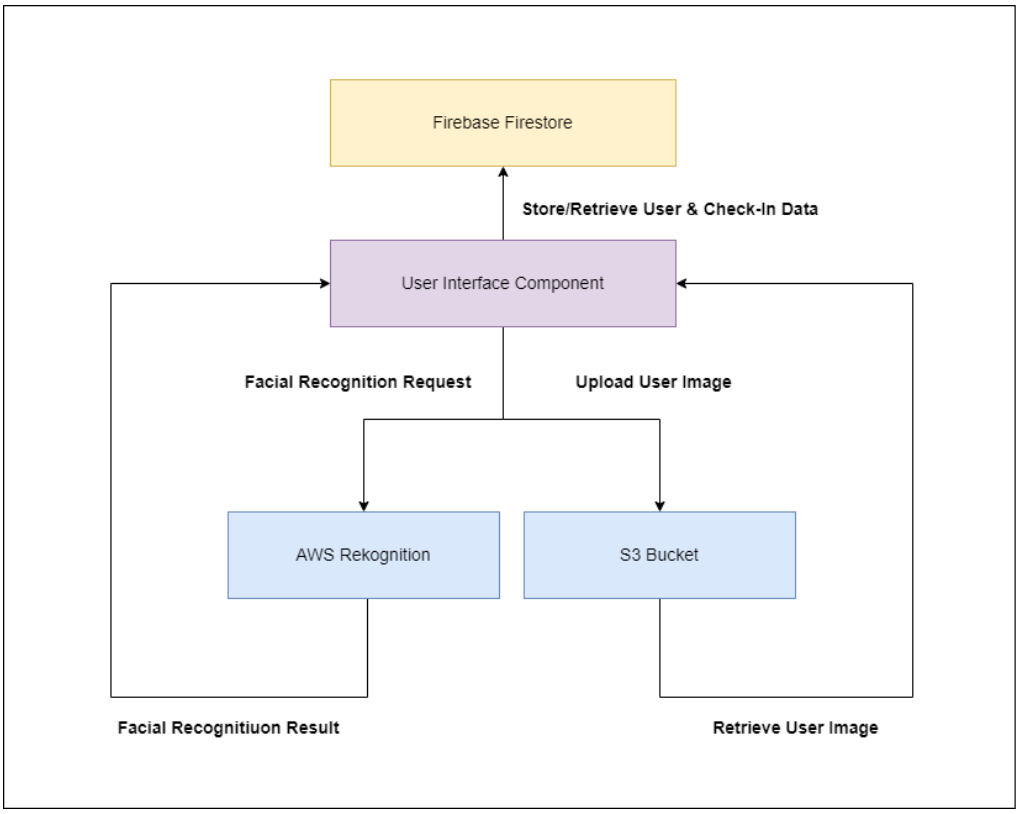


Figure 6.5 : AWS Rekognition and S3 Bucket Components and Design

As illustrated in Figure 6.5, AWS Rekognition and S3 Bucket work together to provide facial recognition capabilities and store user images securely. AWS Rekognition processes user images and extracts facial features to enable accurate identity verification during check-ins. Meanwhile, the S3 Bucket stores user images, allowing for efficient retrieval and reducing the load on the application server.

6.2.2 Interactions between Components

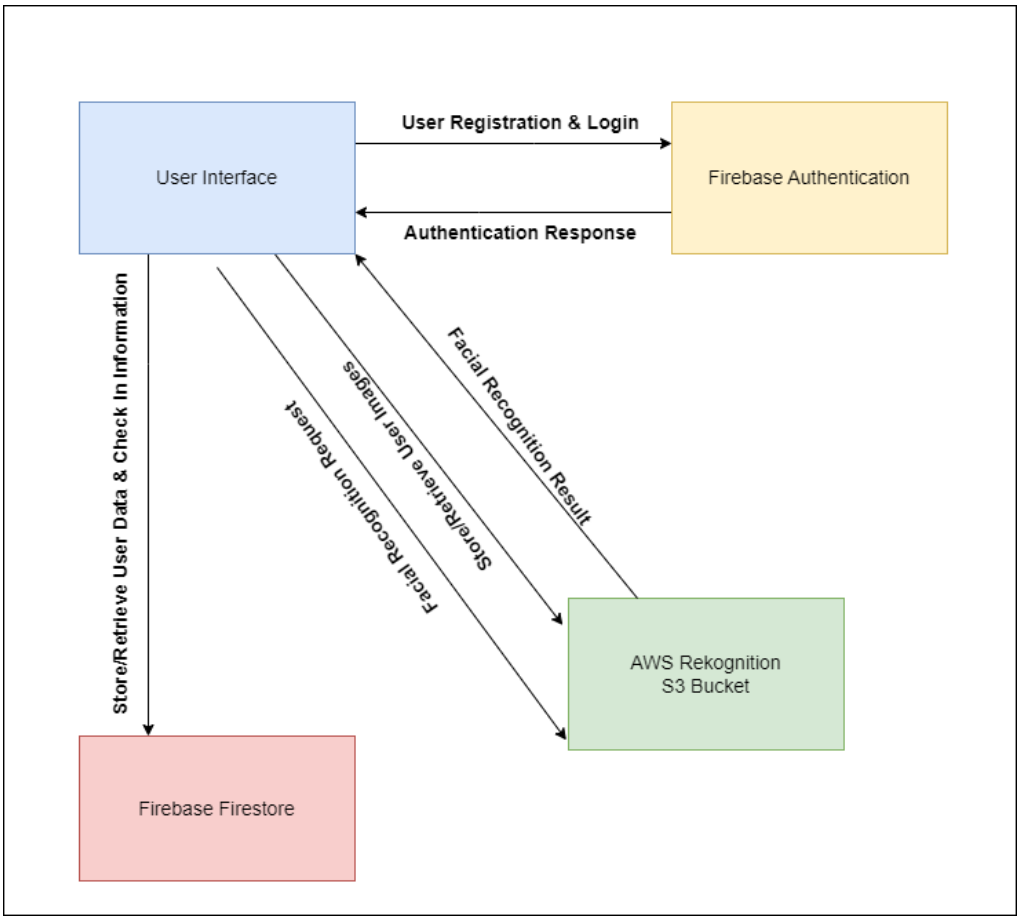


Figure 6.6 : Interactions between Components

Figure 6.6 shows the interactions between the system components. The User Interface communicates with Firebase Authentication to manage user registration and login processes. It also interacts with Firebase Firestore to store and retrieve user data and check-in information. AWS Rekognition and S3 Bucket are used to provide facial recognition capabilities and secure storage for user images. The User Interface sends images to AWS Rekognition for facial feature extraction and comparison during the check-in process. Once the facial features are extracted and compared, the results are sent back to the User Interface to determine whether the check-in is successful or not. User images are stored and retrieved from the S3 Bucket as needed.

6.3 User Interface Design

The figures below shows the user interface designed in this application and screenshots depicting the potential interface of this application are attached below along with their respective descriptions,.

Splash Screen

Figure 6.7 showcases the Splash Screen of the application, which greets users upon launching the app. This screen features the application's logo and branding, creating an inviting and visually appealing introduction. As the application loads its essential resources in the background, the Splash Screen ensures a smooth transition to the next phase of user interaction, such as the Sign Up or Login Screen.

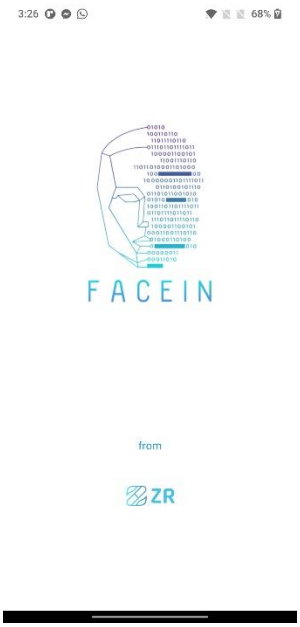


Figure 6.7 : Splash Screen

Sign Up Screen

The screenshots below depict the application's account registration process, which users must complete to access the application's functionalities. Figure 6.8 displays the account registration fields, which require users to provide necessary information. After successfully validating the user's input format and completing the registration, the application directs the user to the Home Screen.

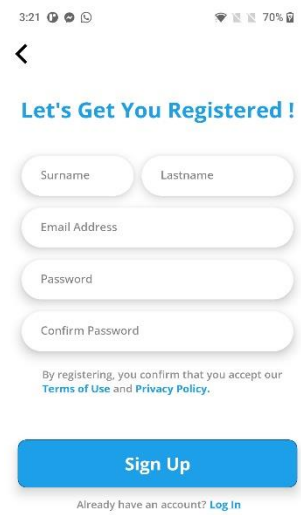


Figure 6.8 : Sign Up Screen

Login Screen

The Login Screen, as shown in Figure 6.8, prompts users to enter their email address and password to access the application. Upon successful validation of the provided credentials, users are navigated to the Home Screen.

Figure 6.9 : Login Screen

Forgotten Password Screen

In case users forget their password, the Forgotten Password Screen, depicted in Figure 6.10, allows them to reset it. By entering their email address, a password reset link is sent to the provided email, enabling users to regain access to their account.

Figure 6.10 : Forgotten Password Screen

Home Screen

The Home Screen, displayed in Figure 6.11, serves as a welcoming interface that offers a visually appealing layout. Although no interactions take place on this screen, it features a bottom navigation tab that allows users to navigate to the Profile Landing Screen and Check-In Menu Screen.

Figure 6.11 : Home Screen

Profile Landing Screen

Figure 6.12 presents the Profile Landing Screen, which contains a small container featuring the user's mini profile picture and full name. Additional options on this screen include a dark theme toggle switch, privacy policy access, app version display, and a log-out button.

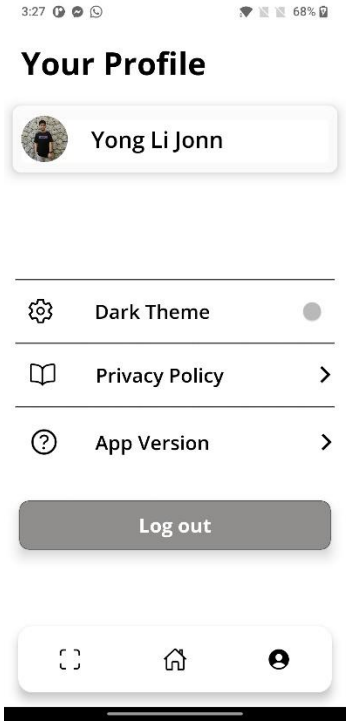


Figure 6.12 : Profile Landing Screen

Profile Screen

The Profile Screen, shown in Figure 6.13, displays the user's information, including their profile picture, full name, email, student ID, and course subject. Additionally, there is a button for users to verify their profile through email verification, ensuring the accuracy and legitimacy of the provided information.

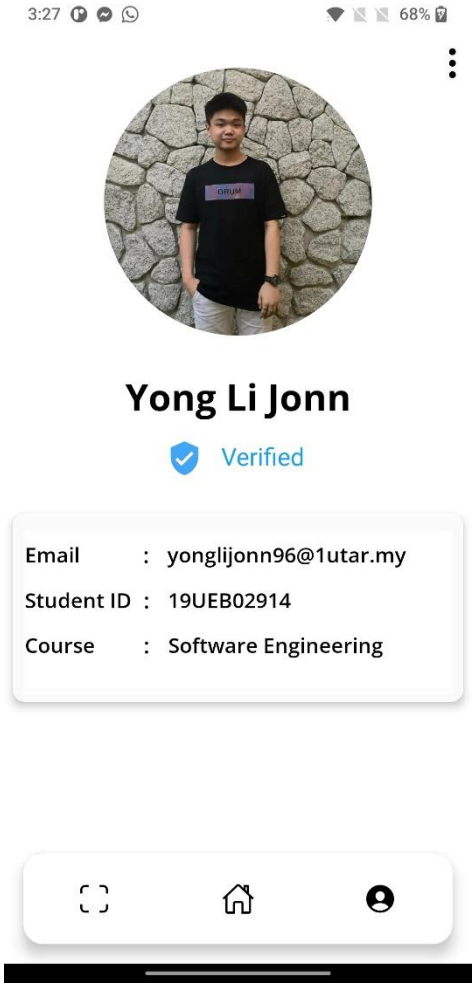


Figure 6.13 : Profile Screen

Edit Profile Screen

The Edit Profile Screen, shown in Figure 6.14, allows users to modify their information, such as changing their profile picture, surname, lastname, student ID, and course subject. To change their profile picture, users can tap on the '+' icon and choose between taking a new picture or selecting an existing one from their photo album. After making a selection, users can click on the "Upload" button to change their profile picture. To update the modified data in Firebase, users must click on the "Save" button, after which the updated user information

will be reflected on the Profile Screen. In the Edit Profile Screen, users can also delete their account by pressing the "Delete" button.

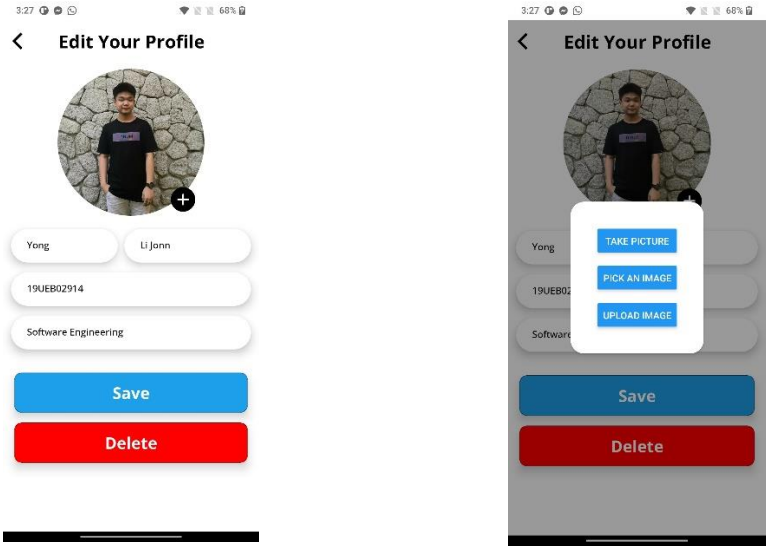


Figure 6.14 : Edit Profile Screen

Check-In Menu Screen

The Check-In Menu Screen features four buttons: Check-in, History, Enroll Face, and Delete Face. Before users have filled in all their information, the Enroll Face function cannot be carried out. Figure 6.15 displays the state of the buttons before face enrollment, with the Check-in and Delete Face buttons disabled. Figure 6.16 shows the state of the buttons after face enrollment, where the Check-in and Delete Face buttons are enabled, and the Enroll Face button is disabled.

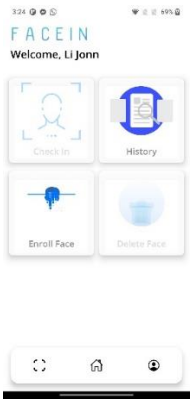


Figure 6.15 : Before Face Enrollment

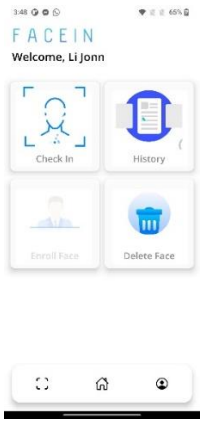


Figure 6.16 : After Face Enrollment

Enroll Face Screen

In the Enroll Face Screen, depicted in Figure 6.17, users can open the camera, capture a photo, and enroll their face for subsequent verification during check-ins.

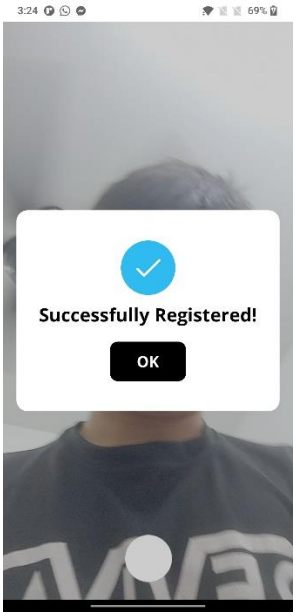


Figure 6.17 : Enroll Face Screen

Check-In Screen

As shown in Figure 6.18, the Check-In Screen allows users to open the camera, capture a photo, and verify whether their face matches the enrolled face. When a match is found, the user's information and check-in details are displayed.

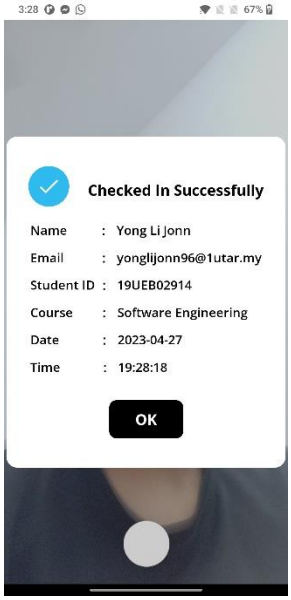


Figure 6.18 : Check In Screen

History Screen

The History Screen, displayed in Figure 6.19, presents a record of the user's check-ins, providing an overview of their attendance history.

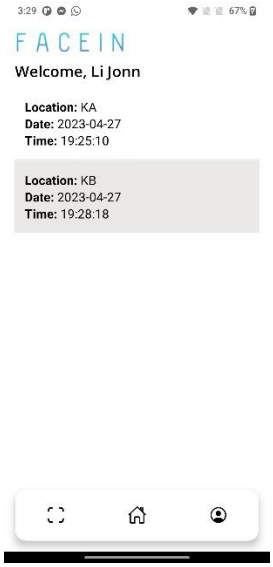


Figure 6.19 : History Screen

Dark Theme Mode

Figure 6.20 illustrates the Dark Theme Mode, where users can customize their experience by switching between dark and light theme modes according to their preferences.

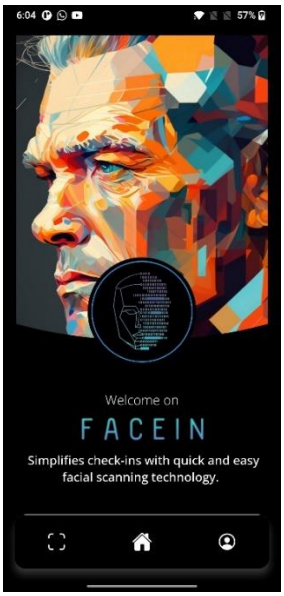


Figure 6.20 : Dark Theme Mode

6.4 Summary

This chapter presents the system design for the In-building Facial Recognition Check-in Application. The design includes the system architecture, database design, and user interface design. The system architecture highlights the interactions between the main components, while the database design focuses on the structure and organization of data. The user interface design covers the layout, navigation, and user experience, and the functional components describe the key features of the system, such as user authentication, user identity management, facial recognition, and displaying checked-in history.

CHAPTER 7

IMPLEMENTATION

7.1 Introduction

This section details the execution carried out for the creation of the In-building Facial Recognition Check-in Application, showcasing each design aspect involved in the implementation along with a concise explanation and workflow. The primary focus of this project is on the user-facing elements of the system, and as such, all modules and designs described in this section pertain to the user side of the application.

7.2 User Authentication Design

Upon reaching the authentication page, users will be presented with two options: logging in or registering for the application. This authentication design is specifically designed to manage user registration and login processes.

7.2.1 Creating Account

a. User registration or account creation function is carried out through Firebase Authentication.
b. Users can register and sign up for the application as a new user using their UTAR email and password, along with additional details such as their surname and last name, if they do not already have an existing account.
c. Upon clicking the sign-up button, a validation process is performed to ensure that the user inputs the correct email format with '@1utar.my' and requires the user's password to contain at least one lowercase character, one uppercase character, one special case character, and a minimum length of 8 characters.
d. Once the user provides all necessary information and submits the form, their details will be saved in the Firebase Firestore collection.

e. Successful account creation and sign-up will allow users to access the home page of the application while updating their profile as verified in the Profile Screen.

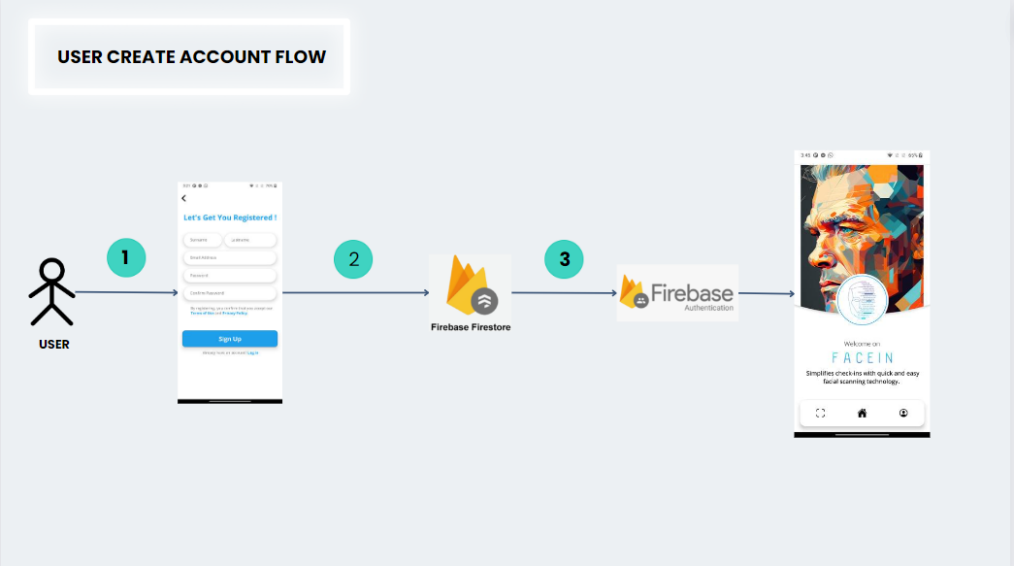


Figure 7.1: User Registration Architecture

7.2.2 Login Account

a. User login is carried out through Firebase Authentication, where users can log in to the application using their registered email and password.

b. The email and password are validated to ensure that they are registered, and this process is carried out by Firebase Authentication.

c. Upon successful verification, the user's details will be retrieved for use within the application.

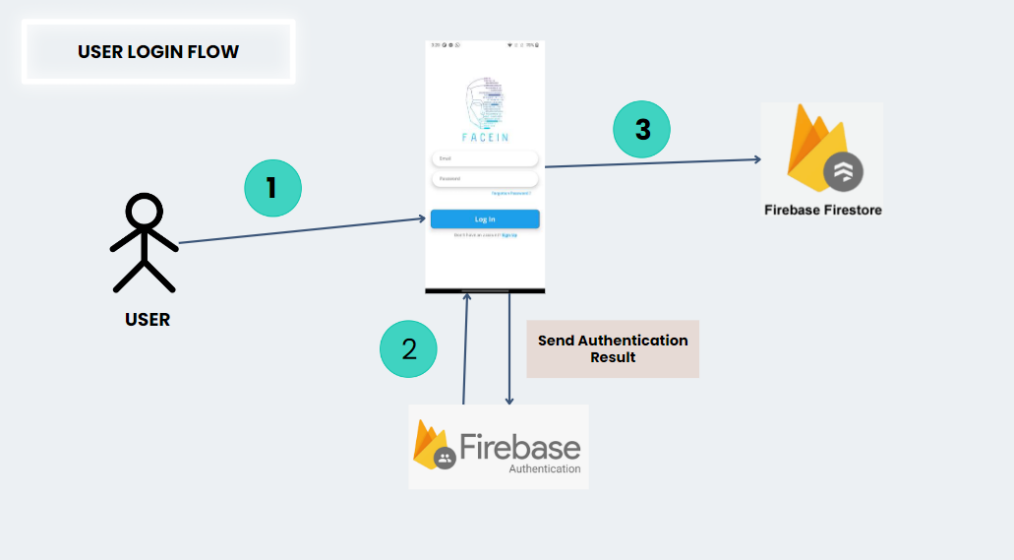


Figure 7.2: User Login Architecture

7.3 User Identity

The following steps and diagram describe and illustrate the process of the user navigating through their Profile Screen.

7.3.1 Profile Screen

a. Users authenticate themselves into the application using Firebase Authentication, which allows them to access the Firebase Firestore to retrieve their respective details using their unique id.

b. Once on the homepage, the user must select the profile icon in the bottom tab to navigate to the Profile Landing Screen.

c. Inside the Profile Landing Screen, the user must click on the profile container consisting of a miniature version of the profile picture and full name.

d. After clicking, the user will be redirected to the Profile Screen, and their information will be retrieved from the Firebase Firestore to display their profile picture, full name, account verification state, email, student ID, and course.

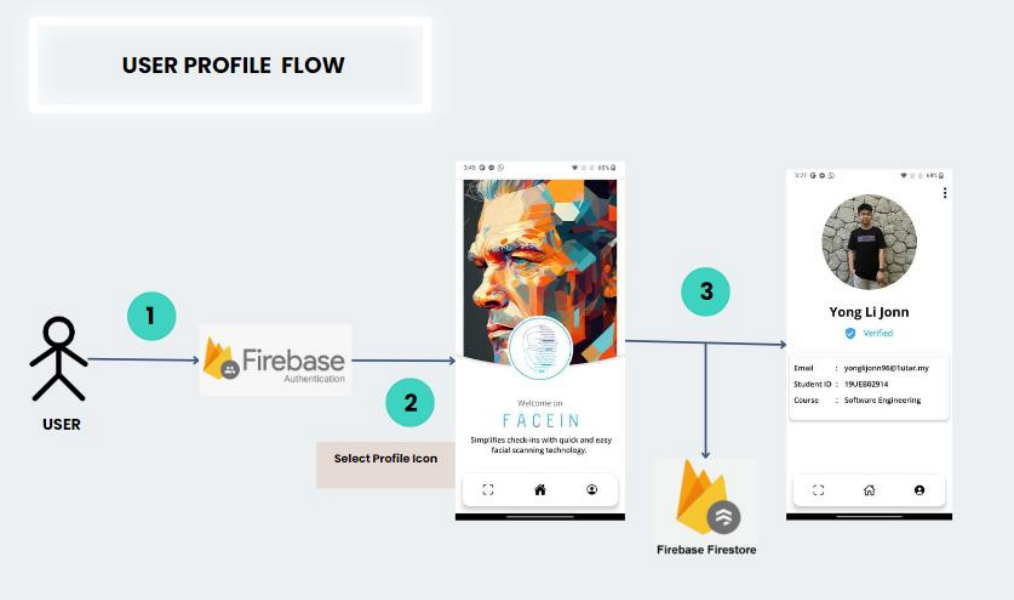


Figure 7.3: User Profile Architecture

7.3.2 Edit Screen

a. Continuing from the Profile Screen, to navigate to the Edit Screen and carry out the edit and update function, the user must select the vertical ellipsis or 3-dotted icon.

b. Once redirected to the Edit Profile Screen, the user must select the type of user information they wish to edit. The type of user information includes the profile picture, surname, last name, student ID, and course.

c. To update the profile picture, the user has the option to either take a picture through the camera or select an image from the album using the react native crop image picker. After making their selection, the user must click on the ‘Upload’ button to successfully upload the image into the Firebase Storage. The profile picture in the Edit Profile Screen will be temporarily changed to reflect the selected choice.

d. If all the user information they wish to update is made, the user must click on the ‘Save’ button to perform the update function.

e. Upon successful editing of the profile, the user will be redirected back to the Profile Screen, where they can view the updated information made from the Edit Screen.

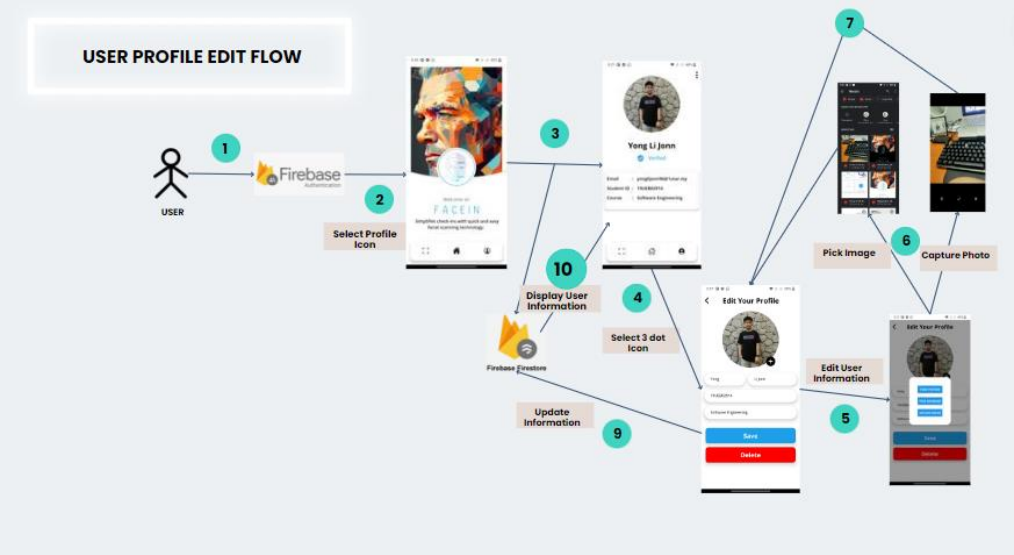


Figure 7.4: User Edit Profile Architecture

7.3.3 Delete Account

a. Continuing from the Edit Screen, there is an alternative function to delete or remove the account from the application.

b. When the user clicks on the 'Delete' button, an alert will pop up to ask for double confirmation before deleting the account.

c. If the ‘Delete’ button is pressed in the alert overlay, the user's information will be removed from the Firestore collection, and their account will be removed from Firebase Authentication.

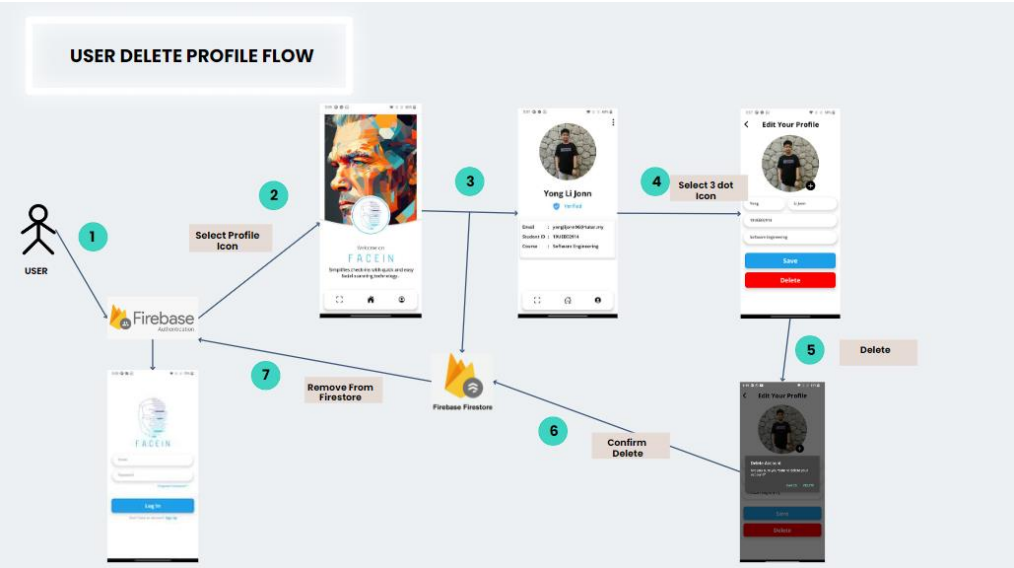


Figure 7.5: User Delete Account Architecture

7.4 Facial Recognition Process

The following steps and diagram describe and illustrate the process of the user going through the facial recognition function and validation processes implemented throughout the process.

7.4.1 Enroll Face via Amazon Rekognition

a. Users authenticate themselves into the application using Firebase Authentication, which allows them to access the Firebase Firestore to retrieve their respective details using their unique id.
b. Once on the homepage, the user must select the check-in in the bottom tab to navigate to the Check-In Menu Screen.
c. Inside the Check-In Menu Screen, the user must enroll their face before performing any other facial recognition function.
d. In order to enroll user's face, user must have all their information filled up in the Profile Screen before performing this function. An alert will prompt upon clicking the 'Enroll Face' button to remind user to fill up user's information before allowed to enroll a face.
e. After all user's information is filled up and fulfil the requirement, upon clicking the 'Enroll Face' button, the react-native camera will be triggered, and user will have to capture their face in order to register or enroll their face.
f. If the Face is not detected upon pressing the capture button, an overlay will popup displaying the message "No Face Detected".
g. Upon Successful Face Enrollment, an overlay will popup displaying the message "Successfully Registered". The captured photo is uploaded to the Amazon S3 Bucket and the users' facial features is extracted from the captured photo.
h. The extracted facial features is then used to create a unique face index that represent the users face and the index is stored into the AWS collection and a Face ID is generated.
i. The generated Face ID will then be stored into the Firestore collection under users unique ID along with other user information.

j. At the same time, after the face enrollment process, the ‘Enroll Face’ button will be disabled, and the ‘Check-In’ button and ‘Delete Face’ button is enabled.

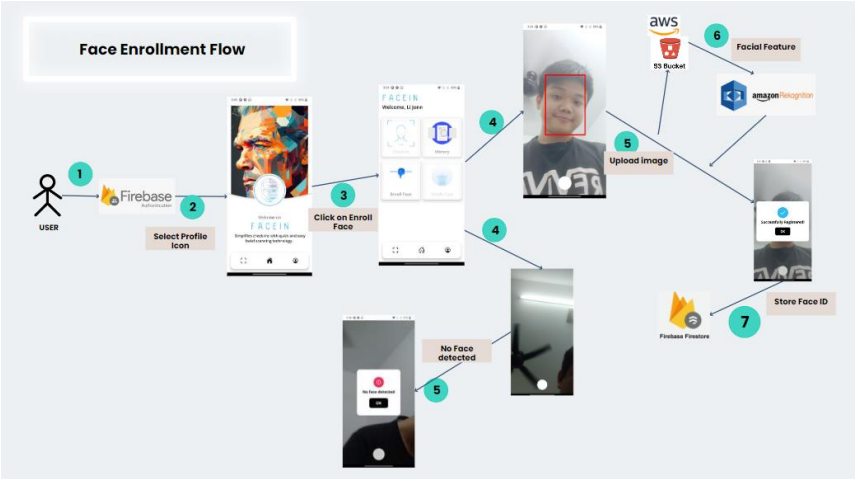


Figure 7.6: Enroll Face Architecture

7.4.2 Remove Face via Amazon Rekognition

a. To remove a registered face, users must first have their face enrolled. Users click the enabled ‘Delete Face’ button, and an alert prompts the user to confirm or cancel the function.

b. When the confirm button, ‘OK’ in the alert prompt is pressed, the ‘deleteFaceFromCollection’ operation is performed and the user’s registered face is removed from the AWS collection, and the saved Face ID in the Firestore Collection is updated to 'null'.

c. Upon successful face deletion, the ‘Enroll Face’ button is enabled again, and the ‘Check-In’ and ‘Delete Face’ buttons are disabled.

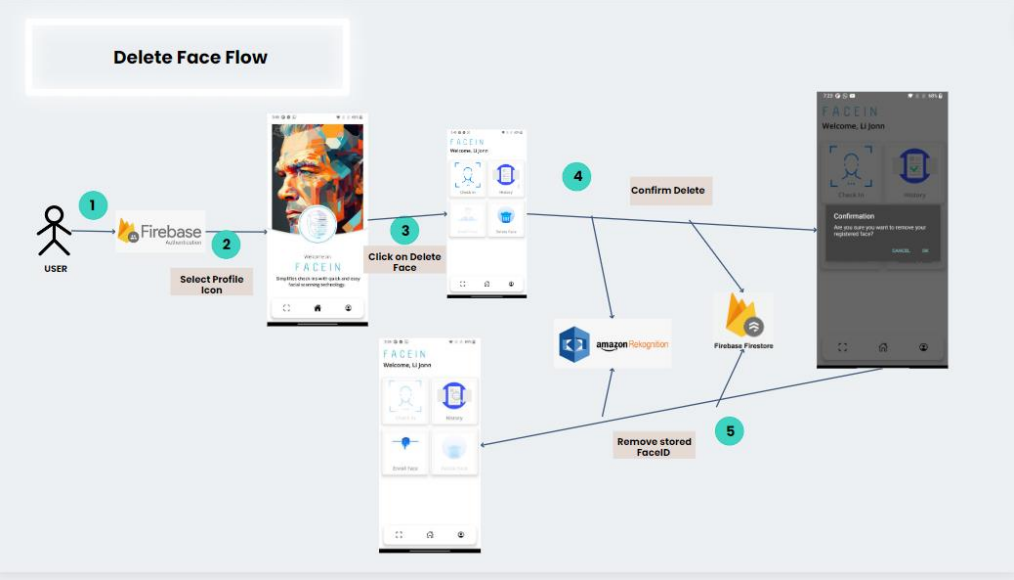


Figure 7.7: Remove Face Architecture

7.4.3 Check-In Process via Amazon Rekognition

a. To perform the check-in or identity verification function, users must first have their face enrolled. Users click the enabled ‘Check In’ button, and an overlay pops up for users to select the location of either the ‘KA’ or ‘KB’ building to check-in.

b. After selecting the building location, the react-native camera is triggered, and users must capture their face to verify their identity. In this process, the ‘searchFaceFromCollection’ operation is performed by searching for a face in the collection using the face in the captured photo.

c. In this process, a validation is also performed to verify the identity of the person scanning the face by comparing the Face ID stored in the Firebase Firestore with the Face ID found in the AWS collection. If the Face ID stored in the Firebase Firestore equals to the Face ID found, then the results indicates as recognized. If it is not equals, then the result is vice versa.

d. If the face is not detected when users press the capture button, an overlay pops up displaying the message “No Face Detected”.

e. If the face is detected but not recognized when users press the capture button, an overlay pops up displaying the message “Failed To Check In, Please Try Again”.

f. If the face is detected and recognized when users press the capture button, an overlay pops up displaying the message “Successfully Checked In” along with other user and check-in information. This information includes the user’s name, email, student ID, course, and check-in details such as location, date, and time.

g. Upon successful check-in, the check-in details are stored in the Firestore sub-collection called ‘checkins’ under the user's collection.

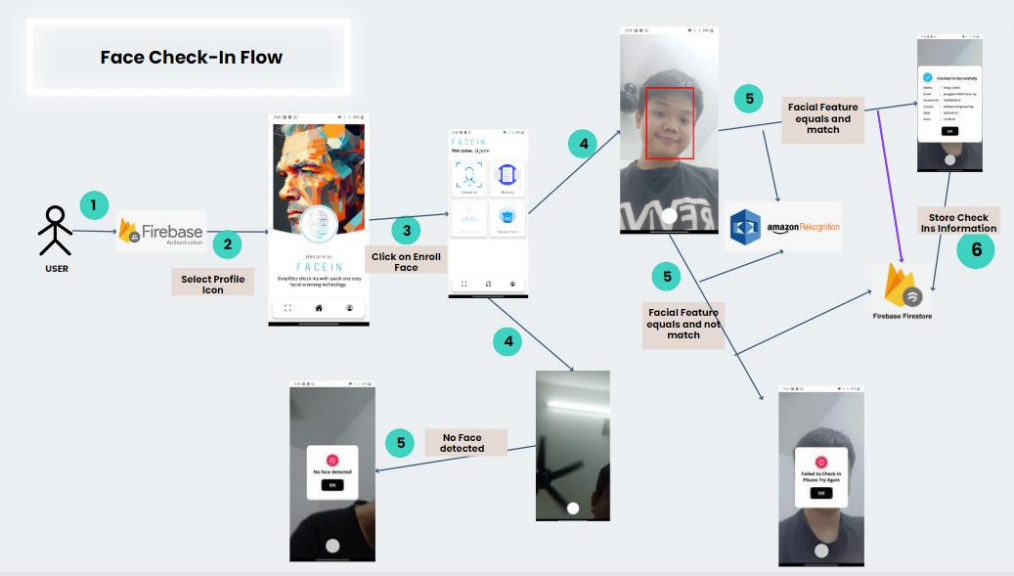


Figure 7.8: Check-In Process Architecture

7.5 Displaying Checked-In History

a. Users can view their history of successful check-ins by clicking the ‘History’ button on the Check-In Menu Screen.

b. If there is a check-in record, the information from the 'checkins' sub-collection is retrieved from Firebase Firestore under the user's collection.

c. The retrieved information is displayed as a list in the History Screen, including location, date, and time.

d. If there are no check-in records, the screen displays ‘no record found’.

e. Any records older than a month are automatically deleted from Firebase Firestore and reflected.

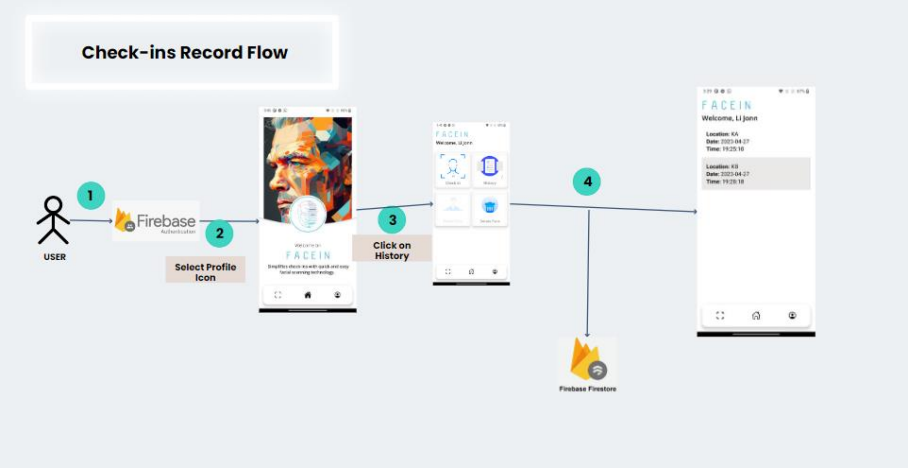


Figure 7.9: Display Checked-In History Architecture

7.6 Summary

This chapter summarized the project details on the implementation of the user-facing elements of the system, including user authentication, user identity, facial recognition, and displaying checked-in history. The authentication design allows users to register and log in using their UTAR email and password. The user identity module includes the Profile Screen, Edit Screen, and Delete Account function. The facial recognition process involves enrolling and removing faces using Amazon Rekognition and performing check-ins by searching for a face in the collection and comparing the Face ID with the one stored in Firebase Firestore. Lastly, the Display Checked-In History function allows users to view their successful check-ins. The chapter provides a detailed explanation of the workflow and architecture for each module.

CHAPTER 8

SYSTEM TESTING

8.1 Introduction

Testing is a crucial part of software development that aims to verify whether the application functions as intended based on its requirements. This chapter will provide a detailed explanation of the testing objectives, scope, and various types of testing involved in the process.

8.2 Testing Objectives

The objectives aimed to accomplish through testing is explained as below:

a. Ensure that the user is able to register into the application.
b. Ensure that the user is able to login into the application.
c. Ensure that the user is able to logout of the application.
d. Ensure that the user is able to edit the user information which includes profile picture, surname, last name, student ID, and course.
e. Ensure that the user is able to upload the profile picture through taking a new picture or selecting existing image from the album.
f. Ensure that the user is able to update the edited user information successfully and reflect the updated information.
g. Ensure that the user is able to delete the account totally from the application.
h. Ensure that the user is able to enroll their face for facial recognition check-in purpose into the AWS collection and Firebase Firestore.
i. Ensure that the user is able to delete the registered face from the AWS collection and Firebase Firestore.
j. Ensure that the user is able to check-in into the building using the facial recognition feature, by matching the facial biometric data captured.
k. Ensure that the user is able view the list of check-ins record in the history.

8.3 Testing Scope

For this application, the components that are tested are listed as follow:

a. Requirements listed on user registration and login.
b. Requirements listed on the functionality of the application.

Components that are not tested are listed as follows:

a. Database testing
b. Performance Testing on the Facial Recognition Process.

8.4 Testing Type

Testing that has been conducted on the application are as follows:

a. Unit Testing on the major functionalities of the application.
b. Integration Testing between the application and integrated pre-trained Model.
c. User Acceptance Testing (UAT) which is conducted together with UTAR Community (students, and lecturer).

8.5 Testing results

This section presents the outcomes of the various tests performed on the application. The tests were conducted to achieve specific objectives and the results are described in detail, including the summary, test steps, test data, expected and actual results, as well as the pass/fail status. Unit testing and User Acceptance Testing (UAT) were carried out on all the objectives, while Integration Testing was conducted on objectives (g), (h), and (i) because they involved interactions with pre-trained model.

8.5.1 Unit Testing

Test Case No	001	**Test Case Name**	Create Account
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		
Executed By	Yong Li Jonn	**Executed Date**	15/4/2023
Pre-Conditions	User has no registered account prior to creating account.		

Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Enter surname, last name, UTAR email, password, and confirm password	1)Enter surname	Yong	A new user is created in the database. User is navigated to the Home Screen.	A new user is created in the database. User is navigated to the Home Screen.	Pass
	2)Enter last name	Li Jonn			
	3)Enter UTAR email	yonglijonn96 @1utar.my			
	4)Enter password	Yong123zr@			
	5)Enter confirm password	Yong123zr@			
Enter surname, last name, email, password, and confirm password	1)Enter null surname		Error messages will prompt in an overlay requesting user to fill all fields.	Error messages will prompt in an overlay requesting user to fill in all fields.	Pass
	2)Enter null last name				
	3)Enter null UTAR email				
	4)Enter null password				
	5)Enter null confirm password				
Enter email	1)Enter email	yonglijonn96 @1utar.my	The entered email is in correct format, proceed to input other field.	The entered email is in correct format, proceed to input other field.	Pass

	2)Enter wrong email format	yonglijonn96 @hotmail.com	Error message “Invalid Email” will prompt in an overlay.	Error message “Invalid Email” will prompt in an overlay.	Pass
Enter password	1) Enter password	Yong123zr@	The entered password is in correct format, proceed to input other field.	The entered password is in correct format, proceed to input other field.	Pass
	2)Enter wrong password format in correct number of characters	123123123	Error message “Password should at least have 1 uppercase, 1 lowercase, and 1 special case character” will prompt in an overlay.	Error message “Password should at least have 1 uppercase, 1 lowercase, and 1 special case character” will prompt in an overlay.	Pass
	3) Enter correct password format with incorrect number of characters	123Jo@	Error message “Password should be at least 6 characters” will prompt in an overlay.	Error message “Password should be at least 8 characters” will prompt in an overlay.	Pass
Enter confirm password	1) Enter confirm password	Yong123zr@	The entered password is in match, proceed to	The entered password is in match, proceed to	Pass

			input other field.	input other field.	
	2)Enter wrong password confirmation	Yongggg123zr@	Error message “Password do not match” will prompt in an overlay.	Error message “Password do not match” will prompt in an overlay.	Pass

Table 8.1: Unit Test Case for Create Account

Test Case No	002		**Test Case Name**	Login Account	
Designed by	Yong Li Jonn		**Design Date**	15/4/2023	
Role	User				
Executed By	Yong Li Jonn		**Executed Date**	15/4/2023	
Pre-Conditions	User has registered an account.				
Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Enter UTAR email, and password,	1)Enter UTAR email	yonglijonn96 @1utar.my	Authentication success. User is navigated to the Home Screen.	Authentication success. User is navigated to the Home Screen.	Pass
	2)Enter password	Yong123zr@			
Enter email, and password	1)Enter null email		Error messages will prompt in an overlay requesting user to enter email and password.	Error messages will prompt in an overlay requesting user to enter email and password.	Pass
	2)Enter null password				
Enter email	1)Enter email that doesn’t exist in the database	yong01 @1utar.my	Error messages displaying “auth/user not found” will	Error messages displaying “auth/user not found” will	Pass

			prompt in an overlay	prompt in an overlay	
	2)Enter wrong email format	yong01 @hotmail.com	Error messages displaying “Invalid Email” will prompt in an overlay	Error messages displaying “Invalid Email” will prompt in an overlay	Pass
Enter password	1)Enter wrong password	11111111	Error messages displaying “Wrong password” will prompt in an overlay	Error messages displaying “Wrong password” will prompt in an overlay	Pass

Table 8.2: Unit Test Case for Login Account

Test Case No	003		**Test Case Name**	Logout Account	
Designed by	Yong Li Jonn		**Design Date**	15/4/2023	
Role	User				
Executed By	Yong Li Jonn		**Executed Date**	15/4/2023	
Pre-Conditions	User has registered an account.				
Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully Logged out of the account by pressing on the ‘Log Out’ button in the Profile Landing Screen	Click on the ‘Log Out’ button	-	The log out function perform successfully. User is navigated to the Login Screen.	The log out function perform successfully. User is navigated to the Login Screen.	Pass

Table 8.3: Unit Test Case for Logout Account

Test Case No	004	**Test Case Name**	Edit User Information
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		
Executed By	Yong Li Jonn	**Executed Date**	15/4/2023
Pre-Conditions	User has successfully logged into the account.		

Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully edit user information which includes profile picture, surname, last name, student ID, and course	1)Test Case No 005	1)Test Case No 005	Edit user information successfully by holding the data value temporary	Edit user information successfully by holding the data value temporary	Pass
	2)Enter surname	Yong			
	3)Enter last name	Li Jonn			
	4)Enter student ID	19UEB02914			
	5)Enter Course	Software Engineering			

Table 8.4: Unit Test Case for Edit User Information

Test Case No	005	**Test Case Name**	Update Profile Picture
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		
Executed By	Yong Li Jonn	**Executed Date**	15/4/2023
Pre-Conditions	User has successfully logged into the account and performing Edit User Information function.		

Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully take a picture and upload as a profile picture	1)Tap on the '+' Icon at the bottom right of the profile picture 2)Click on 'Take	Captured profile picture using react-native-crop-picker	The captured photo will be uploaded into the Firebase Storage and will reflect the captured photo as the new profile picture temporary in	The captured photo will be uploaded into the Firebase Storage and will reflect the captured photo as the new profile picture	Pass

	Picture Button' 3)Capture a Photo 4)Click on the 'Upload Image'		the Edit Screen.	temporary in the Edit Screen.	
Successfully pick a picture from the album and upload as a profile picture	1)Tap on the '+' Icon at the bottom right of the profile picture 2)Click on 'Pick an Image' Button 3)Select an image 4)Click on the 'Upload Image'	Picked picture from the album using react-native-crop-picker	The picked photo will be uploaded into the Firebase Storage and will reflect the captured photo as the new profile picture temporary in the Edit Screen.	The picked photo will be uploaded into the Firebase Storage and will reflect the captured photo as the new profile picture temporary in the Edit Screen.	Pass

Table 8.5: Unit Test Case for Upload Profile Picture

Test Case No	006	**Test Case Name**	Update User Information
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		
Executed By	Yong Li Jonn	**Executed Date**	15/4/2023
Pre-Conditions	User has successfully logged into the account and performed Edit User Information function.		

Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully update the edited user information which includes profile picture, surname, last name, student ID, and course	1)Edit the user information that wished to be changed 2) Click on the 'Save' button	The edited user information that consist of profile picture, surname, last name, student ID, and course	The edited user information is updated in the Firebase Firestore and reflected on the Profile Screen.	The edited user information is updated in the Firebase Firestore and reflected on the Profile Screen.	Pass
Successfully update the edited and not edited user information which includes profile picture, surname, last name, student ID, and course	1)Edit the user information that wished to be changed 2) Leave the user information to remain the same by not editing it 3) Click on the 'Save' button	The edited user information and not edited user information that consist of profile picture, surname, last name, student ID, and course	The edited user information and not edited user information is updated in the Firebase Firestore and reflected on the Profile Screen.	The edited user information or not edited user information is updated in the Firebase Firestore and reflected on the Profile Screen.	Pass

Table 8.6: Unit Test Case for Update User Information

Test Case No	007	**Test Case Name**	Delete Account
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		

Executed By	Yong Li Jonn	**Executed Date**	15/4/2023
Pre-Conditions	User has successfully logged into the account and navigating on the Edit Profile Screen.		

Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully delete the account from the application by removing all user information and user credential confirm delete the account.	1)Click on the 'Delete' button 2) Click on the 'DELETE' button in the alert popup	The user information and user credential.	The user information is deleted from the Firebase Firestore, and user credential is deleted from the Firebase Authentication. The user is navigated to the Login Screen.	The user information is deleted from the Firebase Firestore, and user credential is deleted from the Firebase Authentication. The user is navigated to the Login Screen.	Pass
Successfully cancel the delete account function by cancel delete the account.	1)Click on the 'Delete' button 2) Click on the 'Cancel' button in the alert popup	The user information and user credential.	The Delete Account function is cancelled, and all the user information and credential remains.	The Delete Account function is cancelled, and all the user information and credential remains.	Pass

Table 8.7: Unit Test Case for Delete Account

Test Case No	008	**Test Case Name**	View Check-Ins Record
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		
Executed By	Yong Li Jonn	**Executed Date**	15/4/2023

Pre-Conditions	User has successfully logged into the account and performed check-in function.				
Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully display the list of check-ins record in History Screen	1)Successfully perform facial recognition check-in in the Check-In Screen 2)Click on the 'History' button in the Check-In Menu Screen 3) View on the list of check-ins	Check-In location, date, and time	The check-ins record are retrieved from the Firestore subcollection 'checkins' and displayed in a list in the History Screen.	The check-ins record are retrieved from the Firestore subcollection 'checkins' and displayed in a list in the History Screen.	Pass
Successfully display no record in History Screen	1)Perform nothing on the check in function or wait for a month for all the checkins record to be deleted automatically 2)Click on the 'History' button in the Check-In Menu Screen	Check-In location, date, and time	No check-in record is found from the Firestore subcollection 'checkins' and displayed as no record in History Screen.	No check-in record is found from the Firestore subcollection 'checkins' and displayed as no record in History Screen.	Pass

	2) View on the list of check-ins				

Table 8.8: Unit Test Case for View Check-Ins Record

8.5.2 Integration Testing

Test Case No	009	**Test Case Name**	Face Enrollment
Designed by	Yong Li Jonn	**Design Date**	15/4/2023
Role	User		
Executed By	Yong Li Jonn	**Executed Date**	15/4/2023
Pre-Conditions	User has successfully logged into the account and filled up all the user information.		

Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
User register their facial data into the AWS collection and store the generated Face ID into the Firebase Firestore.	1)User filled in all the user information 2)Click on the 'Enroll Face' button in the Check-In Menu Screen 3) Capture a photo with face detected using react-native-camera	The captured photo, the facial feature, and the Face ID	An overlay with the message "Successfully Registered ! " is displayed. The captured photo is uploaded to the AWS S3 Bucket. The facial metadata extracted is indexed into the AWS collection. The generated Face ID is stored into the	An overlay with the message "Successfully Registered ! " is displayed. The captured photo is uploaded to the AWS S3 Bucket. The facial metadata extracted is indexed into the AWS collection. The generated Face ID is stored into the	Pass

			Firebase Firestore User redirect to the ‘Check In Menu’ Screen The ’Check In’ button and ‘Delete Face’ button is enabled.	Firebase Firestore User redirect to the ‘Check In Menu’ Screen The ’Check In’ button and ‘Delete Face’ button is enabled.	
User perform Face Enrollment function without showing face	1)User filled in all the user information 2)Click on the ‘Enroll Face’ button in the Check-In Menu Screen 3) Capture a photo without face detected using react-native-camera	The captured photo, the facial feature, and the Face ID	An overlay with the message “No Face Detected ! “ is displayed.	An overlay with the message “No Face Detected ! “ is displayed.	Pass
User register their facial data into the AWS collection and store the generated Face ID into the Firebase	1)User filled in all the user information 2)Click on the ‘Enroll Face’ button in the Check-In Menu Screen	The captured photo, the facial feature, and the Face ID	An overlay with the message “No Face Detected ! “ is displayed.	An overlay with the message “Successfully Registered ! “ is displayed. The captured photo is uploaded to	Fail

Firestore by capturing image from another device.	3) Capture a photo with face detected in a photo from another device using react-native-camera			the AWS S3 Bucket. The facial metadata extracted is indexed into the AWS collection. The generated Face ID is stored into the Firebase Firestore User redirect to the 'Check In Menu' Screen The 'Check In' button and 'Delete Face' button is enabled.	

Table 8.9: Integration Test Case for Face Enrollment

Test Case No	010		**Test Case Name**	Delete Registered Face	
Designed by	Yong Li Jonn		**Design Date**	15/4/2023	
Role	User				
Executed By	Yong Li Jonn		**Executed Date**	15/4/2023	
Pre-Conditions	User has successfully logged into the account and enrolled a face				
Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
Successfully delete the	1) User enrolled a face	The user registered	The registered face is deleted	The registered face is deleted	Pass

registered face from the AWS collection and remove the stored Face ID in the Firestore Firebase	2)Click on the 'Delete Face' button 2) Click on the 'DELETE' button in the alert popup	facial data in AWS collection and stored Face ID in the Firebase Firestore.	from the AWS collection, and the Face ID is set to null in the Firebase Firestore.	from the AWS collection, and the Face ID is set to null in the Firebase Firestore.	
Successfully cancel Delete Face function by cancel removing the registered face.	1)User enrolled a face 2)Click on the 'Delete Face' button 2) Click on the 'Cancel' button in the alert popup	The user registered facial data in AWS collection and stored Face ID in the Firebase Firestore.	The Delete Face function is cancelled, the registered face remain in the AWS collection, and the Face ID remain stored in the Firebase Firestore.	The Delete Face function is cancelled, the registered face remain in the AWS collection, and the Face ID remain stored in the Firebase Firestore.	Pass

Table 8.10: Integration Test Case for Delete Registered Face

Test Case No	011		**Test Case Name**	Facial Recognition Check In	
Designed by	Yong Li Jonn		**Design Date**	15/4/2023	
Role	User				
Executed By	Yong Li Jonn		**Executed Date**	15/4/2023	
Pre-Conditions	User has successfully logged into the account and enrolled a face				
Summary	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**	**Status**
User check-in into the selected building by performing	1)User Enrolled a Face	The captured photo, the facial feature, the	The facial feature from the captured photo is extracted.	The facial feature from the captured photo is extracted.	Pass

facial recognition check-in	2)Click on the 'Check In' button in the Check-In Menu Screen 3)Select the location 'KA' or 'KB'. 4) Capture a photo with face detected using react-native-camera	Face ID, the matched Face ID in the Firestore with the Face ID in AWS collection, user information, and check-in information.	The facial metadata extracted is searched through the collection. The Face ID of the founded face from the collection is matched with the Face ID stored in the Firestore An overlay with the user information and check in information is displayed. The check in information is stored as a subcollection in the Firestore. User redirect to the 'Check in Menu' Screen	The facial metadata extracted is searched through the collection. The Face ID of the founded face from the collection is matched with the Face ID stored in the Firestore An overlay with the user information and check in information is displayed. The check in information is stored as a subcollection in the Firestore. User redirect to the 'Check in Menu' Screen	
User perform Check In function without	1)User Enrolled a Face	The captured photo, the facial feature, the	An overlay with the message "No Face	An overlay with the message "No Face	Pass

showing face	2)Click on the 'Check In' button in the Check-In Menu Screen 3)Select the location 'KA' or 'KB'. 4) Capture a photo without face detected using react-native-camera	Face ID, the matched Face ID in the Firestore with the Face ID in AWS collection, user information, and check-in information.	Detected ! " is displayed.	Detected ! " is displayed.	
User check-in into the selected building by performing facial recognition check-in as in capturing image from another device.	1)User Enrolled a Face 2)Click on the 'Check In' button in the Check-In Menu Screen 3)Select the location 'KA' or 'KB'. 4) Capture a photo with face detected in a photo from another device using react-native-camera	The captured photo, the facial feature, the Face ID, the matched Face ID in the Firestore with the Face ID in AWS collection, user information, and check-in information.	An overlay with the message "No Face Detected ! " is displayed.	The facial feature from the captured photo is extracted. The facial metadata extracted is searched through the collection. The Face ID of the founded face from the collection is matched with the Face ID stored in the Firestore An overlay with the user	Fail

				information and check in information is displayed. The check in information is stored as a subcollection in the Firestore. User redirect to the ‘Check in Menu’ Screen	

Table 8.11: Integration Test Case for Facial Recognition Check In

8.5.3 User Acceptance Testing (UAT)

The testing objectives outlined in section 8.2 were used by Yong Li Jonn to design the tests listed in the table below on 15th April 2023. In order to ensure accuracy and credibility of the test results, the tests were conducted physically in the 'KA' and 'KB' block of UTAR buildings. These tests were conducted by the UTAR community that include 14 students and 1 lecturer on 18th April to 20th April 2023. To verify the results, the tester's name, student ID, course, and UTAR email were taken as proof and documented in Appendix B. The results of each test, indicating success or failure, are represented by an ‘nS’ as the number of students, ‘nL’ as number of lecturers, followed by an 'X' symbol. For failed tests, remarks explaining the cause of failure were provided.

Test Subject	**(n) Pass**	**Fail**	**Remarks**
Create Account	(14S) (1L) X		
Login Account	(14S) (1L) X		
Logout Account	(14S) (1L) X		

Update Profile Picture	(14S) (1L) X		
Update User Information	(14S) (1L) X		
Delete Account	(14S) (1L) X		
View Check-Ins Record	(14S) (1L) X		
Face Enrollment by Capturing Image with Own Device	(14S) (1L) X		
Face Enrollment by Capturing Image with a Face from another Device		**(14S) (1L) X**	The function for differentiating a real face or a face from an image is not currently implemented and will be planned as a future work
Delete Registered Face	(14S) (1L) X		
Facial Recognition Check In by Capturing Image with own Device	(14S) (1L) X		
Facial Recognition Check In by Capturing Image with a face from another device		**(14S) (1L) X**	The function for differentiating a real face or a face from an image is not currently implemented and will be planned as a future work

Table 8.12: User Acceptance Testing (UAT) Results

8.6 Summary

In summary, this chapter provides an overview of the testing phase for the In Building Facial Recognition Check-In application. It covers the testing objectives, scope, and types of testing conducted, as well as the results obtained from each test. The chapter also includes the test cases for both unit and integration testing and the results of the user acceptance testing conducted by the UTAR Community.

CHAPTER 9

CONCLUSION & FUTURE DEVELOPMENTS

9.1 Introduction

This section details the achievement of the project objectives outlined in section 1.5.2, including the implementation of solutions, development of the application, and completion of testing. Additionally, it explores potential avenues for future improvement and integration with advanced technologies and features.

9.2 Achieving Objectives

The objectives outlined in section 1.5.2 have been successfully accomplished through the development and testing of the In-Building Facial Recognition Check-in System. The following details how each objective was met:

1. The facial recognition system was developed using a pre-trained machine learning model (Amazon Rekognition), resulting in a high level of accuracy in identifying and verifying individuals based on their facial features. Extensive testing and optimization were conducted to achieve a recognition accuracy of over 95%, meeting the specified target.
2. An Android mobile application was created, providing users with a seamless and intuitive interface. The application allows users to create accounts, enroll their faces, and perform check-ins to designated buildings. Additionally, users can view their check-in records and manage their personal information, ensuring a user-friendly experience that meets their needs.
3. A Firebase back-end database was successfully integrated into the system, ensuring secure storage of user information, enrollment data, and check-in records. The database provides efficient data retrieval and management capabilities, enabling seamless access to user information and check-in records in real-time.
4. User acceptance testing was conducted with a sample group of at least 15 individuals from the UTAR community, including students, lecturers,

and staff. Feedback was gathered to assess user satisfaction, identify any issues or areas for improvement, and make necessary adjustments to enhance the overall user experience.

9.3 Limitations

Despite the successful achievement of the project objectives, the In-Building Facial Recognition Check-in System has some limitations that should be acknowledged.

Firstly, the system is currently limited to real-time facial recognition using the device's camera and does not have the capability to differentiate between a live face and a face in a still image. This limitation may result in potential security risks, as students could potentially check-in successfully by displaying their face from a photo on another device during the scanning process. To address this concern, future developments could implement liveness detection functionality, ensuring that the system accurately identifies live faces and prevents unauthorized check-ins.

Secondly, the application does not include location detection, meaning it cannot determine if the user's device is located in the selected building or area during the check-in process. Consequently, the check-in will not fail based on the device's location, potentially allowing users to check-in from unauthorized locations. To mitigate this limitation, integrating location detection capabilities in future iterations would enhance the accuracy and effectiveness of the check-in system.

Lastly, the current application is only available on the Android platform, which may exclude users with devices operating on other platforms, such as iOS. Developing cross-platform compatibility in future updates would increase the application's accessibility and cater to a broader range of students.

9.4 Future Developments

To enhance the In-Building Facial Recognition Check-in System and further optimize its functionality, several improvements and additional features are proposed for future development:

a. Liveness Detection: Implementing liveness detection functionality would allow the system to differentiate between a live face and a face in a still image, increasing the security and reliability of the check-in process.
b. Location Detection: Implementing location detection capabilities would enable the application to determine if the user's device is within the selected building or area during the check-in process. This would ensure that users are physically present when checking in, further enhancing the system's accuracy and usefulness.
c. Integration with other UTAR systems: Expanding the scope of the application to include integration with other university systems, such as attendance tracking during lectures, tutorials and practical, library access, or event registration. Thus, this would create a more comprehensive and unified experience for UTAR students.
d. Cross-platform compatibility: Developing the application for additional platforms, such as iOS, would increase its accessibility, portability and make it available to a wider range of students with different devices.
e. Continuous Improvement: Regularly updating the pre-trained machine learning model with new data will enhance its performance over time and ensure that the facial recognition system remains accurate and effective as technology and user requirements evolve.

9.5 Summary

In conclusion, the In-Building Facial Recognition Check-in System has successfully achieved its project objectives, providing UTAR students with a seamless check-in process. Despite some limitations, such as the lack of liveness detection, location detection, and cross-platform compatibility, the system demonstrates promising performance and usability. Future developments, including implementing liveness detection, location detection, integration with other UTAR systems, and cross-platform compatibility, will address these limitations and further optimize the system. With continued development and improvement, the In-Building Facial Recognition Check-in System has the potential to significantly enhance the check-in process at UTAR, creating a more comprehensive and unified experience for students and serving as a valuable asset to the university and its community.

REFERENCES

Alita, S., 2019. Malaysia Set to Implement Facial Recognition System to Combat Crime. [online] Available at: <https://opengovasia.com/malaysia-set-to-implement-facial-recognition-system-to-combat-crime/> [Accessed 20 August 2022].

Amazon Web Services., n. d. *Amazon Rekognition Documentation.* [online] Available at: <https://docs.aws.amazon.com/rekognition/index.html> [Accessed 24 April 2023].

Arc, 2018. *Convolutional Neural Network*. [online] Available at: <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05> [Accessed 30 August 2022].

Bischoff, P., 2021. *Facial recognition technology (FRT): 100 countries analyzed*. [online] Available at: <https://www.comparitech.com/blog/vpn-privacy/facial-recognition-statistics/> [Accessed 28 July 2022].

Bischoff, P., 2021a. *The Global Map of Facial Recognition Technologies*, digital image, viewed 28 July 2022, <https://www.comparitech.com/blog/vpn-privacy/facial-recognition-statistics/>

Bischoff, P., 2021b. *Facial Recognition Use by Countries: Full Score [40]*, digital image, viewed 28 July 2022, <https://www.comparitech.com/blog/vpn-privacy/facial-recognition-statistics/>

C, A., 2020. An Overview on Convolutional Neural Networks. [online] Available at: <https://medium.com/swlh/an-overview-on-convolutional-neural-networks-ea48e76fb186> [Accessed 31 August 2022].

Canterbury AI, 2021. *Face Recognition prevails*. [online] Available at: <https://canterbury.ai/facial-recognition-how-wearing-masks-can-affect-facial-recognition-application/> [Accessed 28 July 2022].

Cambridge Dictionary, 'Facial Recognition', *A Cambridge Advanced Learner's Dictionary*, 4th edn, Cambridge University Press, Cambridge.

Chan, M., 2020. *Not all tech will play out says Sunway Education Group*. [online] Available at: <https://techwireasia.com/2019/03/not-all-tech-will-play-out-says-sunway-education-group/> [Accessed 28 July 2022].

Chen, P., Geng, X., Zou, M., Xu, Q. and Tan, D., 2020, March. Development and Optimization of Check-in System Based on Face Recognition Technology. In *IOP Conference Series: Materials Science and Engineering* (Vol. 782, No. 5, p. 052022). IOP Publishing.

Demchenko, M., 2020. *Phases in Rapid Application Development*, digital image, viewed 28 July 2022, <https://ncube.com/blog/what-is-rapid-application-development>

eID, 2021. *FACE RECOGNITION SYSTEM*. [online] Available at: <https://www.electronicid.eu/en/blog/post/face-recognition/en> [Accessed 28 July 2022].

Elisha, O., 2022, *The Top 10 Computer Vision APIs in 2022*. [online] Available at: < https://viso.ai/computer-vision/computer-vision-apis/> [Accessed 23 April 2023].

Goodfellow, I., Bengio, Y. and Courville, A., 2018. *Deep learning. MIT press*.

Google Cloud., n. d. *Cloud Vision API Documentation.* [online] Available at: <https://cloud.google.com/vision/docs> [Accessed 24 April 2023].

Hammim, R., 2020. *SK Taman Perling 1 uses facial recognition scanners to mark pupils' attendance.* [online] Available at: <https://www.nst.com.my/news/nation/2020/01/552737/sk-taman-perling-1-uses-facial-recognition-scanners-mark-pupils> [Accessed 28 July 2022].

i-Scoop, n.d. *Building management in the age of IP and IoT (interview)*. [online] Available at: <https://www.i-scoop.eu/internet-of-things-iot/building-management-systems-iot/> [Accessed 20 August 2022].

Kaspersky, n.d. *What is facial recognition?* [online] Available at: <https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition> [Accessed 28 July 2022].

Khan, M., Chakraborty, S., Astya, R. and Khepra, S., 2019, October. Face detection and recognition using OpenCV. In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 116-119). IEEE.

Klosowski, T., 2020. *Facial Recognition Is Everywhere. Here's What We Can Do About It.* [online] Available at: <https://www.malaysiaairports.com.my/media-centre/news/facial-recognition-will-replace-boarding-pass-faster-and-safer-passenger> [Accessed 20 August 2022].

Kumar, A., 2022. *Different Types of CNN Architectures.* [online] Available at: <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/> [Accessed 21 August 2022].

Kushal, M., BV, K.K., MJ, C.K. and Pappa, M., 2020, July. ID Card Detection with Facial Recognition using Tensorflow and OpenCV. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 742-746). IEEE.

Lewis, J. A., 2021. *How Does Facial Recognition Work?* [online] Available at: < https://www.csis.org/analysis/how-does-facial-recognition-work> [Accessed 28 July 2022].

Lucid Content Team, n.d. *4 Phases of Rapid Application Development Methodology*. [online] Available at: <https://www.lucidchart.com/blog/rapid-application-development-methodology> [Accessed 29 August 2022].

Maksymenko, S., 2019. *How to build a face detection and recognition system*. [online] Available at: <https://towardsdatascience.com/how-to-build-a-face-detection-and-recognition-system-f5c2cdfbeb8c> [Accessed 20 August 2022].

Malaysia Airport, 2021. *Facial Recognition Will Replace Boarding Pass for Faster and Safer Passenger Authentication at KL International Airport.* [online] Available at: <https://www.malaysiaairports.com.my/media-centre/news/facial-recognition-will-replace-boarding-pass-faster-and-safer-passenger> [Accessed 20 August 2022].

Martin, M., 2022a. *What is Waterfall Model in SDLC? Advantages and Disadvantages*. [online] Available at: <https://www.guru99.com/what-is-sdlc-or-waterfall-model.html> [Accessed 21 August 2022].

Martin, M., 2022b. *What is RAD Model? Phases, Advantages and Disadvantages*. [online] Available at: <https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html> [Accessed 21 August 2022].

Microsoft., n. d. a. *Computer Vision – Azure Cognitive Services.* [online] Available at: <https://learn.microsoft.com/en-us/azure/cognitive-services/computer-vision/> [Accessed 24 April 2023].

Microsoft., n. d. b. *Limited Access to Face API.* [online] Available at: <https://learn.microsoft.com/en-us/legal/cognitive-services/computer-vision/limited-access-identity> [Accessed 24 April 2023].

Mishra, M., 2020. *Convolutional Neural Network, Explained*. [online] Available at: <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939> [Accessed 30 August 2022].

NEC, 2021. *What is the future of facial recognition technology in 2022 and beyond?* [online] Available at: <https://www.nec.co.nz/market-leadership/publications-media/what-is-the-future-of-facial-recognition-technology-in-2022-and-beyond/> [Accessed 20 August 2022].

NEC, 2022. *A brief history of Facial Recognition*. [online] Available at: <https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial-recognition/> [Accessed 28 July 2022].

Norstrom, P. and Consulting, A., 2021. Has Covid increased public faith in facial recognition? *Biometric Technology Today*, *2021*(11-12), pp.5-8.

Petrescu, R.V., 2019. Face recognition as a biometric application. *Journal of Mechatronics and Robotics*, *3*, pp.237-257.

Pingel, J., 2017. *Math Works – Convolutional Neural Network*. Available at: <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html> [Accessed 29 August 2022].

Ríos-Sánchez, B., Costa-da-Silva, D., Martín-Yuste, N. and Sánchez-Ávila, C., 2019. Deep Learning for Facial Recognition on Single Sample per Person Scenarios with Varied Capturing Conditions. *Applied Sciences*, *9*(24), p.5474.

Salama Abdelminaam, D., Almansori, A.M., Taha, M. and Badr, E., 2020. A deep facial recognition system using computational intelligent algorithms. *Plos one*, *15*(12), p.e0242269.

Shah, A., 2022. *TOP 10 FACIAL RECOGNITION APPS IN 2022*. [online] Available at: <https://www.nec.co.nz/market-leadership/publications-media/what-is-the-future-of-facial-recognition-technology-in-2022-and-beyond/> [Accessed 20 August 2022].

Sharma, S., 2017. *Activation Functions in Neural Networks*. [online] Available at: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> [Accessed 31 August 2022].

Shashkina, V., 2022. *Business guide to facial recognition: benefits, applications, and issues to consider*. [online] Available at: <https://itrexgroup.com/blog/facial-recognition-benefits-applications-challenges/> [Accessed 28 July 2022].

Sightcorp, n.d. *How is Deep Learning used in Facial Recognition?*. [online] Available at: <https://sightcorp.com/knowledge-base/face-recognition-deep-learning/> [Accessed 28 July 2022].

Statista, n.d. *Facial recognition market size worldwide in 2020 and 2025*. [online] Available at: <https://www.statista.com/statistics/1275334/global-facial-recognition-market-size/> [Accessed 20 August 2022].

Thales, 2021. *Top facial recognition technologies.* [online] Available at: <https://www.thalesgroup.com/en/markets/digital-identity-and security/government/biometrics/facial-recognition> [Accessed 20 August 2022].

The Sumsuber, 2021. *SingPass: All You Need to Know in Under 500 Words*. [online] Available at: <https://sumsub.com/blog/singpass/> [Accessed 20 August 2022].

Vadlapati, J., Velan, S.S. and Varghese, E., 2021, July. Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE.

Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, *9*(4), pp.611-629.

Yang, H. and Han, X., 2020. Face recognition attendance system based on real-time video processing. *IEEE Access*, *8*, pp.159143-159150.

Zhang, T.C., 2021. On the Face of It: The Use of Facial Recognition Check-in Technology. *Rosen Research Review*, *2*(3), p.11.

Zulfiqar, M., Syed, F., Khan, M.J. and Khurshid, K., 2019, July. Deep face recognition for biometric authentication. In *2019 international conference on electrical, communication, and computer engineering (ICECCE)* (pp. 1-6). IEEE.

APPENDICES

Appendix A: Gantt Chart

Gantt Chart

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17
1.	Problem Identification and Project Planning	2022-06-29																	
2.	Literature Review: 1) Research relevant details on Machine Learning and Deep Learning 2) Implementation of Facial Recognition in Society	2022-07-24																	
3.	Do sourcing to collect relevant databases for the research	2022-08-07																	
4.	Preliminary Results	2022-08-04																	
5.	Completion of Project I report writing and presentation	2022-09-08																	

Gantt Chart (Project I)

Gantt Chart

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17
1.	Project II Planning	2023-02-12																	
2.	Amendment on the report of Project I	2023-02-12																	
3.	Project Implementation (Starting Phase): - Environment Setup - Building App UI - Database Setup - Connecting Application to Database	2023-03-05																	
4.	Project Implementation (Development Phase): -Setup Real-time Database -Insert Data into Database -Integration of Pre-trained Model into the Application -Proper linkage of Main FR Functions with Interface -Improve User Interface Design	2023-03-26																	
5.	Poster Design	2023-04-12																	
6.	Project Implementation (Final Phase): - Final Optimization on the FR functions - Application Deployment	2023-04-23																	
7.	- Conduct User Acceptance Testing - Code Refinement	2023-04-23																	
8.	Completion of Project II Report Writing	2023-04-28																	
9.	Completion of Project II Presentation	2023-05-05																	

Gantt Chart (Project II)

Appendix B: User Acceptance Tester Information

Tester	UAT001 (Lecturer)	UAT 002	UAT 003
	Name: Dr Fatimah Audah binti Md Zaki Role: Supervisor in FYP	Name: Elson Jap Jia Fong Student ID: 19UEB02251 Email: elsonjap0506@1utar.my Course: Actuarial Science	Name: Wong Yao Jun Student ID: 19UEB03049 Email: wongyaojun@1utar.my Course: Actuarial Science
			
Tester	UAT 04	UAT 005	UAT 006
	Name: Chau Choon Wei Student ID:18UEB02339 Email: choonwei00@1utar.my Course: Civil Engineering	Name: Yit Kok Kean Student ID: 19UEB03585 Email: yitkokkean@1utar.my Course: Materials and Manufacturing	Name: Yee Jia En Student ID: 19UKB06474 Email: xjiaenx2002@1utar.my Course: Accounting
Tester	UAT 007	UAT 008	UAT 009
	Name: Joanne Lim Zhi Yee Student ID: 19UEB04000 Email: jlzy21@1utar.my Course: Software Engineering	Name: Lim Kee Yuan Student ID: 19UEB03537 Email: keeyuannn.92@1utar.my Course: Mechatronic Engineering	Name: Yeo Ya Qi Student ID: 19UEB01604 Email: cookie.yeo@1utar.my Course: Civil Engineering
Tester	UAT 010	UAT 011	UAT 011
	Name: Vignnesh Ravindran Student ID: 1902683 Email: ravin.vignnesh@1utar.my Course: Software Engineering	Name: Bradley Kueh Shi Kwang Student ID: 19UJB05506 Email: bradleykueh58@1utar.my Course: Bachelor of Corporate Communications	Name: Chong Hau Yong Student ID: 19UEB03617 Email: hauyong520@1utar.my Course: Software Engineering
Tester	UAT 013	UAT 014	UAT 015
	Name: Ng Tze Heng Student ID: 19UKB03411 Email: ntzeheng@1utar.my Course: Accounting	Name: Cheong Yu Jian Student ID: 19UEB01152 Email: kencheong09@1utar.my Course: Civil Engineering	Name: Chong Yi Fan Student ID: 19UEB04870 Email: chongyifan001@1utar.my Course: Civil Engineering