FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

BY

NG SUET ENG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology

(Kampar Campus)

JUN 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

Academic Session: JUN 2023

I ___________________________Ng Suet Eng_____________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

SuLee

___________________________ ___________________________

(Author's signature) (Supervisor's signature)

Address:
110-3, Lorong Makmur 11A
Off Persiaran Agacia 1
Bandar Agaria,
31910 Kampar Perak

_______Su Lee Seng________
Supervisor's name

Date: 14th September 2023

Date: 14th September 2023

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 14/09/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ___ *Ng Suet Eng* ___ (ID No: __***20ACB06580***__) has completed this final year project entitled "*FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS*" under the supervision of __Mr. Su Lee Seng__ (Supervisor) from the Department of __________________________, Faculty of Information and Communications Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Ng Suet Eng*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___________________________

Name : Ng Suet Eng

Date : 14th September 2023

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude and appreciation to my supervisors, Mr. Su Lee Seng, for providing me with the wonderful opportunity to work on a face recognition project, which marks my first step towards building a career in the field of AI and ML. I am incredibly thankful to you.

I would also like to express my deep gratitude to a very special person in my life, Ms. Chai Siew Khuan, for her unwavering patience, unconditional support, and love, and for standing by my side during challenging times. Lastly, I am immensely grateful to my parents and family for their constant love, support, and encouragement throughout the course of my journey.

ABSTRACT

This project explores the growing domain of facial recognition technology, known for accurately identifying people based on their unique facial features. This technology has gained popularity across various industries for preventing identity fraud. In the context of exam centres, where confirming candidates' identities is vital, facial recognition can play a crucial role.

The main focus of this project is to prevent fraud in exam centres. The system's key requirement is accurately identifying individuals. It also needs to be fast for quick verification. The project uses a mix of hardware and software tools. Hardware includes cameras and laptops, meanwhile software tools include PyCharm Community, OpenCV and the Haar-Cascade Algorithm are used for development. By implementing this facial recognition system, candidates would need to identify their identity before entering exam centres, as it is to reducing cheating and impersonation. The system aims to verify candidates swiftly and accurately, improving the overall exam process.

In conclusion, this project to develop a facial recognition system for exam centres is a step towards enhancing exam integrity and preventing identity fraud. Through advanced technology, the project aims to accurately identify candidates, reduce fraud, and provide insights into exam performance. It could have a big impact on education and lead to future advancements in exams and identity verification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

%	percentage

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
NCEE	National College Entrance Examination
PCA	Principle Component Analysis
2D	2 Dimension
3D	3 Dimension
R-CNN	Region-based Convolutional Neural Network
SVM	Support Vector Machine
AdaBoost	Adaptive Boosting
POFA	Pictures of Facial Affect
LBP	Local Binary Patterns
NIST	National Institute of Standards and Technology
FRVT	Facial Recognition Vendor Test
ORM	Object-Relational Mapping
MMOD CNN	Maximum-Margin Object Detection Convolutional Neural Network
SDKs	Software Development Kits
GUIs	Graphical User Interfaces
UI	User Interface
PDPA	Personal Data Protection Act 2010

Chapter 1

Introduction

Facial recognition stands as a cutting-edge technology that employs advanced algorithms to identify individuals by comparing facial attributes from photos or videos against a database of faces. This biometric identification method captures unique facial characteristics like gender, age, emotions, facial features, and expressions. Specifically, features such as the nose bridge, eye spacing, and chin contour can rapidly and accurately determine an individual's identity using this technology. An impressive aspect of facial recognition is its capacity to function effectively even in busy and dynamic environments. It excels at identifying individuals in real-time, adapting to varying lighting conditions, and changing facial expressions or appearances. This adaptability renders it highly useful for identity verification across diverse scenarios.

The concept behind facial recognition involves converting facial feature images into mathematical expressions, enabling comparison for similarity determination. In general, these filters are created through the application of deep learning techniques, which involving neural networks within AI for data processing. Furthermore, natural variations such as hairstyle changes, facial expressions, accessories, lighting conditions, and more can pose challenges to recognition. [1] Thus, robust algorithms are necessary, capable of accounting for these variations to ensure reliable results. The learning algorithms are trained on substantial datasets of facial images, allowing them to learn intricate patterns and representations. These algorithms generate numerical facial feature representations to form a unique "fingerprint" or template of an individual's face. During recognition, captured face templates are compared to those in a database to establish similarity and identification.

Woodrow W Bledsoe is acknowledged as one of the initial trailblazers in the field of facial recognition technology. In the period spanning from 1964 to 1966, he spearheaded a team of researchers that conducted experiments to investigate the capability of computers to identify human faces [2]. The team utilized a basic scanner to capture facial attributes like the hairline, eyes, and nose, although the computer did not achieve success in finding matches [2]. Through his research, Bledsoe proposed that the complexity of the face recognition conundrum stems from the inherent variability in factors such as lighting conditions, angles, facial expressions, aging, and more [2]. Bledsoe's contributions laid the cornerstone for subsequent breakthroughs in facial recognition technology, and his insights into the challenges of face recognition continue to hold significance even in the present day.

Today, facial recognition technology is used in many fields like security, immigration, education, and healthcare. But its widespread use has raised concerns about fairness and privacy. People worry about who owns the data, how accurate it is, and how private our information remains. The technology isn't perfect – it can sometimes make mistakes or identify the wrong person. As it enters new markets, rules need to be in place to make sure it's used properly and respects people's privacy. This means having clear guidelines on how data is collected, stored, and used. Balancing the benefits of facial recognition with protecting people's rights is a big challenge that needs careful thought and action from both regulators and tech developers.

Although facial recognition technology has the capability to improve public safety and security by deterring crimes and managing immigration, its implementation must be paired with effective regulations and supervision to prevent its misapplication. Without proper checks in place, there exists a potential for misuse or infringing upon individual privacy. As facial recognition technology evolves and enters new sectors beyond government applications, the essential ethical issues concerning data ownership, accuracy, and privacy persist. Policymakers, institutions, and developers should take these ethical concerns into account and put in place suitable measures to minimize risks and guarantee the conscientious and ethical utilization of facial recognition technology.

1.1 Problem Statement and Motivation

As technology and innovation continue to advance at a rapid pace, the integration of biometric facial recognition is becoming more prevalent across a wide range of industries. Despite the potential advantages it offers, this adoption has also triggered doubts and skepticism. Several challenges are anticipated:

Frauds in the form of impersonation Happening at Exam Location

Impersonation fraud within examination venues pertains to instances where individuals attempt to manipulate or disrupt the examination process by assuming false identities. This fraudulent activity can manifest in a variety of ways and can be executed by candidates or external entities who should not legitimately take the exam. A prevalent approach involves a candidate enlisting an intermediary, often called a proxy candidate, to sit the exam on their behalf. The proxy candidate may bear a resemblance to the real candidate or employ counterfeit identification documents to impersonate their identity. Cheating in the form of impersonation is one of the common problems encounters in exam systems. In instances of impersonation fraud within exam systems, the absence of a dependable identity verification system is frequently recognized as the core issue. [3] Moreover, subsequent to addressing the concern of impersonation is the need to ascertain the authentication status of the candidate. If the candidate's identity is verified, their attendance must be documented [3]. In the absence of adequate authentication protocols, it becomes more feasible for individuals to engage in cheating through impersonation. This can result in inaccurate attendance records, where the presence of a person is recorded even if they did not actually attend the exam. Such erroneous attendance records can yield serious consequences, particularly when verifying an individual's presence at a specific venue holds significance for legal or criminal matters, such as certifications, job qualifications, or legal inquiries. To counteract the challenge of impersonation fraud in exam settings, it is crucial to implement robust identity verification measures. This can encompass techniques like biometric authentication, such as facial recognition, and cross-checking identification documents against a reliable database. Furthermore, the act of documenting and officially notifying the relevant officials can play a role in identifying and stopping cases of impersonation fraud. In general, combating the problem of impersonation fraud during exams necessitates a comprehensive strategy involving dependable methods of confirming identity, utilizing technological solutions, and upholding rigorous examination procedures. This approach is vital for preserving the honesty of the examination process and guaranteeing an equitable assessment of candidates' competencies and understanding.

Very Time Consuming to perform Student Verifications due to high number of Examination Candidates

The process of manually confirming the identities of a large number of exam candidates can be exceptionally time-intensive for a variety of reasons. Initially, it mandates skilled personnel to visually compare the physical attributes of each student with their provided identification documents, such as ID cards or passports. This procedure can be gradual and monotonous, particularly when dealing with a substantial candidate pool. Additionally, manual verification may encompass multiple stages, including validating signatures, cross-referencing data, or confirming other identification particulars. These steps can exacerbate the already time-consuming nature of the process. The supplementary steps necessitate meticulous examination and attention to detail, which can contribute to delays in the overall verification timeline.

Moreover, the considerable count of exam candidates might result in prolonged lines or waiting periods, leading to logistical complexities and inefficiencies in managing candidate flow. This can further elongate the time required for verification, potentially causing disruptions in the exam timetable. Moreover, manual verification is susceptible to human errors, encompassing misidentifications or mismatches of student identities. These inaccuracies can arise from factors like fatigue, distractions, or inherent biases, consequently further extending the verification process and introducing inaccuracies to the final outcomes. Conversely, the adoption of an automated identity validation system, like biometric facial recognition, can notably accelerate the procedure. Facial recognition technology has the capability to correlate a student’s facial features rapidly and precisely with their pre-recorded information, negating the necessity for manual visual confirmation. This advancement can significantly decrease the time and energy needed for authenticating the identities of exam candidates, leading to a more efficient and smooth exam operation characterized by heightened precision and dependability.

The National College Entrance Examination (NCEE), commonly known as "Gaokao," is a highly competitive admission test in China, which is attempted by millions of high school graduates or candidates with equivalent educational backgrounds annually.[4] In the year 2022, an astonishing 11.93 million students participated in the nationwide college entrance exams held in July. [5] According to China's national examination regulations, the process of confirming students' identities and conducting security checks generally begin 50 minutes before the commencement of the examination [5]. This encompasses visually confirming the identity of each student by comparing their physical features with their provided identification papers, like ID cards or exam slips. Additionally, conducting supplementary security checks is essential to guarantee the equity and honesty of the exam process. Due to the immense number of candidates participating in the exam, manually authenticating the identities of such a substantial group of students within a restricted time frame can be a challenging and time-intensive undertaking. Accomplishing the verification procedure for every student necessitates a substantial allocation of human labor and resources. This includes inspecting identification documents, affirming other identification particulars, and overseeing the movement of candidates. Apart from China's NCEE, other global examinations like the Test of English as a Foreign Language (TOEFL) and the Scholastic Assessment Test (SAT) also attract a substantial cohort of candidates. For example, TOEFL alone has an approximate global candidate count of 2.3 million, while the SAT is undertaken by more than 1.7 million high school graduates on a yearly basis. The manual authentication of identities for such an immense number of candidates can be exceptionally time-intensive and demanding in terms of resources. The usual procedure involves visually confirming the identity of each student by scrutinizing their identification documents, validating other pertinent identity details, and managing the organization of candidates. These tasks necessitate a considerable investment of time, manpower, and expenses.

When emergencies arise and candidates misplace their verification slips prior to exams, confirming their identities manually can introduce considerable time and exertion to the procedure. This could lead to avoidable postponements and heightened pressure for both candidates and exam organizers. Nonetheless, by utilizing biometric facial recognition technology, candidates can undergo swift and precise verification through their distinctive facial attributes, obviating the necessity for protracted manual authentication protocols. This approach streamlines the verification process, lessens the need for personnel and expenses, and guarantees a seamless and effective examination encounter, even in scenarios where verification slips are misplaced due to unforeseen circumstances.

Low Integrity and Effectiveness of Human Verification

In the conventional approach to confirming student identities, numerous potential inaccuracies or complications could emerge during the process. An example of this might occur if the documents presented by the students themselves is out-of-date or if the specifics don't line up with the information kept in official databases. Furthermore, human verification personnel are susceptible to errors, especially when confronted with resemblances in physical attributes among students, like comparable heights, hairstyles, or eyewear choices. This likeness can lead to instances of mistaken identity or erroneous outcomes.

These discrepancies and incongruities within the verification procedure can wield a substantial impact on its efficiency. Data integrity is also a critical concern in the verification process. Loss, corruption, or compromise of data can significantly impact the accuracy and reliability of the verification results. This condition may also compromise the dataset's overall security and reliability, leading to inaccurate findings and possible security breaches. Biometric facial recognition technology, on the other hand, provides a more dependable and accurate method of verifying students. Within the context of confirming student identities, the resemblances in physical characteristics among students, including elements like heights, hairstyles, eyewear choices, and other factors, can present a difficulty for verification personnel. This scenario can give rise to errors in human recognition, potentially leading to instances of mistaken identity or erroneous results. To illustrate, there is a possibility that verification staff might misidentify a student as another due to the resemblances in their physical traits, consequently producing verification results that are not accurate. [6]

1.2 Project Objectives

The primary objective of this thesis is to introduce an innovative application that capitalizes on facial recognition technology for the explicit purpose of identifying and validating candidates' presence within examination venues, all through the analysis of digital images. This endeavor encompasses the creation of an automated framework adept at detecting facial profiles within images, subsequently extracting intricate facial attributes, and ultimately executing the task of face recognition. The overarching goals of this project encompass:

Reduce Fraudulent in the Form of Impersonation Cases in Exam Location

With the increasing prevalence of cheating in various types of exams, ranging from quizzes to nationwide exams. Candidates have gotten more inventive in finding ways to gain an unfair advantage, ranging from simple tricks to sophisticated methods like hiring someone to take exams on their behalf. To address this problem at exam locations, facial recognition technology has emerged as a promising solution. Through the utilization of sophisticated biometrics powered by advanced Machine Learning (ML), facial recognition technology can adeptly register students for examinations, authenticate their identities, and guarantee their attendance during tests in a secure and reliable manner [8].

One of the key advantages of facial recognition technology is its immunity to identity theft attempts. The technology can properly recognize and match face data with the stored data, making it very difficult for candidates to impersonate others due to each person's distinctive facial traits. This proficiency substantially diminishes the likelihood of exam impersonation, where candidates aim to deceive by enlisting substitutes to undertake exams in their stead. The incorporation of facial recognition technology within examination venues holds the potential to significantly elevate the integrity and trustworthiness of the examination procedure. It offers a reliable and secure method of authenticating candidates' identities, eliminating fraudulent actions, and maintaining a level playing field for all candidates. Furthermore, the employment of facial recognition technology can assist to protect the reputation of the educational system by discouraging cheating and fostering academic integrity.

<u>Reduce the Time Taken and Increase the Effectiveness of the Verification and Validation Processes of Candidates</u>

Facial recognition technology holds the capability to streamline the authentication and validation procedures for candidates through the automated identification and acknowledgment of individuals, obviating the necessity for manual authentication by examination personnel. Traditional methods reliant on manual authentication, frequently entailing the scrutiny of identification documents and their alignment with the candidate's appearance, can consume substantial time and demand considerable labor, particularly in situations encompassing a significant number of candidates. By employing a facial recognition system, this procedure can be automated, leading to a reduction in the time allocated to candidate verification and validation processes, thereby enhancing the real-time efficiency of facial detection and recognition.

Furthermore, facial recognition technology facilitates an elevated degree of precision and reliability in contrast to conventional approaches. Human inaccuracies, encompassing errors in aligning identification documents or erroneously identifying candidates founded on their physical traits, can be mitigated or eradicated through the application of facial recognition systems. This can yield a more efficient process of candidate authentication and validation, guaranteeing accurate access authorization to examination sites and diminishing the potential for impersonation or deceitful undertakings.

Furthermore, a facial recognition system has the potential to amplify the efficiency of the validation procedure through the secure retention and handling of data, thereby safeguarding the integrity and auditability of the data. This precautionary measure aids in deterring any tampering or unauthorized alteration of data, which in turn ensures the credibility and dependability of the validation outcomes. The incorporation of biometrics driven by machine learning within facial recognition technology contributes to precise and swift facial expression recognition, thereby rendering it more challenging for candidates to engage in cheating or employ deceptive tactics during examinations, like enlisting the help of a friend or surrogate to undertake the test.

By April 2020, significant advancements had been made in the precision of face identification algorithms, achieving an error rate as low as 0.08%. This stands in stark contrast to the 4.1% error rate observed in the leading algorithm of 2014, as detailed in the findings of the National Institute of Standards and Technology (NIST). [9] This minimal error rate within facial recognition technology has significantly truncated the time-consuming task of confirming candidates' identities. The effectiveness of the authentication and validation processes is heightened, as the precise and dependable facial recognition system can swiftly and accurately confirm candidates' identities, thus curtailing the time investment necessary for the verification stage. The heightened accuracy of facial recognition technology permits the optimal utilization of resources, encompassing personnel, equipment, software, and time. The system can adeptly handle a substantial volume of candidates in real-time, all the while upholding the accuracy and reliability of the authentication process. This serves to streamline the examination administration by diminishing the time expended on identity confirmation, thereby facilitating the more efficient allocation of resources and elevating the general efficacy of the verification and validation protocols.

Increase Data Integrity of Face Recognition

Facial recognition technology, conversely, depends on algorithmic processes driven by data, which can enhance the integrity of data by diminishing the likelihood of human mistakes and predispositions. Through the utilization of facial recognition for identity authentication within examination venues, the precision and trustworthiness of the verification procedure can be elevated, consequently heightening the integrity of the data. The algorithms of facial recognition can autonomously identify and isolate facial attributes from digital images, subsequently contrasting them with pre-established data to validate the student's identity. This mechanized procedure curtails the possibilities of human errors, biases, and incongruities that may be linked with manual verification techniques, thus strengthening the validity of the authentication process.

To ensure the accuracy and integrity of facial recognition data, it is essential to train the AI or ML models using carefully selected datasets that are not overfitting or underfitting and are unbiased. This requires meticulous data preparation, which includes removing biases, noise, and inconsistencies from the training data. Furthermore, the training procedure should encompass a wide array of facial images that encompass diverse ethnicities, genders, ages, and expressions, in order to preclude bias and enhance the model's resilience. Through the meticulous curation and training of facial recognition models utilizing top-notch and diverse datasets, the soundness and effectiveness of the data employed for identification and authentication can be elevated. This guarantees the dependability, precision, and impartiality of the facial recognition system, ultimately fostering elevated data integrity throughout the verification process.

In ideal circumstances, facial recognition software can attain nearly flawless precision. Facial recognition technology has made substantial strides in recent times, consistently lowering its error rates. In assessments like the NIST's Facial Recognition Vendor Test (FRVT), verification algorithms are utilized to match participants against well-defined reference images, like passport photos. These types of comparisons can yield accuracy levels as remarkable as 99.97% [9]. This underscores the efficiency of facial recognition algorithms in precisely recognizing and confirming individuals. The AI or ML model responsible for training facial recognition must guarantee that the chosen training dataset is neither over nor underfitting and is also free from biases. As a result, this approach enhances the excellence and reliability of the data.

It's important to highlight that preserving data integrity is a continuous endeavor. New data might need to be consistently curated and integrated into the training procedure to accommodate changing situations and enhance the precision and equity of the facial recognition system. Consistent supervision, assessment, and enhancements to the training data play a crucial role in maintaining the system's effectiveness and reliability when it comes to identifying and authenticating individuals in examination venues and various other scenarios.

1.3 Project Scope and Direction

The project's culmination is an application for facial recognition designed to validate the identities of exam candidates during examination sessions. Essentially, the project entails collecting facial photos of exam candidates exhibiting diverse characteristics, including different angles, lighting conditions, shadows, expressions, and other factors if feasible. The scope of the project includes the task of recognizing exam candidates in photographs, maybe in real-time circumstances as well. In this endeavor, the recognition process begins with employing cameras equipped with facial detection technology to locate and pinpoint the facial image of an exam candidate, even when they are positioned straight ahead or in a side profile within the image. The undertaking might necessitate the establishment of essential infrastructure within exam venues, which could encompass the installation of cameras and associated hardware, along with the configuration of software for facial recognition capabilities. This process might also entail resolving any physical, technical, or logistical challenges posed by varying examination environments, including differences in lighting conditions and camera perspectives.

This undertaking would involve obtaining appropriate consent from the authorities, adhering to privacy regulations, and assuring the security and reliability of the acquired data. The model would then undergo training using this dataset to familiarize itself with and distinguish the distinctive facial attributes of individual exam candidates. The project would encompass the creation and implementation of a robust identity verification mechanism using facial recognition technology. This might entail capturing facial images of exam participants both during registration and during the actual exams, subsequently matching these images against stored facial templates to verify their identity. In this process, the facial image is analyzed. This facial recognition technology would rely on 2D (two-dimensional) representations, as it's more convenient to compare 2D images within a database compared to 3D images. The application will analyze facial geometry to identify the unique facial traits that set each face apart. Based on these facial characteristics, the process involves transforming analog information, such as a facial image, into a digital representation of data that's essentially distilled into a mathematical formula. A determination is made by assessing whether the resulting facial print corresponds to the image stored within the facial recognition database.

The project might entail the integration of the facial recognition system with the pre-existing exam management system, which comprises aspects like enrollment, scheduling, and result management. Within this endeavor, there would be an exhaustive process of testing and validating the facial recognition system to guarantee its precision, dependability, and security. This could entail conducting thorough tests across various examination venues, validating the system against known identities, and tracking its performance over an extended period. Any identified issues or errors during this testing phase would necessitate resolution.

On the whole, the project's scope and trajectory concerning facial recognition for identity verification within examination venues would adopt a comprehensive approach. This would encompass the selection of appropriate technology, the establishment of necessary infrastructure, the accumulation and training of data, the identity verification procedure, the integration with pre-existing systems, rigorous testing and validation, as well as the delivery of training and accompanying documentation.

1.4 Contributions

The system outlined in this project aims to create an automated mechanism for detecting and recognizing faces to ensure accurate candidate identification within an examination framework. Employing facial recognition technology can introduce an extra stratum of security by precisely confirming the identity of individuals partaking in exams, relying on their distinct facial attributes. This mechanism holds the potential to deter instances of impersonation and cheating during exams, thereby guaranteeing that only authorized individuals gain access to the examination process. Facial recognition facilitates speedy, automated, and smooth verification, circumventing issues such as the loss or theft of identification credentials.

This project aims to enhance the effectiveness of verifying exam candidates' identities, primarily by expediting the process through the implementation of facial recognition. This technology can streamline the identity verification procedure within examination sites, resulting in swifter and more efficient outcomes when compared to traditional methods like manual ID checks. Automated facial recognition can promptly align exam takers' facial images with stored templates, diminishing the necessity for manual interference and conserving time. Furthermore, facial recognition technology can improve accessibility for candidates with disabilities, who may have difficulty providing physical identification credentials. Face recognition technology can give a non-intrusive and convenient way for students to confirm their identification, making tests more inclusive and accessible.

In addition, facial recognition technology can bring about a substantial improvement in accessibility for candidates with disabilities. Let's consider a student who has mobility impairments and might struggle to reach their wallets, bags, or pockets to retrieve physical identification documents. In this context, conventional methods might be both inconvenient and frustrating. However, with the integration of facial recognition, this student could easily confirm their identity by looking at the camera, which is particularly beneficial when the algorithm requires complete facial features for accurate identification. This mechanism offers a non-intrusive and user-friendly approach to identity verification, making the exam environment more inclusive and accommodating.

Next, implementing automated identity verification through facial recognition can alleviate the administrative load on exam administrators and staff. This approach has the potential to diminish the necessity for labor-intensive tasks like manual record-keeping, data entry, and document management associated with identity validation. This shift in workload allocation can release valuable resources, enabling these personnel to focus on other crucial responsibilities. With facial recognition's ability to provide a seamless and user-friendly experience for exam takers, it eradicates the requirement for physical documents and manual identity inspections. Consequently, this advancement can mitigate the stress and apprehension often experienced by exam candidates. From the perspective of the institution, incorporating facial recognition for identity verification in exams aligns with important rules and standards that ensure fair exams, data security, and confidentiality. This approach demonstrates the institution's commitment to conducting verification in a way that follows the rules, protecting the privacy and rights of exam candidates.

Facial recognition technology holds promise for future advancements, including its fusion with other biometric methods, refinement of accuracy, and extension to different examination venues. These possibilities offer avenues for continuous innovation and enhancement in the realms of exam administration and identity validation. Employing facial recognition for identity validation in examination sites brings forth numerous benefits, encompassing heightened security, elevated efficiency, improved accessibility for those with disabilities, decreased administrative load, data reliability, enriched user experience, adherence to regulations, and prospective innovation. This integration helps to prevent exam impersonation and dishonesty, makes identity checks easier, reduces admin tasks, ensures exam fairness, and paves the way for innovation.

1.5 Report Organization

This report will include of seven sections, namely the introduction, a review of the literature, the system's methodology or approach, the design of the system, its implementation, an evaluation, and discussion of the system, and finally, the conclusion.

Chapter 1 covers various aspects such as project background, project introduction, problem statements, objectives, and contributions. These elements collectively provide a comprehensive overview of the project.

Chapter 2, the literature review, encompasses relevant historical background, along with an exploration of existing systems, technologies, and facial recognition algorithms that are currently in use.

Chapter 3, titled "System Method or Approach," focuses on clarifying the methodology used for the system development process, specifically emphasizing the adoption of the Prototyping model, along with the presentation of the system's design diagram.

Chapter 4, covers "System Design," provides an overview of the initial stages of the proposed project. This chapter will cover aspects such as the system block diagram, a description of the system's flow, and the design of interactions among system components.

Chapter 5, designated as "System Implementation," covers the establishment of both hardware and software configurations. This chapter details the initial setup and configuration of the Integrated Development and Learning Environment (IDLE), along with a description of the system's operation, accompanied by reference screenshots.

Chapter 6, is about "System Evaluation and Discussion," entails an examination of the system's performance and evaluation, encompassing feedback and reviews obtained after testing. This chapter also addresses the project's encountered challenges and issues, concluding with recommendations.

Chapter 7 is the conclusion of the project serves as a summarization of the proposed system.

Chapter 2

Literature Review

2.1 Different Approaches of Face Recognition

Face recognition has captivated a variety of backgrounds, including face recognition, facial patterns, computer graphics, computer vision, neural networks, and psychology. However, a few techniques for facial recognition are as follows: [7]

1. Holistic Matching Methods
2. Feature-based (structural) Methods
3. Hybrid Methods Holistic Matching Methods

In holistic matching, this strategy employs the complete facial image as the unprocessed input for a recognition system. The holistic matching technique can be classified into linear and non-linear projection techniques. [7] In the case of non-linear projection, the initial image is transformed into a high-dimensional space where facial features are streamlined, and then conventional linear methods are employed. In linear projection methods like principal component analysis, the input images undergo scrutiny based on their grayscale values. [7] In order to train the system, multiple perspectives of individuals are amassed within a database. Each of these images will be translated into a lengthy vector using grayscale representation. When it comes to recognition, the vectors corresponding to pertinent faces are matched against all stored vectors in the database. The vector image in the database that most closely resembles the unfamiliar face is determined as the recognized face. Given the potentially extensive scale of each vector, variations in lighting conditions and noise might affect the accuracy of grayscale levels. The assembly of images serves as a training dataset for image comparison and the generation of eigenfaces. At the moment, image information can be employed to derive Eigenfaces using a mathematical method called Principal Component Analysis (PCA). [10]
Below is the figure illustrating the flow of the eigenface-based algorithm.

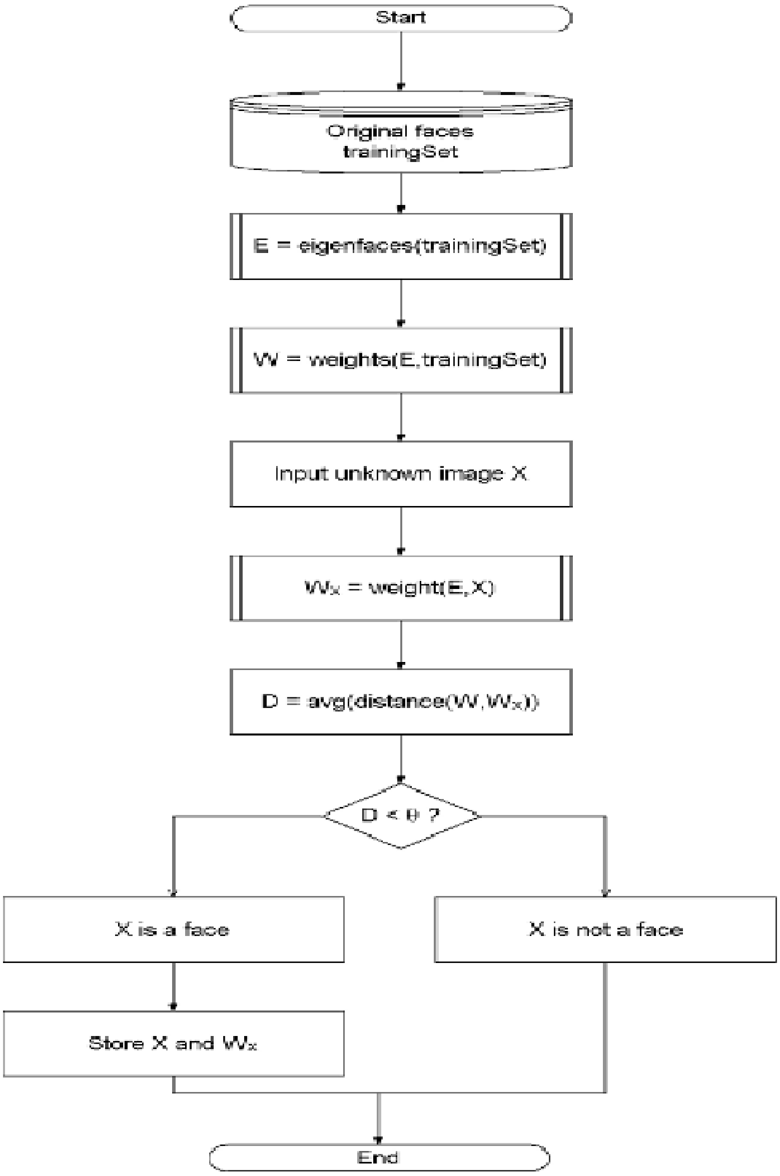


Figure 2.1: Flow chart of eigenface-based algorithm

Features-based (structural) Methods [10]:

This approach entails the initial extraction of the positions and specific characteristics within limited areas, encompassing geometric and/or visual data related to facial components like eyes, nose, and mouth. These isolated characteristics are subsequently fed into a structural classifier. A common challenge in feature extraction is the “restoration” of features, which arises when features are not visible due to large variations, such as variations in head posture when comparing frontal and profile images. The methods used to differentiate among the three distinct extraction techniques are outlined below: (I)Generic methods utilizing edges, lines, and curves, (II)Feature-template-based method, and (III)Structural matching methods that integrating geometric restrictions on the features. Because the system employs depth and a measurement axis, even a side view offers sufficient data for precise identification. The standard process for the 3D system encompasses various stages, such as detection, position determination, measurement, representation, and matching.

Hybrid Methods:
A hybrid approach to face recognition merges both holistic and feature-based (structural) methods. Generally, the facial image is captured in 3D, and then the system employs an algorithm to compute facial feature data.

2.1.1 Haar – Cascade Classifier Algorithm

Paul Viola and Michael Jones introduced the Haar-Cascade classifier as a proficient tool for object detection. [11] According to Sander Soo (2014), a Haar-like feature analyses
the adjacent rectangular sections in a detection window at a fixed position, sums up pixel intensities in each area, and subsequently computes the disparity between these totals. [11] The outcome will be utilized for classifying the image's subsections. The Haar-Cascade classifier will conduct training using both positive and negative images, enabling it to identify objects through the learned patterns. This classifier can recognize various human features, including complete body, lower body, eyes, and front-facing face.

2.1.2 R-CNN (Region-based Convolutional Neural Network)

In 2014, a group of scientists at UC Berkeley developed the R-CNN, a deep convolutional network capable of identifying 80 distinct object categories within photographs. [12] R-CNN consists of three main components: firstly, it generates 2,000 region suggestions through the Selective Search approach. After resizing these regions to a predefined consistent size, a 4,096-dimensional feature vector is extracted from each proposed region. Lastly, it employs a pre-trained SVM (Support Vector Machine) algorithm to classify the region proposal as either the background or one of the object categories.

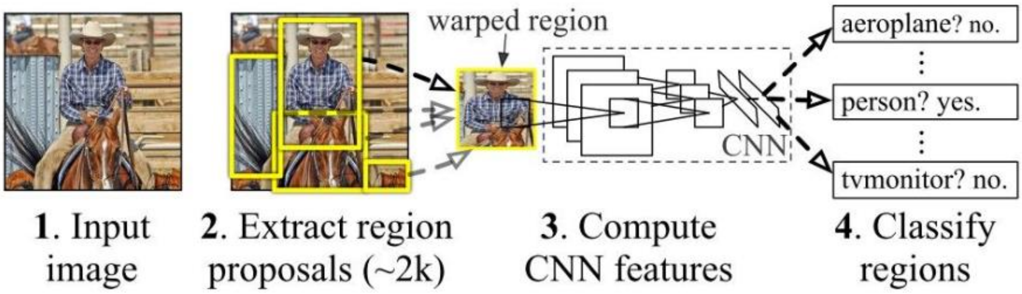


Figure 2.1.2: R-CNN

2.1.3 Adaboost (Adaptive Boosting)

Adaboost is a method that can swiftly process images with remarkable precision in detection. It's a machine learning algorithm that constructs a robust classifier by combining weak classifiers in multiple stages. To achieve impressive face detection performance, Adaboost-based face detection employs Haar 2D features, "integral pictures," and a "cascade" approach [13]. In the investigation of the POFA (Pictures of Facial Affect) dataset, perfect accuracy of 100% has been achieved in face detection. Figure 2.1.3 below illustrates the identified face, highlighted with a red rectangle [13].

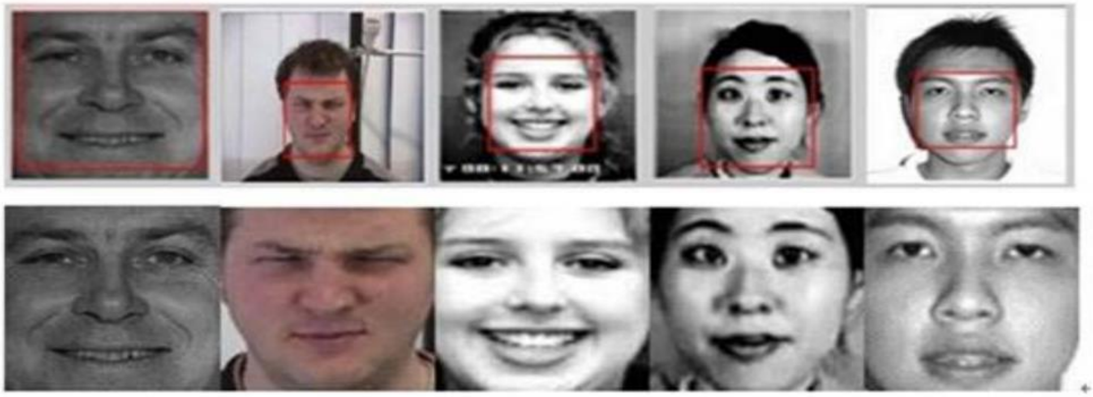


Figure 2.1.3: Face Detection based on Adaboost

(The regions indicated by the red rectangles are the results of detected face area)

2.1.4 LBP (Local Binary Patterns)

LBP (Local Binary Pattern) is a simple and effective texture operator that assigns binary labels to image pixels by comparing the neighborhood of each pixel against a threshold, resulting in a binary numerical representation [14]. In the beginning, nearby pixels within a predefined patch receive an integration value if their grayscale level surpasses that of the central pixel; otherwise, a value of zero is assigned. Then, the central pixel is assigned a binary digit based on the integration values of its neighboring pixels. The classic LBP operator employs a 3 x 3 patch, producing an 8-digit binary code derived from the neighboring pixel values. After labeling all the pixels in an image, the final step involves creating a histogram using the resulting LBP array.

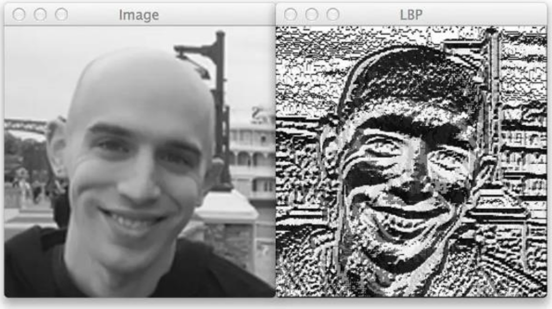


Figure 2.1.4: An example of computing the LBP outcome (right) from the original image (left).

2.2 Application of Face Recognition

2.2.1 Android’s Face Unlock

Apart from the traditional use of PINs and the evolving fingerprint scanners, facial recognition technology has become a common security feature in smartphones. Nowadays, nearly all mobile phones incorporate facial recognition technology as an alternative to PINs or fingerprints for unlocking. However, there is a security concern with this standard facial recognition method because it solely relies on the front-facing camera in combination with a 2D facial recognition algorithm. This makes it cost-effective and convenient to capture an image of the user's facial features using only two dimensions. Nonetheless, since this is merely a 2D image, it becomes vulnerable to a potential thief who could easily deceive the system and unlock the phone with a simple snapshot of the user's face [15]. In 2018, Samsung introduced the Galaxy S9 or S9+ smartphones featuring a security feature called "Intelligent Scan," which seems to blend iris and facial recognition technologies [16]. Initially, the phone performs a facial recognition scan as the first step in the unlocking process. If this facial recognition scan fails to unlock the device, it then proceeds to check the user's irises. In cases where neither of these methods succeeds, Intelligent Scan makes an effort to verify the user's identity by combining both approaches. Importantly, this entire process occurs almost instantaneously [17]. One challenge faced by iris scanning technology is its effectiveness under bright sunlight conditions [18]. In such situations, the device turns to verifying the user's identity through facial recognition using Intelligent Scan. Conversely, in low-light conditions, facial recognition may not perform at its best, prompting the phone to utilize the infrared iris scanner instead [18]. These processes all take place rapidly, making Intelligent Scan a swift and convenient method for unlocking the device. It's worth noting that Intelligent Scan isn't a new form of biometric authentication by itself; rather, it's an intelligent algorithm that selects the most suitable authentication method based on the prevailing environmental conditions [18].

2.2.2 Apple Face ID

Samsung introduced its facial recognition unlock feature prior to Apple's Face ID on the iPhone X, but it's widely perceived as being less secure than Apple's technology [17]. Face ID's process involves using an infrared floodlight to illuminate the user's face, and this illumination remains unaffected by the ambient lighting conditions since it operates in the non-visible spectrum. Afterward, a grid of 30,000 infrared laser points is projected and subsequently reflected by the floodlight [15]. Rather than taking a picture of the infrared light pattern, the specialized infrared camera in Face ID detects slight variations in the matrix point reflections caused by tiny movements of the user's face. This process enables the capture of extremely accurate 3D depth information [15]. iPhone devices then compare these scans or mathematical models of the user's face with the one that has been previously set up and stored on the device to determine if there is a match. If a match is found, it unlocks the phone or authorizes an Apple Pay payment [19]. Upon the initial release of Face ID, Apple emphasized its commitment to preventing the system from being deceived by methods such as using photographs. To bolster security, Apple collaborated with expert Hollywood mask makers and makeup artists to train the model and ensure Face ID's resistance to such bypass attempts [19]. Face ID mandates user attention for unlocking, so if a user's eyes are closed or they are not looking at the device, it won't unlock. Additionally, the company stated that the likelihood of someone unlocking another person's iPhone with Face ID is as low as one in a million [19].

2.2.3 DeepFace

DeepFace is a sophisticated facial recognition system developed by a team of researchers at Facebook, employing deep learning techniques. The team utilizes this facial recognition system for tagging images on Facebook [20]. This procedure entails the comparison of a human face captured from a digital image or a video frame with a Facebook database containing multiple faces to identify a match. The illustration below demonstrates that DeepFace is an 8-layered convolutional neural network model, with each layer designated by a combination of a letter and a number. The numbers range from 1 to 8, indicating the layer's index, while the letters convey the layer's type. Specifically, "C" represents a convolutional layer, "M" indicates maximum pooling, "L" signifies a locally connected layer, and "F" denotes a fully connected layer [20]. Although it has a relatively limit number of layers, DeepFace boasts an extensive parameter size. Nevertheless, the implementation of DeepFace raises certain apprehensions, including issues related to privacy (capturing people's movements without consent), bias (potential discrimination against individuals of varying races, ages, or genders), and misuse (the misuse of facial images for malicious or illegal purposes) [20].

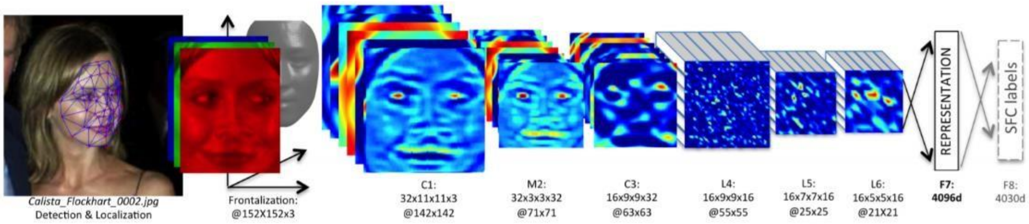


Figure 2.2.3: DeepFace Model Structure

2.2.4 FaceNet

Back in 2015, Schroff and his team at Google introduced FaceNet, a deep neural network designed to extract distinctive facial characteristics from images of individuals' faces [21]. FaceNet operates by taking an image of a person's face as input and producing a 128-dimensional vector that encapsulates the essential facial features. This vector, often referred to as an 'embedding' in machine learning, condenses the distinctive traits of the face into a concise numerical representation [21]. The figure below shows the algorithm of FaceNet.

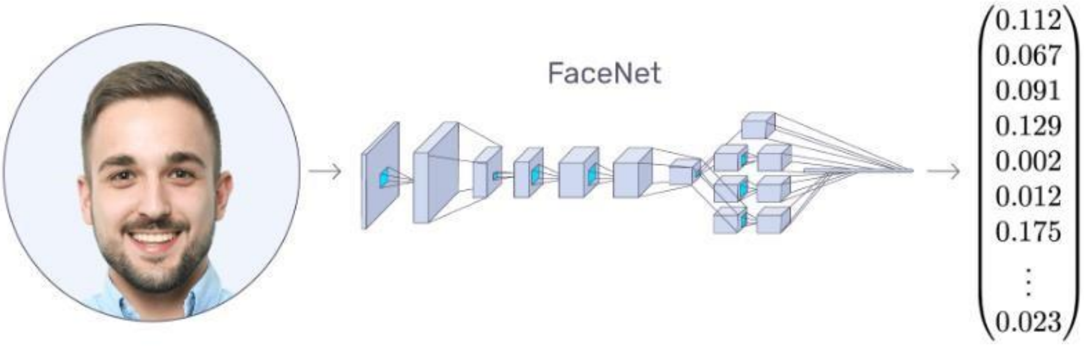


Figure 2.2.4: FaceNet Algorithm (takes image of a face as input and outputs the embedding vector)

Embeddings enable the representation of a facial image as coordinates on a Cartesian coordinate system. To train FaceNet, a substantial dataset of facial photos is necessary. During the training process, FaceNet initializes random vectors for each image, leading to scattered data points on the plot [21]. It then proceeds to randomly choose an anchor image, a positive image (belonging to the same person), and a negative image (belonging to a different person). The network's parameters are adjusted in such a way that the positive image becomes closer to the anchor image than the negative image, a technique known as 'Triplet Loss' [21]. This iterative process continues until all images of the same person are close to each other and far from images of different individuals, resulting in learned embeddings that effectively capture facial features [21].

2.3 Strength and Weakness of Face Recognition Applications

	Strength	Weakness
Android's Face Unlock	• Provides a quick and convenient method for unlocking the device by merely gazing at it. • User convenience with a simple setup and straightforward operation. • Incorporation with a range of applications and services, including those related to banking and payments.	• Availability on Android devices can vary depending on the specific hardware model. • Exhibits reduced reliability in demanding situations like low-light settings, extreme angles, or alterations in the user's appearance.
Apple Face ID	• High security with complex depth-sensing camera system that captures and examines more than 30,000 infrared dots. • Faster and more efficient than fingerprint biometrics using capacitive-based technology and optical sensors.	• Privacy worries arise when third-party app developers utilize the system to gather facial data. • The accuracy is compromised by various environmental factors like fluctuations in lighting, different poses, and alterations in facial expressions.

DeepFace	• Achieves accuracy rates that exceed the performance of humans. • Have a substantial user population and a vast dataset for training purposes. • Quick and effective real-time processing capabilities.	• Privacy concerns involving the collection and retention of facial data. • Significant potential for fraudulent and criminal activities due to vulnerabilities.
Face Net	• Capability of accurately identify faces regardless of different facial expressions and angles. • Capable of functioning effectively under varying lighting conditions. • Capable of recognizing faces that have facial markings. • Can identify partial or half-faces accurately.	• Unable to provide accurate matches when images are edited with text overlaid on them. • Tends to place excessive emphasis on the similarity of eyewear when individuals are photographed wearing glasses in the images.

Table 2.3: Strength and Weakness of Face Recognition Applications

Chapter 3

System Methodology/Approach

3.1 Methodology

The chosen approach for this project is the Prototyping Model, a methodology that entails the creation, testing, and iterative refinement of application prototypes until a desirable outcome is attained. This iterative process sets the stage for the subsequent development of the complete software or product based on the well-defined and refined requirements. This approach is most effective when not all project requirements are fully understood upfront. It entails an iterative and exploratory trial-and-error process as part of the development journey. As the project is carried out by individual rather than in a team setting, Prototype Model is ideal for individual programming projects because it accommodates evolving requirements, encourages quick progress, and simplifies development, all while ensuring the software aligns with the developer's vision.

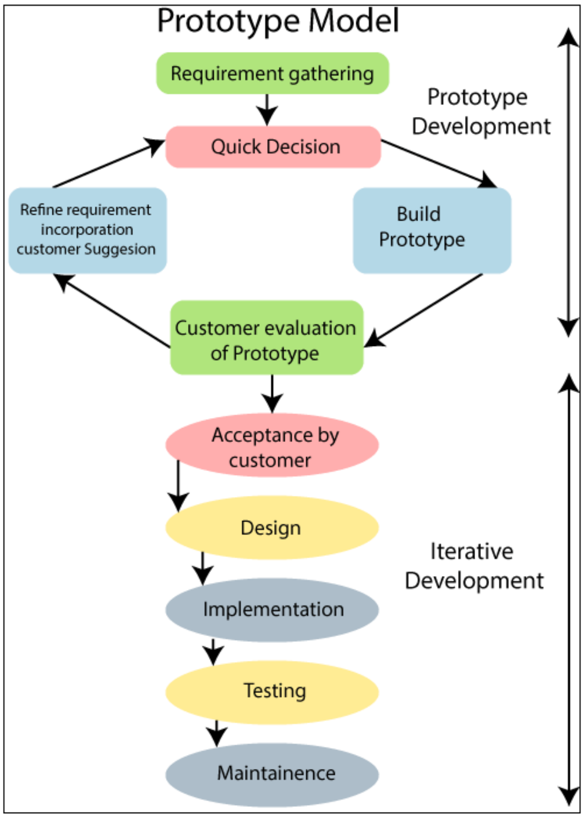


Figure 3.1: Prototype Model

Main phases of Prototyping Model:

1. **Requirements Gathering**

   In this phase, the initial project requirements are gathered. These requirements include aspects such as user identification, user enrollment, hardware requirements, user interface, and more.

2. **Prototype Development**

   The Prototype development phase includes a rapid design process. This stage involves creating a basic system design, although it may not be the final, comprehensive design. For instance, I constructed a face recognition attendance system application using PyCharm Community and Firebase as an online database to store relevant data.

3. **User Evaluation of Prototype and Suggestions to Refine Requirements**

   In this phase, the proposed system will be introduced to the Utarian for an initial evaluation. I will gather feedback from both my friends and my supervisor regarding the prototype testing. Subsequently, I will enhance the prototype based on this feedback until the users are content with the result. Once the final prototype is approved, we will proceed to develop the ultimate system.

4. **Implementation and Testing**

   In this final phase, once the final face recognition attendance system application is constructed based on the approved final prototype, it undergoes comprehensive testing to ensure its functionality and reliability. After successful testing, it is deployed for production use.

3.2 Use Case Diagram

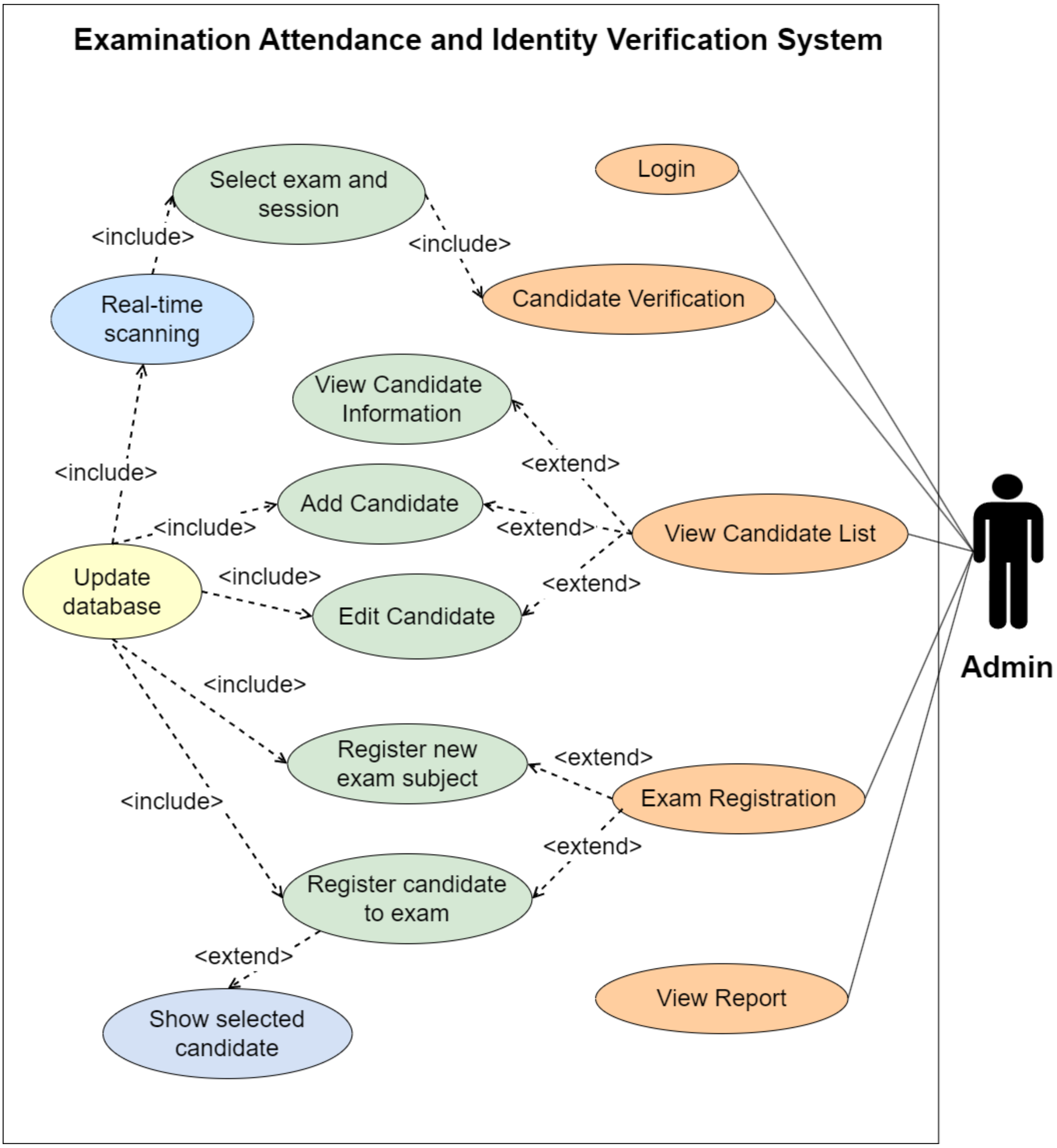


Figure 3.2: Overall Use Case Diagram

3.3 Use Case Description

Use Case Name: Login	ID: 1	Importance Level: High
Primary Actor: Admin, System	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Admin - wants to login to his/her account.		
Brief Description: This use case describes how we handle the process of login to the users' account.		
Trigger: User wants to view his/her account details. Type: External		
Relationships: Association: Admin Include: type user Id and password Extend: separate to admin login and student login		
Normal Flow of Events: 1. Admin needs to enter user Id and password. 2. The system will check whether this user Id exists in the system or not. 3. The system will check whether the user's Id and password correct or not. 4. The system will check and validate credentials. 5. The system will load all the system function of the admin's account and direct access to admin main page.		
Sub Flows: Not applicable		
Alternate / Exceptional Flows: 1a. Admin can press "Show Password" if would like to display the entered password.		

Table 3.3.1: Use Case – Login Module

Use Case Name: Add Candidate	ID: 2	Importance Level: High
Primary Actor: Admin, System	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Admin - wants to add new candidate into the system		
Brief Description: This use case describes how we handle the process of admin to add new candidate into the system.		
Trigger: Admin wants to add new candidate information. Type: External		
Relationships: Association: Admin Include: Not applicable Extend: Not applicable		
Normal Flow of Events: 1. To begin the process of adding a new candidate to the system, admin need to navigate to the 'Candidate List' tab and selects the 'Add Candidate' button. 2. Upon choosing 'Add Candidate,' an entry form promptly appears on the screen. 3. Then, inputting essential details such as the candidate's name, ID, faculty, starting year of intake, current studying year information. 4. If the information is confirmed, clicking 'Add,' the candidate's information is securely and instantly updated in the Firebase database.		
Sub Flows: Not applicable		
Alternate / Exceptional Flows: Not applicable		

Table 3.3.2: Use Case – Add Candidate Module

<table>
<tr><td>Use Case Name: Register Candidate to Exam Subject</td><td>ID: 3</td><td>Importance Level: Medium</td></tr>
<tr><td>Primary Actor: Admin, System</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
• Admin – register candidates to selected subject and session</td></tr>
<tr><td colspan="3">Brief Description: This use case describes how admin handle registration of candidates to selected subject and session.</td></tr>
<tr><td colspan="3">Trigger: Admin wants to register candidate to selected subject and session.
Type: External</td></tr>
<tr><td colspan="3">Relationships:
Association: Admin
Include: Not applicable
Extend: Not applicable</td></tr>
<tr><td colspan="3">Normal Flow of Events:
1. Admin needs to direct to the Exam Registration page.
2. Then, admin need to click on the ‘Register Candidate to Exam’ button at the bottom of the page.
3. Admin needs to select desired subject and session.
4. Admin must click on the ‘Confirm’ button as the selected candidate correctly added under the session child node inside Firebase.
5. Admin can register for the candidate by select on the candidate name and click on the ‘Add Candidate’ button.
6. If admin would like to view the selected candidates. There is a ‘Show Candidate’ button to display the selected candidates list.</td></tr>
<tr><td colspan="3">Sub Flows: Not applicable</td></tr>
<tr><td colspan="3">Alternate / Exceptional Flows: Not applicable</td></tr>
</table>

Table 3.3.3: Use Case – Register Candidate to Exam Subject Module

Use Case Name: Verify Candidate Identity	ID: 4	Importance Level: High
Primary Actor: Admin, System	Use Case Type: Detail, Essential	
Stakeholders and Interests: • Admin – allow candidate to scan face for identity verification in exam		
Brief Description: This use case describes how we handle the process of admin allow candidate to scan their faces for student identity verification		
Trigger: Admin allow candidate to do facial recognition to verify identity. Type: External		
Relationships: Association: Admin Include: Real-time scanning, update database Extend: Not applicable		
Normal Flow of Events: 1. Admins have to open camera and on the scan face function. 2. Candidates have to stand in certain range for real time scanning for identity verification. 3. Once candidate's face has been recognized, their attendance will be taken at the same time. 4. A confirmation notification will display and attendance information will update to Firebase. 5. If candidate face structure n is unable to be recognized, then it will remain at the real time scanning phase.		
Sub Flows: Not applicable		
Alternate / Exceptional Flows: Not applicable		

Table 3.3.4: Use Case – Verify Candidate Identity Module

3.4 Activity Diagram

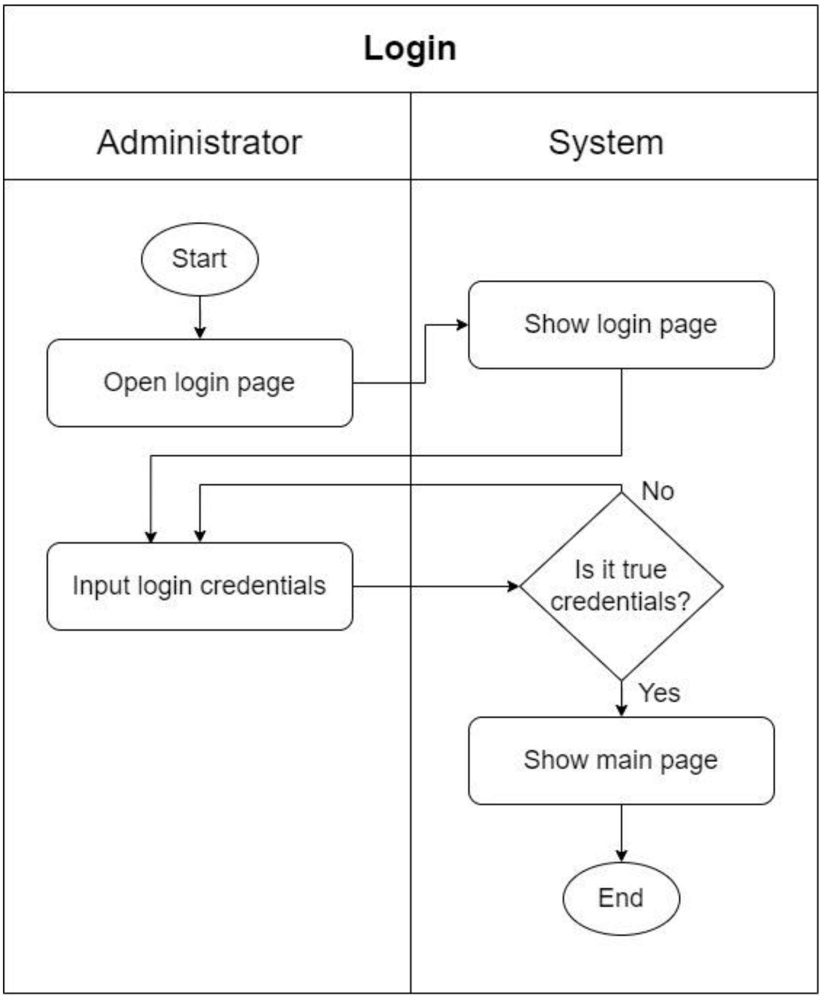


Figure 3.4.1: Login Activity Diagram

In the login activity diagram, admin need to access to the login page before entering the system. Then, it begins with the user entering their user ID and password in the login form. The system then checks these credentials for correctness. If the user ID and password are valid, the system grants access, and it will be directed to the main page. However, if the credentials are incorrect, an error message is displayed, and the user is prompted to re-enter their information. This loop continues until the user enters the correct credentials or decides to exit the login process.

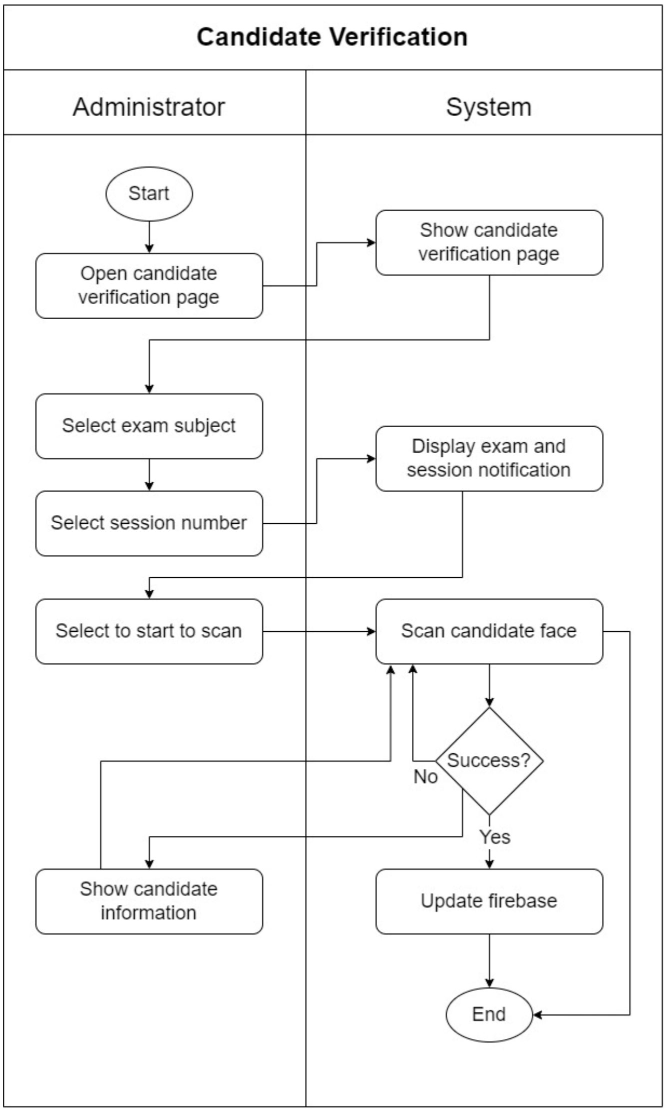


Figure 3.4.2: Candidate Verification Activity Diagram

In the candidate verification activity diagram, the admin must first access the candidate verification page before proceeding to scan candidates' attendance. Within the candidate verification page, they are required to select the exam subject and corresponding session. Upon confirming this information, a notification will appear to confirm the selection. Subsequently, the admin can commence scanning candidates' faces for attendance. If the scanning and verification process is successful, the system displays the candidate's information and the recorded time will update to Firebase, confirming their attendance for the selected exam and session. In case of unsuccessful scanning, the admin remains on the scanning page to make additional attempts. Finally, when the scanning session is complete, they have the option to choose to terminate the scanning process.

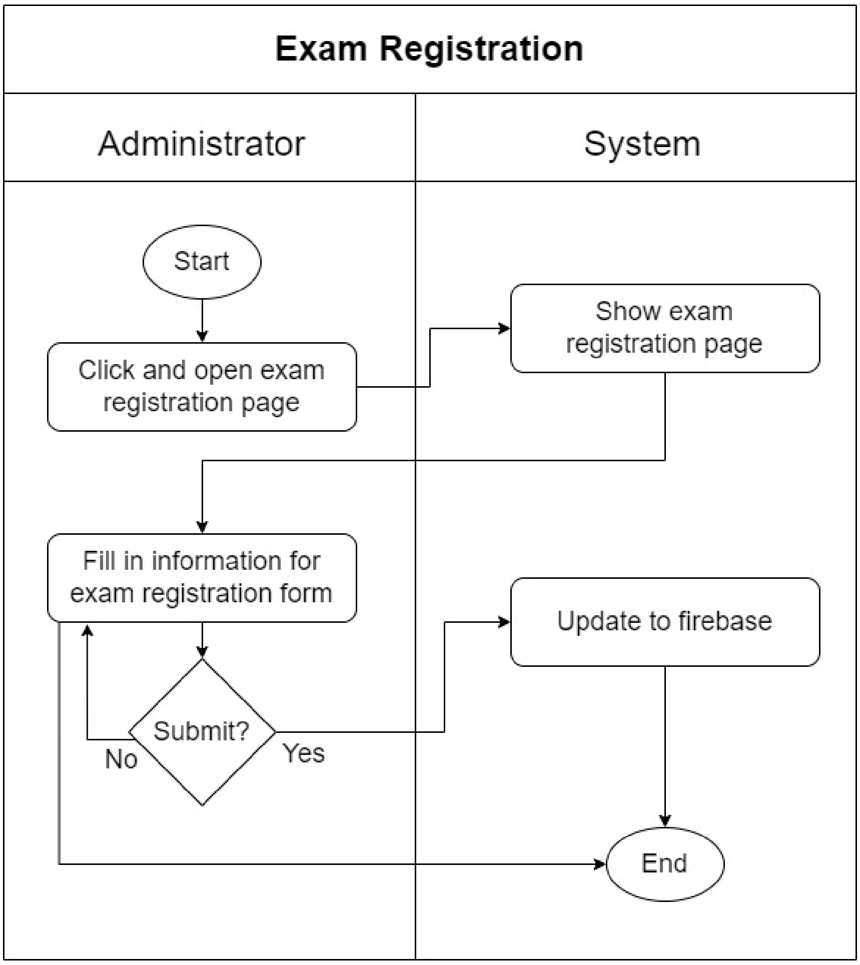

Exam Registration
Administrator
System
Start
Click and open exam registration page
Show exam registration page
Fill in information for exam registration form
Update to firebase
Submit?
No
Yes
End

Figure 3.4.3: Exam Registration Activity Diagram

In the exam registration activity diagram, the administrative user is responsible for adding new exam subjects to the system. To initiate this process, the admin first accesses the exam registration page within the system. On this page, they proceed to input essential information about the exam, including the subject name, date, number of sessions, and the exam location. If the provided information is confirmed, the user can proceed by clicking the 'submit' button to register the subject. Subsequently, the details of the exam subject are stored in Firebase for future reference. In the event that the information is not confirmed, the user has the option to continue editing it. Additionally, the user can choose to continue the session if they intend to add another exam subject, or they can opt to conclude the exam registration session if no further additions are needed.

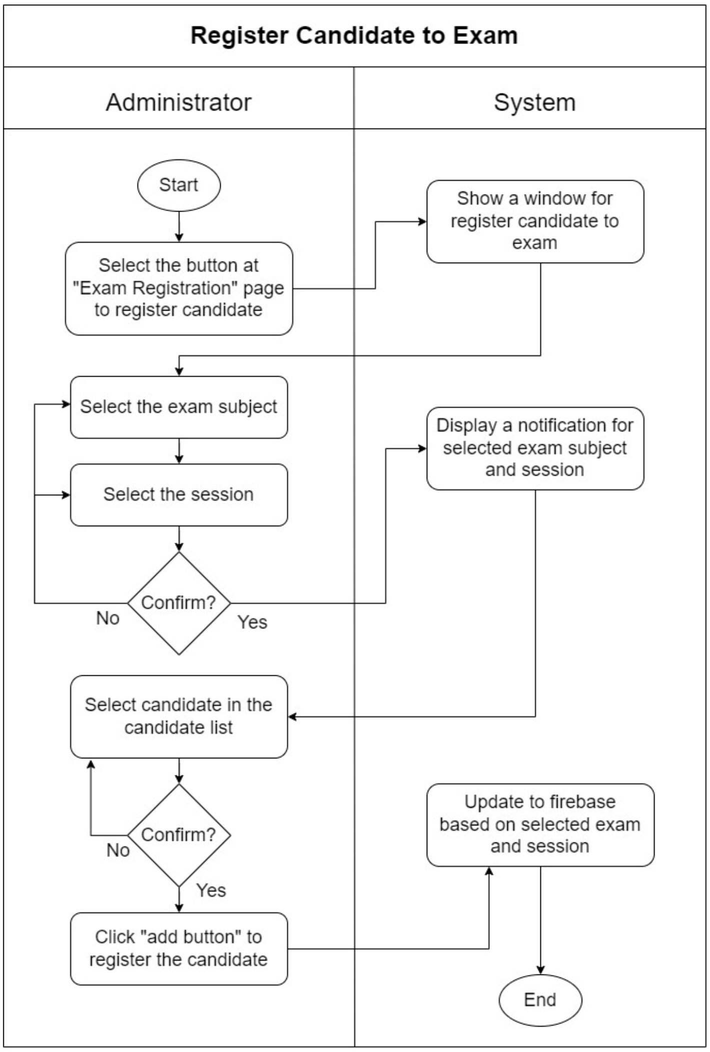

Register Candidate to Exam
Administrator
System
Start
Show a window for register candidate to exam
Select the button at "Exam Registration" page to register candidate
Select the exam subject
Display a notification for selected exam subject and session
Select the session
Confirm?
No
Yes
Select candidate in the candidate list
Update to firebase based on selected exam and session
Confirm?
No
Yes
Click "add button" to register the candidate
End

Figure 3.4.4: Register Candidate to Exam Activity Diagram

For the activity diagram of register candidate to exam, the task needs to be carried out under the "Exam Registration" page. The initial step involves the administrator's selection of the exam subject and session. Once the choice is made, the administrator proceeds by clicking the confirmation button, triggering the appearance of a notification displaying the selected subject and session for verification. If, at this point, an erroneous selection is identified, the administrator has the flexibility to rectify it by re-selecting the exam subject or session. Following confirmation, the administrator can choose candidates from a list and add them to the registration process by clicking "Add Candidate." This updates candidate information in the Firebase database, securely linking it to the chosen exam subject and session for streamlined management and tracking.

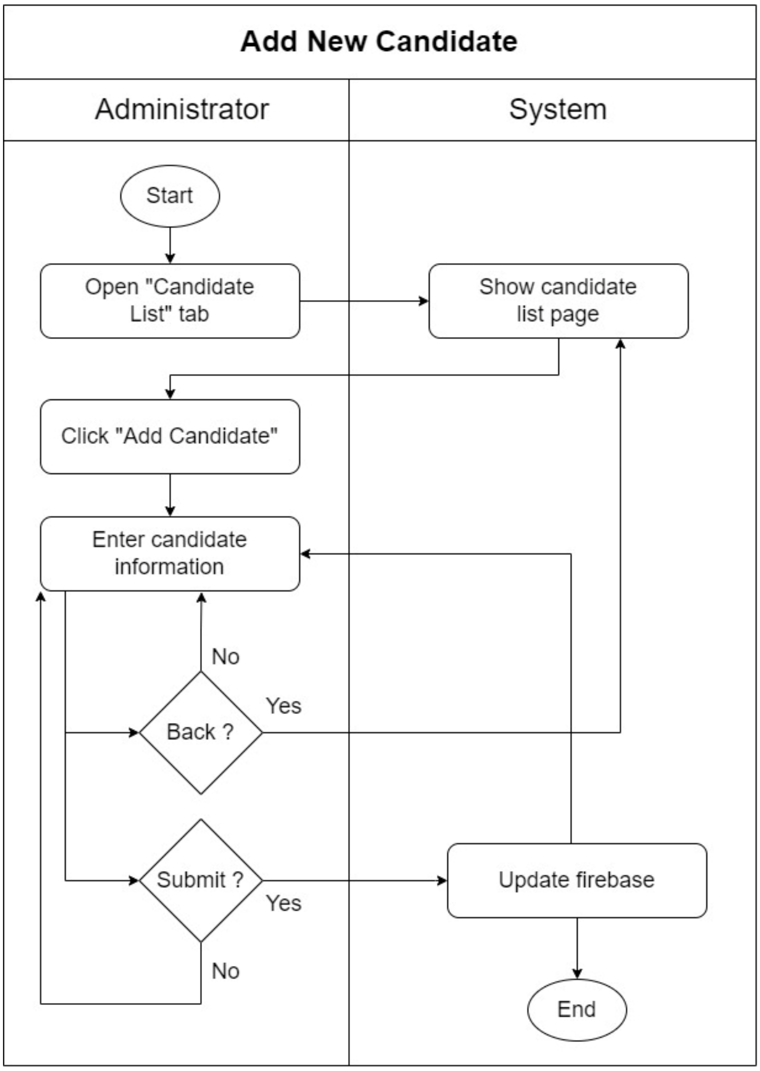


Figure 3.4.5: Add New Candidate Activity Diagram

Within the "Add Candidate" activity diagram, the administrative process commences by navigating to the "Candidate List" tab to initiate the addition of new candidates to the system. Clicking on the "Add Candidate" button empowers the admin to input the candidate's pertinent information. Once the details are confirmed, there is an option to submit the form, resulting in the data being seamlessly updated in the Firebase database.

Furthermore, the admin retains the flexibility to either proceed to the next candidate entry or choose to continue editing if any information requires modification or verification. If the admin wishes to exit the "Add Candidate" page and return to the candidate list, simple click on the "back" button facilitates a smooth transition back to the previous page. The admin has the option to end the session if they no longer need to add new candidates.

Chapter 4

System Design

4.1 System Block Diagram

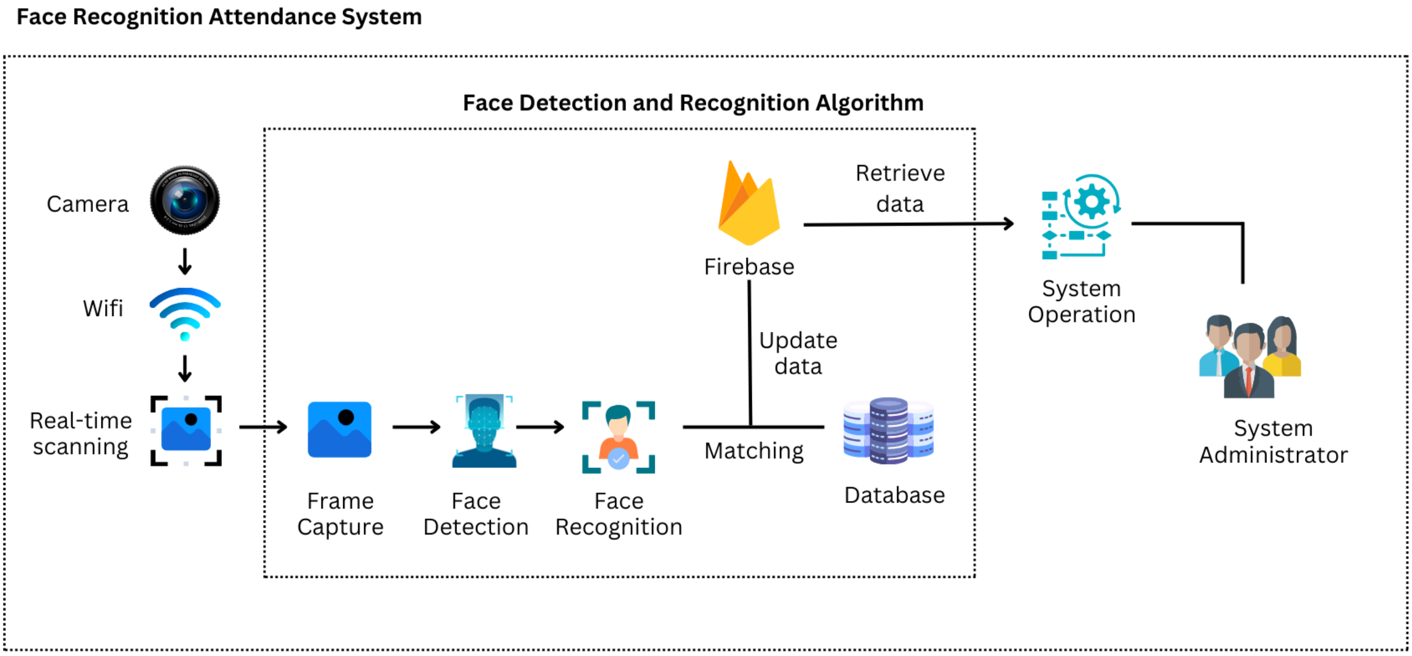


Figure 4.1: System Block Diagram

In the system design block diagram, the process begins with a camera that connects to a Wi-Fi network. Upon system readiness, the administrator gains the ability to initiate attendance tracking by activating the real-time facial scanning feature. During this real-time scanning process, the frame captures the candidate's face, which is then subjected to face detection and recognition algorithms. Initially, the system identifies the candidate's face, and subsequently, it matches the recognized face with the entries in the database. The relevant information, including candidate details and the timestamp of the recording, is then updated in Firebase. Simultaneously, during the recognition process, facial landmark encoding is saved in a pickle file format located within the database. This data stored in Firebase can later be retrieved and applied in the system operations. Subsequently, the system's administrator, who is the user, can utilize this collected information for various purposes within the system.

4.2 Data Preprocessing

The face recognition system relies on the OpenCV and face_recognition libraries as its core components. To enable face recognition, a dataset of candidate photos is gathered, forming the foundation for comparison. Using OpenCV, the webcam stream is continuously captured, and each frame is processed to enhance efficiency. Frames are resized to reduce computational load and converted to RGB color format for consistency. The pivotal face recognition step involves encoding detected faces using the face_recognition library. This encoding serves as a unique fingerprint for each face. By comparing these encodings with reference encodings from the dataset, the system identifies and matches faces. When a match occurs, it signifies a successful face recognition event.

In the context of face recognition in PyCharm, "MMOD CNN" typically refers to a "Maximum-Margin Object Detection Convolutional Neural Network" used to build a face detection and recognition system. It is a type of deep learning model used for object detection tasks, and specifically designed to locate and classify objects within an image. The object detection techniques start by applying a binary classifier to small sections (sub-windows) of an image. This involve scanning an image with a small window and determining whether the content within that window contains an object of interest, such as a face, or not. After applying the binary classifier to many overlapping sub-windows of an image, it ends up with multiple detections, potentially covering the same object. Afterward, a process called non-maximum suppression is used to eliminate redundant detections in overlapping sub-windows, keeping only the most confident ones.

As the problem of the number of possible sub-windows in an image can be extremely large, especially in high-resolution images or datasets with many objects to detect. To make the detection process computationally feasible, classifiers are typically trained and applied only on a subset of these sub-windows. To reduce computational complexity, this subset is chosen strategically by using sampling techniques. For the issue that the practice of sub-sampling sub-windows for training and detection can lead to sub-optimal detection performance. By training the classifier on only a subset of sub-windows, it may overlook important information or object variations present in non-sampled sub-windows. Hence, this may result in less accurate object detection.

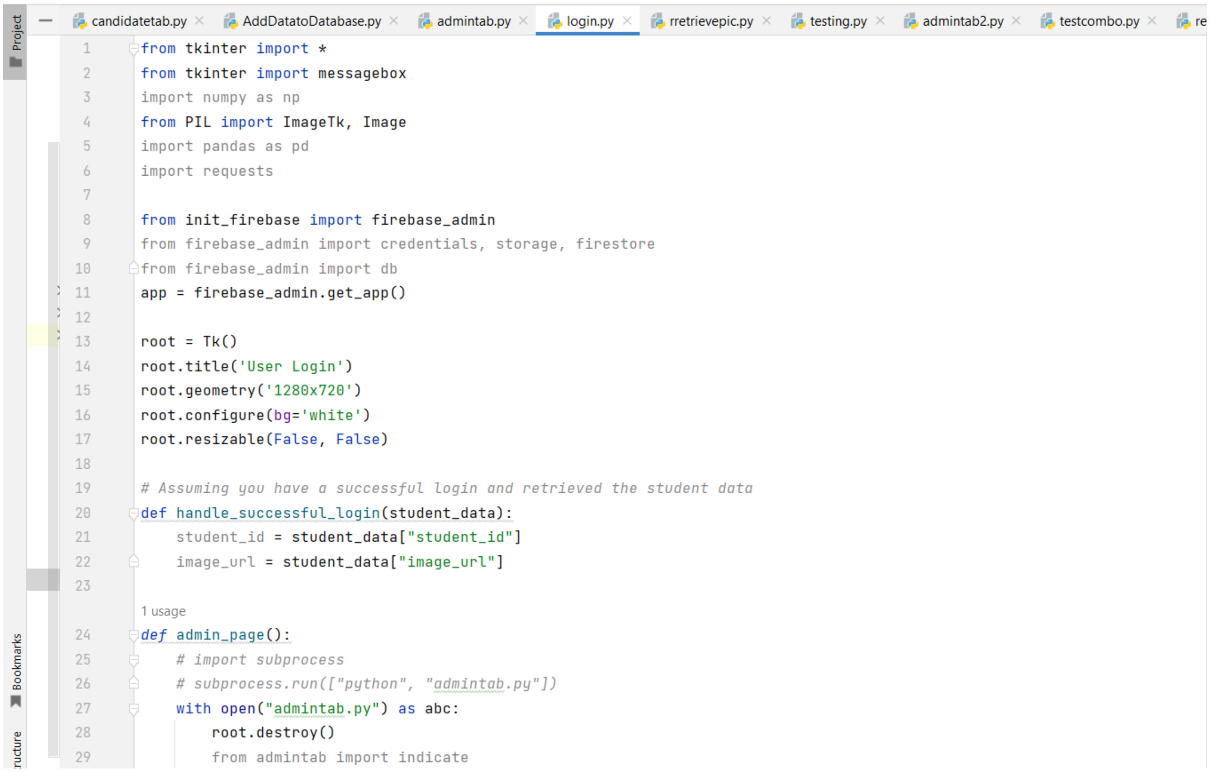
In the process of locating and identifying facial landmark, a specific pre-trained model, shape predictor is being used for image processing tasks, which is ‘shape_predictor_5_face_landmarks’ and ‘shape_predictor_68_face_landmarks’. For ‘shape_predictor_5_face_landmarks’, this model is designed to locate and identify five key facial landmarks on a human face. These landmarks are distinct locations or characteristics on the face, and their placements can be critical for a variety of applications such as face alignment, emotion analysis, and facial expression detection. The five facial landmarks detected by the model includes: ‘Left Eye Corner’, ‘Right Eye Corner’, ‘Nose Tip’, ‘Left Mouth Corner’, and ‘Right Mouth Corner’. These facial landmarks are expressed as (x, y) coordinates within the image, enabling precise localization and examination of distinct facial features. The utility of facial landmark detection spans from enhancing the precision of facial recognition systems to facilitating a variety of facial analysis tasks, including the recognition of emotions, estimating head poses, and more. For ‘shape_predictor_68_face_landmarks’, this model is designed to locate and identify 68 different facial landmarks on a human face. Facial landmark detection encompasses identifying specific points in various facial regions. These regions include landmarks along the eyebrows to capture their shape and position, points around the eyes (including the corners and eyelids), landmarks along the nose (covering the tip and sides), points outlining the lips (including the corners and various points along the lip contour), and landmarks along the jawline and chin.

4.3 System Flow Description

The face recognition attendance system is a cutting-edge solution for automating attendance tracking. It starts by capturing a person's face through a camera and checks it against enrolled individuals. If a match is found, attendance is recorded with a timestamp.

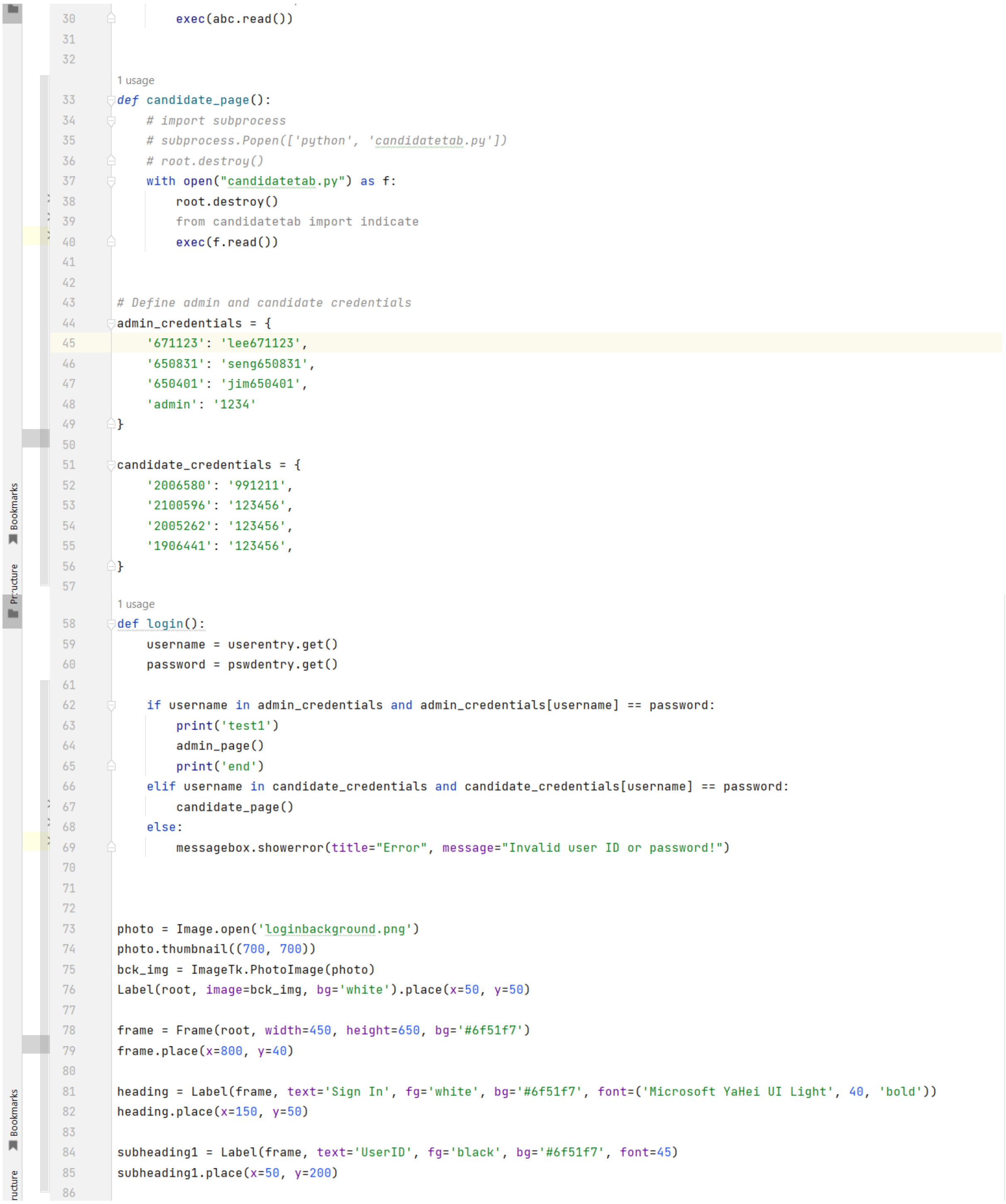
4.3.1 Login.py

The Python file Login.py is responsible for handling the login functionality. It verifies user credentials and, upon successful verification, directs the admin to the main admin page.



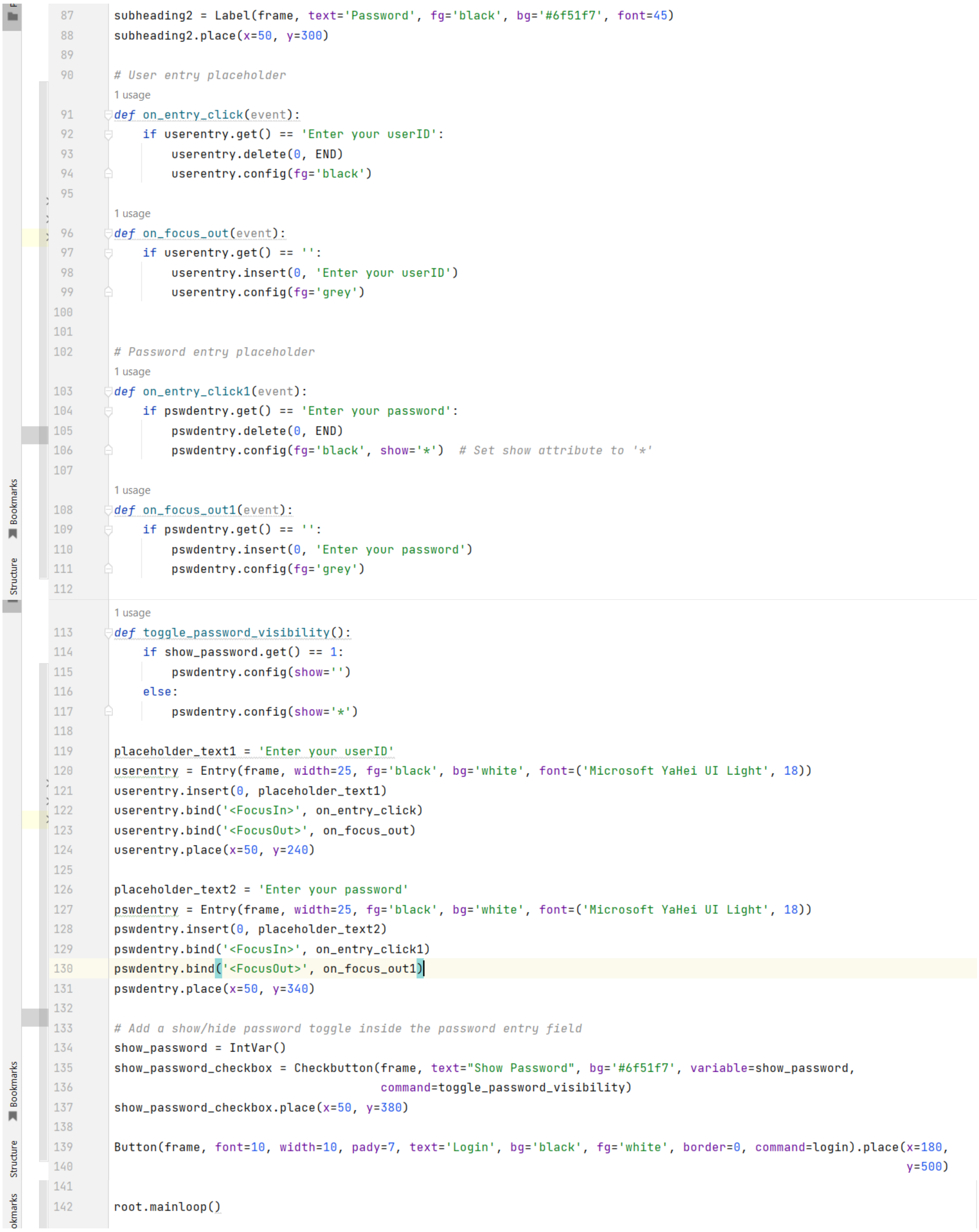
```
from tkinter import *
from tkinter import messagebox
import numpy as np
from PIL import ImageTk, Image
import pandas as pd
import requests

from init_firebase import firebase_admin
from firebase_admin import credentials, storage, firestore
from firebase_admin import db
app = firebase_admin.get_app()

root = Tk()
root.title('User Login')
root.geometry('1280x720')
root.configure(bg='white')
root.resizable(False, False)

# Assuming you have a successful login and retrieved the student data
def handle_successful_login(student_data):
    student_id = student_data["student_id"]
    image_url = student_data["image_url"]

1 usage
def admin_page():
    # import subprocess
    # subprocess.run(["python", "admintab.py"])
    with open("admintab.py") as abc:
        root.destroy()
        from admintab import indicate
```

Figure 4.3.1.1: Login.py Diagram (1/3)



```
        exec(abc.read())


1 usage
def candidate_page():
    # import subprocess
    # subprocess.Popen(['python', 'candidatetab.py'])
    # root.destroy()
    with open("candidatetab.py") as f:
        root.destroy()
        from candidatetab import indicate
        exec(f.read())


# Define admin and candidate credentials
admin_credentials = {
    '671123': 'lee671123',
    '650831': 'seng650831',
    '650401': 'jim650401',
    'admin': '1234'
}

candidate_credentials = {
    '2006580': '991211',
    '2100596': '123456',
    '2005262': '123456',
    '1906441': '123456',
}

1 usage
def login():
    username = userentry.get()
    password = pswdentry.get()

    if username in admin_credentials and admin_credentials[username] == password:
        print('test1')
        admin_page()
        print('end')
    elif username in candidate_credentials and candidate_credentials[username] == password:
        candidate_page()
    else:
        messagebox.showerror(title="Error", message="Invalid user ID or password!")



photo = Image.open('loginbackground.png')
photo.thumbnail((700, 700))
bck_img = ImageTk.PhotoImage(photo)
Label(root, image=bck_img, bg='white').place(x=50, y=50)

frame = Frame(root, width=450, height=650, bg='#6f51f7')
frame.place(x=800, y=40)

heading = Label(frame, text='Sign In', fg='white', bg='#6f51f7', font=('Microsoft YaHei UI Light', 40, 'bold'))
heading.place(x=150, y=50)

subheading1 = Label(frame, text='UserID', fg='black', bg='#6f51f7', font=45)
subheading1.place(x=50, y=200)

```

Figure 4.3.1.2: Login.py Diagram (2/3)

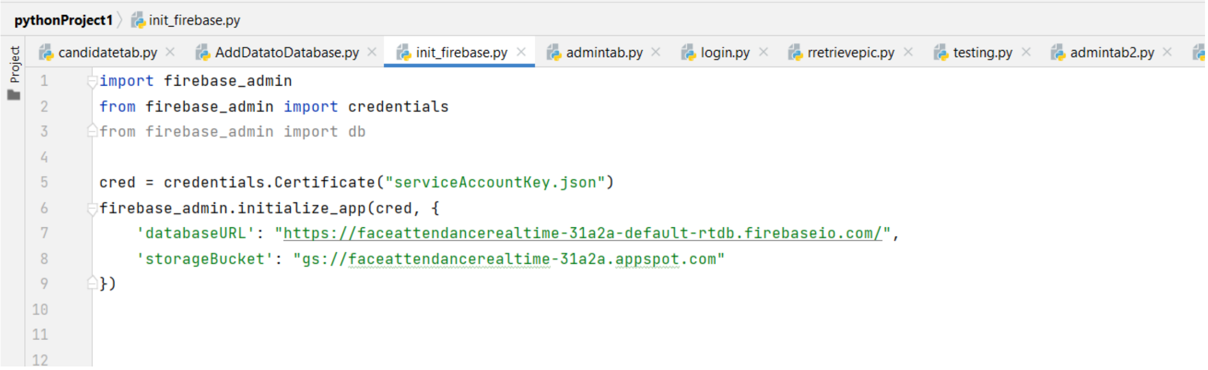


```
subheading2 = Label(frame, text='Password', fg='black', bg='#6f51f7', font=45)
subheading2.place(x=50, y=300)

# User entry placeholder
1 usage
def on_entry_click(event):
    if userentry.get() == 'Enter your userID':
        userentry.delete(0, END)
        userentry.config(fg='black')

1 usage
def on_focus_out(event):
    if userentry.get() == '':
        userentry.insert(0, 'Enter your userID')
        userentry.config(fg='grey')


# Password entry placeholder
1 usage
def on_entry_click1(event):
    if pswdentry.get() == 'Enter your password':
        pswdentry.delete(0, END)
        pswdentry.config(fg='black', show='*')  # Set show attribute to '*'

1 usage
def on_focus_out1(event):
    if pswdentry.get() == '':
        pswdentry.insert(0, 'Enter your password')
        pswdentry.config(fg='grey')

1 usage
def toggle_password_visibility():
    if show_password.get() == 1:
        pswdentry.config(show='')
    else:
        pswdentry.config(show='*')

placeholder_text1 = 'Enter your userID'
userentry = Entry(frame, width=25, fg='black', bg='white', font=('Microsoft YaHei UI Light', 18))
userentry.insert(0, placeholder_text1)
userentry.bind('<FocusIn>', on_entry_click)
userentry.bind('<FocusOut>', on_focus_out)
userentry.place(x=50, y=240)

placeholder_text2 = 'Enter your password'
pswdentry = Entry(frame, width=25, fg='black', bg='white', font=('Microsoft YaHei UI Light', 18))
pswdentry.insert(0, placeholder_text2)
pswdentry.bind('<FocusIn>', on_entry_click1)
pswdentry.bind('<FocusOut>', on_focus_out1)
pswdentry.place(x=50, y=340)

# Add a show/hide password toggle inside the password entry field
show_password = IntVar()
show_password_checkbox = Checkbutton(frame, text="Show Password", bg='#6f51f7', variable=show_password,
                                     command=toggle_password_visibility)
show_password_checkbox.place(x=50, y=380)

Button(frame, font=10, width=10, pady=7, text='Login', bg='black', fg='white', border=0, command=login).place(x=180,
                                                                                                              y=500)

root.mainloop()
```

Figure 4.3.1.3: Login.py Diagram (3/3)

4.3.2 Init_firebase.py

The Python file init_firebase.py is responsible for initializing Google Firebase. It is designed to be called or imported into other Python files as needed to establish the Firebase connection and configuration.

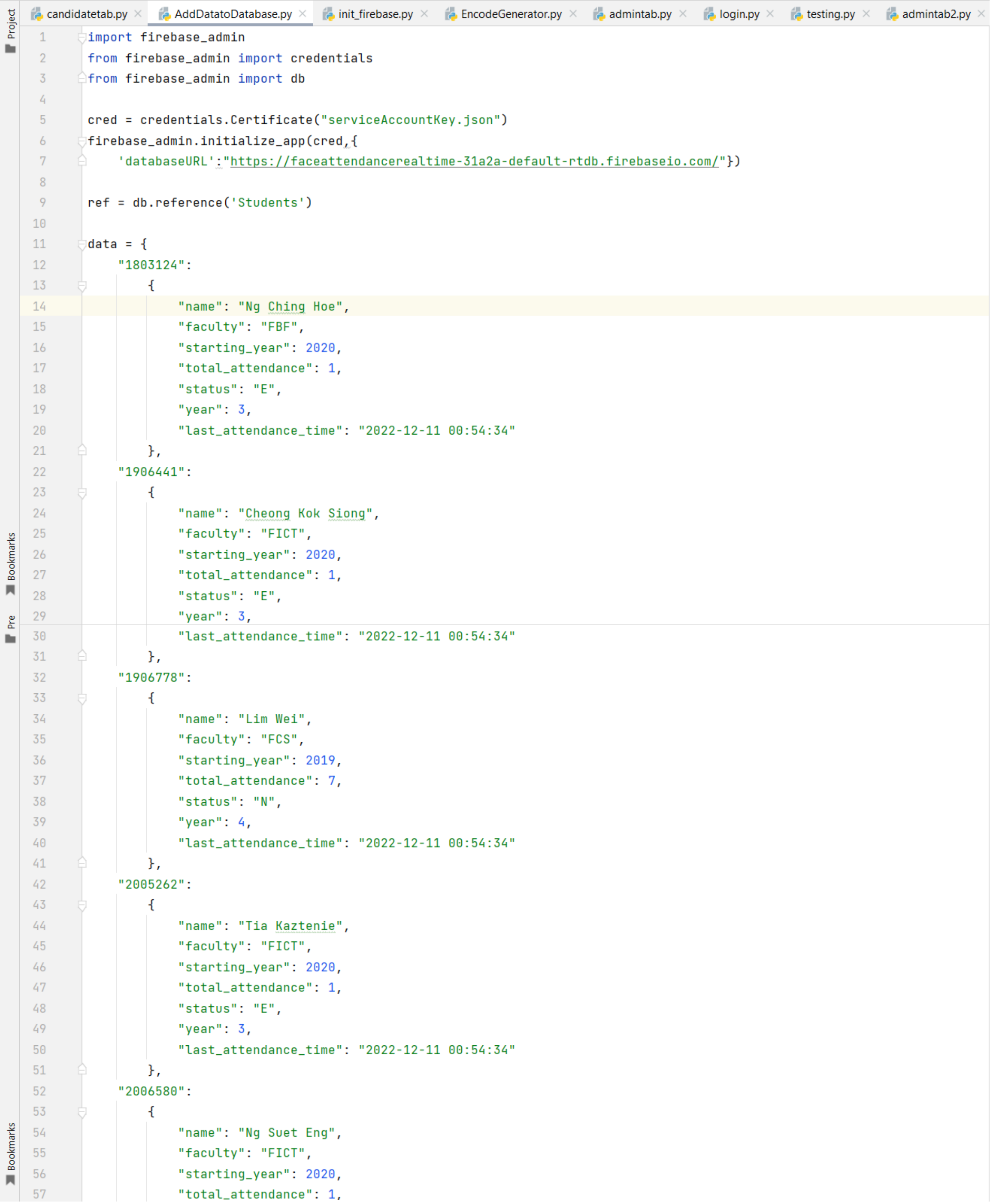


```python
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://faceattendancerealtime-31a2a-default-rtdb.firebaseio.com/",
    'storageBucket': "gs://faceattendancerealtime-31a2a.appspot.com"
})
```

Figure 4.3.2.1: Init_firebase.py Diagram (1/1)

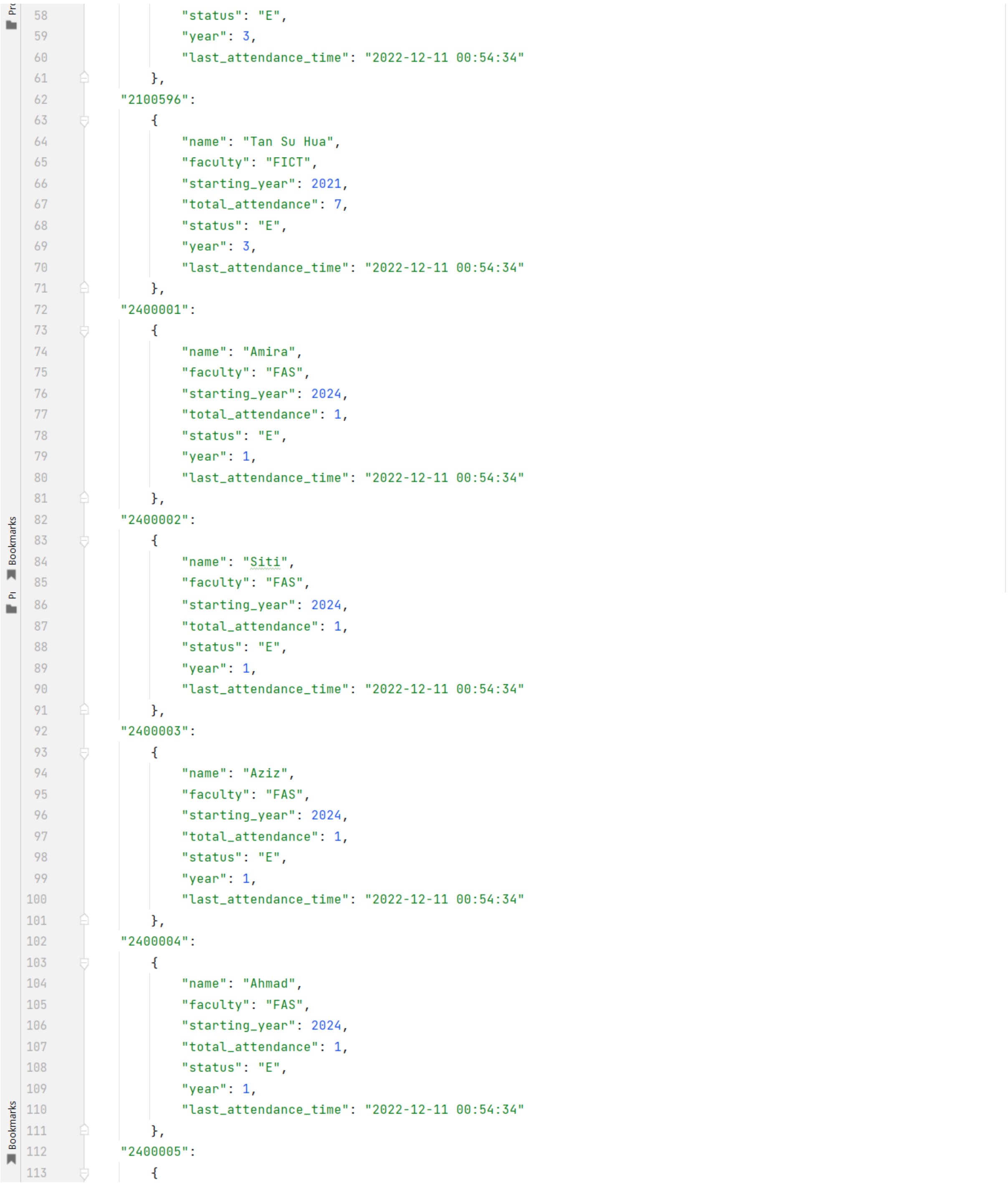
4.3.3 AddDatatoDatabase.py

The Python file AddDataToDatabase.py is used to add candidate information to Firebase. As a demonstration, it includes the addition of 15 sample candidate records to Firebase.



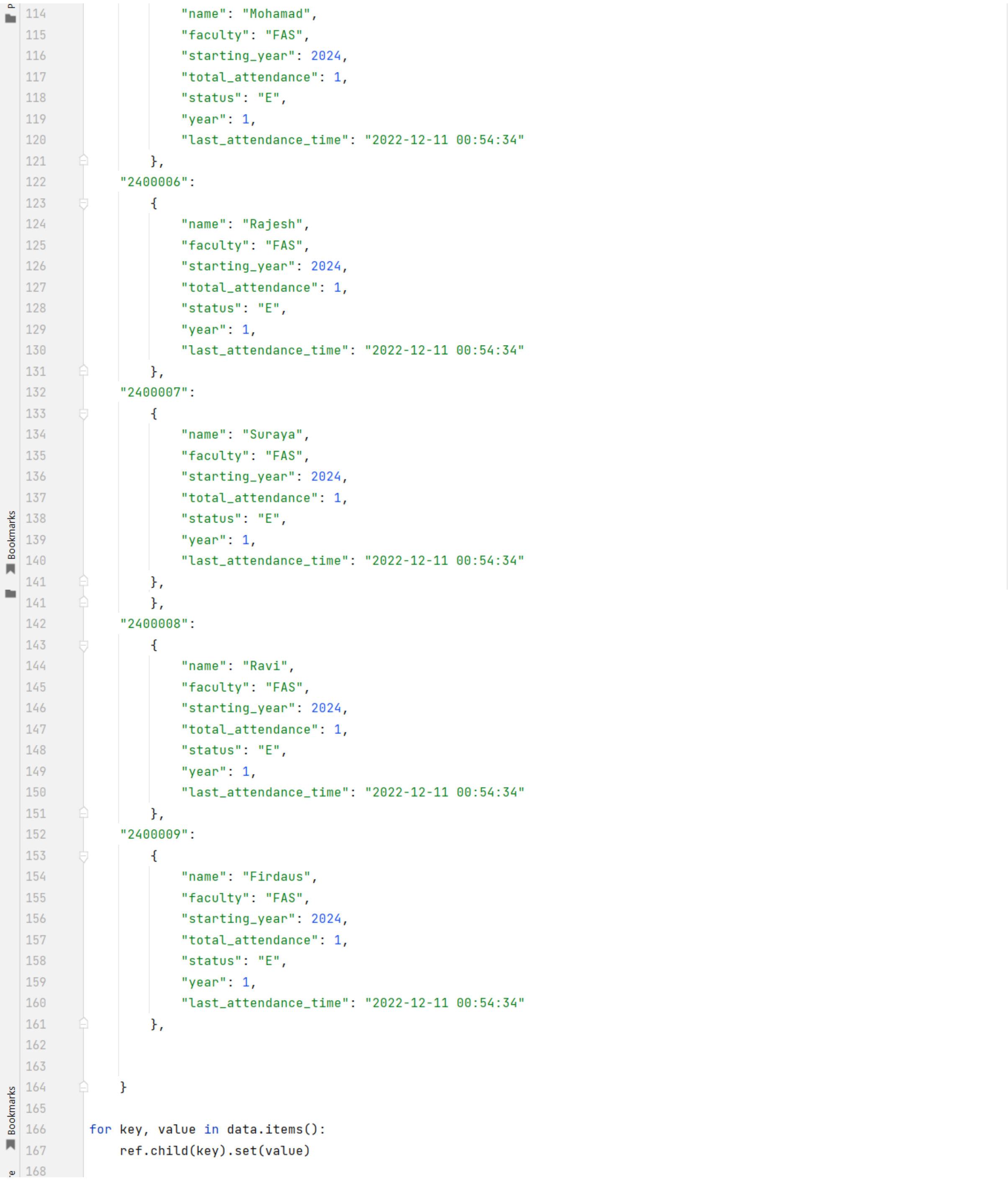
```python
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred,{
    'databaseURL':"https://faceattendancerealtime-31a2a-default-rtdb.firebaseio.com/"})

ref = db.reference('Students')

data = {
    "1803124":
        {
            "name": "Ng Ching Hoe",
            "faculty": "FBF",
            "starting_year": 2020,
            "total_attendance": 1,
            "status": "E",
            "year": 3,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "1906441":
        {
            "name": "Cheong Kok Siong",
            "faculty": "FICT",
            "starting_year": 2020,
            "total_attendance": 1,
            "status": "E",
            "year": 3,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "1906778":
        {
            "name": "Lim Wei",
            "faculty": "FCS",
            "starting_year": 2019,
            "total_attendance": 7,
            "status": "N",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2005262":
        {
            "name": "Tia Kaztenie",
            "faculty": "FICT",
            "starting_year": 2020,
            "total_attendance": 1,
            "status": "E",
            "year": 3,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2006580":
        {
            "name": "Ng Suet Eng",
            "faculty": "FICT",
            "starting_year": 2020,
            "total_attendance": 1,
```

Figure 4.3.3.1: AddDatatoDatabase.py Diagram (1/3)



```
            "status": "E",
            "year": 3,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2100596":
        {
            "name": "Tan Su Hua",
            "faculty": "FICT",
            "starting_year": 2021,
            "total_attendance": 7,
            "status": "E",
            "year": 3,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400001":
        {
            "name": "Amira",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400002":
        {
            "name": "Siti",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400003":
        {
            "name": "Aziz",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400004":
        {
            "name": "Ahmad",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400005":
        {
```

Figure 4.3.3.2: AddDatatoDatabase.py Diagram (2/3)

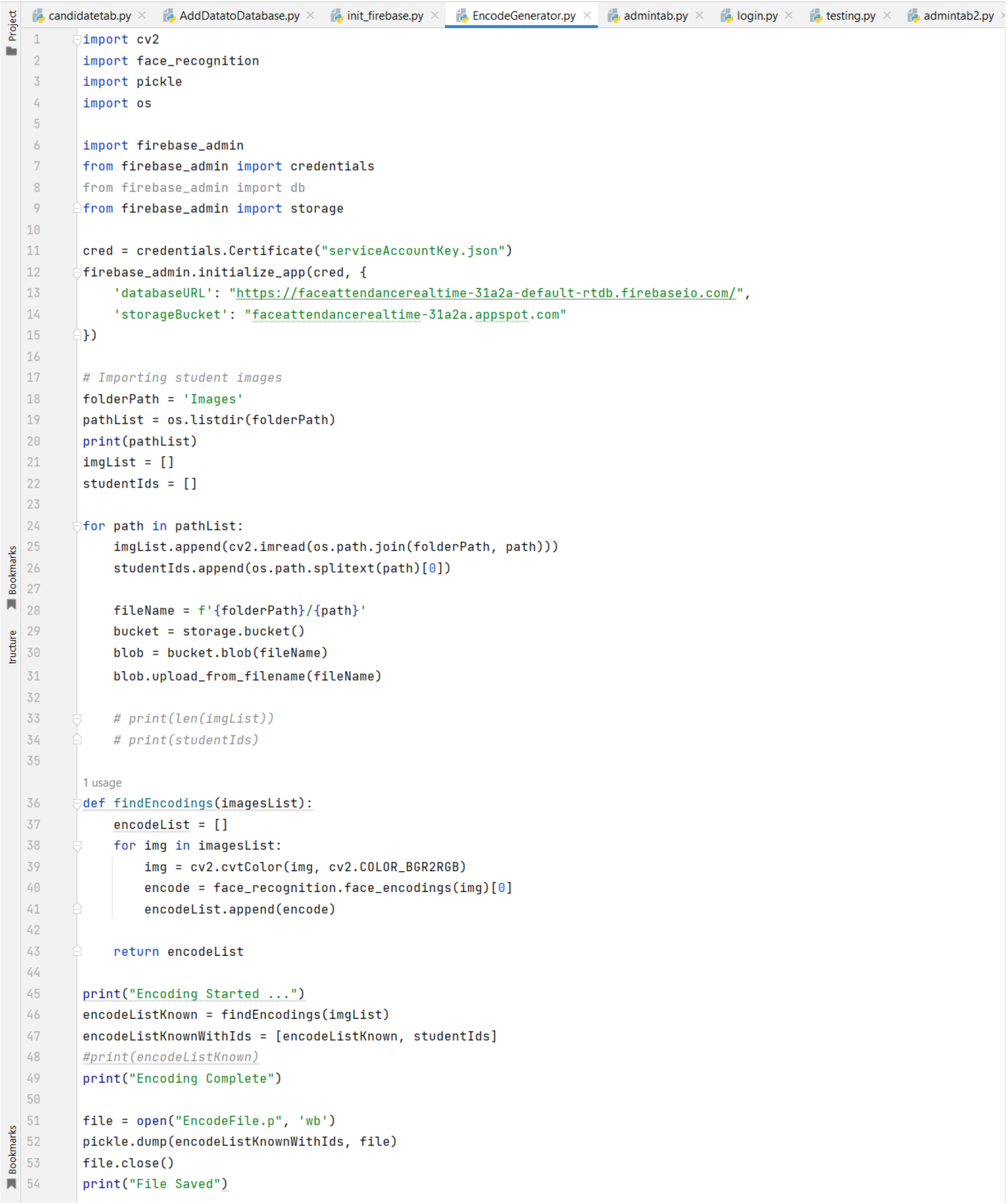


```
            "name": "Mohamad",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400006":
        {
            "name": "Rajesh",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400007":
        {
            "name": "Suraya",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
        },
    "2400008":
        {
            "name": "Ravi",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "2400009":
        {
            "name": "Firdaus",
            "faculty": "FAS",
            "starting_year": 2024,
            "total_attendance": 1,
            "status": "E",
            "year": 1,
            "last_attendance_time": "2022-12-11 00:54:34"
        },


    }

for key, value in data.items():
    ref.child(key).set(value)
```

Figure 4.3.3.3: AddDatatoDatabase.py Diagram (3/3)

4.3.4 EncodeGenerator.py

To encode the facial features of a candidate's profile image for the recognition process, an encoding algorithm is employed. This algorithm extracts and represents the unique facial features of the candidate, facilitating subsequent recognition tasks.

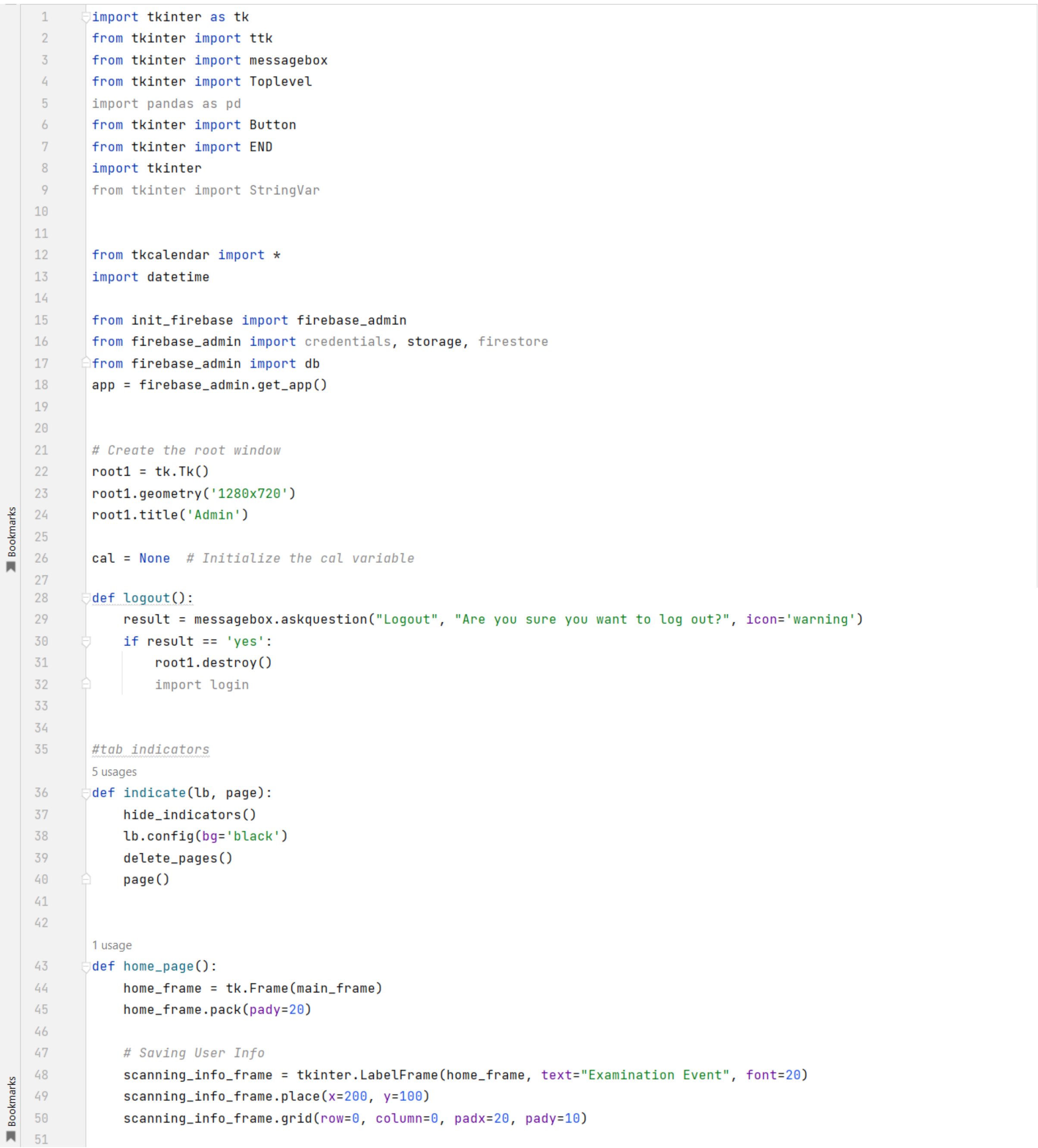


```python
import cv2
import face_recognition
import pickle
import os

import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://faceattendancerealtime-31a2a-default-rtdb.firebaseio.com/",
    'storageBucket': "faceattendancerealtime-31a2a.appspot.com"
})

# Importing student images
folderPath = 'Images'
pathList = os.listdir(folderPath)
print(pathList)
imgList = []
studentIds = []

for path in pathList:
    imgList.append(cv2.imread(os.path.join(folderPath, path)))
    studentIds.append(os.path.splitext(path)[0])

    fileName = f'{folderPath}/{path}'
    bucket = storage.bucket()
    blob = bucket.blob(fileName)
    blob.upload_from_filename(fileName)

    # print(len(imgList))
    # print(studentIds)

def findEncodings(imagesList):
    encodeList = []
    for img in imagesList:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)

    return encodeList

print("Encoding Started ...")
encodeListKnown = findEncodings(imgList)
encodeListKnownWithIds = [encodeListKnown, studentIds]
#print(encodeListKnown)
print("Encoding Complete")

file = open("EncodeFile.p", 'wb')
pickle.dump(encodeListKnownWithIds, file)
file.close()
print("File Saved")
```

Figure 4.3.4.1: EncodeGenerator.py Diagram (1/1)

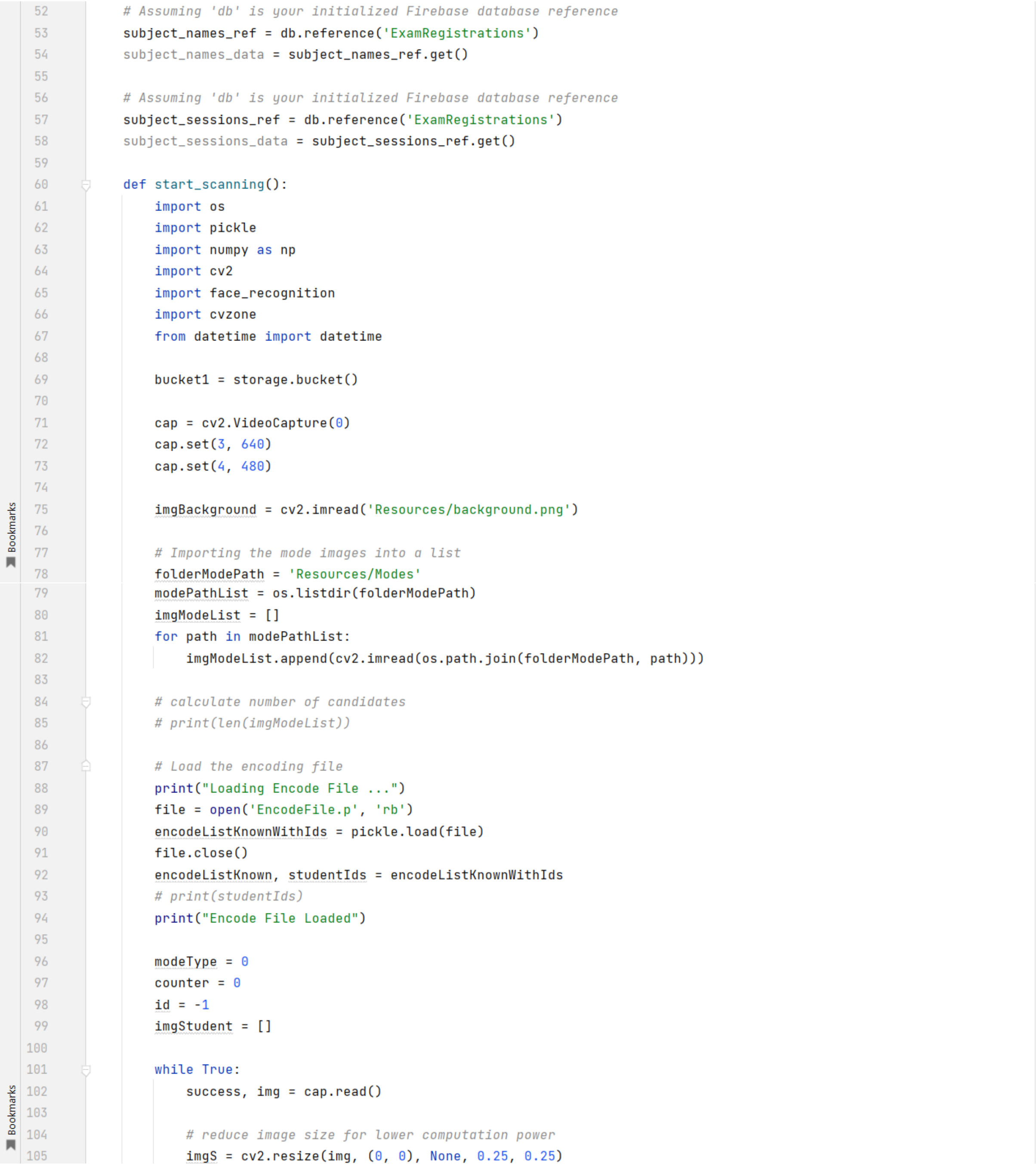
4.3.5 AdminTab.py

This AdminTab.py include some primary function of administrator. The primary responsibilities of an administrator encompass several crucial functions, which include candidate verification, maintaining the candidate list, facilitating examination registration, and overseeing report generation.



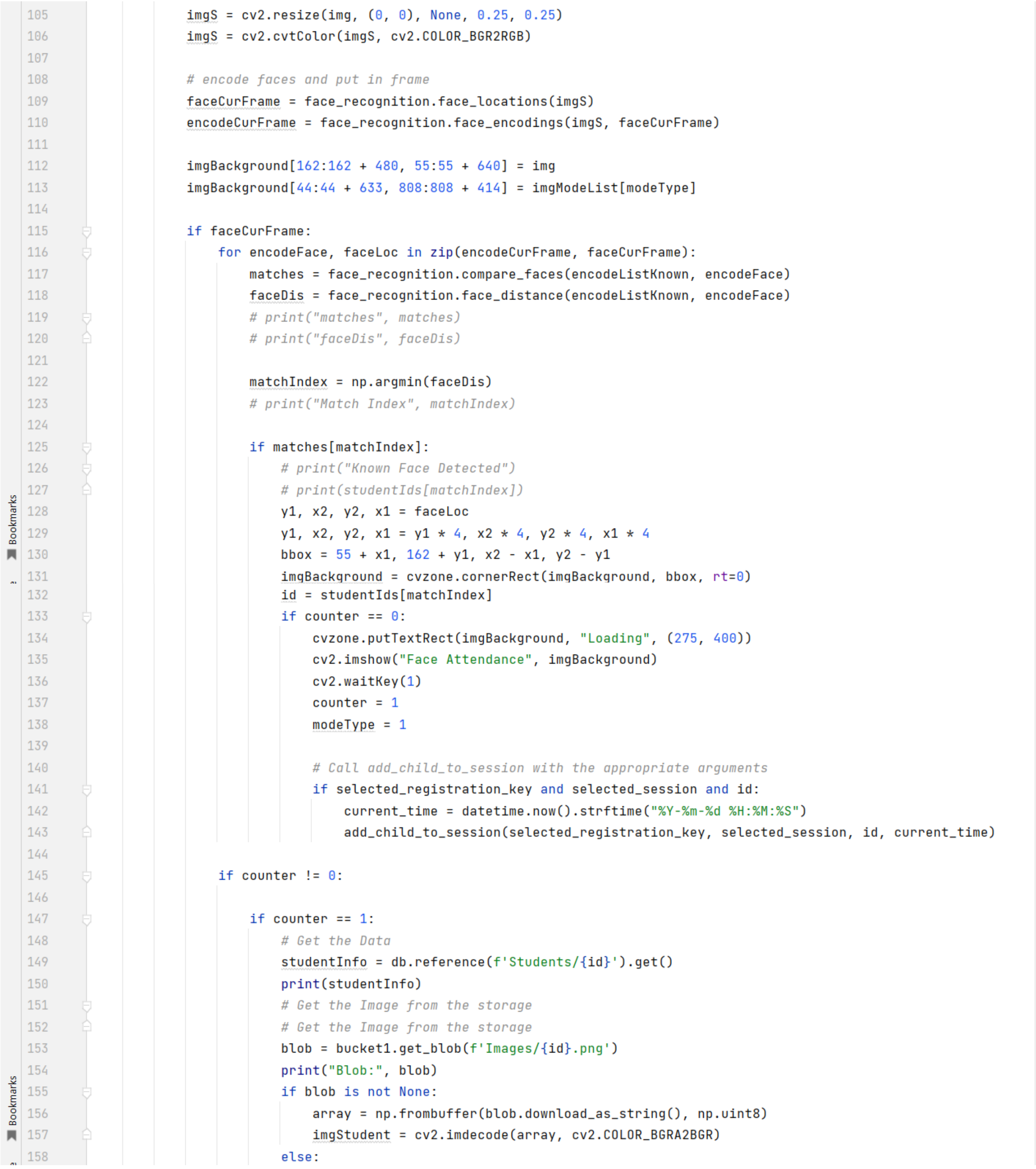
```python
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from tkinter import Toplevel
import pandas as pd
from tkinter import Button
from tkinter import END
import tkinter
from tkinter import StringVar


from tkcalendar import *
import datetime

from init_firebase import firebase_admin
from firebase_admin import credentials, storage, firestore
from firebase_admin import db
app = firebase_admin.get_app()


# Create the root window
root1 = tk.Tk()
root1.geometry('1280x720')
root1.title('Admin')

cal = None  # Initialize the cal variable

def logout():
    result = messagebox.askquestion("Logout", "Are you sure you want to log out?", icon='warning')
    if result == 'yes':
        root1.destroy()
        import login


#tab indicators
5 usages
def indicate(lb, page):
    hide_indicators()
    lb.config(bg='black')
    delete_pages()
    page()


1 usage
def home_page():
    home_frame = tk.Frame(main_frame)
    home_frame.pack(pady=20)

    # Saving User Info
    scanning_info_frame = tkinter.LabelFrame(home_frame, text="Examination Event", font=20)
    scanning_info_frame.place(x=200, y=100)
    scanning_info_frame.grid(row=0, column=0, padx=20, pady=10)

```

Figure 4.3.5.1: AdminTab.py Diagram (1/15)



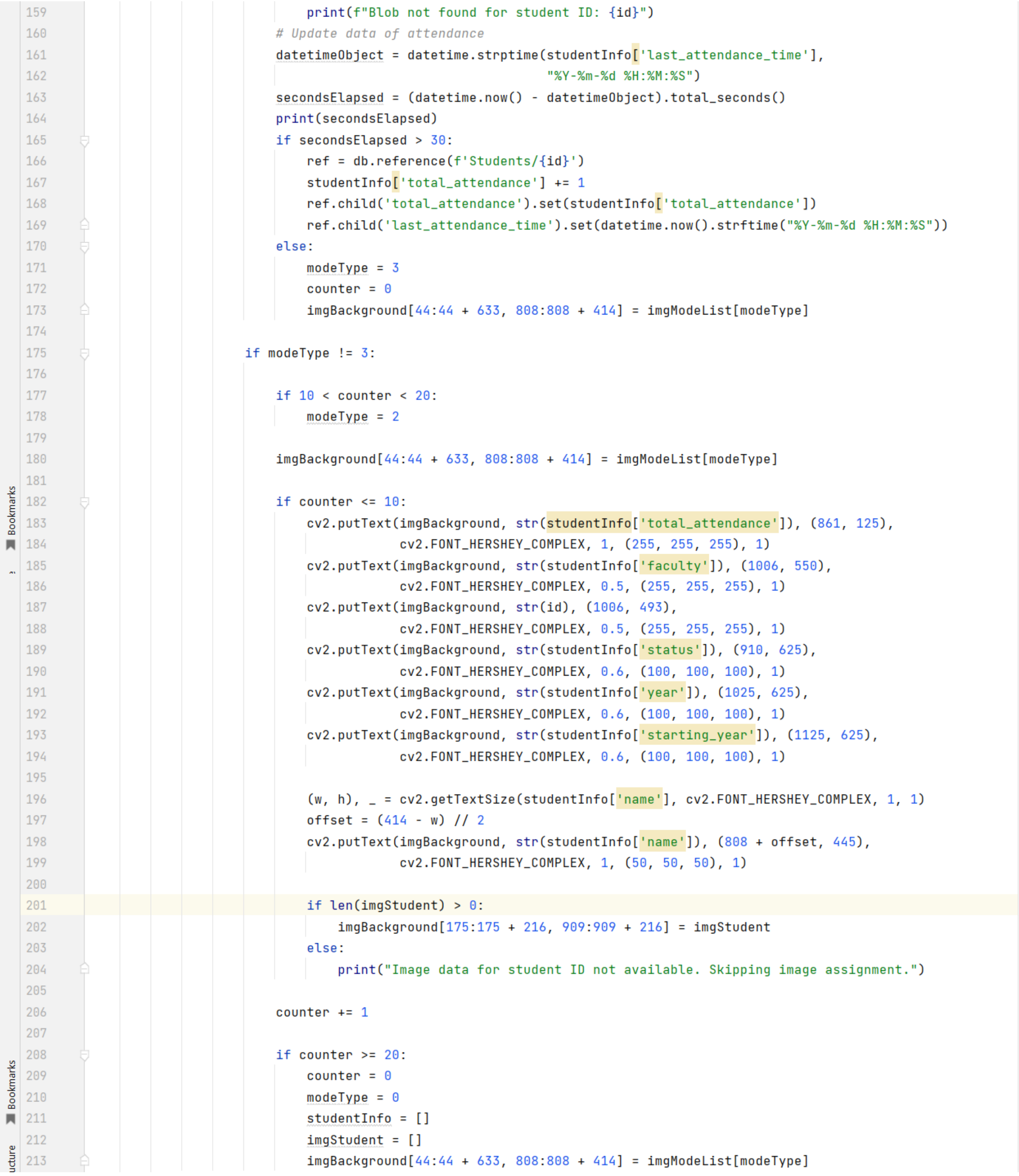
```
# Assuming 'db' is your initialized Firebase database reference
subject_names_ref = db.reference('ExamRegistrations')
subject_names_data = subject_names_ref.get()

# Assuming 'db' is your initialized Firebase database reference
subject_sessions_ref = db.reference('ExamRegistrations')
subject_sessions_data = subject_sessions_ref.get()

def start_scanning():
    import os
    import pickle
    import numpy as np
    import cv2
    import face_recognition
    import cvzone
    from datetime import datetime

    bucket1 = storage.bucket()

    cap = cv2.VideoCapture(0)
    cap.set(3, 640)
    cap.set(4, 480)

    imgBackground = cv2.imread('Resources/background.png')

    # Importing the mode images into a list
    folderModePath = 'Resources/Modes'
    modePathList = os.listdir(folderModePath)
    imgModeList = []
    for path in modePathList:
        imgModeList.append(cv2.imread(os.path.join(folderModePath, path)))

    # calculate number of candidates
    # print(len(imgModeList))

    # Load the encoding file
    print("Loading Encode File ...")
    file = open('EncodeFile.p', 'rb')
    encodeListKnownWithIds = pickle.load(file)
    file.close()
    encodeListKnown, studentIds = encodeListKnownWithIds
    # print(studentIds)
    print("Encode File Loaded")

    modeType = 0
    counter = 0
    id = -1
    imgStudent = []

    while True:
        success, img = cap.read()

        # reduce image size for lower computation power
        imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
```

Figure 4.3.5.2: AdminTab.py Diagram (2/15)



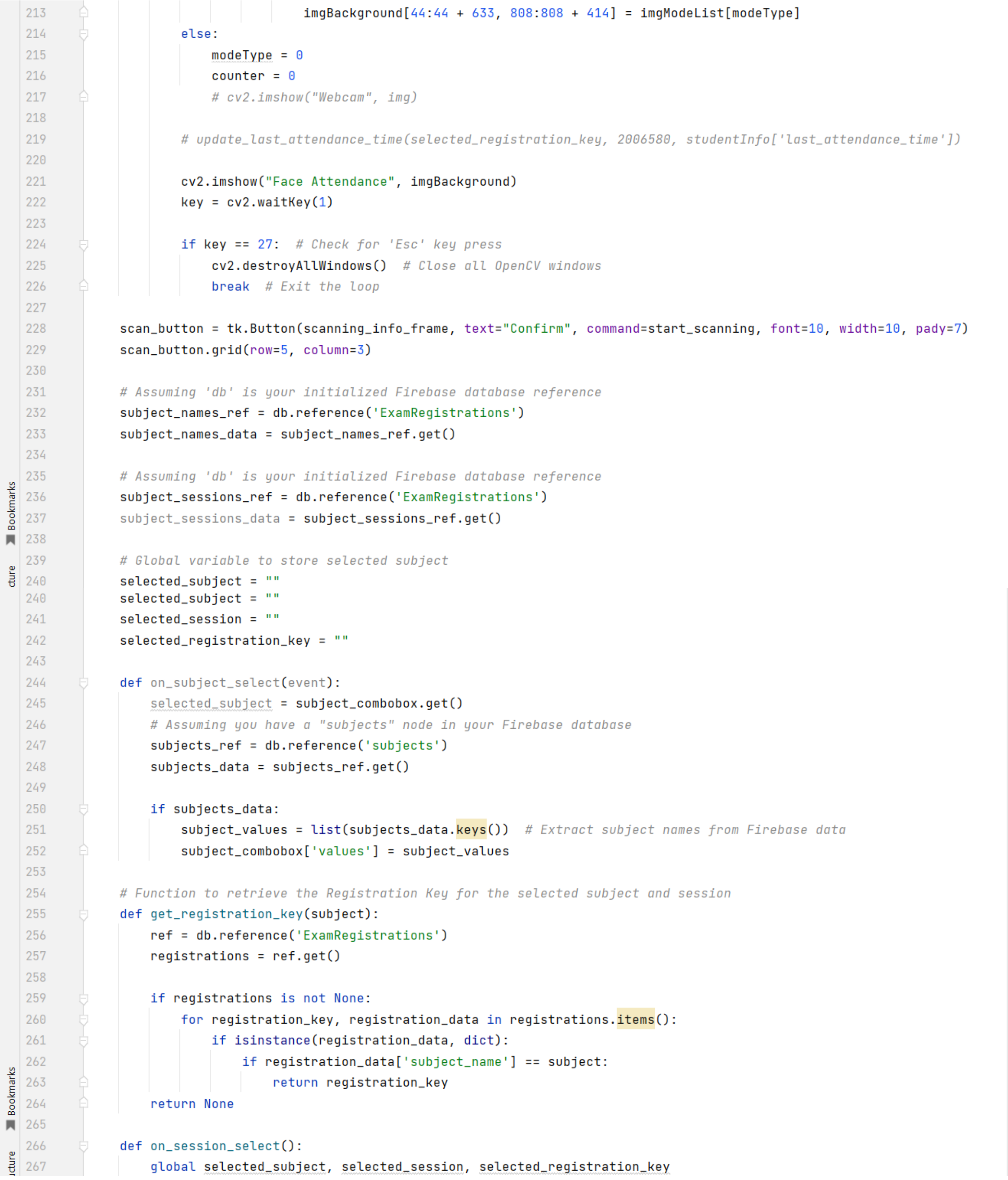
```
        imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
        imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

        # encode faces and put in frame
        faceCurFrame = face_recognition.face_locations(imgS)
        encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)

        imgBackground[162:162 + 480, 55:55 + 640] = img
        imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

        if faceCurFrame:
            for encodeFace, faceLoc in zip(encodeCurFrame, faceCurFrame):
                matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
                faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
                # print("matches", matches)
                # print("faceDis", faceDis)

                matchIndex = np.argmin(faceDis)
                # print("Match Index", matchIndex)

                if matches[matchIndex]:
                    # print("Known Face Detected")
                    # print(studentIds[matchIndex])
                    y1, x2, y2, x1 = faceLoc
                    y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
                    bbox = 55 + x1, 162 + y1, x2 - x1, y2 - y1
                    imgBackground = cvzone.cornerRect(imgBackground, bbox, rt=0)
                    id = studentIds[matchIndex]
                    if counter == 0:
                        cvzone.putTextRect(imgBackground, "Loading", (275, 400))
                        cv2.imshow("Face Attendance", imgBackground)
                        cv2.waitKey(1)
                        counter = 1
                        modeType = 1

                        # Call add_child_to_session with the appropriate arguments
                        if selected_registration_key and selected_session and id:
                            current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
                            add_child_to_session(selected_registration_key, selected_session, id, current_time)

            if counter != 0:

                if counter == 1:
                    # Get the Data
                    studentInfo = db.reference(f'Students/{id}').get()
                    print(studentInfo)
                    # Get the Image from the storage
                    # Get the Image from the storage
                    blob = bucket1.get_blob(f'Images/{id}.png')
                    print("Blob:", blob)
                    if blob is not None:
                        array = np.frombuffer(blob.download_as_string(), np.uint8)
                        imgStudent = cv2.imdecode(array, cv2.COLOR_BGRA2BGR)
                    else:
```

Figure 4.3.5.3: AdminTab.py Diagram (3/15)



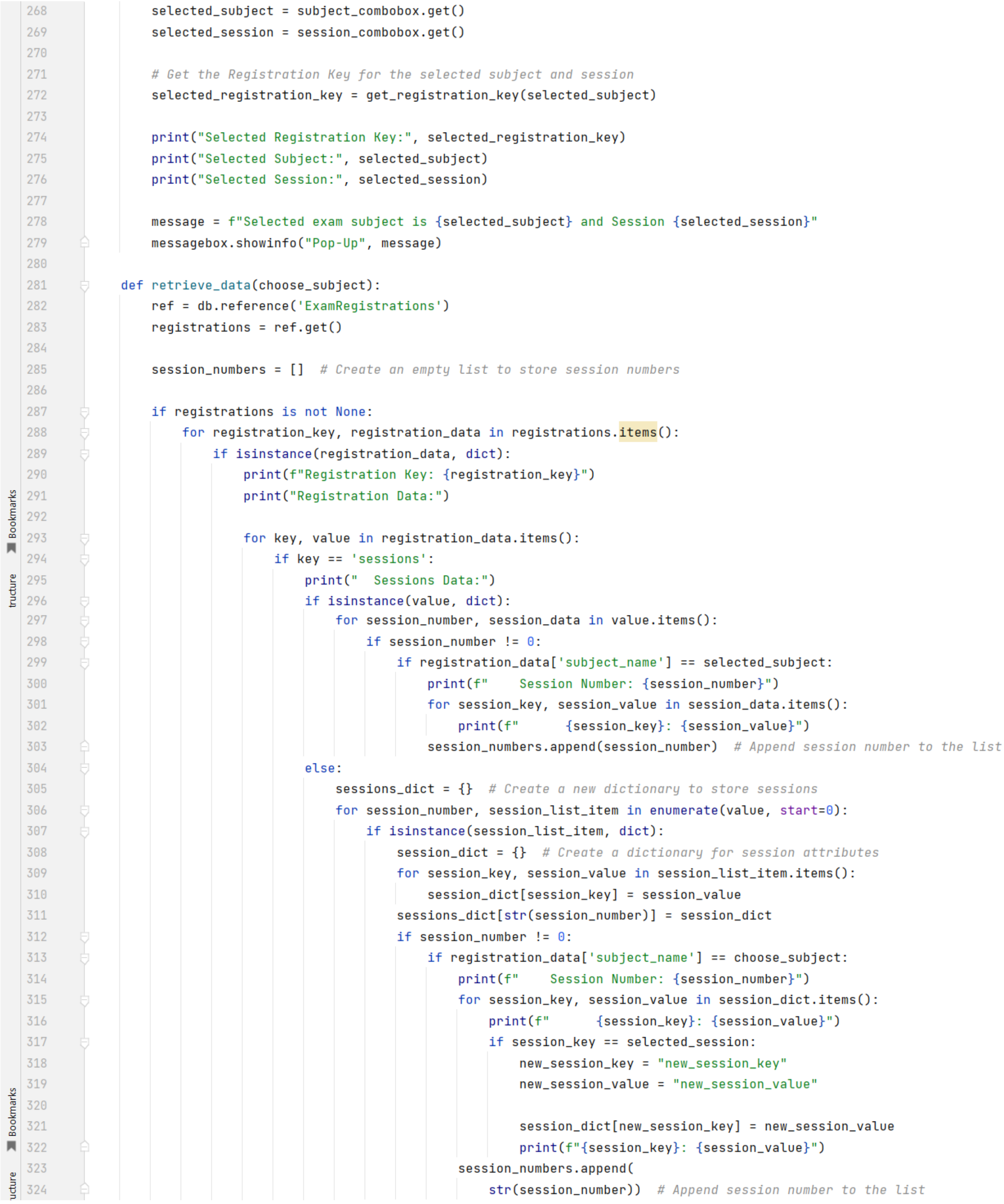
```
        print(f"Blob not found for student ID: {id}")
    # Update data of attendance
    datetimeObject = datetime.strptime(studentInfo['last_attendance_time'],
                                       "%Y-%m-%d %H:%M:%S")
    secondsElapsed = (datetime.now() - datetimeObject).total_seconds()
    print(secondsElapsed)
    if secondsElapsed > 30:
        ref = db.reference(f'Students/{id}')
        studentInfo['total_attendance'] += 1
        ref.child('total_attendance').set(studentInfo['total_attendance'])
        ref.child('last_attendance_time').set(datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
    else:
        modeType = 3
        counter = 0
        imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

if modeType != 3:

    if 10 < counter < 20:
        modeType = 2

    imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

    if counter <= 10:
        cv2.putText(imgBackground, str(studentInfo['total_attendance']), (861, 125),
                    cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 1)
        cv2.putText(imgBackground, str(studentInfo['faculty']), (1006, 550),
                    cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)
        cv2.putText(imgBackground, str(id), (1006, 493),
                    cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)
        cv2.putText(imgBackground, str(studentInfo['status']), (910, 625),
                    cv2.FONT_HERSHEY_COMPLEX, 0.6, (100, 100, 100), 1)
        cv2.putText(imgBackground, str(studentInfo['year']), (1025, 625),
                    cv2.FONT_HERSHEY_COMPLEX, 0.6, (100, 100, 100), 1)
        cv2.putText(imgBackground, str(studentInfo['starting_year']), (1125, 625),
                    cv2.FONT_HERSHEY_COMPLEX, 0.6, (100, 100, 100), 1)

        (w, h), _ = cv2.getTextSize(studentInfo['name'], cv2.FONT_HERSHEY_COMPLEX, 1, 1)
        offset = (414 - w) // 2
        cv2.putText(imgBackground, str(studentInfo['name']), (808 + offset, 445),
                    cv2.FONT_HERSHEY_COMPLEX, 1, (50, 50, 50), 1)

        if len(imgStudent) > 0:
            imgBackground[175:175 + 216, 909:909 + 216] = imgStudent
        else:
            print("Image data for student ID not available. Skipping image assignment.")

    counter += 1

    if counter >= 20:
        counter = 0
        modeType = 0
        studentInfo = []
        imgStudent = []
        imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]
```

Figure 4.3.5.4: AdminTab.py Diagram (4/15)



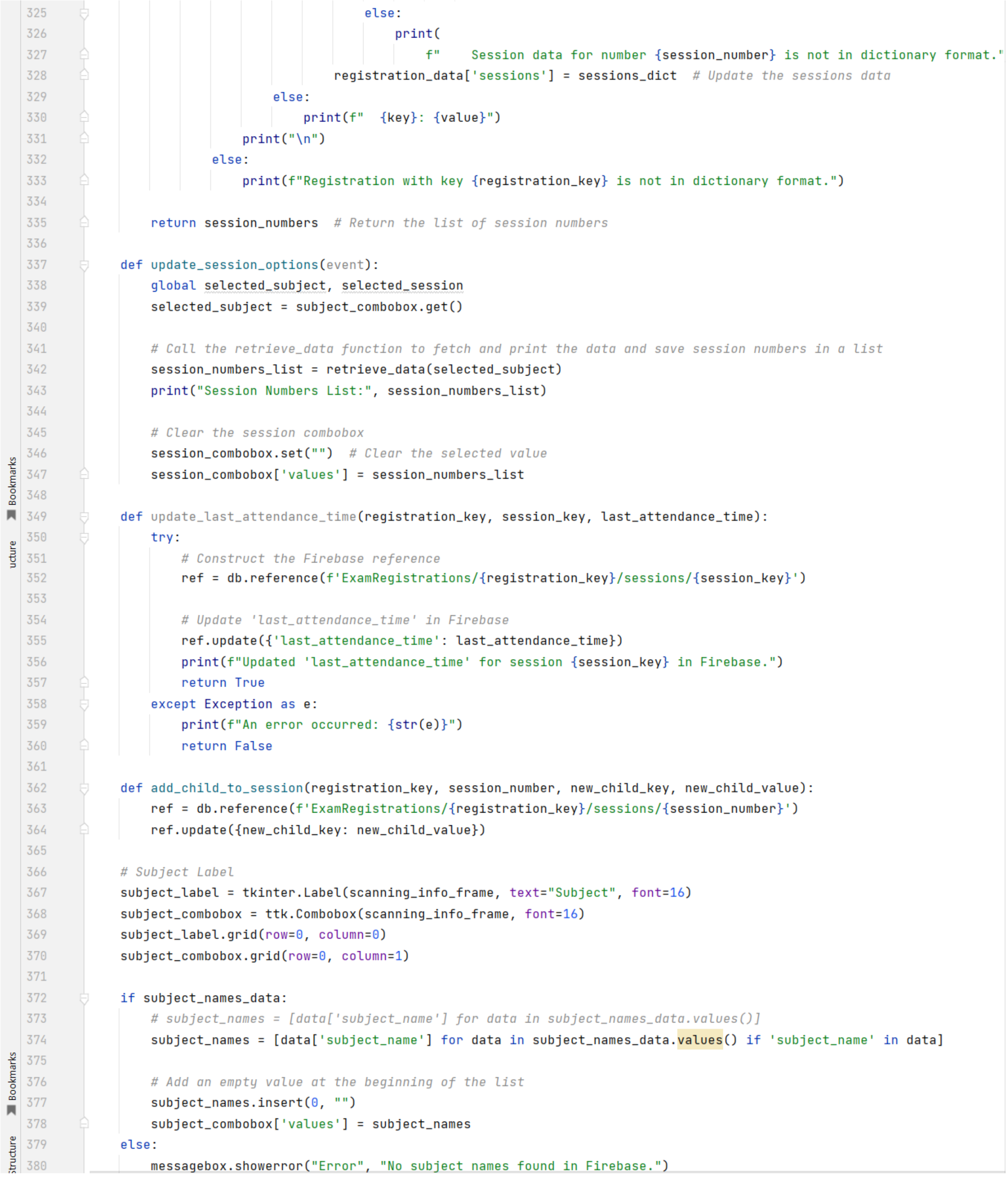
```
                        imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]
        else:
            modeType = 0
            counter = 0
            # cv2.imshow("Webcam", img)

        # update_last_attendance_time(selected_registration_key, 2006580, studentInfo['last_attendance_time'])

        cv2.imshow("Face Attendance", imgBackground)
        key = cv2.waitKey(1)

        if key == 27:  # Check for 'Esc' key press
            cv2.destroyAllWindows()  # Close all OpenCV windows
            break  # Exit the loop

scan_button = tk.Button(scanning_info_frame, text="Confirm", command=start_scanning, font=10, width=10, pady=7)
scan_button.grid(row=5, column=3)

# Assuming 'db' is your initialized Firebase database reference
subject_names_ref = db.reference('ExamRegistrations')
subject_names_data = subject_names_ref.get()

# Assuming 'db' is your initialized Firebase database reference
subject_sessions_ref = db.reference('ExamRegistrations')
subject_sessions_data = subject_sessions_ref.get()

# Global variable to store selected subject
selected_subject = ""
selected_subject = ""
selected_session = ""
selected_registration_key = ""

def on_subject_select(event):
    selected_subject = subject_combobox.get()
    # Assuming you have a "subjects" node in your Firebase database
    subjects_ref = db.reference('subjects')
    subjects_data = subjects_ref.get()

    if subjects_data:
        subject_values = list(subjects_data.keys())  # Extract subject names from Firebase data
        subject_combobox['values'] = subject_values

# Function to retrieve the Registration Key for the selected subject and session
def get_registration_key(subject):
    ref = db.reference('ExamRegistrations')
    registrations = ref.get()

    if registrations is not None:
        for registration_key, registration_data in registrations.items():
            if isinstance(registration_data, dict):
                if registration_data['subject_name'] == subject:
                    return registration_key
    return None

def on_session_select():
    global selected_subject, selected_session, selected_registration_key
```

Figure 4.3.5.5: AdminTab.py Diagram (5/15)



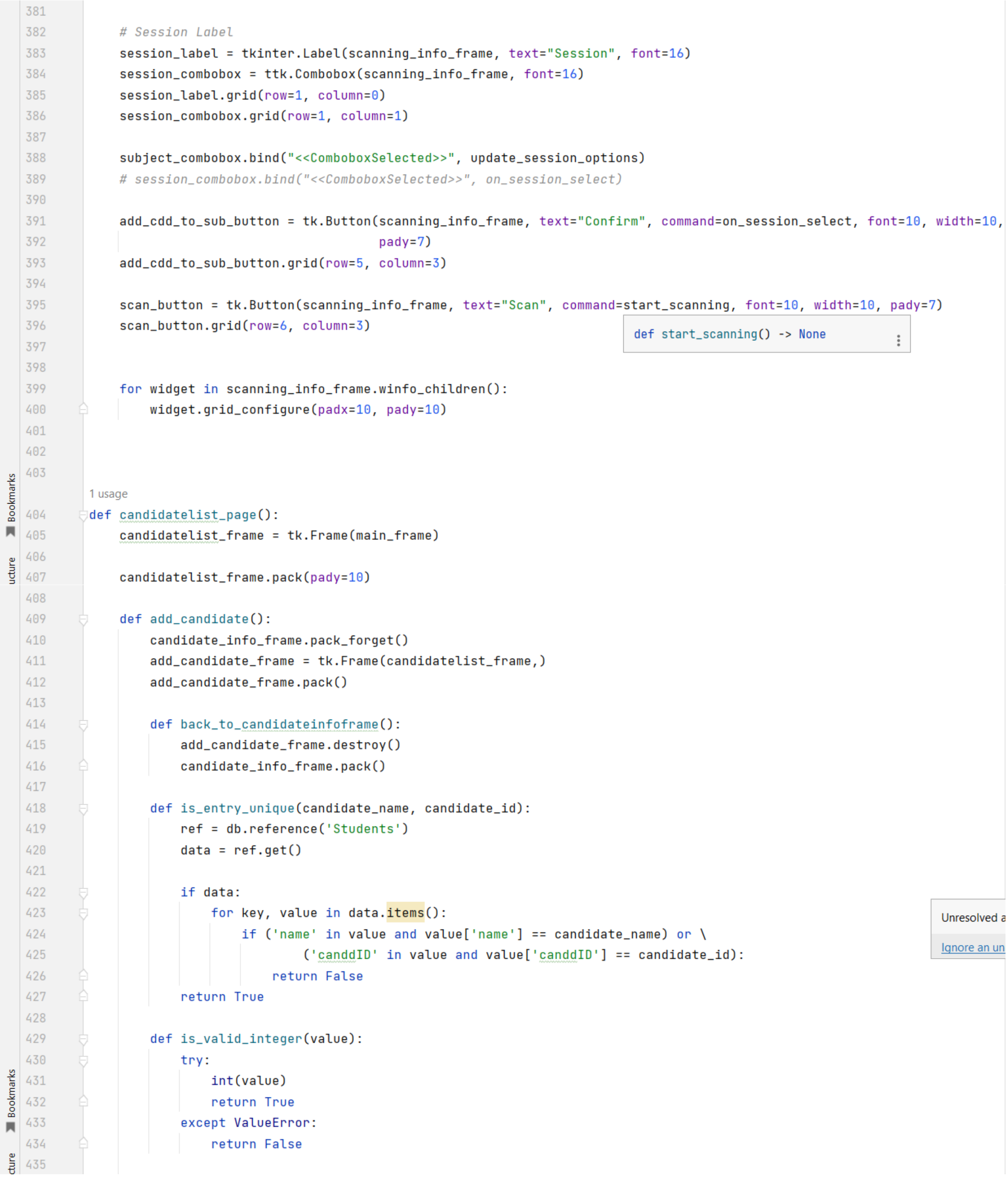
```
    selected_subject = subject_combobox.get()
    selected_session = session_combobox.get()

    # Get the Registration Key for the selected subject and session
    selected_registration_key = get_registration_key(selected_subject)

    print("Selected Registration Key:", selected_registration_key)
    print("Selected Subject:", selected_subject)
    print("Selected Session:", selected_session)

    message = f"Selected exam subject is {selected_subject} and Session {selected_session}"
    messagebox.showinfo("Pop-Up", message)

def retrieve_data(choose_subject):
    ref = db.reference('ExamRegistrations')
    registrations = ref.get()

    session_numbers = []  # Create an empty list to store session numbers

    if registrations is not None:
        for registration_key, registration_data in registrations.items():
            if isinstance(registration_data, dict):
                print(f"Registration Key: {registration_key}")
                print("Registration Data:")

                for key, value in registration_data.items():
                    if key == 'sessions':
                        print("  Sessions Data:")
                        if isinstance(value, dict):
                            for session_number, session_data in value.items():
                                if session_number != 0:
                                    if registration_data['subject_name'] == selected_subject:
                                        print(f"    Session Number: {session_number}")
                                        for session_key, session_value in session_data.items():
                                            print(f"      {session_key}: {session_value}")
                                        session_numbers.append(session_number)  # Append session number to the list
                        else:
                            sessions_dict = {}  # Create a new dictionary to store sessions
                            for session_number, session_list_item in enumerate(value, start=0):
                                if isinstance(session_list_item, dict):
                                    session_dict = {}  # Create a dictionary for session attributes
                                    for session_key, session_value in session_list_item.items():
                                        session_dict[session_key] = session_value
                                    sessions_dict[str(session_number)] = session_dict
                                    if session_number != 0:
                                        if registration_data['subject_name'] == choose_subject:
                                            print(f"    Session Number: {session_number}")
                                            for session_key, session_value in session_dict.items():
                                                print(f"      {session_key}: {session_value}")
                                                if session_key == selected_session:
                                                    new_session_key = "new_session_key"
                                                    new_session_value = "new_session_value"

                                                    session_dict[new_session_key] = new_session_value
                                                    print(f"{session_key}: {session_value}")
                                            session_numbers.append(
                                                str(session_number))  # Append session number to the list
```

Figure 4.3.5.6: AdminTab.py Diagram (6/15)



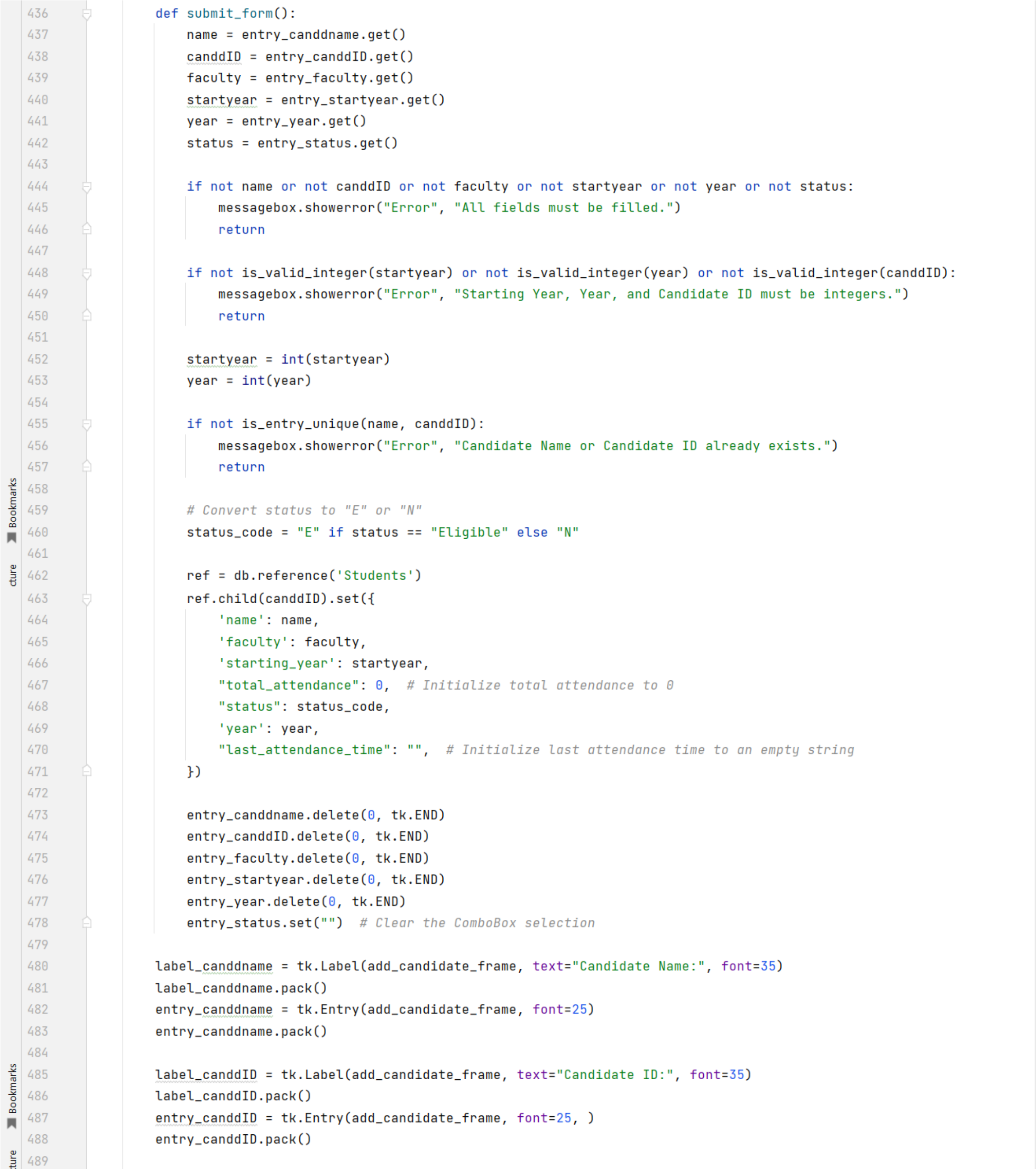
```
                                else:
                                    print(
                                        f"    Session data for number {session_number} is not in dictionary format."
                            registration_data['sessions'] = sessions_dict  # Update the sessions data
                    else:
                        print(f"  {key}: {value}")
                print("\n")
            else:
                print(f"Registration with key {registration_key} is not in dictionary format.")

    return session_numbers  # Return the list of session numbers

def update_session_options(event):
    global selected_subject, selected_session
    selected_subject = subject_combobox.get()

    # Call the retrieve_data function to fetch and print the data and save session numbers in a list
    session_numbers_list = retrieve_data(selected_subject)
    print("Session Numbers List:", session_numbers_list)

    # Clear the session combobox
    session_combobox.set("")  # Clear the selected value
    session_combobox['values'] = session_numbers_list

def update_last_attendance_time(registration_key, session_key, last_attendance_time):
    try:
        # Construct the Firebase reference
        ref = db.reference(f'ExamRegistrations/{registration_key}/sessions/{session_key}')

        # Update 'last_attendance_time' in Firebase
        ref.update({'last_attendance_time': last_attendance_time})
        print(f"Updated 'last_attendance_time' for session {session_key} in Firebase.")
        return True
    except Exception as e:
        print(f"An error occurred: {str(e)}")
        return False

def add_child_to_session(registration_key, session_number, new_child_key, new_child_value):
    ref = db.reference(f'ExamRegistrations/{registration_key}/sessions/{session_number}')
    ref.update({new_child_key: new_child_value})

# Subject Label
subject_label = tkinter.Label(scanning_info_frame, text="Subject", font=16)
subject_combobox = ttk.Combobox(scanning_info_frame, font=16)
subject_label.grid(row=0, column=0)
subject_combobox.grid(row=0, column=1)

if subject_names_data:
    # subject_names = [data['subject_name'] for data in subject_names_data.values()]
    subject_names = [data['subject_name'] for data in subject_names_data.values() if 'subject_name' in data]

    # Add an empty value at the beginning of the list
    subject_names.insert(0, "")
    subject_combobox['values'] = subject_names
else:
    messagebox.showerror("Error", "No subject names found in Firebase.")
```

Figure 4.3.5.7: AdminTab.py Diagram (7/15)



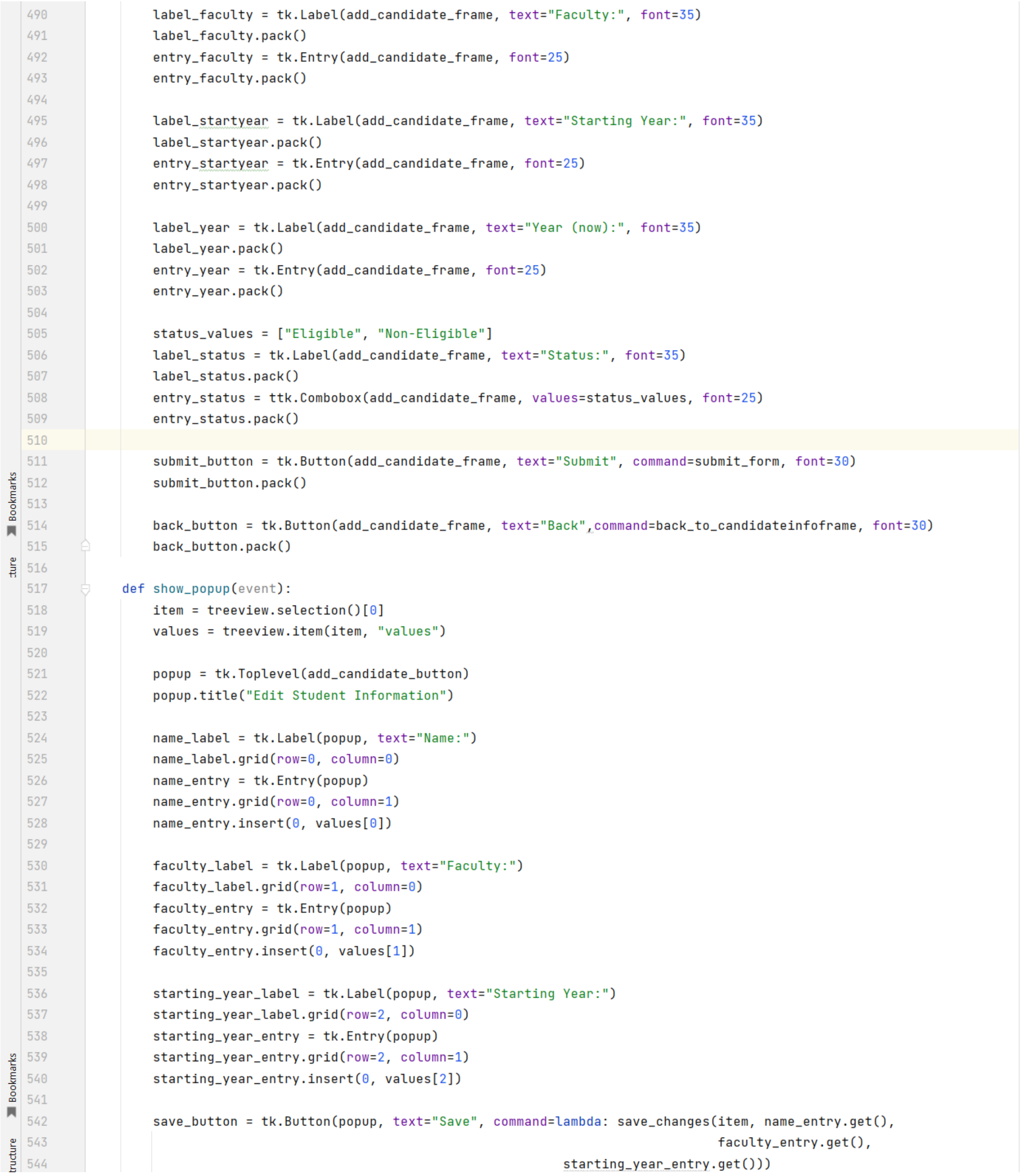
```python

    # Session Label
    session_label = tkinter.Label(scanning_info_frame, text="Session", font=16)
    session_combobox = ttk.Combobox(scanning_info_frame, font=16)
    session_label.grid(row=1, column=0)
    session_combobox.grid(row=1, column=1)

    subject_combobox.bind("<<ComboboxSelected>>", update_session_options)
    # session_combobox.bind("<<ComboboxSelected>>", on_session_select)

    add_cdd_to_sub_button = tk.Button(scanning_info_frame, text="Confirm", command=on_session_select, font=10, width=10,
                                      pady=7)
    add_cdd_to_sub_button.grid(row=5, column=3)

    scan_button = tk.Button(scanning_info_frame, text="Scan", command=start_scanning, font=10, width=10, pady=7)
    scan_button.grid(row=6, column=3)


    for widget in scanning_info_frame.winfo_children():
        widget.grid_configure(padx=10, pady=10)



def candidatelist_page():
    candidatelist_frame = tk.Frame(main_frame)

    candidatelist_frame.pack(pady=10)

    def add_candidate():
        candidate_info_frame.pack_forget()
        add_candidate_frame = tk.Frame(candidatelist_frame,)
        add_candidate_frame.pack()

        def back_to_candidateinfoframe():
            add_candidate_frame.destroy()
            candidate_info_frame.pack()

        def is_entry_unique(candidate_name, candidate_id):
            ref = db.reference('Students')
            data = ref.get()

            if data:
                for key, value in data.items():
                    if ('name' in value and value['name'] == candidate_name) or \
                            ('canddID' in value and value['canddID'] == candidate_id):
                        return False
            return True

        def is_valid_integer(value):
            try:
                int(value)
                return True
            except ValueError:
                return False

```

Figure 4.3.5.8: AdminTab.py Diagram (8/15)



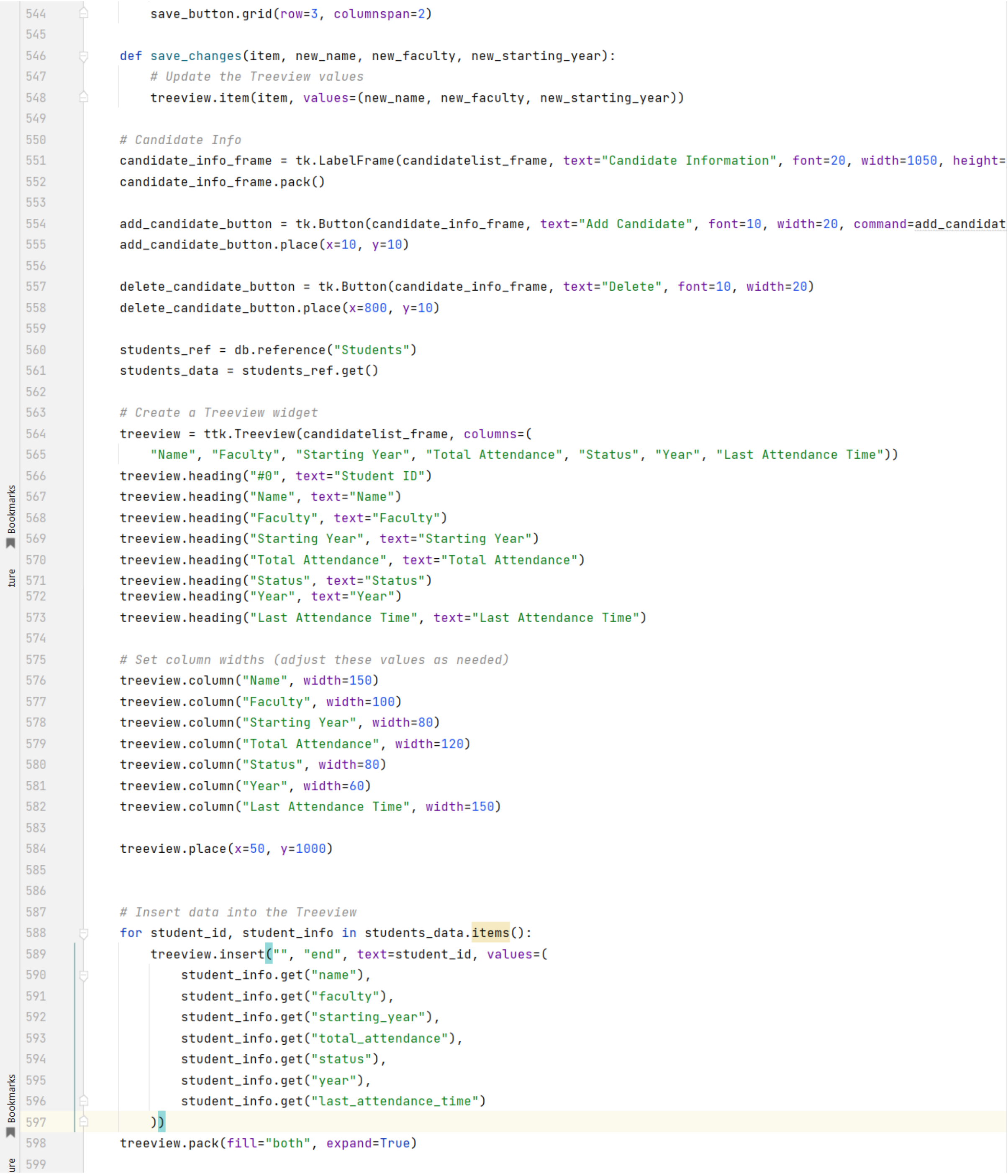
```
def submit_form():
    name = entry_canddname.get()
    canddID = entry_canddID.get()
    faculty = entry_faculty.get()
    startyear = entry_startyear.get()
    year = entry_year.get()
    status = entry_status.get()

    if not name or not canddID or not faculty or not startyear or not year or not status:
        messagebox.showerror("Error", "All fields must be filled.")
        return

    if not is_valid_integer(startyear) or not is_valid_integer(year) or not is_valid_integer(canddID):
        messagebox.showerror("Error", "Starting Year, Year, and Candidate ID must be integers.")
        return

    startyear = int(startyear)
    year = int(year)

    if not is_entry_unique(name, canddID):
        messagebox.showerror("Error", "Candidate Name or Candidate ID already exists.")
        return

    # Convert status to "E" or "N"
    status_code = "E" if status == "Eligible" else "N"

    ref = db.reference('Students')
    ref.child(canddID).set({
        'name': name,
        'faculty': faculty,
        'starting_year': startyear,
        "total_attendance": 0,  # Initialize total attendance to 0
        "status": status_code,
        'year': year,
        "last_attendance_time": "",  # Initialize last attendance time to an empty string
    })

    entry_canddname.delete(0, tk.END)
    entry_canddID.delete(0, tk.END)
    entry_faculty.delete(0, tk.END)
    entry_startyear.delete(0, tk.END)
    entry_year.delete(0, tk.END)
    entry_status.set("")  # Clear the ComboBox selection

label_canddname = tk.Label(add_candidate_frame, text="Candidate Name:", font=35)
label_canddname.pack()
entry_canddname = tk.Entry(add_candidate_frame, font=25)
entry_canddname.pack()

label_canddID = tk.Label(add_candidate_frame, text="Candidate ID:", font=35)
label_canddID.pack()
entry_canddID = tk.Entry(add_candidate_frame, font=25, )
entry_canddID.pack()
```

Figure 4.3.5.9: AdminTab.py Diagram (9/15)



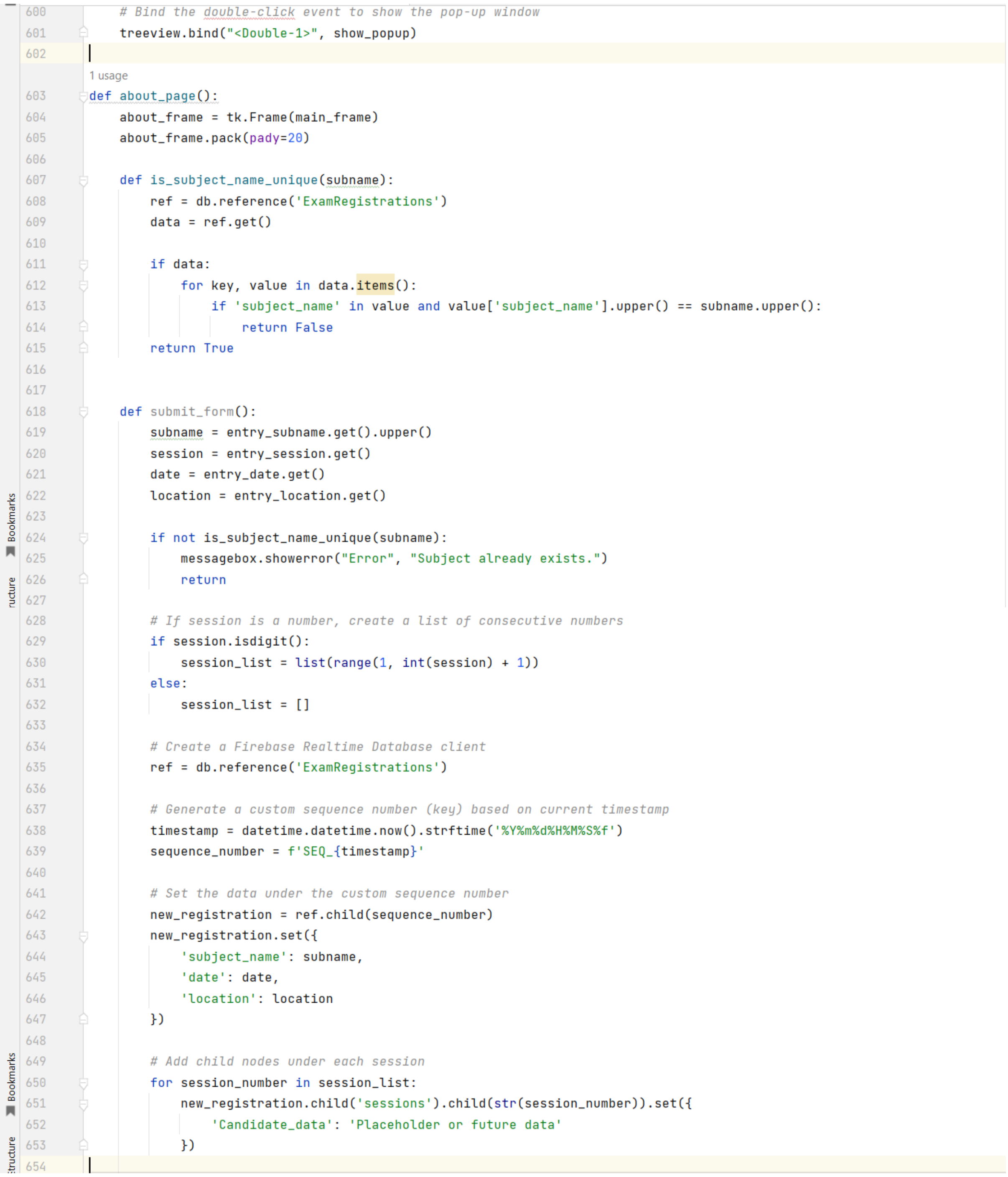
```python
        label_faculty = tk.Label(add_candidate_frame, text="Faculty:", font=35)
        label_faculty.pack()
        entry_faculty = tk.Entry(add_candidate_frame, font=25)
        entry_faculty.pack()

        label_startyear = tk.Label(add_candidate_frame, text="Starting Year:", font=35)
        label_startyear.pack()
        entry_startyear = tk.Entry(add_candidate_frame, font=25)
        entry_startyear.pack()

        label_year = tk.Label(add_candidate_frame, text="Year (now):", font=35)
        label_year.pack()
        entry_year = tk.Entry(add_candidate_frame, font=25)
        entry_year.pack()

        status_values = ["Eligible", "Non-Eligible"]
        label_status = tk.Label(add_candidate_frame, text="Status:", font=35)
        label_status.pack()
        entry_status = ttk.Combobox(add_candidate_frame, values=status_values, font=25)
        entry_status.pack()

        submit_button = tk.Button(add_candidate_frame, text="Submit", command=submit_form, font=30)
        submit_button.pack()

        back_button = tk.Button(add_candidate_frame, text="Back",command=back_to_candidateinfoframe, font=30)
        back_button.pack()

    def show_popup(event):
        item = treeview.selection()[0]
        values = treeview.item(item, "values")

        popup = tk.Toplevel(add_candidate_button)
        popup.title("Edit Student Information")

        name_label = tk.Label(popup, text="Name:")
        name_label.grid(row=0, column=0)
        name_entry = tk.Entry(popup)
        name_entry.grid(row=0, column=1)
        name_entry.insert(0, values[0])

        faculty_label = tk.Label(popup, text="Faculty:")
        faculty_label.grid(row=1, column=0)
        faculty_entry = tk.Entry(popup)
        faculty_entry.grid(row=1, column=1)
        faculty_entry.insert(0, values[1])

        starting_year_label = tk.Label(popup, text="Starting Year:")
        starting_year_label.grid(row=2, column=0)
        starting_year_entry = tk.Entry(popup)
        starting_year_entry.grid(row=2, column=1)
        starting_year_entry.insert(0, values[2])

        save_button = tk.Button(popup, text="Save", command=lambda: save_changes(item, name_entry.get(),
                                                                                 faculty_entry.get(),
                                                             starting_year_entry.get()))
```

Figure 4.3.5.10: AdminTab.py Diagram (10/15)



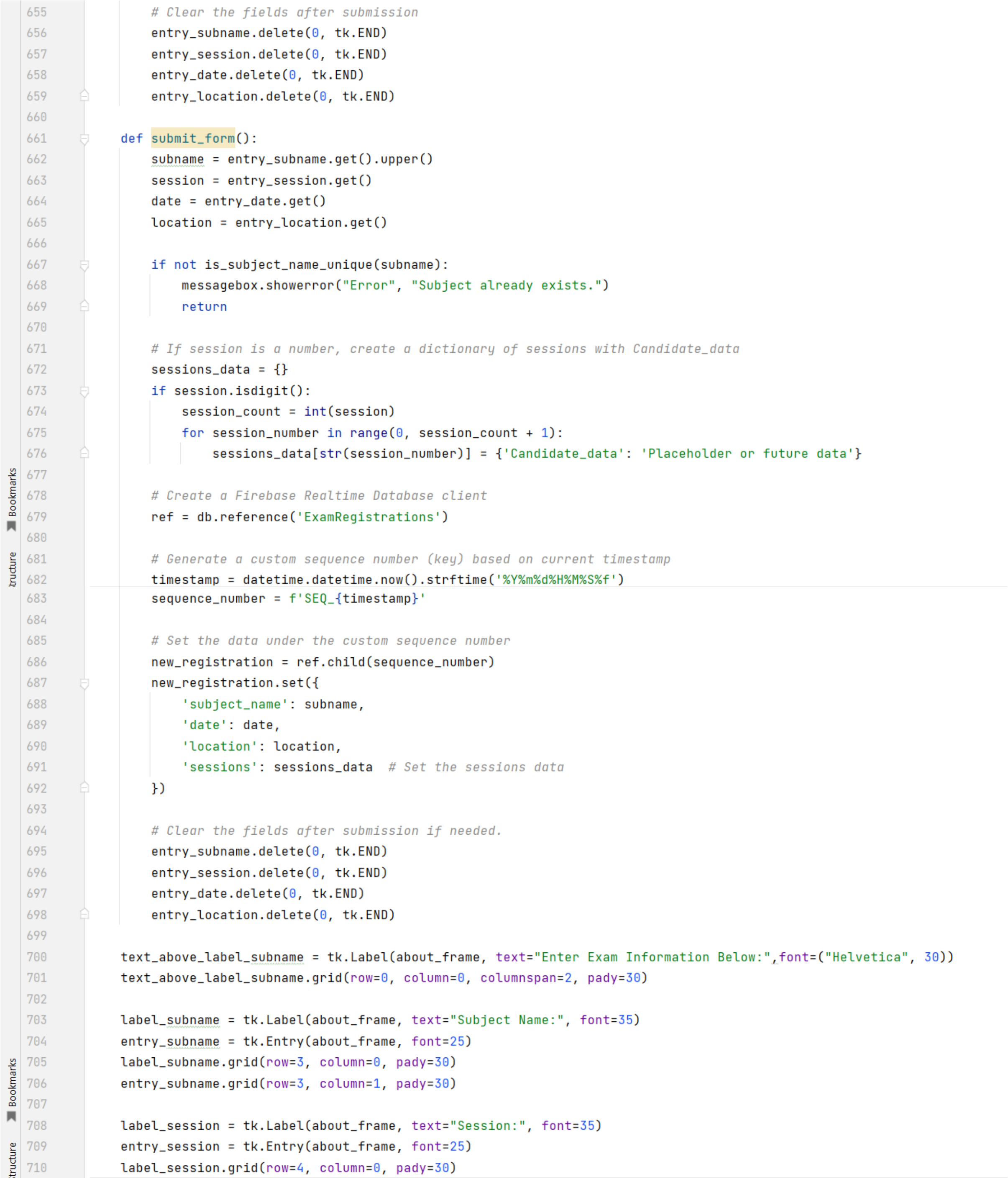
```
    save_button.grid(row=3, columnspan=2)

def save_changes(item, new_name, new_faculty, new_starting_year):
    # Update the Treeview values
    treeview.item(item, values=(new_name, new_faculty, new_starting_year))

# Candidate Info
candidate_info_frame = tk.LabelFrame(candidatelist_frame, text="Candidate Information", font=20, width=1050, height=
candidate_info_frame.pack()

add_candidate_button = tk.Button(candidate_info_frame, text="Add Candidate", font=10, width=20, command=add_candidat
add_candidate_button.place(x=10, y=10)

delete_candidate_button = tk.Button(candidate_info_frame, text="Delete", font=10, width=20)
delete_candidate_button.place(x=800, y=10)

students_ref = db.reference("Students")
students_data = students_ref.get()

# Create a Treeview widget
treeview = ttk.Treeview(candidatelist_frame, columns=(
    "Name", "Faculty", "Starting Year", "Total Attendance", "Status", "Year", "Last Attendance Time"))
treeview.heading("#0", text="Student ID")
treeview.heading("Name", text="Name")
treeview.heading("Faculty", text="Faculty")
treeview.heading("Starting Year", text="Starting Year")
treeview.heading("Total Attendance", text="Total Attendance")
treeview.heading("Status", text="Status")
treeview.heading("Year", text="Year")
treeview.heading("Last Attendance Time", text="Last Attendance Time")

# Set column widths (adjust these values as needed)
treeview.column("Name", width=150)
treeview.column("Faculty", width=100)
treeview.column("Starting Year", width=80)
treeview.column("Total Attendance", width=120)
treeview.column("Status", width=80)
treeview.column("Year", width=60)
treeview.column("Last Attendance Time", width=150)

treeview.place(x=50, y=1000)


# Insert data into the Treeview
for student_id, student_info in students_data.items():
    treeview.insert("", "end", text=student_id, values=(
        student_info.get("name"),
        student_info.get("faculty"),
        student_info.get("starting_year"),
        student_info.get("total_attendance"),
        student_info.get("status"),
        student_info.get("year"),
        student_info.get("last_attendance_time")
    ))
treeview.pack(fill="both", expand=True)
```

Figure 4.3.5.11: AdminTab.py Diagram (11/15)



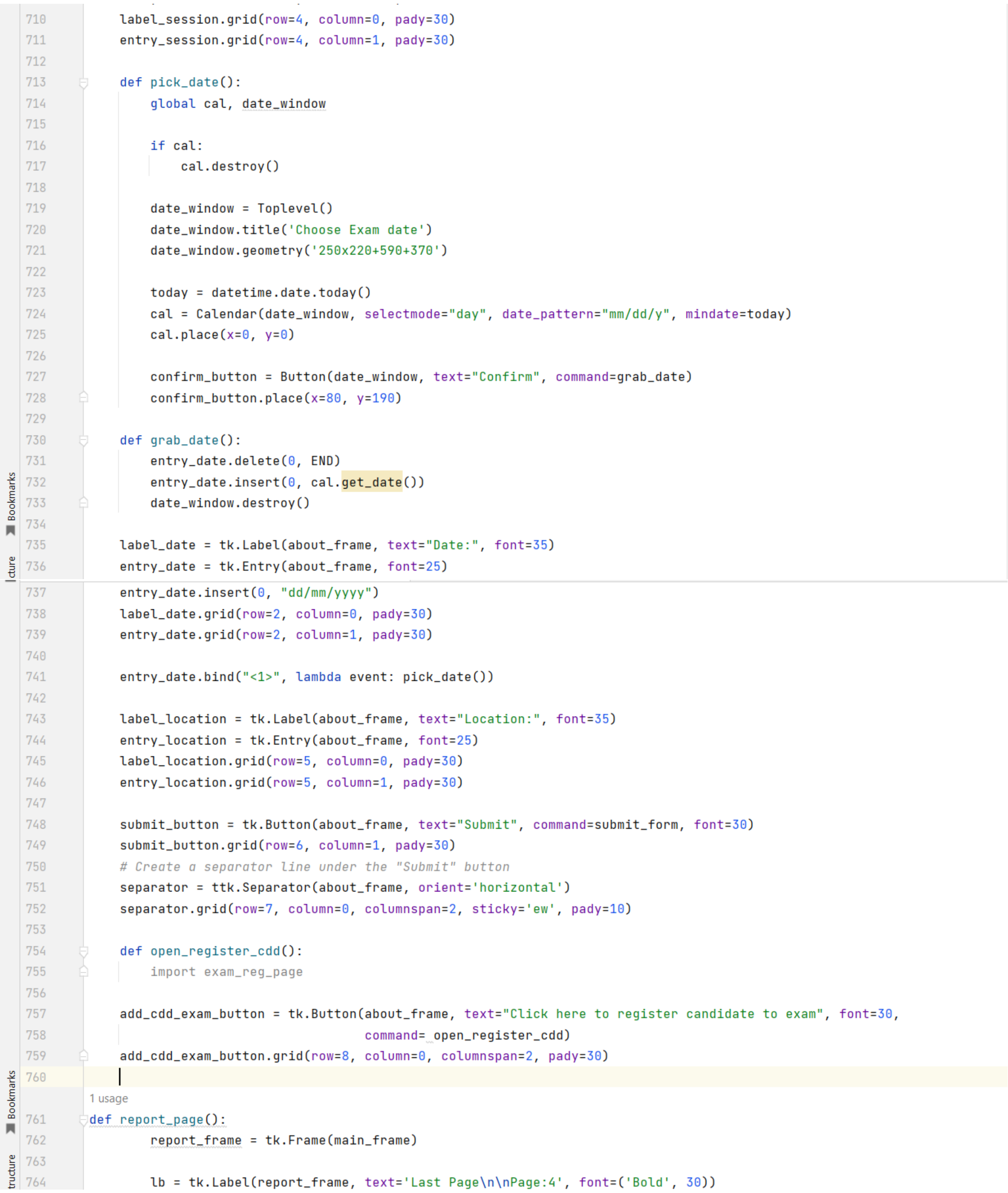
```
    # Bind the double-click event to show the pop-up window
    treeview.bind("<Double-1>", show_popup)

1 usage
def about_page():
    about_frame = tk.Frame(main_frame)
    about_frame.pack(pady=20)

    def is_subject_name_unique(subname):
        ref = db.reference('ExamRegistrations')
        data = ref.get()

        if data:
            for key, value in data.items():
                if 'subject_name' in value and value['subject_name'].upper() == subname.upper():
                    return False
        return True


    def submit_form():
        subname = entry_subname.get().upper()
        session = entry_session.get()
        date = entry_date.get()
        location = entry_location.get()

        if not is_subject_name_unique(subname):
            messagebox.showerror("Error", "Subject already exists.")
            return

        # If session is a number, create a list of consecutive numbers
        if session.isdigit():
            session_list = list(range(1, int(session) + 1))
        else:
            session_list = []

        # Create a Firebase Realtime Database client
        ref = db.reference('ExamRegistrations')

        # Generate a custom sequence number (key) based on current timestamp
        timestamp = datetime.datetime.now().strftime('%Y%m%d%H%M%S%f')
        sequence_number = f'SEQ_{timestamp}'

        # Set the data under the custom sequence number
        new_registration = ref.child(sequence_number)
        new_registration.set({
            'subject_name': subname,
            'date': date,
            'location': location
        })

        # Add child nodes under each session
        for session_number in session_list:
            new_registration.child('sessions').child(str(session_number)).set({
                'Candidate_data': 'Placeholder or future data'
            })
```

Figure 4.3.5.12: AdminTab.py Diagram (12/15)



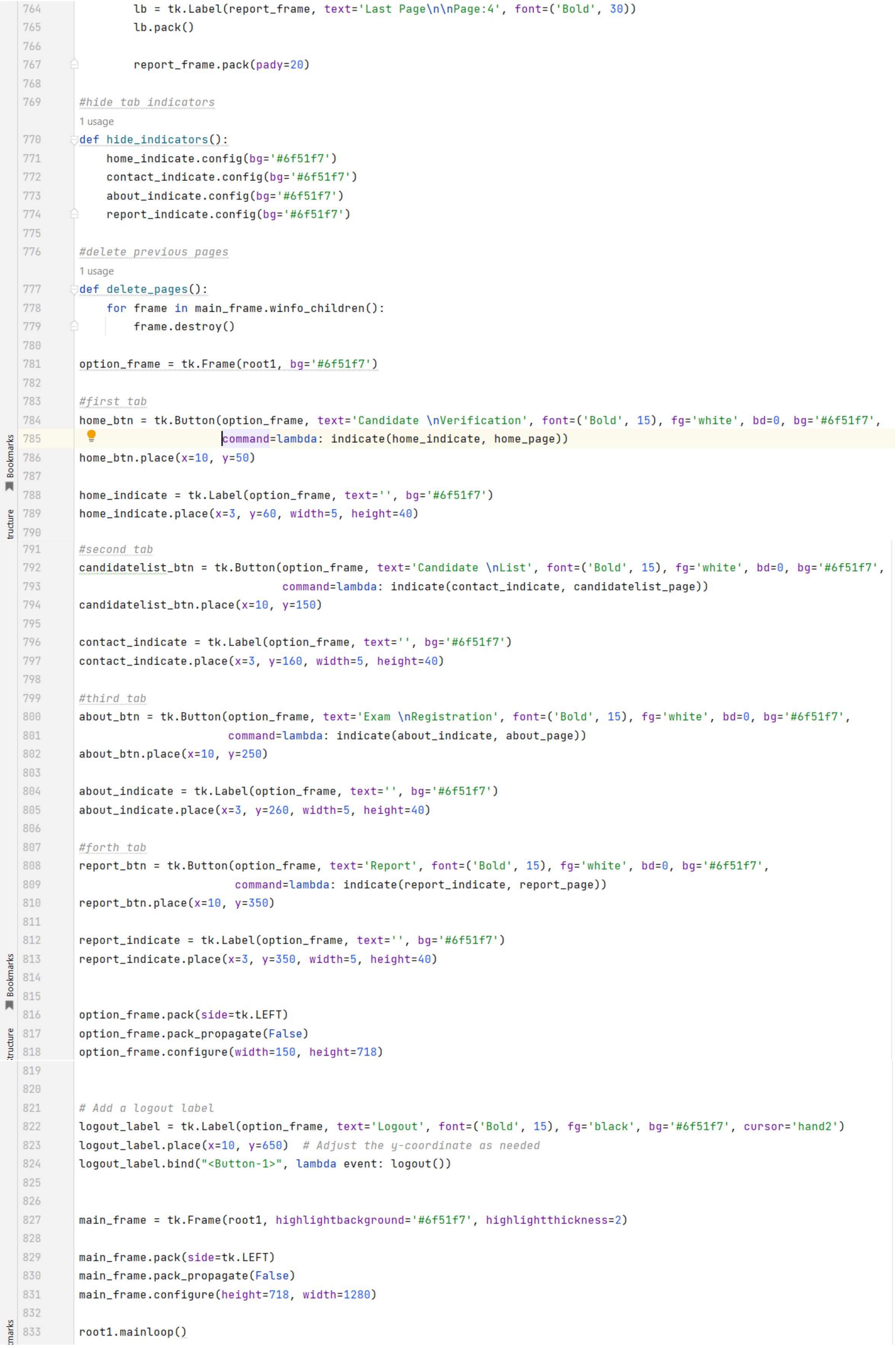
```
        # Clear the fields after submission
        entry_subname.delete(0, tk.END)
        entry_session.delete(0, tk.END)
        entry_date.delete(0, tk.END)
        entry_location.delete(0, tk.END)

    def submit_form():
        subname = entry_subname.get().upper()
        session = entry_session.get()
        date = entry_date.get()
        location = entry_location.get()

        if not is_subject_name_unique(subname):
            messagebox.showerror("Error", "Subject already exists.")
            return

        # If session is a number, create a dictionary of sessions with Candidate_data
        sessions_data = {}
        if session.isdigit():
            session_count = int(session)
            for session_number in range(0, session_count + 1):
                sessions_data[str(session_number)] = {'Candidate_data': 'Placeholder or future data'}

        # Create a Firebase Realtime Database client
        ref = db.reference('ExamRegistrations')

        # Generate a custom sequence number (key) based on current timestamp
        timestamp = datetime.datetime.now().strftime('%Y%m%d%H%M%S%f')
        sequence_number = f'SEQ_{timestamp}'

        # Set the data under the custom sequence number
        new_registration = ref.child(sequence_number)
        new_registration.set({
            'subject_name': subname,
            'date': date,
            'location': location,
            'sessions': sessions_data  # Set the sessions data
        })

        # Clear the fields after submission if needed.
        entry_subname.delete(0, tk.END)
        entry_session.delete(0, tk.END)
        entry_date.delete(0, tk.END)
        entry_location.delete(0, tk.END)

    text_above_label_subname = tk.Label(about_frame, text="Enter Exam Information Below:",font=("Helvetica", 30))
    text_above_label_subname.grid(row=0, column=0, columnspan=2, pady=30)

    label_subname = tk.Label(about_frame, text="Subject Name:", font=35)
    entry_subname = tk.Entry(about_frame, font=25)
    label_subname.grid(row=3, column=0, pady=30)
    entry_subname.grid(row=3, column=1, pady=30)

    label_session = tk.Label(about_frame, text="Session:", font=35)
    entry_session = tk.Entry(about_frame, font=25)
    label_session.grid(row=4, column=0, pady=30)
```

Figure 4.3.5.13: AdminTab.py Diagram (13/15)



```
    label_session.grid(row=4, column=0, pady=30)
    entry_session.grid(row=4, column=1, pady=30)

    def pick_date():
        global cal, date_window

        if cal:
            cal.destroy()

        date_window = Toplevel()
        date_window.title('Choose Exam date')
        date_window.geometry('250x220+590+370')

        today = datetime.date.today()
        cal = Calendar(date_window, selectmode="day", date_pattern="mm/dd/y", mindate=today)
        cal.place(x=0, y=0)

        confirm_button = Button(date_window, text="Confirm", command=grab_date)
        confirm_button.place(x=80, y=190)

    def grab_date():
        entry_date.delete(0, END)
        entry_date.insert(0, cal.get_date())
        date_window.destroy()

    label_date = tk.Label(about_frame, text="Date:", font=35)
    entry_date = tk.Entry(about_frame, font=25)
    entry_date.insert(0, "dd/mm/yyyy")
    label_date.grid(row=2, column=0, pady=30)
    entry_date.grid(row=2, column=1, pady=30)

    entry_date.bind("<1>", lambda event: pick_date())

    label_location = tk.Label(about_frame, text="Location:", font=35)
    entry_location = tk.Entry(about_frame, font=25)
    label_location.grid(row=5, column=0, pady=30)
    entry_location.grid(row=5, column=1, pady=30)

    submit_button = tk.Button(about_frame, text="Submit", command=submit_form, font=30)
    submit_button.grid(row=6, column=1, pady=30)
    # Create a separator line under the "Submit" button
    separator = ttk.Separator(about_frame, orient='horizontal')
    separator.grid(row=7, column=0, columnspan=2, sticky='ew', pady=10)

    def open_register_cdd():
        import exam_reg_page

    add_cdd_exam_button = tk.Button(about_frame, text="Click here to register candidate to exam", font=30,
                                    command= open_register_cdd)
    add_cdd_exam_button.grid(row=8, column=0, columnspan=2, pady=30)

1 usage
def report_page():
        report_frame = tk.Frame(main_frame)

        lb = tk.Label(report_frame, text='Last Page\n\nPage:4', font=('Bold', 30))
```

Figure 4.3.5.14: AdminTab.py Diagram (14/15)

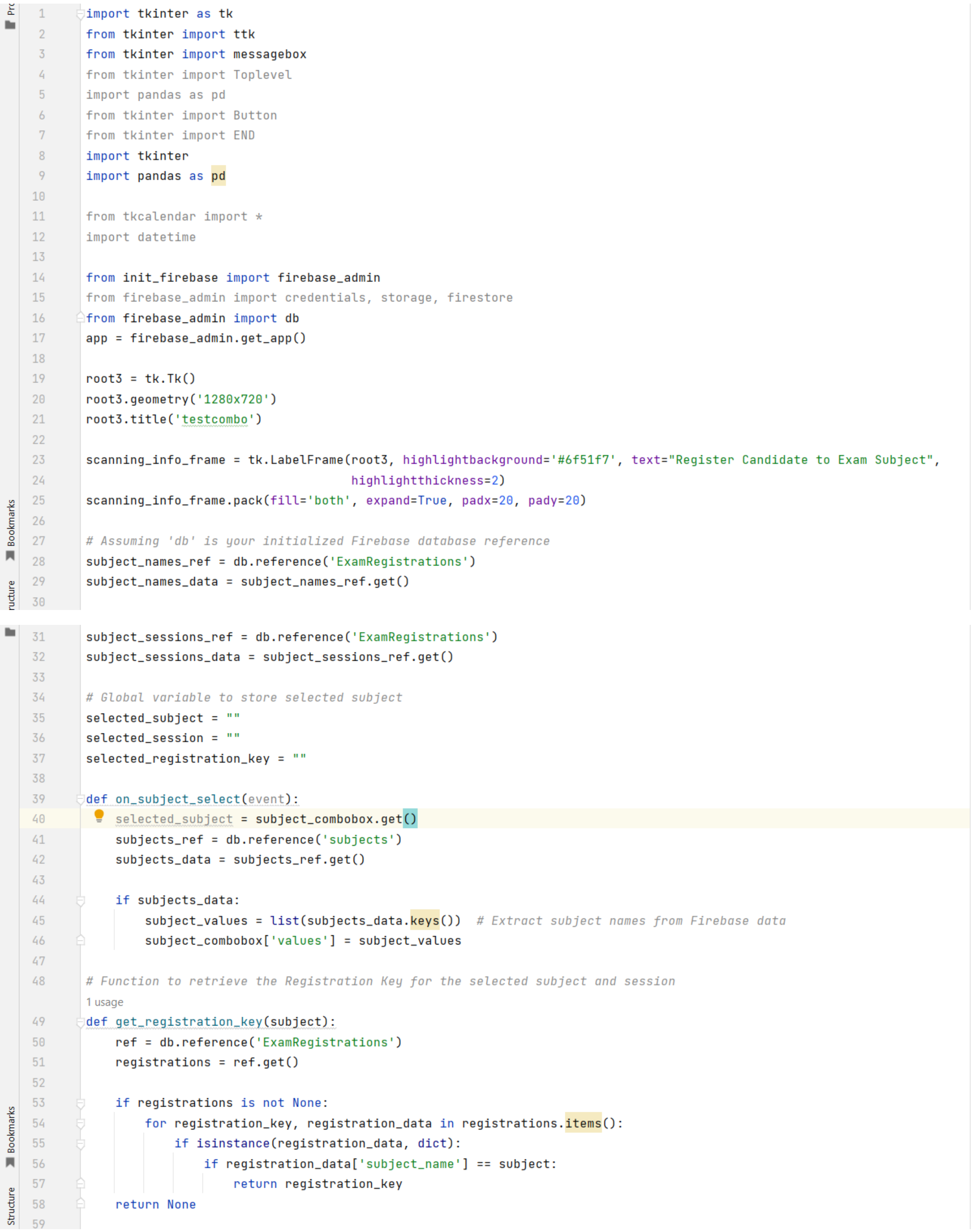


```
        lb = tk.Label(report_frame, text='Last Page\n\nPage:4', font=('Bold', 30))
        lb.pack()

        report_frame.pack(pady=20)

#hide tab indicators
1 usage
def hide_indicators():
    home_indicate.config(bg='#6f51f7')
    contact_indicate.config(bg='#6f51f7')
    about_indicate.config(bg='#6f51f7')
    report_indicate.config(bg='#6f51f7')

#delete previous pages
1 usage
def delete_pages():
    for frame in main_frame.winfo_children():
        frame.destroy()

option_frame = tk.Frame(root1, bg='#6f51f7')

#first tab
home_btn = tk.Button(option_frame, text='Candidate \nVerification', font=('Bold', 15), fg='white', bd=0, bg='#6f51f7',
                     command=lambda: indicate(home_indicate, home_page))
home_btn.place(x=10, y=50)

home_indicate = tk.Label(option_frame, text='', bg='#6f51f7')
home_indicate.place(x=3, y=60, width=5, height=40)

#second tab
candidatelist_btn = tk.Button(option_frame, text='Candidate \nList', font=('Bold', 15), fg='white', bd=0, bg='#6f51f7',
                              command=lambda: indicate(contact_indicate, candidatelist_page))
candidatelist_btn.place(x=10, y=150)

contact_indicate = tk.Label(option_frame, text='', bg='#6f51f7')
contact_indicate.place(x=3, y=160, width=5, height=40)

#third tab
about_btn = tk.Button(option_frame, text='Exam \nRegistration', font=('Bold', 15), fg='white', bd=0, bg='#6f51f7',
                      command=lambda: indicate(about_indicate, about_page))
about_btn.place(x=10, y=250)

about_indicate = tk.Label(option_frame, text='', bg='#6f51f7')
about_indicate.place(x=3, y=260, width=5, height=40)

#forth tab
report_btn = tk.Button(option_frame, text='Report', font=('Bold', 15), fg='white', bd=0, bg='#6f51f7',
                       command=lambda: indicate(report_indicate, report_page))
report_btn.place(x=10, y=350)

report_indicate = tk.Label(option_frame, text='', bg='#6f51f7')
report_indicate.place(x=3, y=350, width=5, height=40)


option_frame.pack(side=tk.LEFT)
option_frame.pack_propagate(False)
option_frame.configure(width=150, height=718)


# Add a logout label
logout_label = tk.Label(option_frame, text='Logout', font=('Bold', 15), fg='black', bg='#6f51f7', cursor='hand2')
logout_label.place(x=10, y=650)  # Adjust the y-coordinate as needed
logout_label.bind("<Button-1>", lambda event: logout())


main_frame = tk.Frame(root1, highlightbackground='#6f51f7', highlightthickness=2)

main_frame.pack(side=tk.LEFT)
main_frame.pack_propagate(False)
main_frame.configure(height=718, width=1280)

root1.mainloop()
```

Figure 4.3.5.15: AdminTab.py Diagram (15/15)

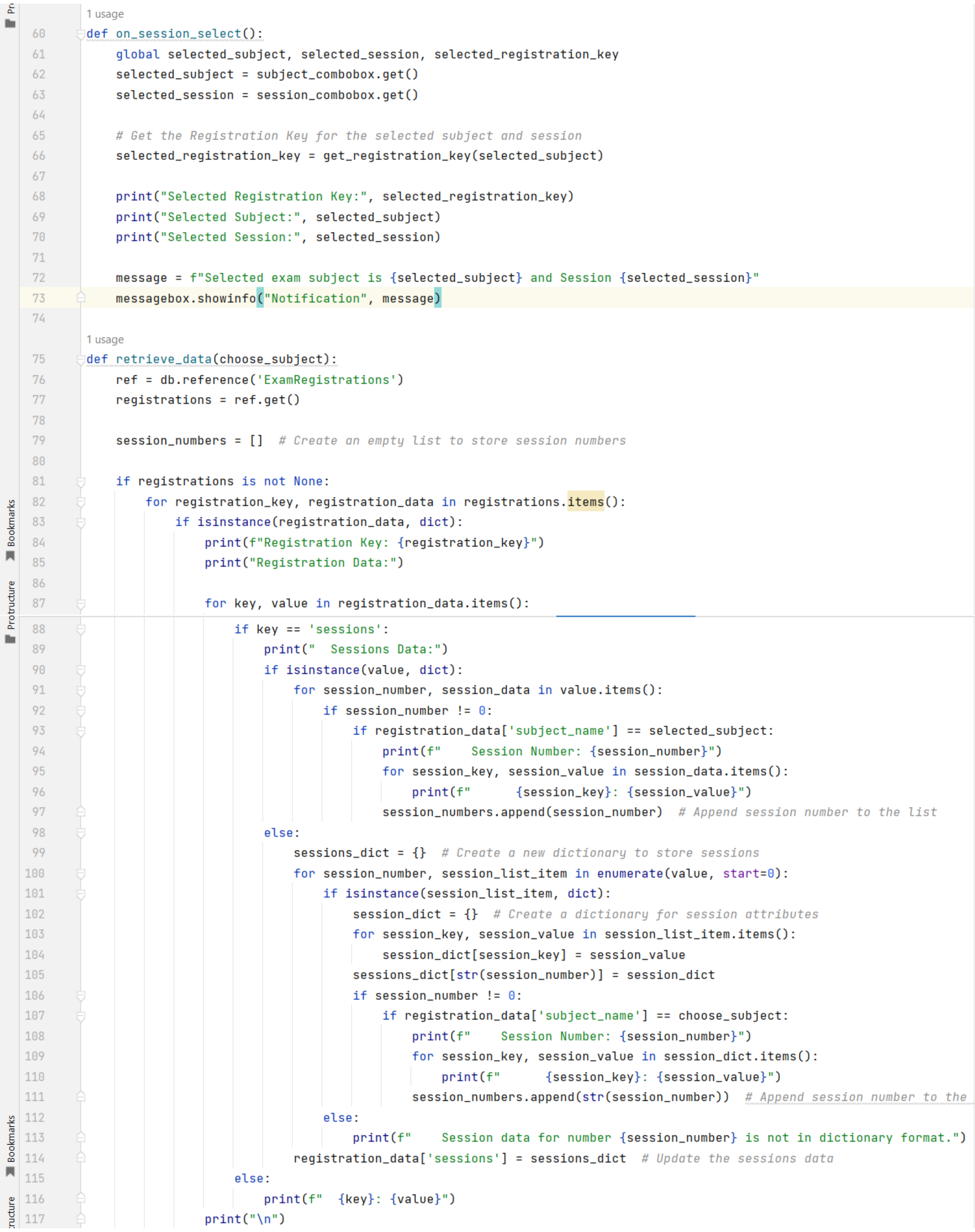
4.3.6 Exam_reg_page.py

The Exam_reg_page.py module is dedicated to the task of registering selected candidates for specific subjects and sessions. In this process, the candidates' IDs and names are stored as child nodes within the session data structure.



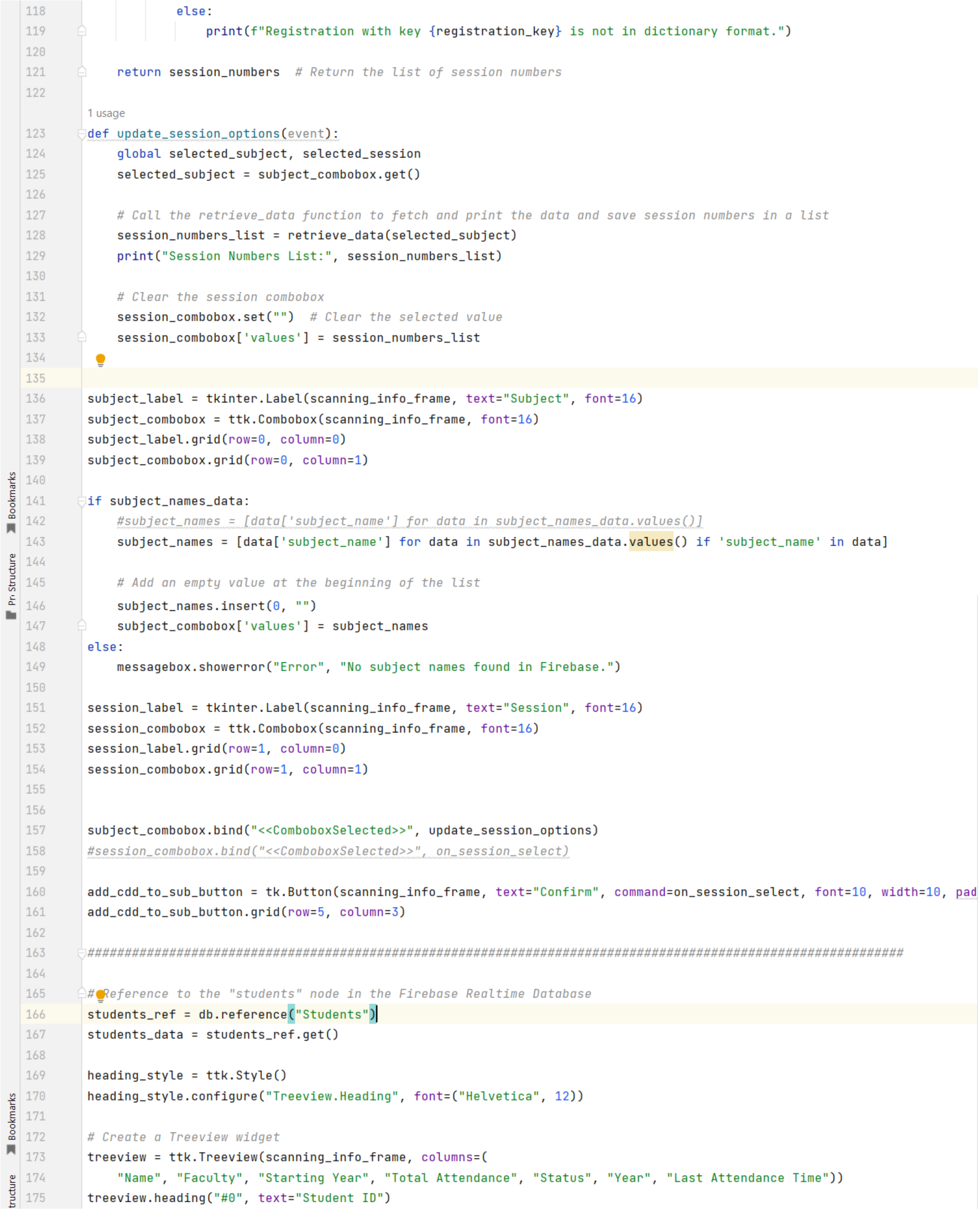
```python
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from tkinter import Toplevel
import pandas as pd
from tkinter import Button
from tkinter import END
import tkinter
import pandas as pd

from tkcalendar import *
import datetime

from init_firebase import firebase_admin
from firebase_admin import credentials, storage, firestore
from firebase_admin import db
app = firebase_admin.get_app()

root3 = tk.Tk()
root3.geometry('1280x720')
root3.title('testcombo')

scanning_info_frame = tk.LabelFrame(root3, highlightbackground='#6f51f7', text="Register Candidate to Exam Subject",
                                    highlightthickness=2)
scanning_info_frame.pack(fill='both', expand=True, padx=20, pady=20)

# Assuming 'db' is your initialized Firebase database reference
subject_names_ref = db.reference('ExamRegistrations')
subject_names_data = subject_names_ref.get()

subject_sessions_ref = db.reference('ExamRegistrations')
subject_sessions_data = subject_sessions_ref.get()

# Global variable to store selected subject
selected_subject = ""
selected_session = ""
selected_registration_key = ""

def on_subject_select(event):
    selected_subject = subject_combobox.get()
    subjects_ref = db.reference('subjects')
    subjects_data = subjects_ref.get()

    if subjects_data:
        subject_values = list(subjects_data.keys())  # Extract subject names from Firebase data
        subject_combobox['values'] = subject_values

# Function to retrieve the Registration Key for the selected subject and session
1 usage
def get_registration_key(subject):
    ref = db.reference('ExamRegistrations')
    registrations = ref.get()

    if registrations is not None:
        for registration_key, registration_data in registrations.items():
            if isinstance(registration_data, dict):
                if registration_data['subject_name'] == subject:
                    return registration_key
    return None
```

Figure 4.3.6.1: Exam_reg_page.py Diagram (1/5)



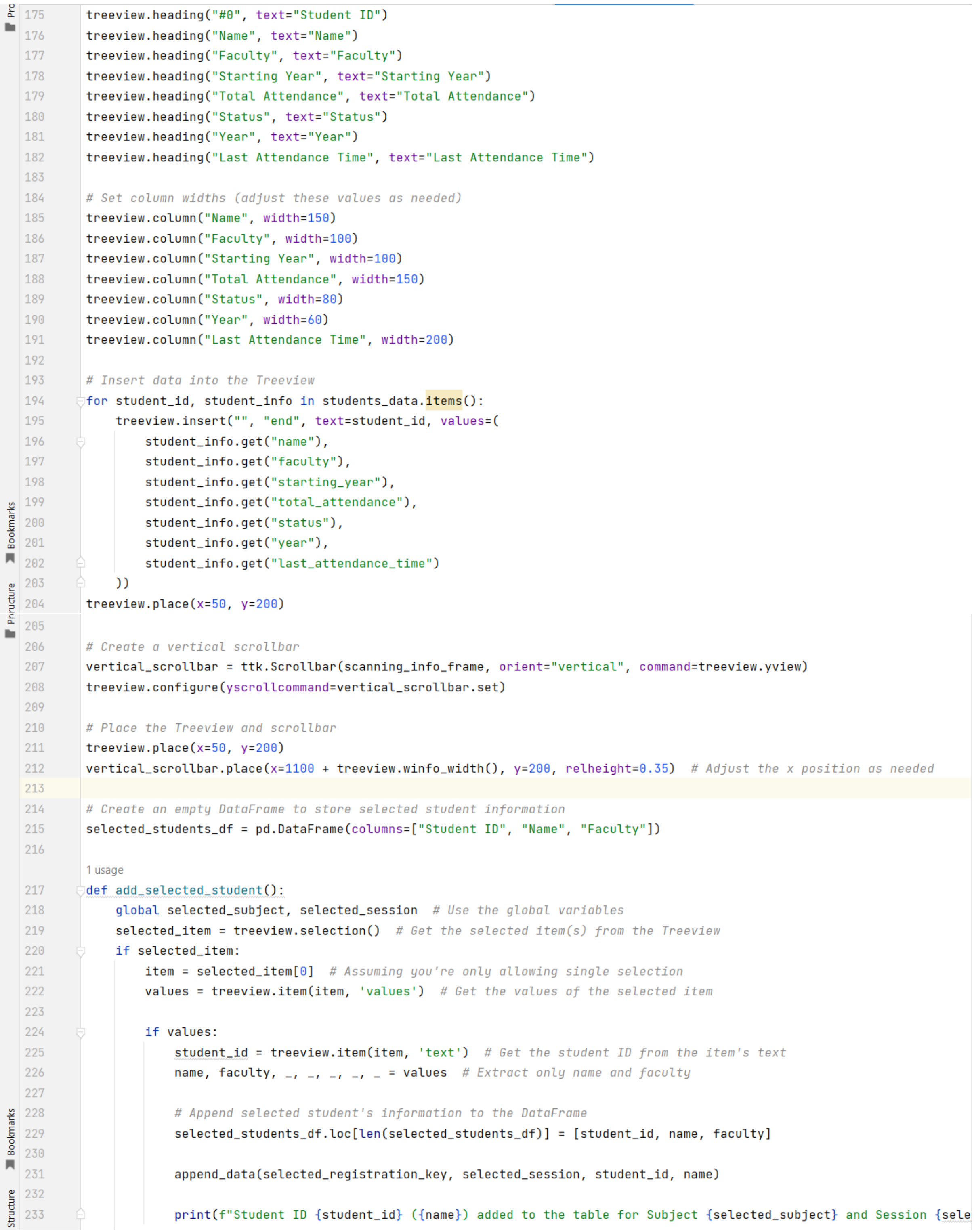
```python
def on_session_select():
    global selected_subject, selected_session, selected_registration_key
    selected_subject = subject_combobox.get()
    selected_session = session_combobox.get()

    # Get the Registration Key for the selected subject and session
    selected_registration_key = get_registration_key(selected_subject)

    print("Selected Registration Key:", selected_registration_key)
    print("Selected Subject:", selected_subject)
    print("Selected Session:", selected_session)

    message = f"Selected exam subject is {selected_subject} and Session {selected_session}"
    messagebox.showinfo("Notification", message)

def retrieve_data(choose_subject):
    ref = db.reference('ExamRegistrations')
    registrations = ref.get()

    session_numbers = []  # Create an empty list to store session numbers

    if registrations is not None:
        for registration_key, registration_data in registrations.items():
            if isinstance(registration_data, dict):
                print(f"Registration Key: {registration_key}")
                print("Registration Data:")

                for key, value in registration_data.items():
                    if key == 'sessions':
                        print("  Sessions Data:")
                        if isinstance(value, dict):
                            for session_number, session_data in value.items():
                                if session_number != 0:
                                    if registration_data['subject_name'] == selected_subject:
                                        print(f"    Session Number: {session_number}")
                                        for session_key, session_value in session_data.items():
                                            print(f"      {session_key}: {session_value}")
                                        session_numbers.append(session_number)  # Append session number to the list
                        else:
                            sessions_dict = {}  # Create a new dictionary to store sessions
                            for session_number, session_list_item in enumerate(value, start=0):
                                if isinstance(session_list_item, dict):
                                    session_dict = {}  # Create a dictionary for session attributes
                                    for session_key, session_value in session_list_item.items():
                                        session_dict[session_key] = session_value
                                    sessions_dict[str(session_number)] = session_dict
                                    if session_number != 0:
                                        if registration_data['subject_name'] == choose_subject:
                                            print(f"    Session Number: {session_number}")
                                            for session_key, session_value in session_dict.items():
                                                print(f"      {session_key}: {session_value}")
                                            session_numbers.append(str(session_number))  # Append session number to the
                                else:
                                    print(f"    Session data for number {session_number} is not in dictionary format.")
                            registration_data['sessions'] = sessions_dict  # Update the sessions data
                    else:
                        print(f"  {key}: {value}")
                print("\n")
```

Figure 4.3.6.2: Exam_reg_page.py Diagram (2/5)



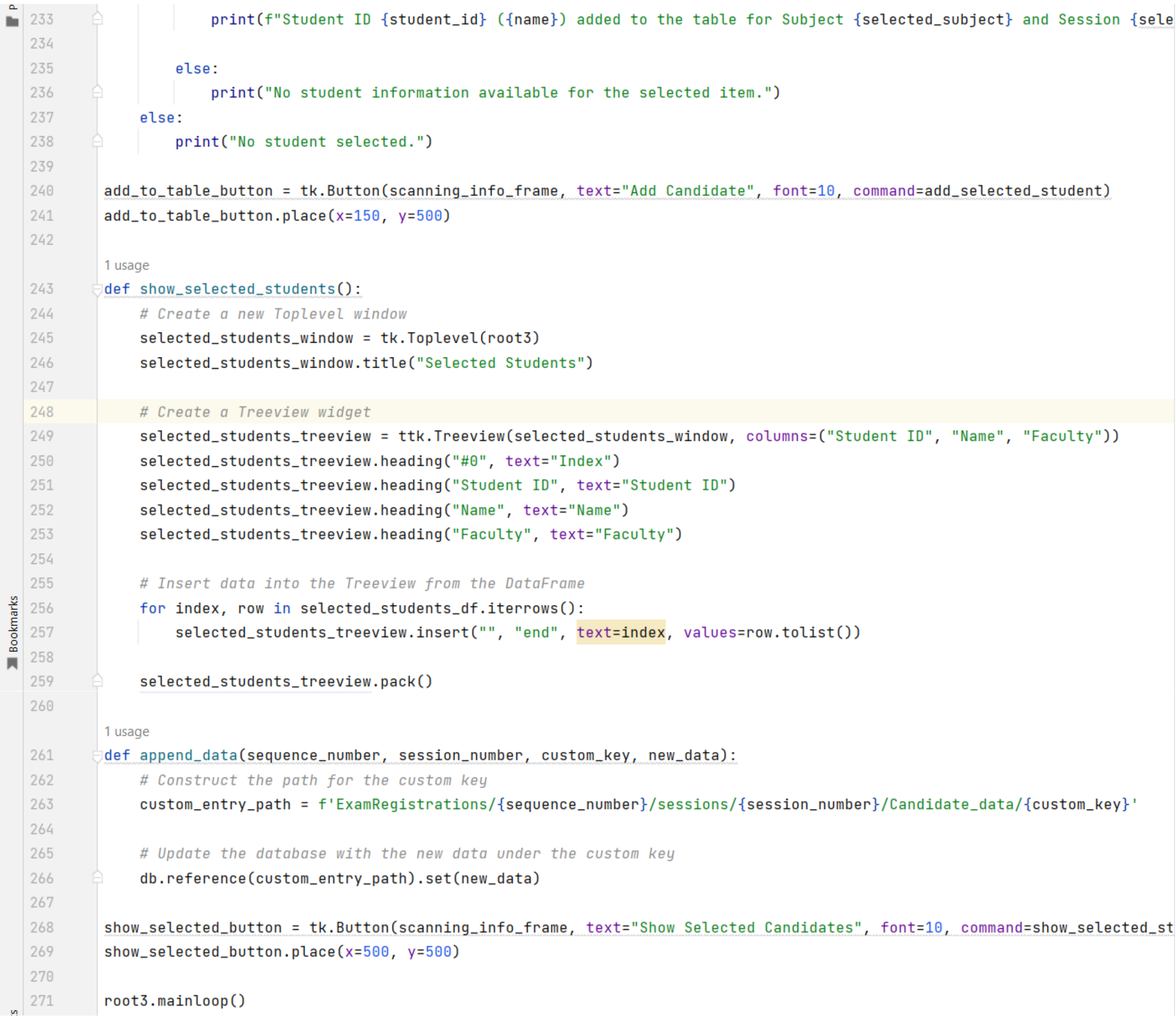
```python
            else:
                print(f"Registration with key {registration_key} is not in dictionary format.")

    return session_numbers  # Return the list of session numbers

def update_session_options(event):
    global selected_subject, selected_session
    selected_subject = subject_combobox.get()

    # Call the retrieve_data function to fetch and print the data and save session numbers in a list
    session_numbers_list = retrieve_data(selected_subject)
    print("Session Numbers List:", session_numbers_list)

    # Clear the session combobox
    session_combobox.set("")  # Clear the selected value
    session_combobox['values'] = session_numbers_list


subject_label = tkinter.Label(scanning_info_frame, text="Subject", font=16)
subject_combobox = ttk.Combobox(scanning_info_frame, font=16)
subject_label.grid(row=0, column=0)
subject_combobox.grid(row=0, column=1)

if subject_names_data:
    #subject_names = [data['subject_name'] for data in subject_names_data.values()]
    subject_names = [data['subject_name'] for data in subject_names_data.values() if 'subject_name' in data]

    # Add an empty value at the beginning of the list
    subject_names.insert(0, "")
    subject_combobox['values'] = subject_names
else:
    messagebox.showerror("Error", "No subject names found in Firebase.")

session_label = tkinter.Label(scanning_info_frame, text="Session", font=16)
session_combobox = ttk.Combobox(scanning_info_frame, font=16)
session_label.grid(row=1, column=0)
session_combobox.grid(row=1, column=1)


subject_combobox.bind("<<ComboboxSelected>>", update_session_options)
#session_combobox.bind("<<ComboboxSelected>>", on_session_select)

add_cdd_to_sub_button = tk.Button(scanning_info_frame, text="Confirm", command=on_session_select, font=10, width=10, pad
add_cdd_to_sub_button.grid(row=5, column=3)

##################################################################################################################

#Reference to the "students" node in the Firebase Realtime Database
students_ref = db.reference("Students")
students_data = students_ref.get()

heading_style = ttk.Style()
heading_style.configure("Treeview.Heading", font=("Helvetica", 12))

# Create a Treeview widget
treeview = ttk.Treeview(scanning_info_frame, columns=(
    "Name", "Faculty", "Starting Year", "Total Attendance", "Status", "Year", "Last Attendance Time"))
treeview.heading("#0", text="Student ID")
```

Figure 4.3.6.3: Exam_reg_page.py Diagram (3/5)



```python
treeview.heading("#0", text="Student ID")
treeview.heading("Name", text="Name")
treeview.heading("Faculty", text="Faculty")
treeview.heading("Starting Year", text="Starting Year")
treeview.heading("Total Attendance", text="Total Attendance")
treeview.heading("Status", text="Status")
treeview.heading("Year", text="Year")
treeview.heading("Last Attendance Time", text="Last Attendance Time")

# Set column widths (adjust these values as needed)
treeview.column("Name", width=150)
treeview.column("Faculty", width=100)
treeview.column("Starting Year", width=100)
treeview.column("Total Attendance", width=150)
treeview.column("Status", width=80)
treeview.column("Year", width=60)
treeview.column("Last Attendance Time", width=200)

# Insert data into the Treeview
for student_id, student_info in students_data.items():
    treeview.insert("", "end", text=student_id, values=(
        student_info.get("name"),
        student_info.get("faculty"),
        student_info.get("starting_year"),
        student_info.get("total_attendance"),
        student_info.get("status"),
        student_info.get("year"),
        student_info.get("last_attendance_time")
    ))
treeview.place(x=50, y=200)

# Create a vertical scrollbar
vertical_scrollbar = ttk.Scrollbar(scanning_info_frame, orient="vertical", command=treeview.yview)
treeview.configure(yscrollcommand=vertical_scrollbar.set)

# Place the Treeview and scrollbar
treeview.place(x=50, y=200)
vertical_scrollbar.place(x=1100 + treeview.winfo_width(), y=200, relheight=0.35)  # Adjust the x position as needed

# Create an empty DataFrame to store selected student information
selected_students_df = pd.DataFrame(columns=["Student ID", "Name", "Faculty"])

1 usage
def add_selected_student():
    global selected_subject, selected_session  # Use the global variables
    selected_item = treeview.selection()  # Get the selected item(s) from the Treeview
    if selected_item:
        item = selected_item[0]  # Assuming you're only allowing single selection
        values = treeview.item(item, 'values')  # Get the values of the selected item

        if values:
            student_id = treeview.item(item, 'text')  # Get the student ID from the item's text
            name, faculty, _, _, _, _, _ = values  # Extract only name and faculty

            # Append selected student's information to the DataFrame
            selected_students_df.loc[len(selected_students_df)] = [student_id, name, faculty]

            append_data(selected_registration_key, selected_session, student_id, name)

            print(f"Student ID {student_id} ({name}) added to the table for Subject {selected_subject} and Session {sele
```

Figure 4.3.6.4: Exam_reg_page.py Diagram (4/5)



```
            print(f"Student ID {student_id} ({name}) added to the table for Subject {selected_subject} and Session {sele

        else:
            print("No student information available for the selected item.")
    else:
        print("No student selected.")

add_to_table_button = tk.Button(scanning_info_frame, text="Add Candidate", font=10, command=add_selected_student)
add_to_table_button.place(x=150, y=500)

1 usage
def show_selected_students():
    # Create a new Toplevel window
    selected_students_window = tk.Toplevel(root3)
    selected_students_window.title("Selected Students")

    # Create a Treeview widget
    selected_students_treeview = ttk.Treeview(selected_students_window, columns=("Student ID", "Name", "Faculty"))
    selected_students_treeview.heading("#0", text="Index")
    selected_students_treeview.heading("Student ID", text="Student ID")
    selected_students_treeview.heading("Name", text="Name")
    selected_students_treeview.heading("Faculty", text="Faculty")

    # Insert data into the Treeview from the DataFrame
    for index, row in selected_students_df.iterrows():
        selected_students_treeview.insert("", "end", text=index, values=row.tolist())

    selected_students_treeview.pack()

1 usage
def append_data(sequence_number, session_number, custom_key, new_data):
    # Construct the path for the custom key
    custom_entry_path = f'ExamRegistrations/{sequence_number}/sessions/{session_number}/Candidate_data/{custom_key}'

    # Update the database with the new data under the custom key
    db.reference(custom_entry_path).set(new_data)

show_selected_button = tk.Button(scanning_info_frame, text="Show Selected Candidates", font=10, command=show_selected_st
show_selected_button.place(x=500, y=500)

root3.mainloop()
```

Figure 4.3.6.5: Exam_reg_page.py Diagram (5/5)

Chapter 5
System Implementation

5.1 Tools

5.1.1 Hardware

In this project, we rely on two primary physical components: a computer, which serves as the central processing unit for various project tasks, and an attached camera, which functions as the scanning module. The camera's primary role is to capture visual data, such as images footage, and process it as part of the project's scanning and data collection processes. This camera is a critical component in ensuring the project's success, as it enables the system to interact with and analyse visual information effectively.

Description	Specifications
Model	Huawei MateBook D 15
Processor	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
Operating System	Windows 10
Graphic	Radeon™ RX Vega 10 Graphics
Memory	4GB DDR4 RAM
Storage	512 GB PCIe SSD

Table 5.1.1: Specifications of laptop

5.1.2 Software

Before embarking on the facial recognition project's development journey, it's crucial to have specific software installed and downloaded on my laptop. The software included are:

- **PyCharm Community Edition 2023.1**
  PyCharm Community Edition is a Python development environment that provides a wide array of features and tools aimed at simplifying the development of Python-based projects. Some of these features include intelligent code completion, debugging tools, testing utilities, and support for version control systems like Git. It's a versatile platform that caters to developers working with different Python frameworks and libraries, making it a valuable tool for Python programming. Moreover, it's accessible on various operating systems, ensuring compatibility for developers using Windows, Mac, or Linux.

- **Firebase Realtime Database**
  Google's Firebase Realtime Database is a cloud-based database created for applications that require real-time data synchronisation across servers, mobile devices, and the web. It employs a JSON-like structure for data storage, enhancing its user-friendliness and adaptability. Real-time data synchronisation, which enables immediate changes between users, is one of its primary characteristics. The database functions even when the internet is down, and when it comes back online, any changes to the data are synchronised. Data privacy is protected by security regulations, and it scales automatically to handle an increase in users and data.

- **Firebase Storage**
  Firebase Storage is a part of Google Firebase, which is a cloud-based storage service tailored for mobile and web apps. It's designed to effortlessly store and deliver user-generated content, like images and videos, simplifying the task for developers to manage without complex infrastructure. It offers software development kits (SDKs) for various platforms, ensuring seamless integration into apps. Additionally, it boasts the capability to handle resumable uploads and downloads, including sizable files, and possesses automatic scalability to accommodate expanding storage needs.

5.1.3 Additional Module and Libraries

- **OpenCV (Open-Source Computer Vision Library)**

  OpenCV is a popular library for computer vision tasks. It offers resources and functionality for image and video analysis, including computer vision techniques, object identification, and image processing.

- **Face-Recognition**

  A Python library that simplifies face recognition tasks. It is frequently used for applications like face authentication and tracking since it can find and identify faces in image or video streams.

- **Firebase-admin**

  A Python library that allows user to interact with Firebase services from a server or backend environment. Firebase, which is a cloud platform that provides various services such as real-time databases, authentication, and cloud storage.

- **PyQt5**

  PyQt5 is a Python package for developing graphical user interfaces (GUIs) for desktop applications. It includes Qt framework bindings, allowing user to create cross-platform apps with a native appearance and feel.

- **Matplotlib**

  Matplotlib is a popular Python data visualisation package. It includes several tools for generating static, animated, and interactive plots and charts.

- **NumPy**

  NumPy is a fundamental library for scientific computing in Python. It supports massive, multi-dimensional arrays and matrices, as well as mathematical procedures for effectively operating on large arrays.

- **Dlib**

  Dlib is a modern C++ toolkit used for object detection, facial landmark detection, and face recognition. Its facial recognition algorithm relies on a deep metric learning strategy, transforming faces into a multi-dimensional space to gauge the similarity by measuring distances between them.

- **Flask**

  Flask is a Python micro web framework that allows developers to create web apps using Python as the primary programming language. It is a quick and versatile framework that allows developers to make multiple architectural options based on their preferences.

- **Pandas**

  Pandas is a Python data manipulation package. It provides data structures such as DataFrames for managing and analysing structured data, making it a powerful data analysis and manipulation tool.

- **Tkcalendar**

  Tkcalendar is a Python package that provides Tkinter with the Calendar and DateEntry widgets. The DateEntry widget is similar to a Combobox, but the drop-down menu is a Calendar to pick a date rather than a list.

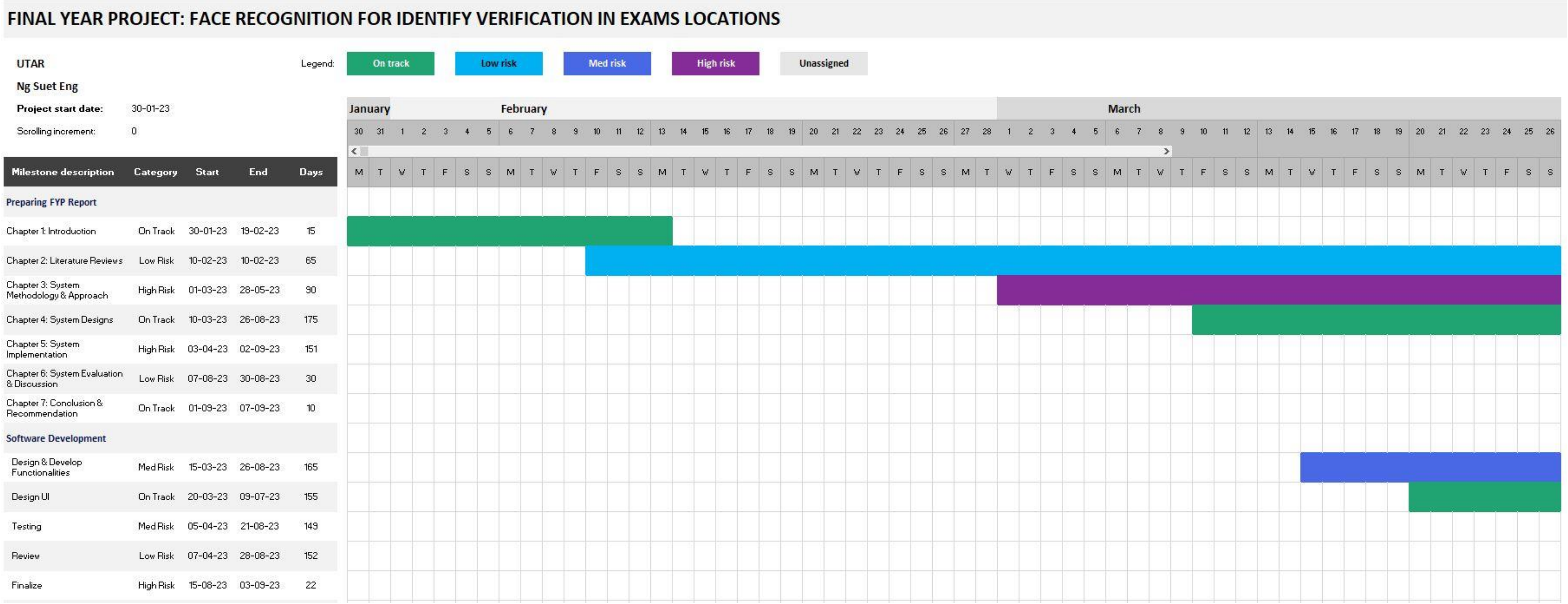
- **Requests**

  Requests is a popular Python module for performing HTTP requests. It makes sending HTTP GET, POST, and other forms of queries to web services and APIs easier.

- **Prettytable**

  Prettytable is a Python library for generating ASCII tables from data. It is often used for formatting and displaying tabular data in a human-readable format such as controlling the width of column padding, text alignment, or table border.

5.2 Timeline



FINAL YEAR PROJECT: FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

UTAR

Ng Suet Eng

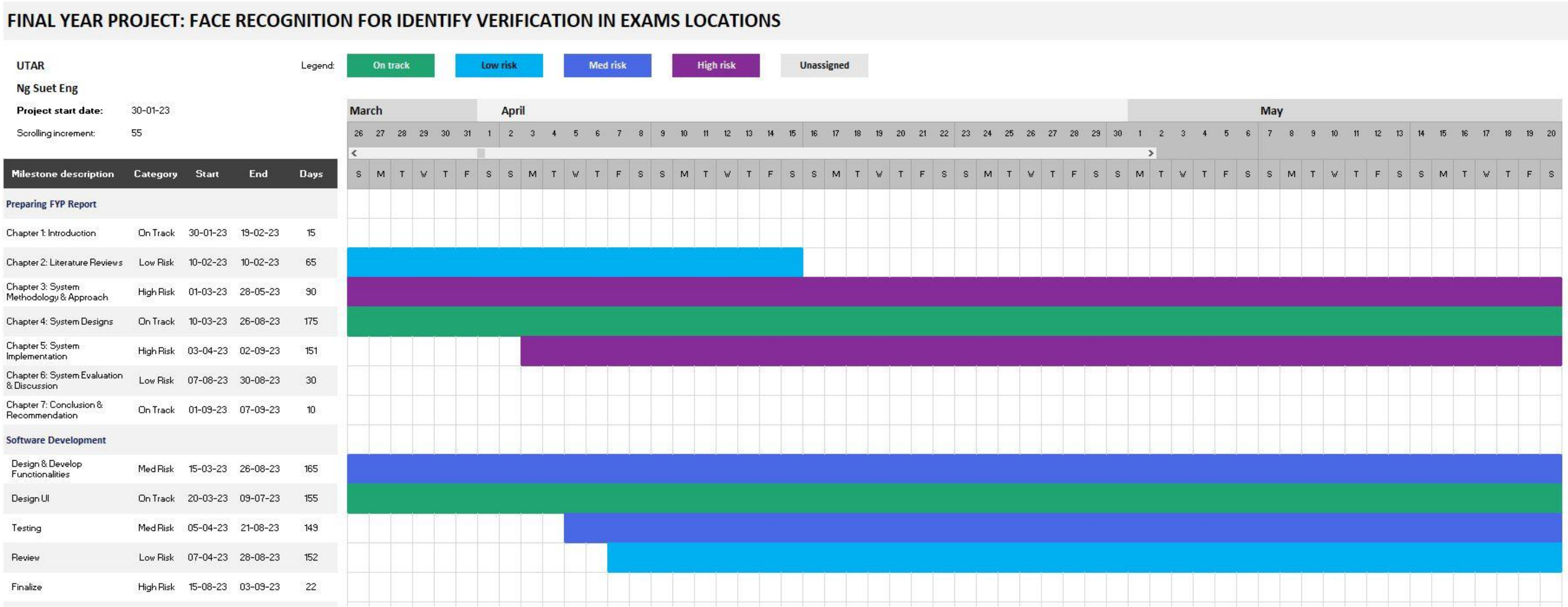
Project start date: 30-01-23

Scrolling increment: 0

Legend: On track | Low risk | Med risk | High risk | Unassigned

Milestone description	Category	Start	End	Days
Preparing FYP Report				
Chapter 1: Introduction	On Track	30-01-23	19-02-23	15
Chapter 2: Literature Reviews	Low Risk	10-02-23	10-02-23	65
Chapter 3: System Methodology & Approach	High Risk	01-03-23	28-05-23	90
Chapter 4: System Designs	On Track	10-03-23	26-08-23	175
Chapter 5: System Implementation	High Risk	03-04-23	02-09-23	151
Chapter 6: System Evaluation & Discussion	Low Risk	07-08-23	30-08-23	30
Chapter 7: Conclusion & Recommendation	On Track	01-09-23	07-09-23	10
Software Development				
Design & Develop Functionalities	Med Risk	15-03-23	26-08-23	165
Design UI	On Track	20-03-23	09-07-23	155
Testing	Med Risk	05-04-23	21-08-23	149
Review	Low Risk	07-04-23	28-08-23	152
Finalize	High Risk	15-08-23	03-09-23	22

Figure 5.2.1: Timeline (1/5)



FINAL YEAR PROJECT: FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

UTAR

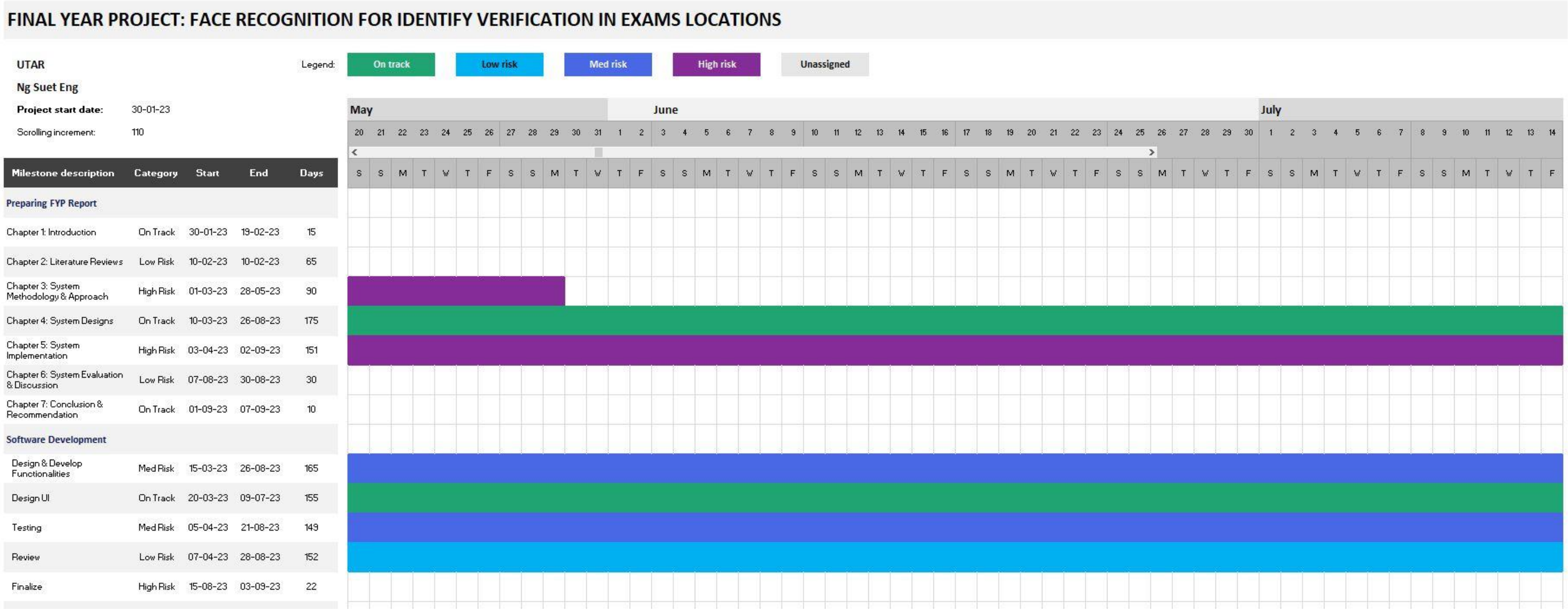
Ng Suet Eng

Project start date: 30-01-23

Scrolling increment: 55

Milestone description	Category	Start	End	Days
Preparing FYP Report				
Chapter 1: Introduction	On Track	30-01-23	19-02-23	15
Chapter 2: Literature Reviews	Low Risk	10-02-23	10-02-23	65
Chapter 3: System Methodology & Approach	High Risk	01-03-23	28-05-23	90
Chapter 4: System Designs	On Track	10-03-23	26-08-23	175
Chapter 5: System Implementation	High Risk	03-04-23	02-09-23	151
Chapter 6: System Evaluation & Discussion	Low Risk	07-08-23	30-08-23	30
Chapter 7: Conclusion & Recommendation	On Track	01-09-23	07-09-23	10
Software Development				
Design & Develop Functionalities	Med Risk	15-03-23	26-08-23	165
Design UI	On Track	20-03-23	09-07-23	155
Testing	Med Risk	05-04-23	21-08-23	149
Review	Low Risk	07-04-23	28-08-23	152
Finalize	High Risk	15-08-23	03-09-23	22

Figure 5.2.2: Timeline (2/5)



FINAL YEAR PROJECT: FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

UTAR

Ng Suet Eng

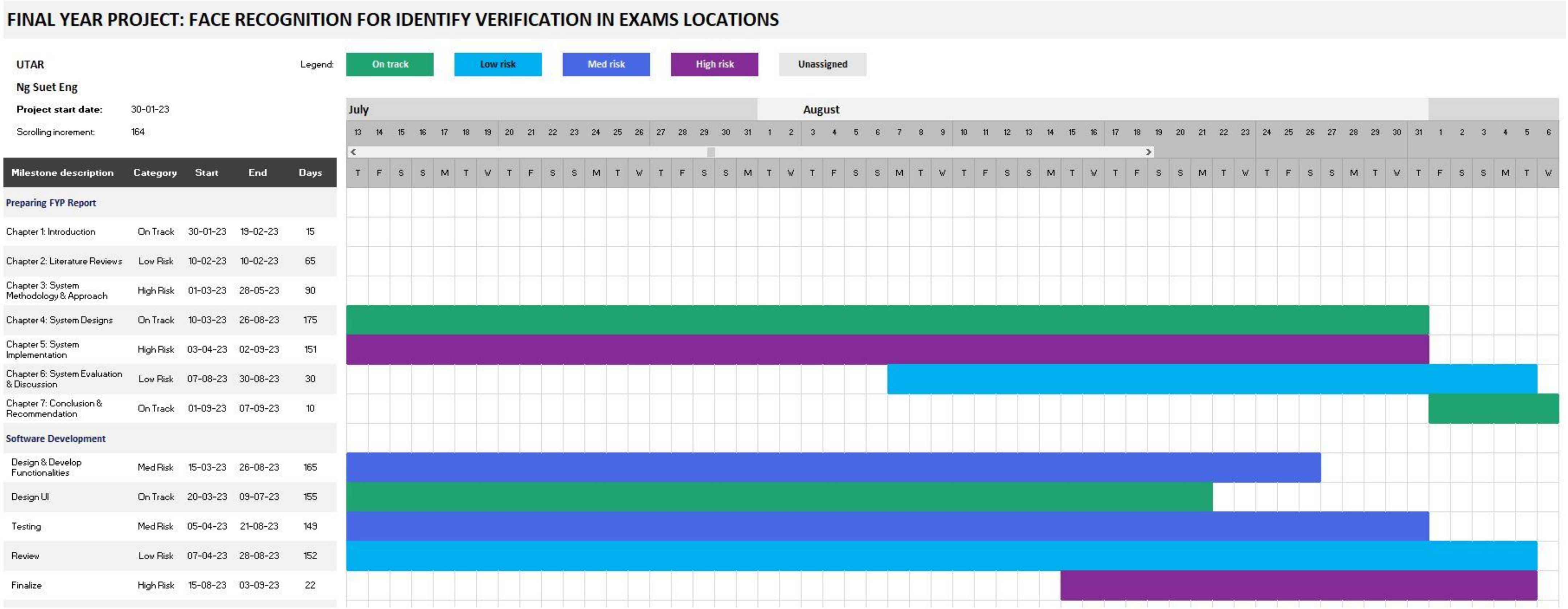
Project start date: 30-01-23

Scrolling increment: 110

Legend: On track | Low risk | Med risk | High risk | Unassigned

Milestone description	Category	Start	End	Days
Preparing FYP Report				
Chapter 1: Introduction	On Track	30-01-23	19-02-23	15
Chapter 2: Literature Reviews	Low Risk	10-02-23	10-02-23	65
Chapter 3: System Methodology & Approach	High Risk	01-03-23	28-05-23	90
Chapter 4: System Designs	On Track	10-03-23	26-08-23	175
Chapter 5: System Implementation	High Risk	03-04-23	02-09-23	151
Chapter 6: System Evaluation & Discussion	Low Risk	07-08-23	30-08-23	30
Chapter 7: Conclusion & Recommendation	On Track	01-09-23	07-09-23	10
Software Development				
Design & Develop Functionalities	Med Risk	15-03-23	26-08-23	165
Design UI	On Track	20-03-23	09-07-23	155
Testing	Med Risk	05-04-23	21-08-23	149
Review	Low Risk	07-04-23	28-08-23	152
Finalize	High Risk	15-08-23	03-09-23	22

Figure 5.2.3: Timeline (3/5)



FINAL YEAR PROJECT: FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

UTAR

Ng Suet Eng

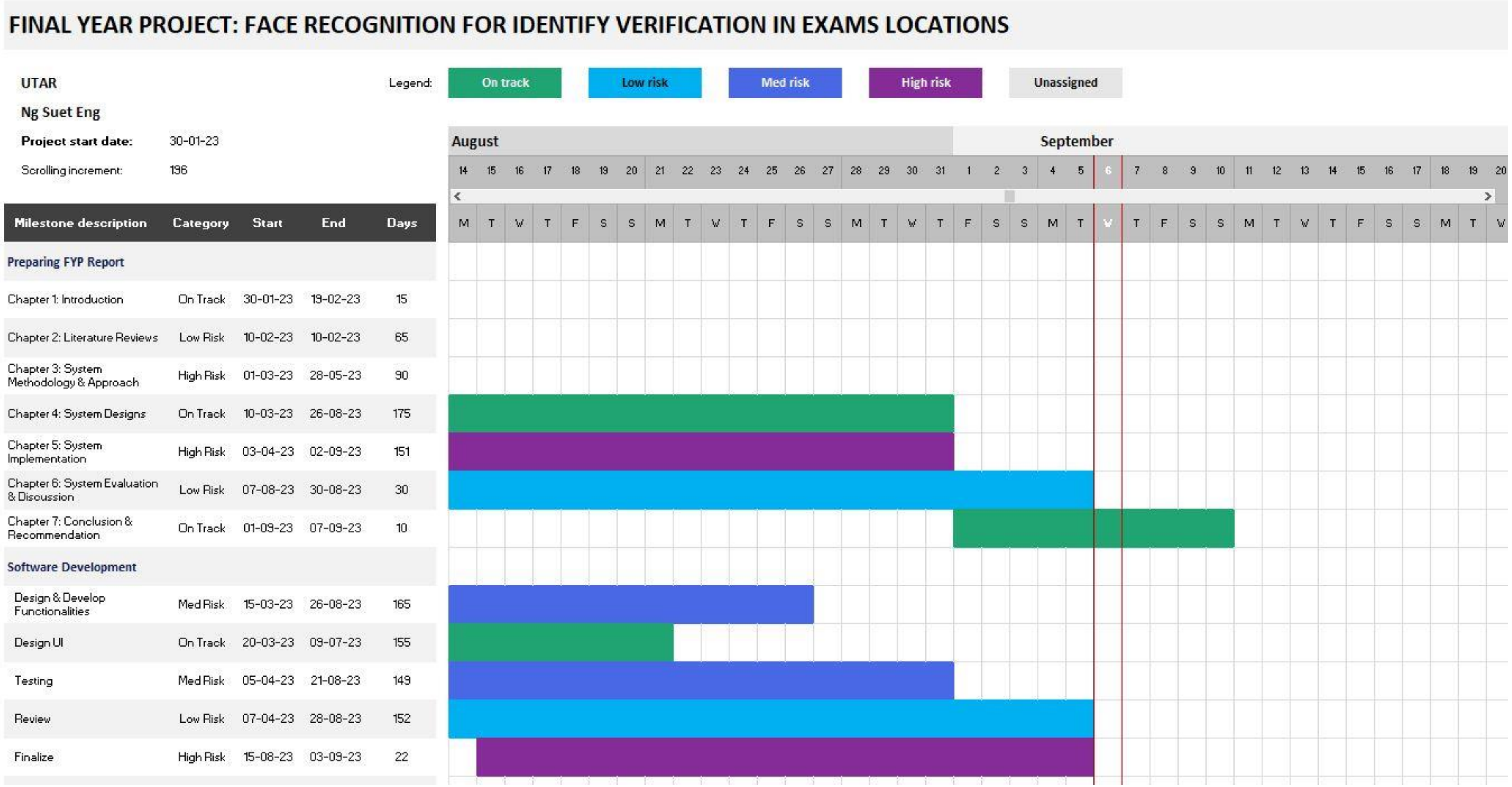
Project start date: 30-01-23

Scrolling increment: 164

Legend: 🟩 On track 🟦 Low risk 🟦 Med risk 🟪 High risk ⬜ Unassigned

Milestone description	Category	Start	End	Days
Preparing FYP Report				
Chapter 1: Introduction	On Track	30-01-23	19-02-23	15
Chapter 2: Literature Reviews	Low Risk	10-02-23	10-02-23	65
Chapter 3: System Methodology & Approach	High Risk	01-03-23	28-05-23	90
Chapter 4: System Designs	On Track	10-03-23	26-08-23	175
Chapter 5: System Implementation	High Risk	03-04-23	02-09-23	151
Chapter 6: System Evaluation & Discussion	Low Risk	07-08-23	30-08-23	30
Chapter 7: Conclusion & Recommendation	On Track	01-09-23	07-09-23	10
Software Development				
Design & Develop Functionalities	Med Risk	15-03-23	26-08-23	165
Design UI	On Track	20-03-23	09-07-23	155
Testing	Med Risk	05-04-23	21-08-23	149
Review	Low Risk	07-04-23	28-08-23	152
Finalize	High Risk	15-08-23	03-09-23	22

Figure 5.2.4: Timeline (4/5)



FINAL YEAR PROJECT: FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

UTAR

Ng Suet Eng

Project start date: 30-01-23

Scrolling increment: 196

Legend: On track | Low risk | Med risk | High risk | Unassigned

Milestone description	Category	Start	End	Days
Preparing FYP Report				
Chapter 1: Introduction	On Track	30-01-23	19-02-23	15
Chapter 2: Literature Reviews	Low Risk	10-02-23	10-02-23	65
Chapter 3: System Methodology & Approach	High Risk	01-03-23	28-05-23	90
Chapter 4: System Designs	On Track	10-03-23	26-08-23	175
Chapter 5: System Implementation	High Risk	03-04-23	02-09-23	151
Chapter 6: System Evaluation & Discussion	Low Risk	07-08-23	30-08-23	30
Chapter 7: Conclusion & Recommendation	On Track	01-09-23	07-09-23	10
Software Development				
Design & Develop Functionalities	Med Risk	15-03-23	26-08-23	165
Design UI	On Track	20-03-23	09-07-23	155
Testing	Med Risk	05-04-23	21-08-23	149
Review	Low Risk	07-04-23	28-08-23	152
Finalize	High Risk	15-08-23	03-09-23	22

Figure 5.2.5: Timeline (5/5)

5.3 Setting and Configuration

Setting up a face recognition attendance system involves a multifaceted process that requires careful attention to hardware, software, and configuration settings. To establish a reliable system, it needs to begin by selecting appropriate hardware components, including a computer, with a camera for capturing faces. Once your hardware is in place, the following steps outline the software and configuration aspects:

First, need to set up the operating system on computer and configure it, ensuring network and security settings are in order. Afterward, setup the face recognition application in the system. Next, establish the camera connection, ensuring it's properly recognized by the system and building the core of system involves preparing a database of images containing the faces of individuals that intend to track attendance for. It must collect an ample number of images per person to ensure accurate recognition.

To enable face detection and recognition, a pre-trained models are using include OpenCV or Dlib. In parallel, Firebase is set up a database to store attendance records, with tables for relevant information such as employee or student IDs, names, timestamps, and more. Enrolling individuals' faces into the system is a crucial step. In the pre-trained models, it typically involves capturing multiple images of each person's face and storing their facial features or embeddings in the computer. The model will further configuration involves fine-tuning the face recognition algorithm by adjusting parameters, thresholds, and settings to balance accuracy and speed.

With these elements in place, it can proceed to track attendance. Implement code to capture video frames from the camera, detect faces within those frames, and compare recognized faces to those in the database, recording attendance data for each recognized face. To complete the system, include features for logging and reporting, registration, and enabling the generation of attendance reports. Security measures should also be implemented to protect against unauthorized access and tampering. Once the system is fully configured, conduct comprehensive testing to ensure accuracy and calibrate it as necessary. Then, deploy the system in desired location, such as an exam venue, ensuring it's operational and connected to the network.

Finally, establish a routine for monitoring system performance and accuracy, and perform regular maintenance tasks, including database backups, software updates, and hardware maintenance. Throughout this process, be mindful of legal and privacy considerations, particularly when dealing with personal biometric data, and ensure compliance with relevant regulations.

5.4 System Operation

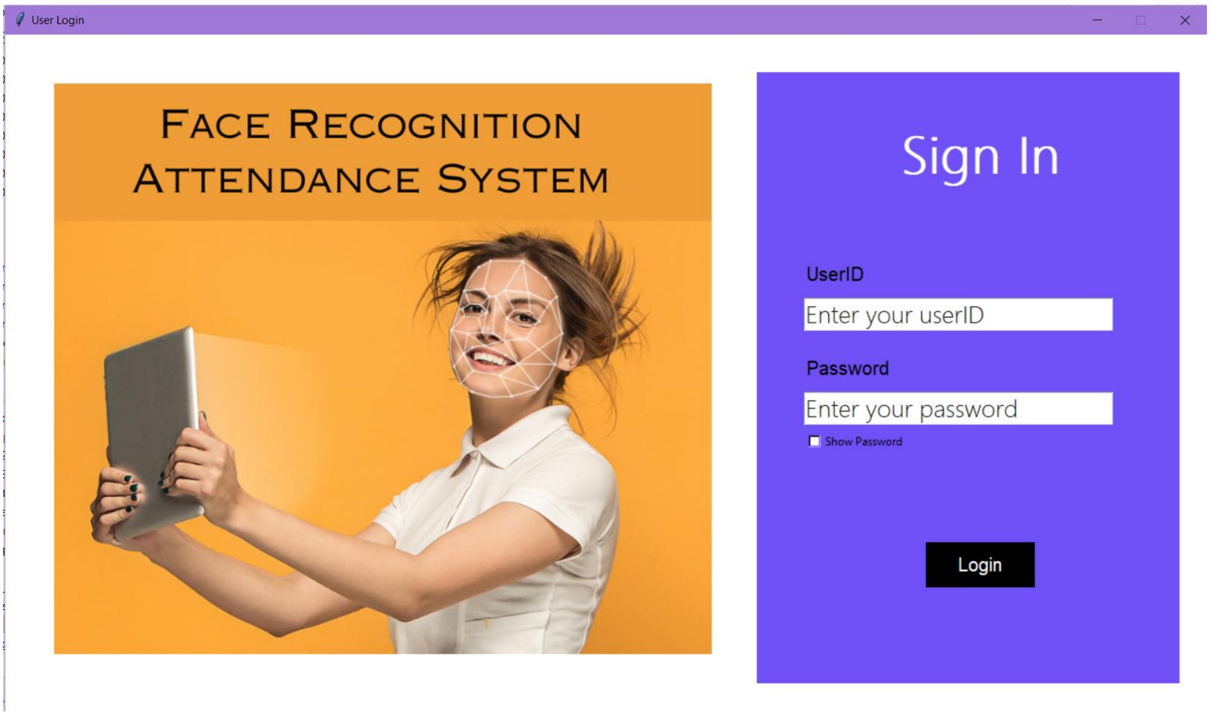


Figure 5.4.1: Login page

This Administrator Login Page is a pivotal gateway to the secure backend of the system, offering exclusive access to individuals with administrative privileges. Administrators are required to provide valid credentials consisting of a unique User ID and a confidential Password. The User ID serves as a crucial identifier, ensuring that only authorized personnel can log in, and passwords typically adhere to stringent security standards, including complexity requirements. Upon entering their credentials, administrators can initiate the login process by clicking the "Submit" button. Additional features often include a 'Show Password' checkbox as an option for displaying the entered password.

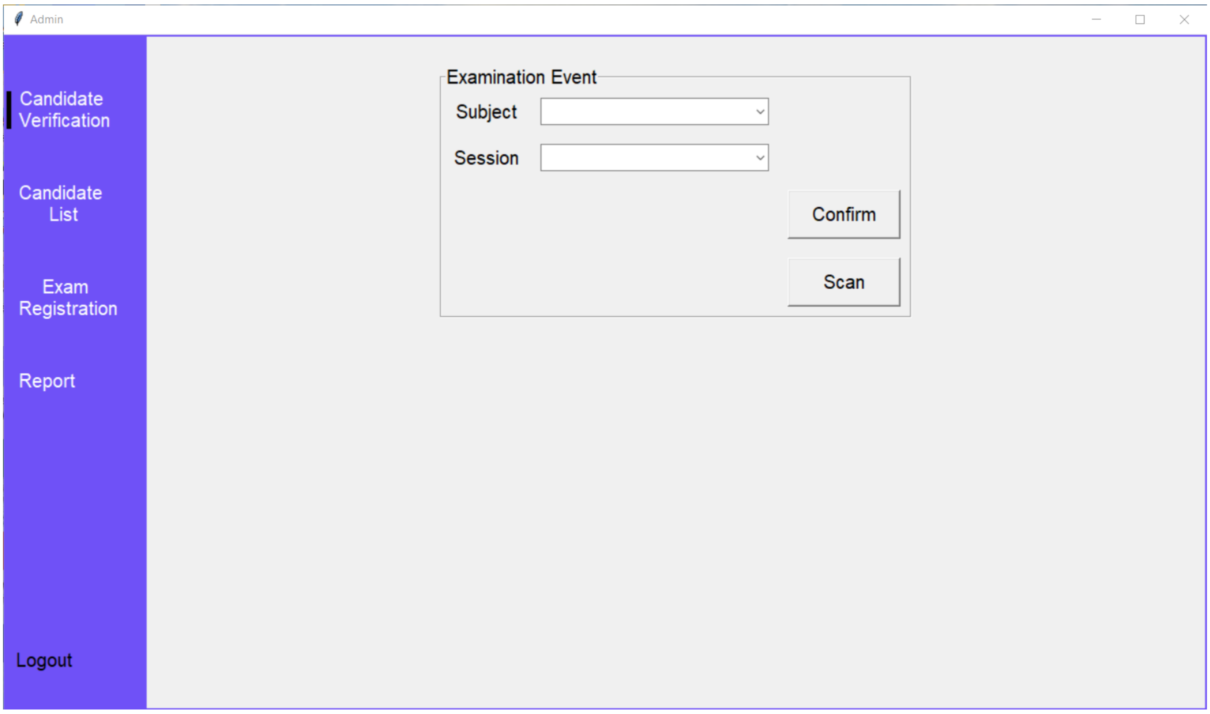
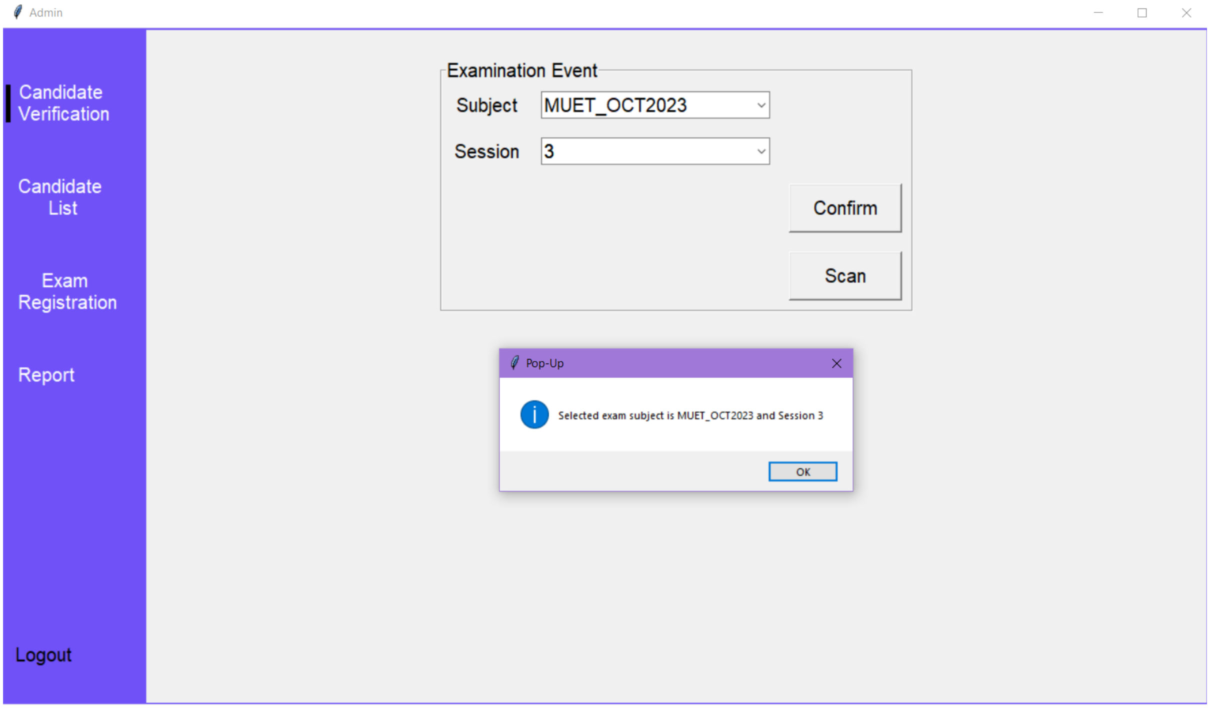


Figure 5.4.2: Main navigation page

After successfully logging in, administrators will be directed to an admin page designed with tab navigation. The main tabs prominently featured are 'Candidate Verification,' 'Candidate List,' 'Exam Registration,' and 'Report.' The image above showcases the 'Candidate Verification' page, which serves as a central hub for managing candidate attendance. To initiate the attendance-taking process, administrators can start by selecting a subject and its corresponding session. Once the selection is confirmed, they can proceed by clicking the 'Confirm' button and then 'Scan' to initiate the scanning process. This streamlined workflow allows administrators to efficiently manage attendance and ensure the smooth operation of the verification process.



Selected exam subject is MUET_OCT2023 and Session 3

Figure 5.4.3: Candidate verification page (with select subject and session)

The image above is an example of subject and session confirmation, followed by a notification popup to inform the user.

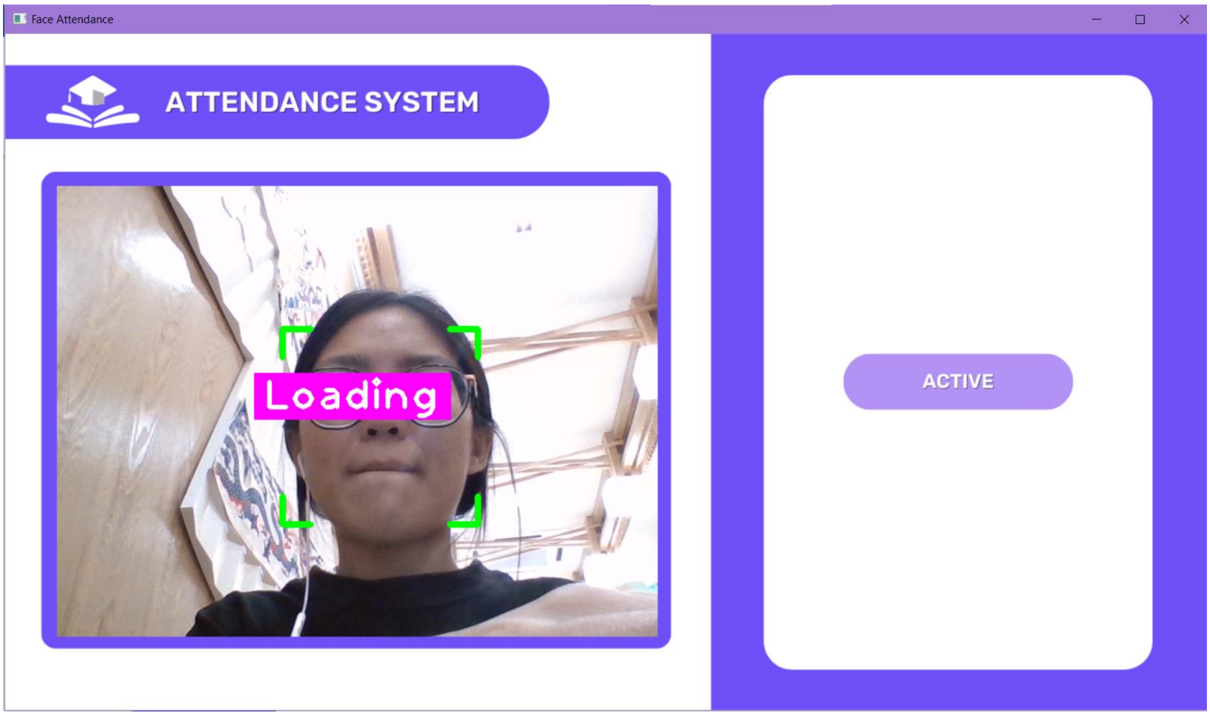


Figure 5.4.4: Scanning page (active)

After clicking the 'Scan' button, an 'Active' scanning page is displayed, signifying that the camera is active and prepared for candidate verification. During the process of detecting and recognizing the candidate's face, a 'Loading' message will be displayed alongside a green box that precisely identifies and allocates the candidate's face.

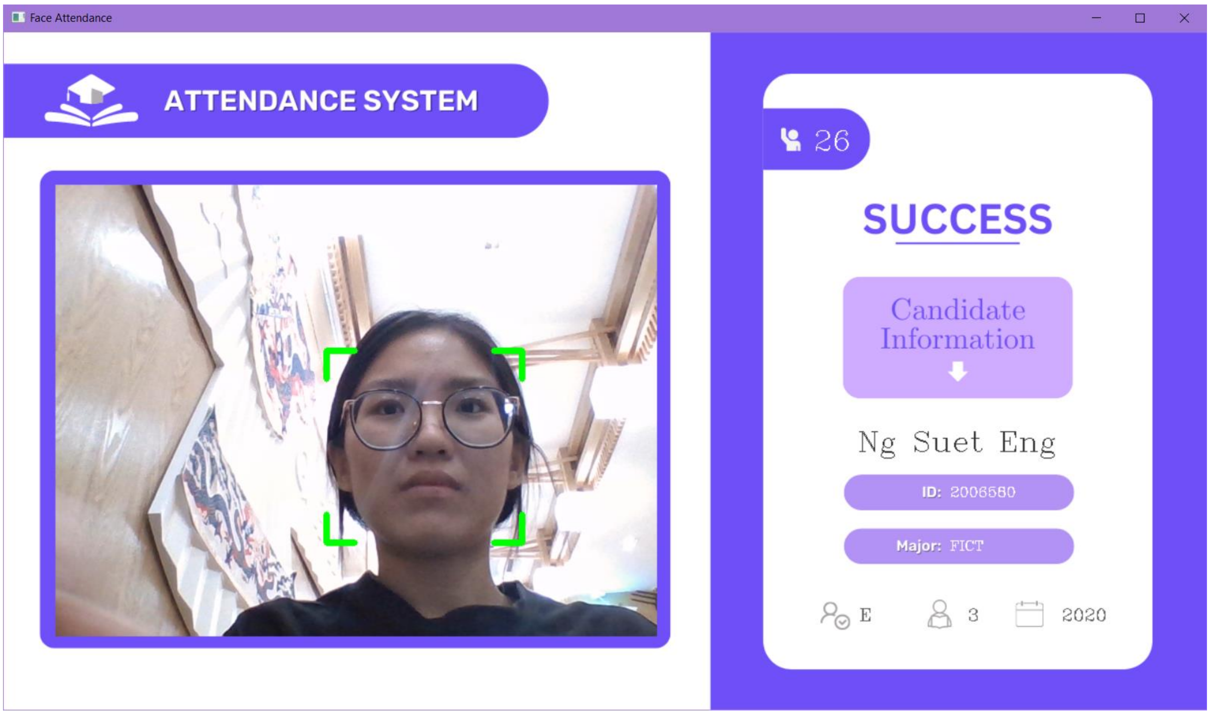


Figure 5.4.5: Scanning page (success)

After a successful recognition, the indication page will prominently display comprehensive candidate information, including the candidate's name, ID, major, status, attendance count, and the intake year.

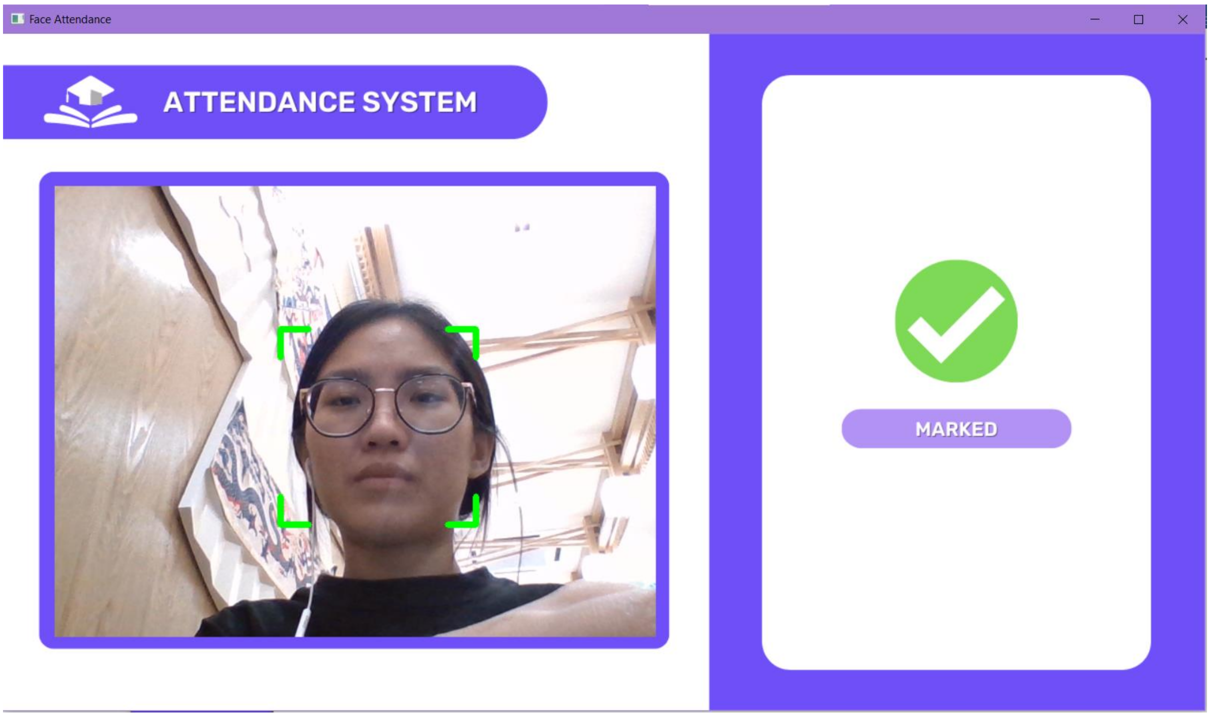


Figure 5.4.6: Scanning page (marked)

When the current attendance record is timestamped and updated in Firebase, the indication page will promptly display a 'Marked' status, signaling the successful completion of the recording process.

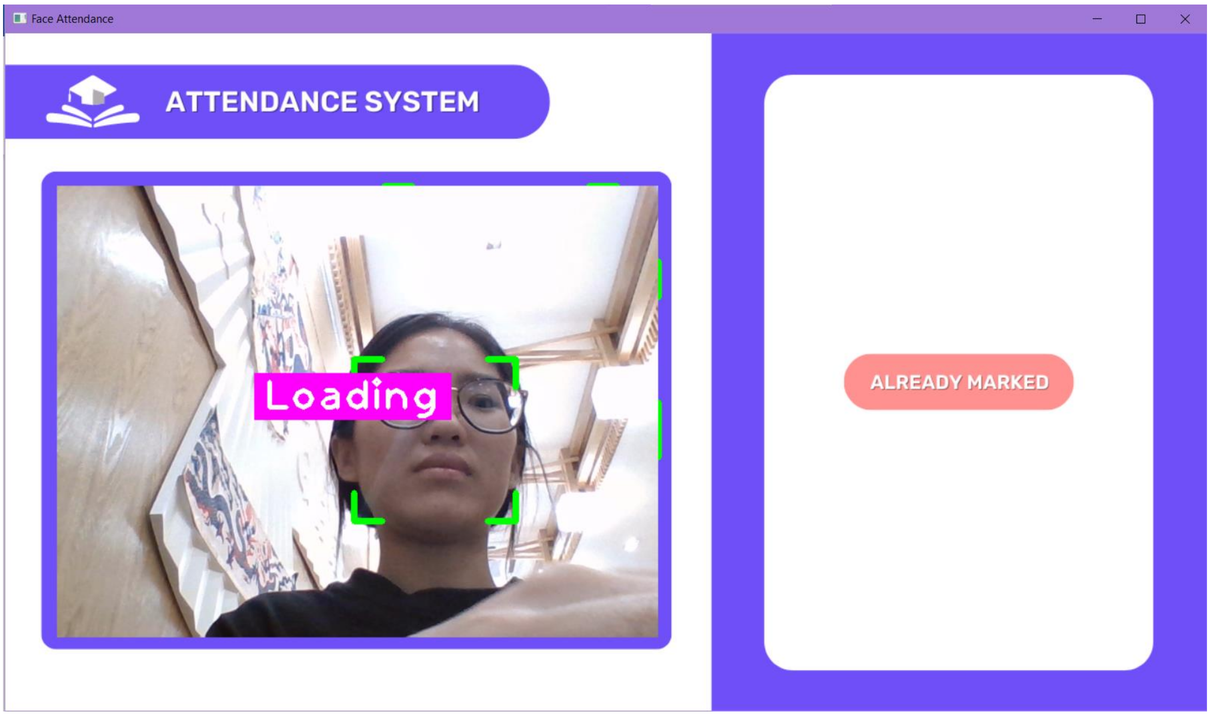
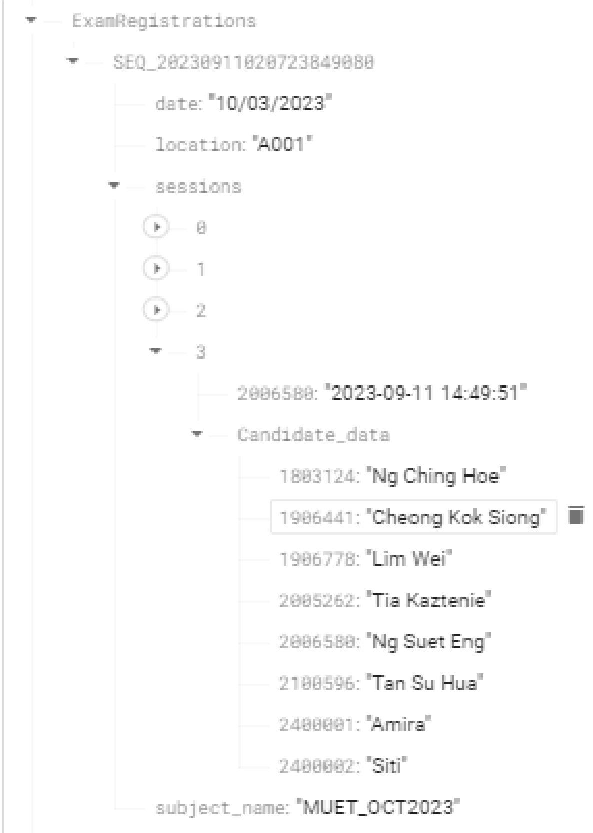


Figure 5.4.7: Scanning page (already marked)

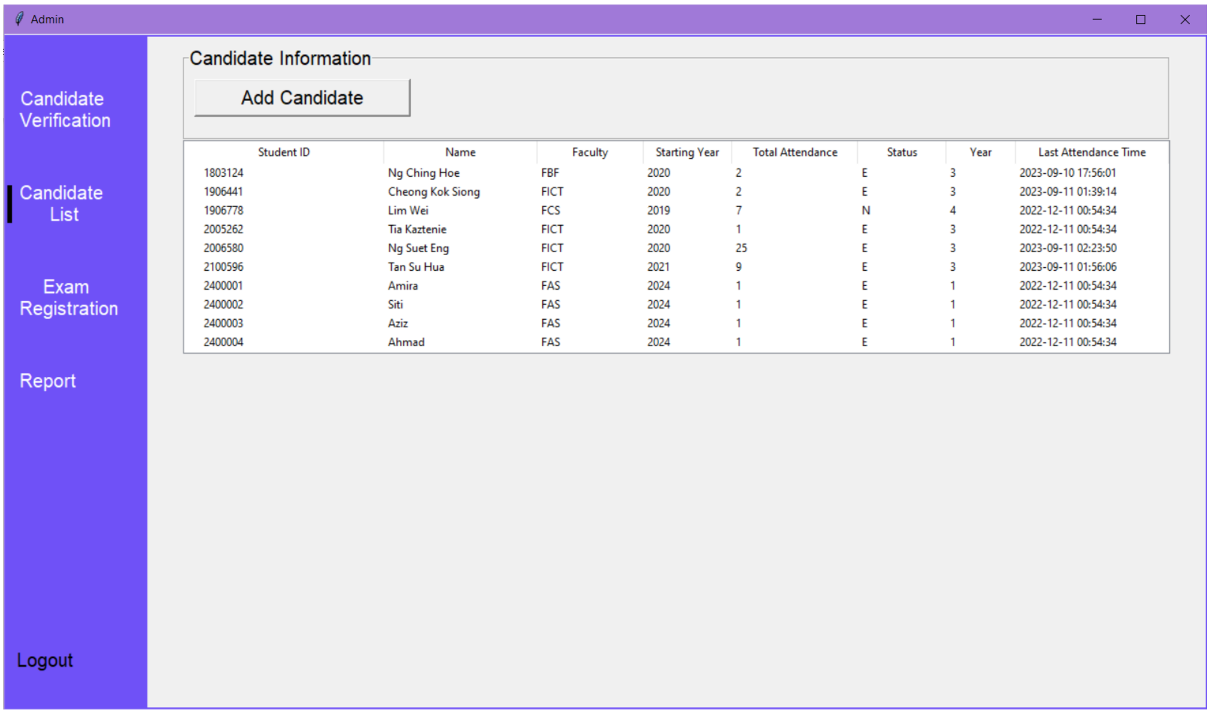
If a candidate remains within the camera frame for 30 seconds after their attendance has been recorded, the system will intelligently display an 'Already Marked' indication to prevent duplicate attendance records and ensure the accuracy of the recording process.



```
ExamRegistrations
  SEQ_20230911020723849080
    date: "10/03/2023"
    location: "A001"
    sessions
      0
      1
      2
      3
        2006580: "2023-09-11 14:49:51"
        Candidate_data
          1803124: "Ng Ching Hoe"
          1906441: "Cheong Kok Siong"
          1906778: "Lim Wei"
          2005262: "Tia Kaztenie"
          2006580: "Ng Suet Eng"
          2100596: "Tan Su Hua"
          2400001: "Amira"
          2400002: "Siti"
    subject_name: "MUET_OCT2023"
```

Figure 5.4.8: Firebase record (already marked)

After successfully capturing and updating the attendance data in Firebase, the diagram above provides a clear visualization of the recorded attendance time, paired with the candidate's unique ID, within the context of the previously selected subject and session.



Student ID	Name	Faculty	Starting Year	Total Attendance	Status	Year	Last Attendance Time
1803124	Ng Ching Hoe	FBF	2020	2	E	3	2023-09-10 17:56:01
1906441	Cheong Kok Siong	FICT	2020	2	E	3	2023-09-11 01:39:14
1906778	Lim Wei	FCS	2019	7	N	4	2022-12-11 00:54:34
2005262	Tia Kaztenie	FICT	2020	1	E	3	2022-12-11 00:54:34
2006580	Ng Suet Eng	FICT	2020	25	E	3	2023-09-11 02:23:50
2100596	Tan Su Hua	FICT	2021	9	E	3	2023-09-11 01:56:06
2400001	Amira	FAS	2024	1	E	1	2022-12-11 00:54:34
2400002	Siti	FAS	2024	1	E	1	2022-12-11 00:54:34
2400003	Aziz	FAS	2024	1	E	1	2022-12-11 00:54:34
2400004	Ahmad	FAS	2024	1	E	1	2022-12-11 00:54:34

Figure 5.4.9: Candidate list page

The second tab within the administrator's page presents a comprehensive table listing candidate information. The displayed fields encompass Candidate ID, Name, Faculty, Intake Year, Total Attendance, Status, Current Academic Year, and the most recent attendance timestamp.

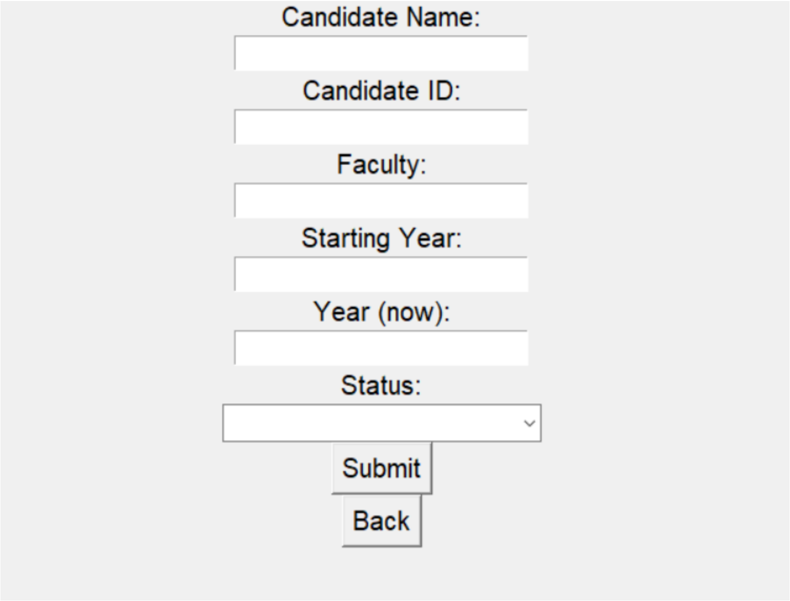
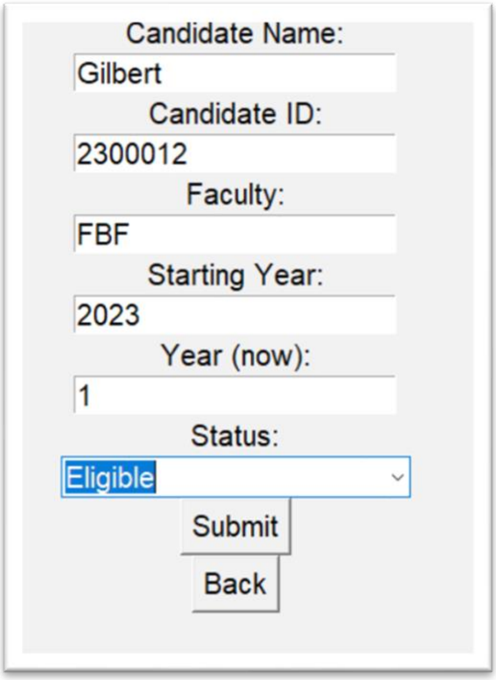
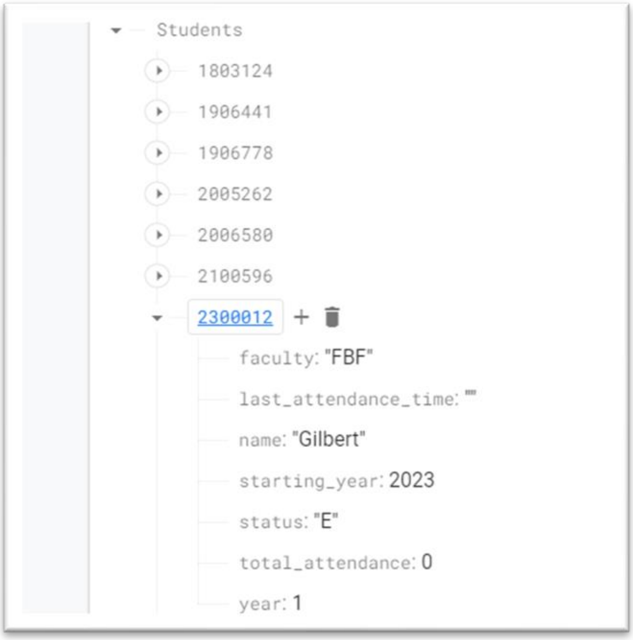


Figure 5.4.10: Add new candidate

When an administrator wishes to add a new candidate to the list, they can simply click the 'Add Candidate' button located at the top right corner. This action triggers the appearance of an entry form, enabling the administrator to input the new candidate's information. Upon confirming the entered details, the administrator can proceed by clicking the 'Submit' button to upload the data to Firebase Cloud Storage. Should the administrator wish to continue adding new candidates, they can do so seamlessly as the form is automatically cleared after a successful submission. Alternatively, the administrator can click the 'Back' button to exit the candidate addition interface, returning to the candidate list view.



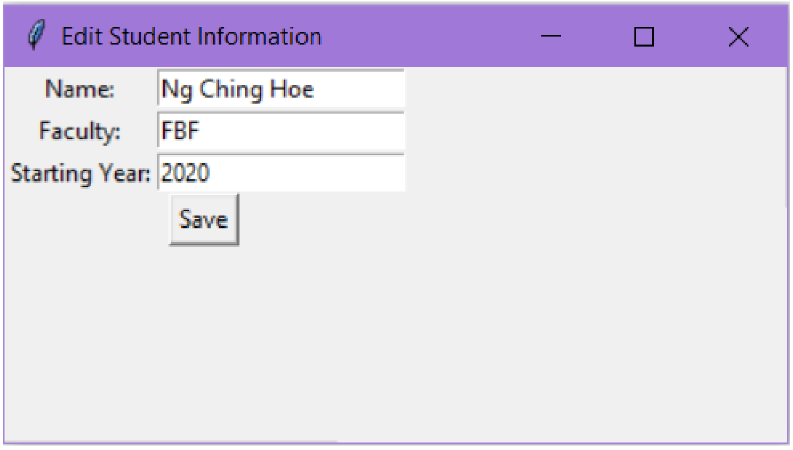
Candidate Name: Gilbert
Candidate ID: 2300012
Faculty: FBF
Starting Year: 2023
Year (now): 1
Status: Eligible



```
Students
  1803124
  1906441
  1906778
  2005262
  2006580
  2100596
  2300012
    faculty: "FBF"
    last_attendance_time: ""
    name: "Gilbert"
    starting_year: 2023
    status: "E"
    total_attendance: 0
    year: 1
```

Figure 5.4.11: Add new candidate (sample) **Figure 5.4.12: Firebase record(sample)**

Here is an illustrative example of the addition of a new candidate named 'Gilbert' to the database. Upon submission, the information seamlessly updates in the database within the 'Student' directory, as depicted in the diagram above. It's worth noting that the 'Last Attendance Time' and 'Total Attendance' fields will initially be null for this newly added candidate, reflecting their recent inclusion in the system.



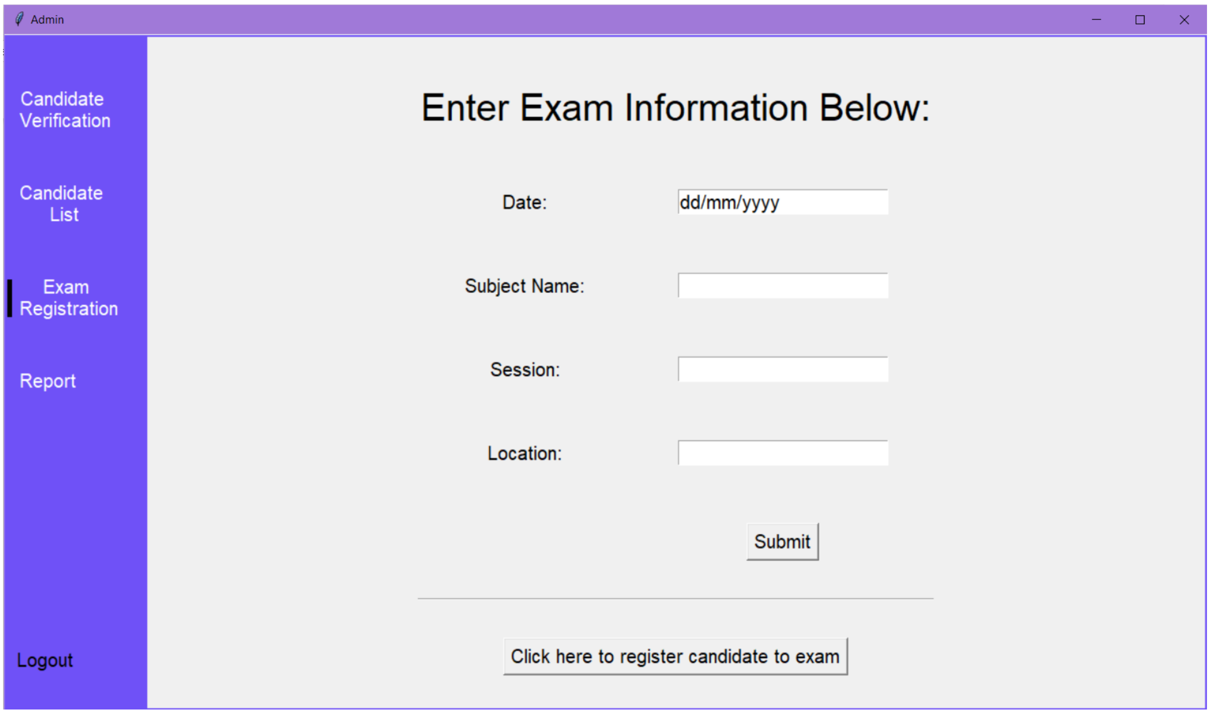
Name: Ng Ching Hoe

Faculty: FBF

Starting Year: 2020

Figure 5.4.13: Edit Candidate Information

If administrators wish to edit candidate information, this can be accomplished by double-clicking the candidate's record within the table list. However, it's important to note that certain fields, particularly confidential information such as the candidate's ID, cannot be edited for security reasons. This ensures the integrity and privacy of sensitive data.



Enter Exam Information Below:

Date: dd/mm/yyyy

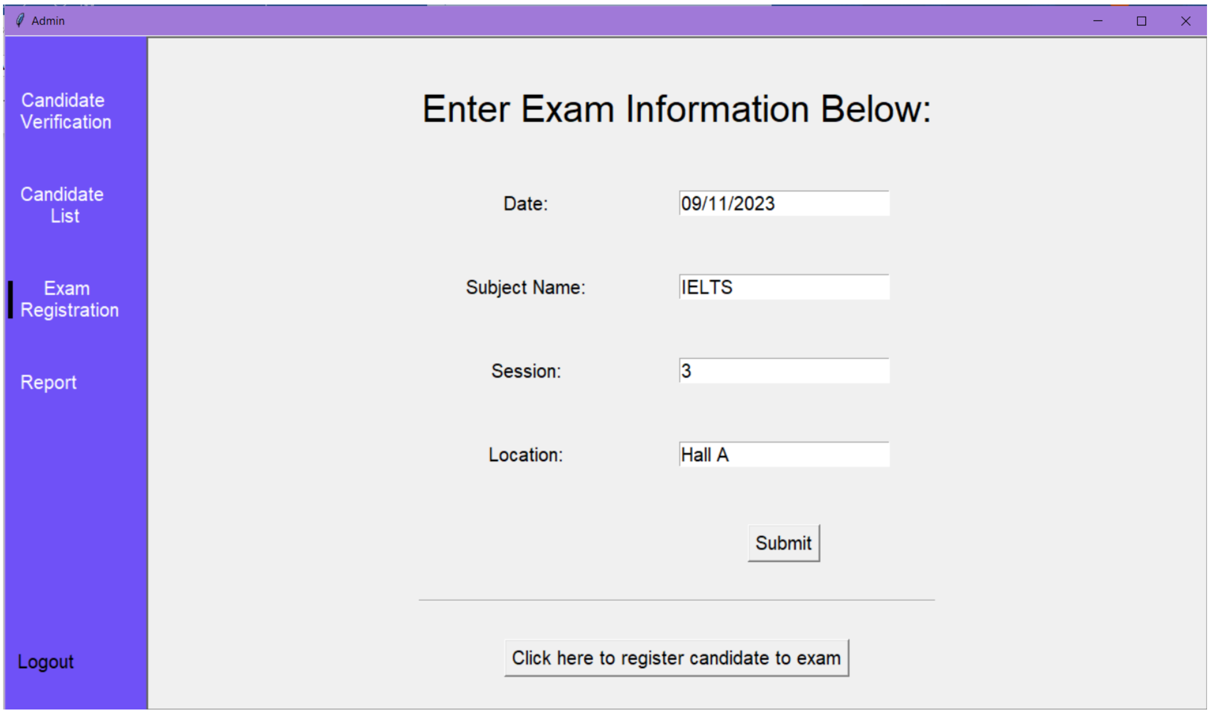
Subject Name:

Session:

Location:

Figure 5.4.14: Exam registration page

Moving on to the 'Exam Registration' tab, its purpose is to facilitate the addition of new exam subjects to the system. To achieve this, administrators are required to input crucial details such as the exam date, subject name, number of sessions, and the exam location. Once all the information is confirmed, administrators can finalize the process by clicking the 'Submit' button, effectively uploading the data to Firebase Cloud Storage.



Enter Exam Information Below:

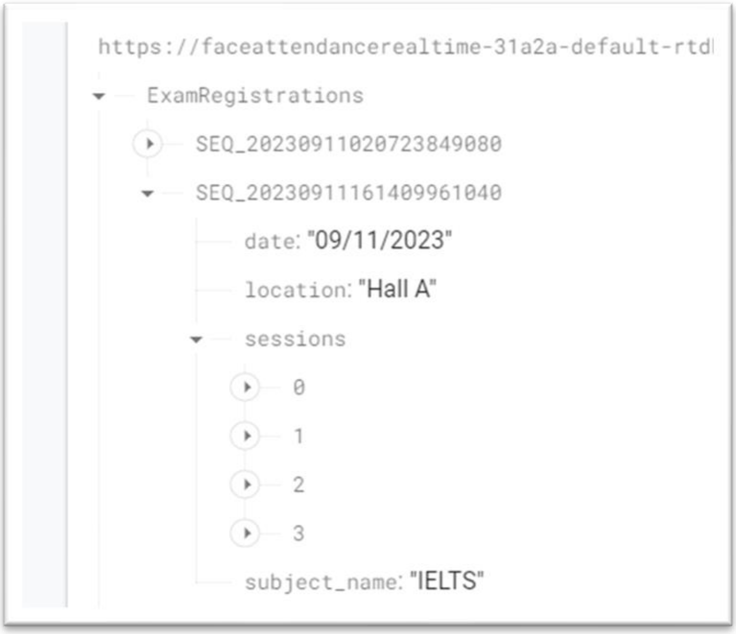
Date: 09/11/2023

Subject Name: IELTS

Session: 3

Location: Hall A

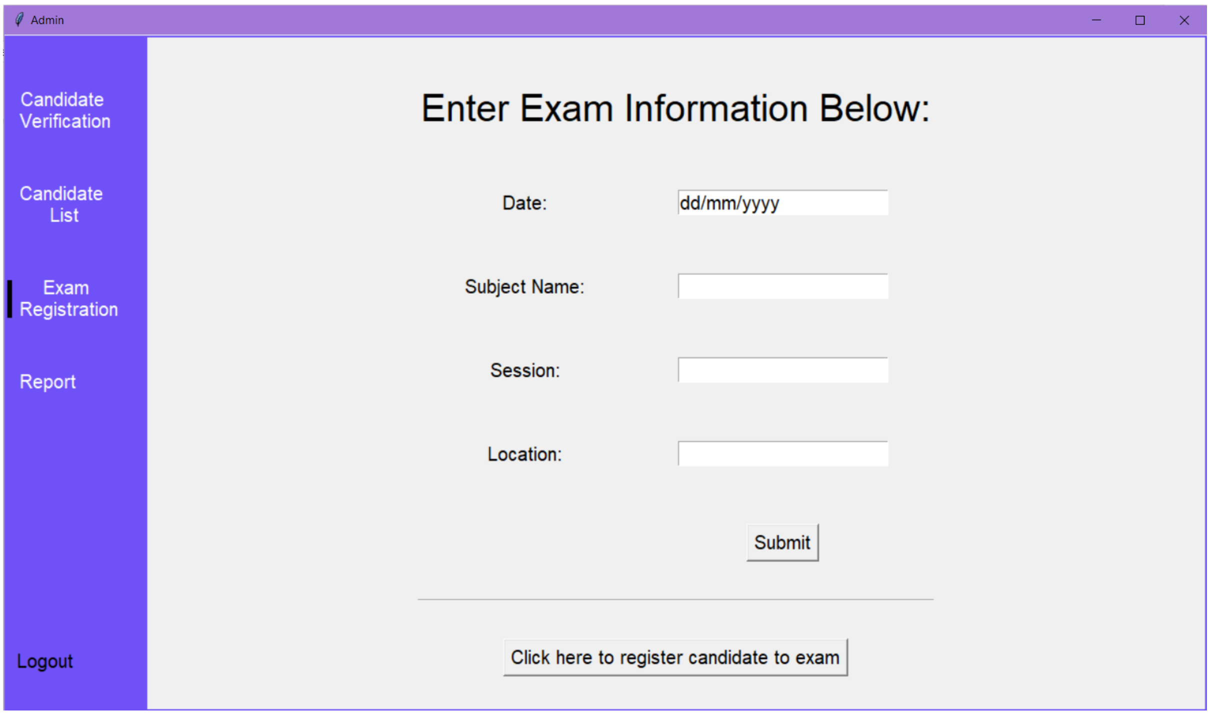
Figure 5.4.15: Exam registration (sample)



```
https://faceattendancerealtime-31a2a-default-rtdl
ExamRegistrations
    SEQ_20230911020723849080
    SEQ_20230911161409961040
        date: "09/11/2023"
        location: "Hall A"
        sessions
            0
            1
            2
            3
        subject_name: "IELTS"
```

Figure 5.4.16: Firebase record (exam registration)

Here's an example of the registration of a new exam subject. The registered exam subject is assigned a unique ID within Firebase, ensuring that the information remains free from duplication. Since there are 3 registered sessions, the child nodes generated under the session will extend up to '3'. However, it's important to note that, due to the indexing format used for session numbers, session '0' will be excluded during the data retrieval process.



Enter Exam Information Below:

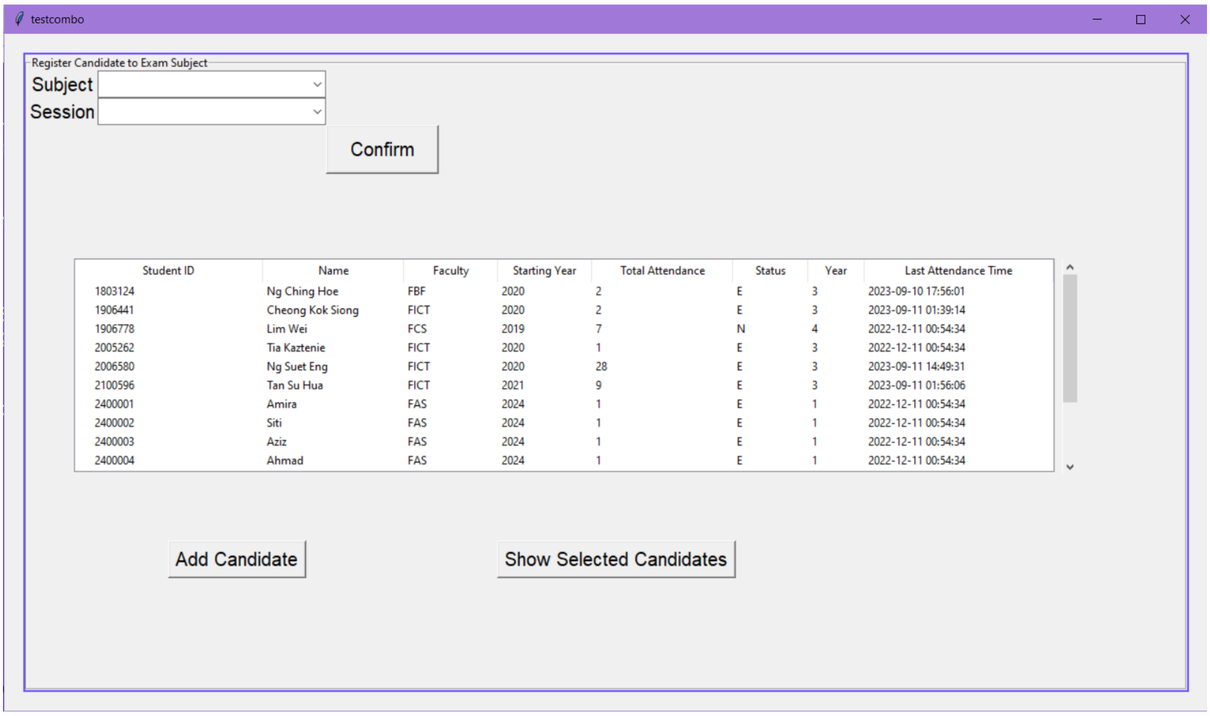
Date: dd/mm/yyyy

Subject Name:

Session:

Location:

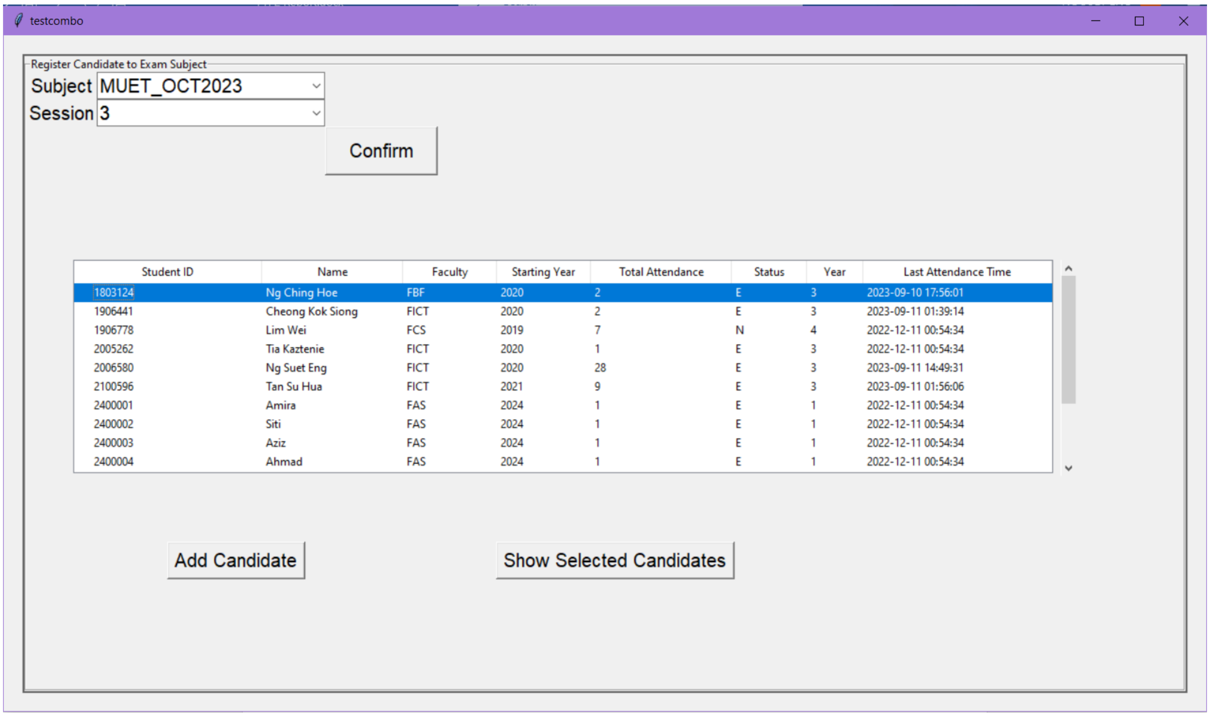
Figure 5.4.17: Exam registration page



Student ID	Name	Faculty	Starting Year	Total Attendance	Status	Year	Last Attendance Time
1803124	Ng Ching Hoe	FBF	2020	2	E	3	2023-09-10 17:56:01
1906441	Cheong Kok Siong	FICT	2020	2	E	3	2023-09-11 01:39:14
1906778	Lim Wei	FCS	2019	7	N	4	2022-12-11 00:54:34
2005262	Tia Kaztenie	FICT	2020	1	E	3	2022-12-11 00:54:34
2006580	Ng Suet Eng	FICT	2020	28	E	3	2023-09-11 14:49:31
2100596	Tan Su Hua	FICT	2021	9	E	3	2023-09-11 01:56:06
2400001	Amira	FAS	2024	1	E	1	2022-12-11 00:54:34
2400002	Siti	FAS	2024	1	E	1	2022-12-11 00:54:34
2400003	Aziz	FAS	2024	1	E	1	2022-12-11 00:54:34
2400004	Ahmad	FAS	2024	1	E	1	2022-12-11 00:54:34

Figure 5.4.18: Register candidate to exam subject

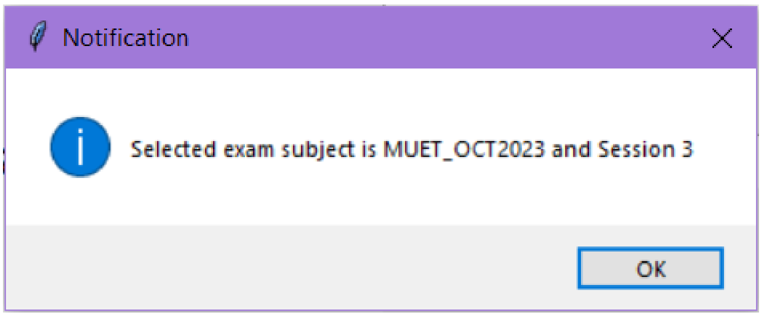
To register candidates for various subjects, administrators can initiate the process by clicking the button located at the bottom of the 'Exam Registration' page. This action will lead them to the 'Register Candidate to Exam Subject' page, where they are required to select the desired subject and session before adding the candidate to the respective examination subject.



Subject MUET_OCT2023
Session 3

Student ID	Name	Faculty	Starting Year	Total Attendance	Status	Year	Last Attendance Time
1803124	Ng Ching Hoe	FBF	2020	2	E	3	2023-09-10 17:56:01
1906441	Cheong Kok Siong	FICT	2020	2	E	3	2023-09-11 01:39:14
1906778	Lim Wei	FCS	2019	7	N	4	2022-12-11 00:54:34
2005262	Tia Kaztenie	FICT	2020	1	E	3	2022-12-11 00:54:34
2006580	Ng Suet Eng	FICT	2020	28	E	3	2023-09-11 14:49:31
2100596	Tan Su Hua	FICT	2021	9	E	3	2023-09-11 01:56:06
2400001	Amira	FAS	2024	1	E	1	2022-12-11 00:54:34
2400002	Siti	FAS	2024	1	E	1	2022-12-11 00:54:34
2400003	Aziz	FAS	2024	1	E	1	2022-12-11 00:54:34
2400004	Ahmad	FAS	2024	1	E	1	2022-12-11 00:54:34

Figure 5.4.19: Register candidate to exam subject (sample)

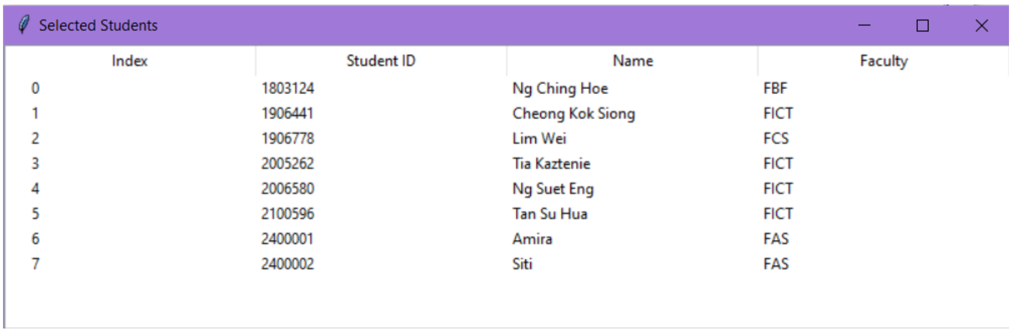


Selected exam subject is MUET_OCT2023 and Session 3

Figure 5.4.20: Notification

Upon selecting the subject and its corresponding session, it is imperative to proceed by clicking the 'Confirm' button to ensure that the candidate is correctly added to the intended subject and session. Following this confirmation, a notification will promptly appear, clearly indicating the selected subject and session for added clarity and reassurance.

Next, administrators should select the candidate from the table list and then click the 'Add Candidate' button to register the candidate for the chosen subject and session. To access the list of added candidates, administrators can simply click on the 'Show Selected Candidates' option, prompting a window to display the registered candidates for convenient viewing.

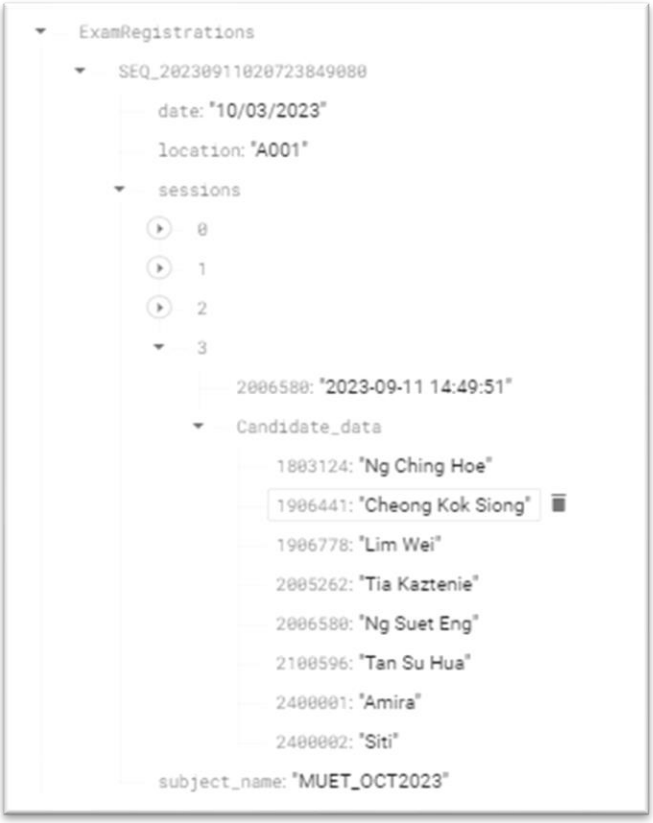


Selected Students

Index	Student ID	Name	Faculty
0	1803124	Ng Ching Hoe	FBF
1	1906441	Cheong Kok Siong	FICT
2	1906778	Lim Wei	FCS
3	2005262	Tia Kaztenie	FICT
4	2006580	Ng Suet Eng	FICT
5	2100596	Tan Su Hua	FICT
6	2400001	Amira	FAS
7	2400002	Siti	FAS

Figure 5.4.21: Show selected candidates

The image above provides an illustrative example of the candidate list following the registration of specific students to the chosen subject and session by the administrator.



```
ExamRegistrations
    SEQ_20230911020723849080
        date: "10/03/2023"
        location: "A001"
        sessions
            0
            1
            2
            3
                2006580: "2023-09-11 14:49:51"
                Candidate_data
                    1803124: "Ng Ching Hoe"
                    1906441: "Cheong Kok Siong"
                    1906778: "Lim Wei"
                    2005262: "Tia Kaztenie"
                    2006580: "Ng Suet Eng"
                    2100596: "Tan Su Hua"
                    2400001: "Amira"
                    2400002: "Siti"
        subject_name: "MUET_OCT2023"
```

Figure 5.4.22: Firebase record (selected candidates)

Simultaneously, this illustrates the storage of data in Firebase Cloud Storage in the form of a nested dictionary format, a structure chosen for enhanced ease of access and management by the backend system when viewing the candidate list.

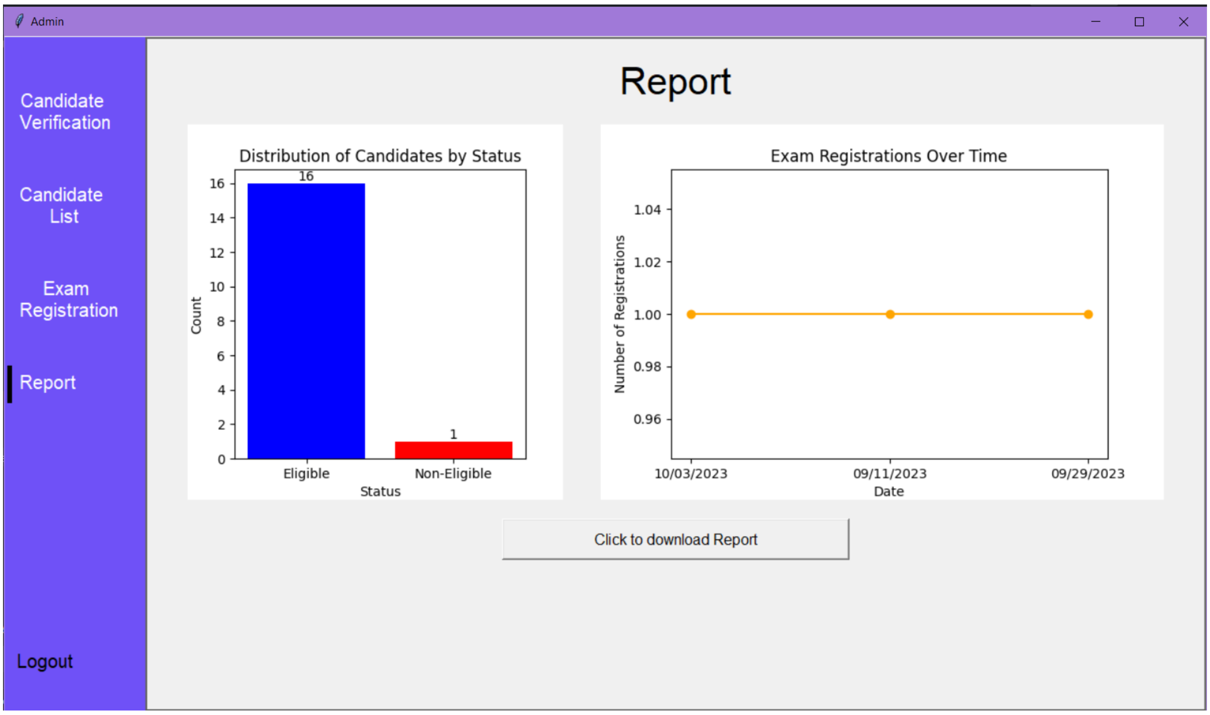
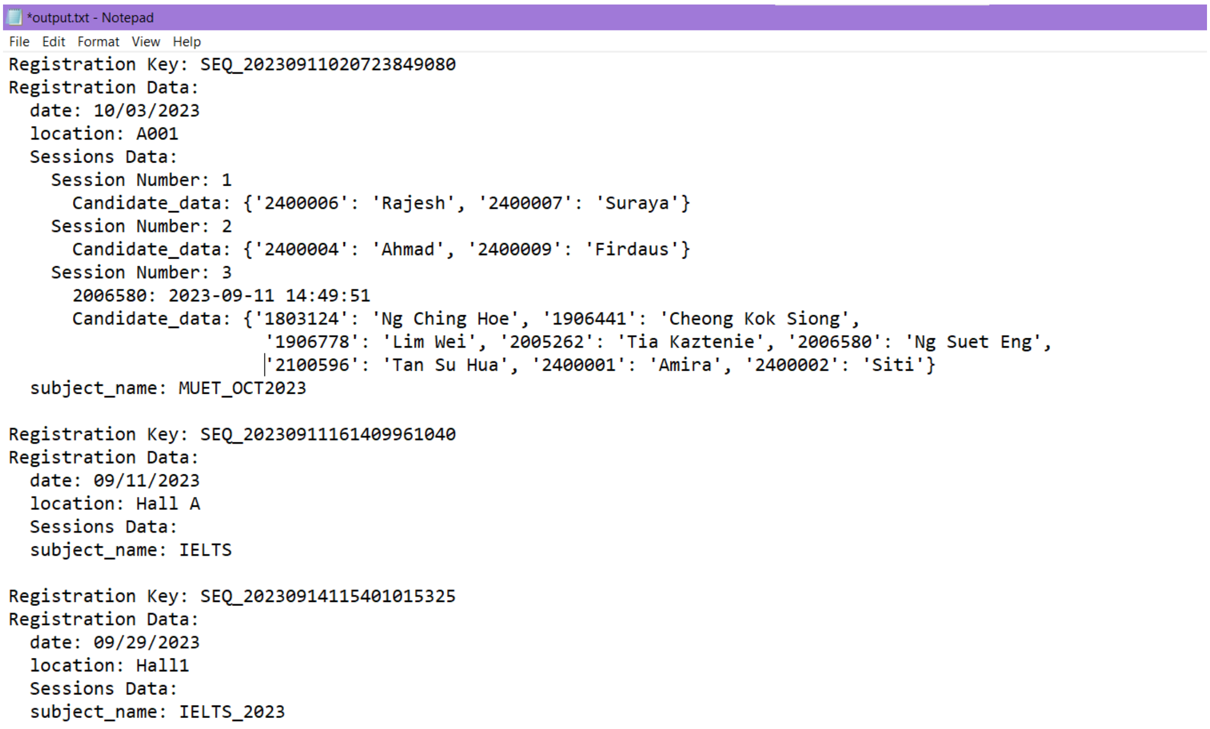


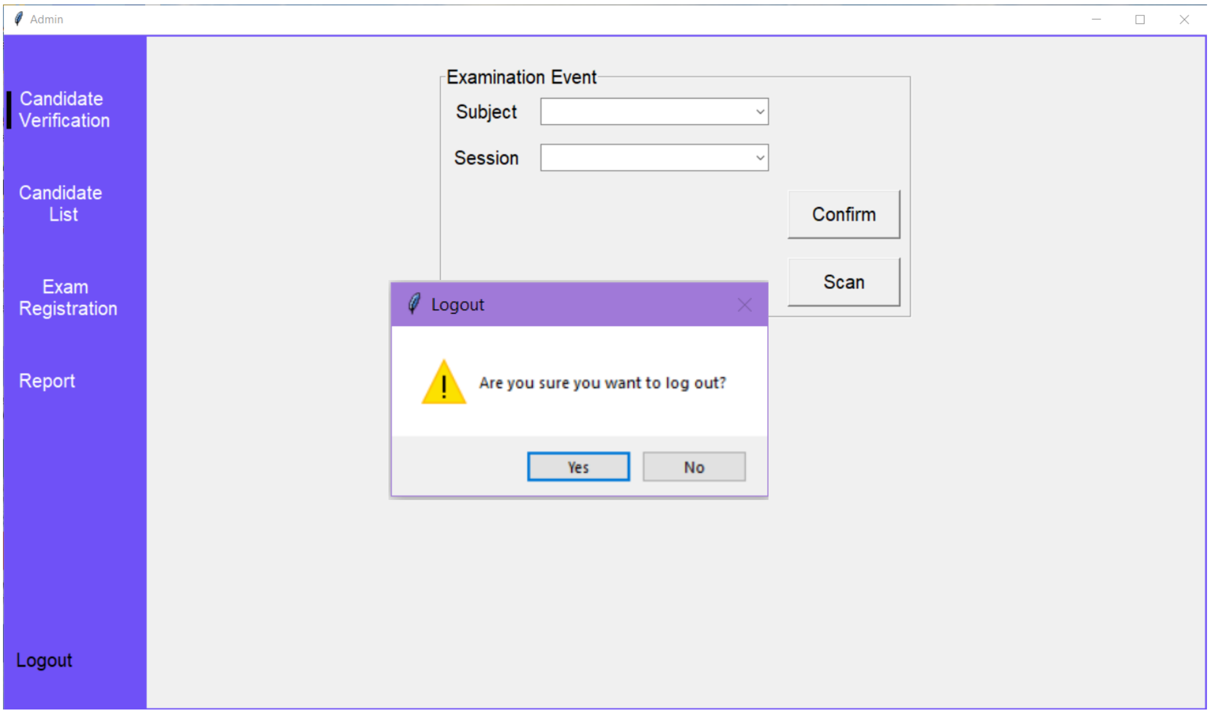
Figure 5.4.23: Report Page



```
Registration Key: SEQ_20230911020723849080
Registration Data:
  date: 10/03/2023
  location: A001
  Sessions Data:
    Session Number: 1
      Candidate_data: {'2400006': 'Rajesh', '2400007': 'Suraya'}
    Session Number: 2
      Candidate_data: {'2400004': 'Ahmad', '2400009': 'Firdaus'}
    Session Number: 3
      2006580: 2023-09-11 14:49:51
      Candidate_data: {'1803124': 'Ng Ching Hoe', '1906441': 'Cheong Kok Siong',
                        '1906778': 'Lim Wei', '2005262': 'Tia Kaztenie', '2006580': 'Ng Suet Eng',
                        '2100596': 'Tan Su Hua', '2400001': 'Amira', '2400002': 'Siti'}
  subject_name: MUET_OCT2023

Registration Key: SEQ_20230911161409961040
Registration Data:
  date: 09/11/2023
  location: Hall A
  Sessions Data:
  subject_name: IELTS

Registration Key: SEQ_20230914115401015325
Registration Data:
  date: 09/29/2023
  location: Hall1
  Sessions Data:
  subject_name: IELTS_2023
```

Figure 5.4.24: Report

In the 'Report' page, the admin is able to view a bar chart displaying the counts of 'Eligible' and 'Non-Eligible' candidates and a line graph that illustrate the number of exam subject registered over time. Subsequently, they can opt to download the report by clicking on the button located beneath the chart. The generated report will encompass all the exam information stored in real-time Firebase under 'ExamRegistrations,' as depicted in Figure 5.4.23.



Are you sure you want to log out?

Figure 5.4.25: Logout confirmation

If administrators no longer require access to the system's functions, they can initiate the logout process by clicking the 'Logout' option located at the bottom of the navigation bar. Subsequently, a confirmation window will appear, prompting administrators to confirm their intent to log out.

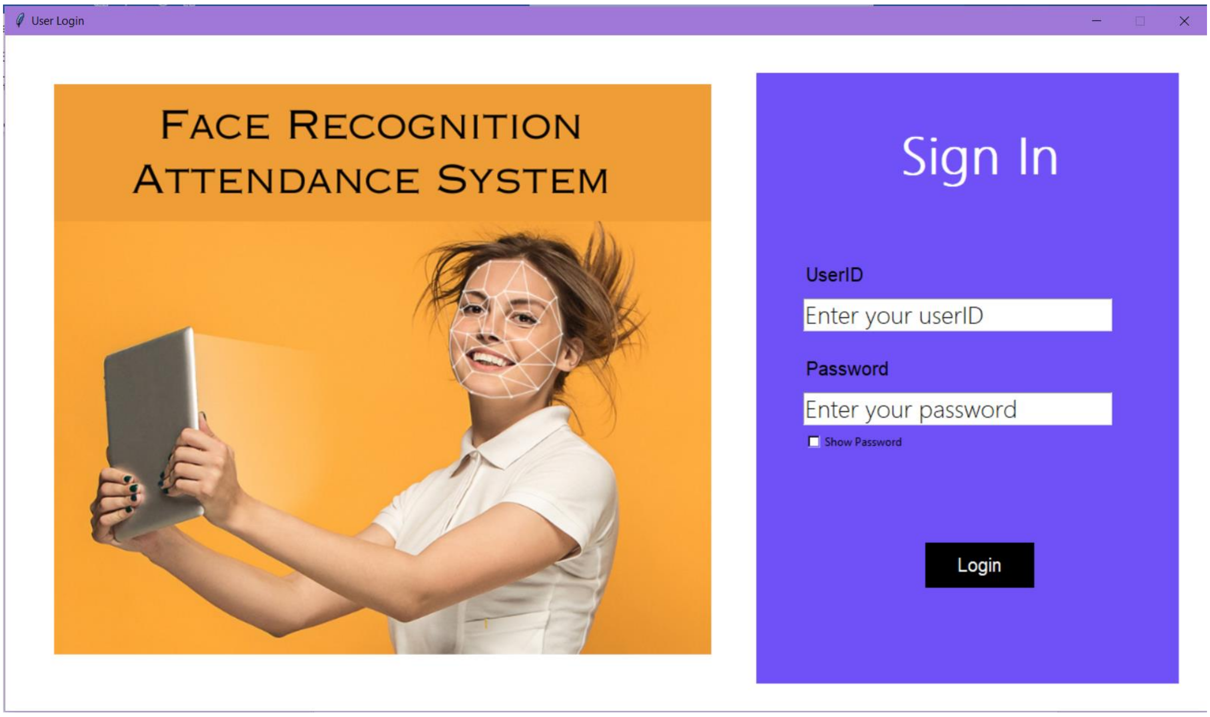


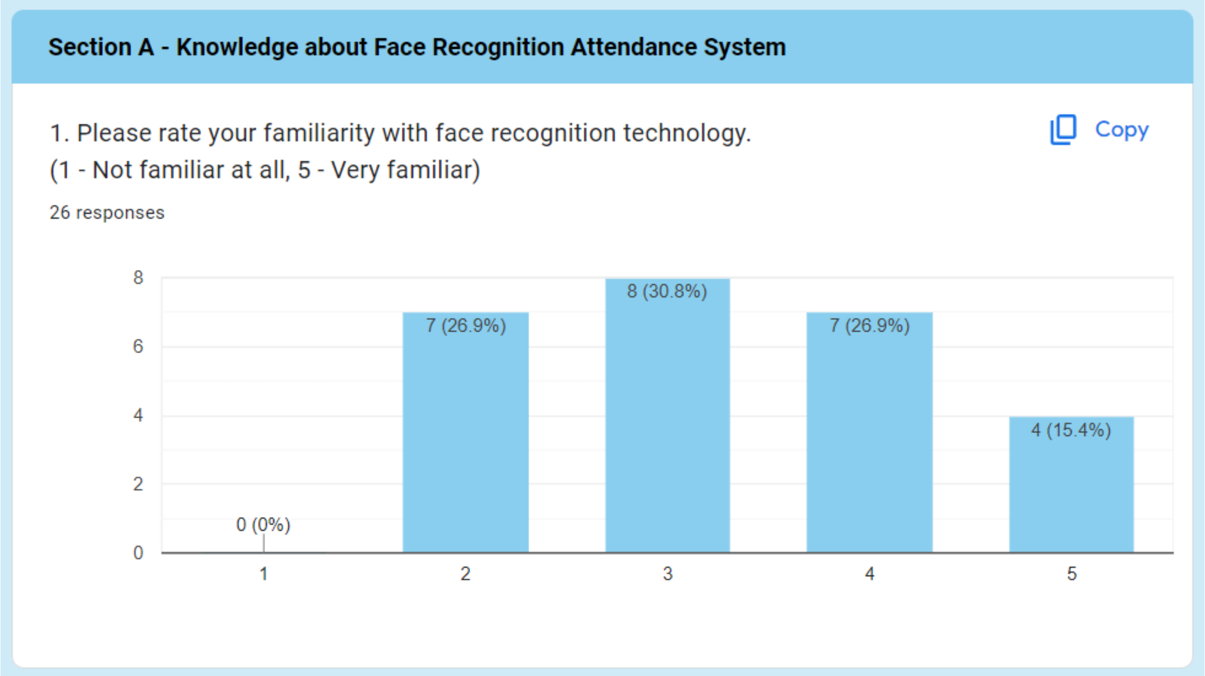
Figure 5.4.26: Login page

Once confirm the logout, it will close the admin window and automatically redirect to the admin login page.

Chapter 6

System Evaluation and Discussion

6.1 System Evaluation



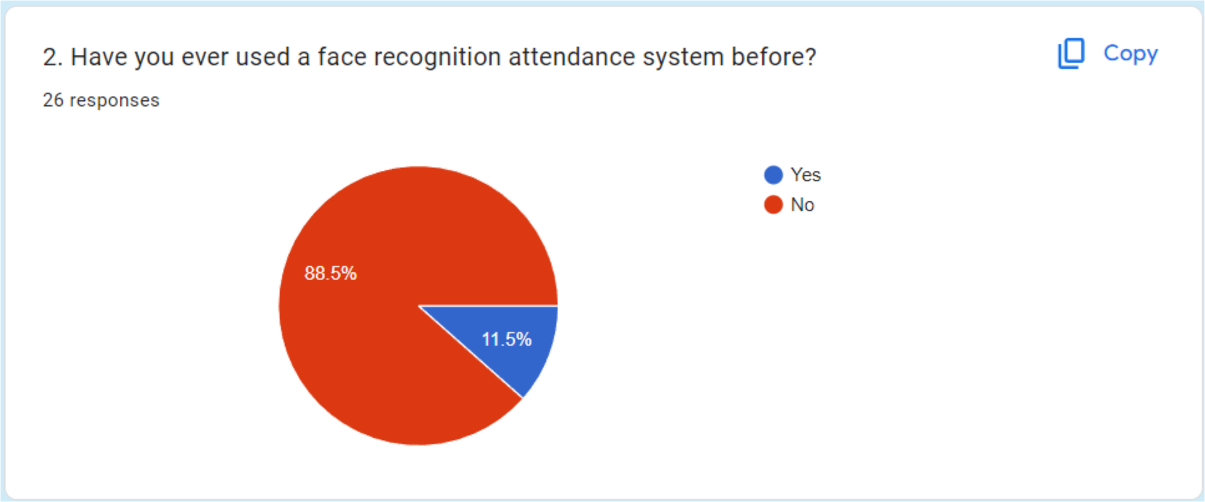
Section A - Knowledge about Face Recognition Attendance System

1. Please rate your familiarity with face recognition technology.
(1 - Not familiar at all, 5 - Very familiar)

26 responses

Figure 6.1.1: Survey Question 1

The initial question aims to gauge respondents' familiarity with facial recognition technology. The responses indicate that out of the 26 participants, 8 of them display a moderate level of familiarity, categorized as level 3. Interestingly, an equal number of respondents fall into both level 2 (indicating a limited familiarity) and level 4 (representing a moderate familiarity). Additionally, a smaller percentage, specifically 15.4% of the respondents (equivalent to 4 individuals), express a high level of familiarity, corresponding to level 5, signifying that they consider themselves very familiar with this technology.

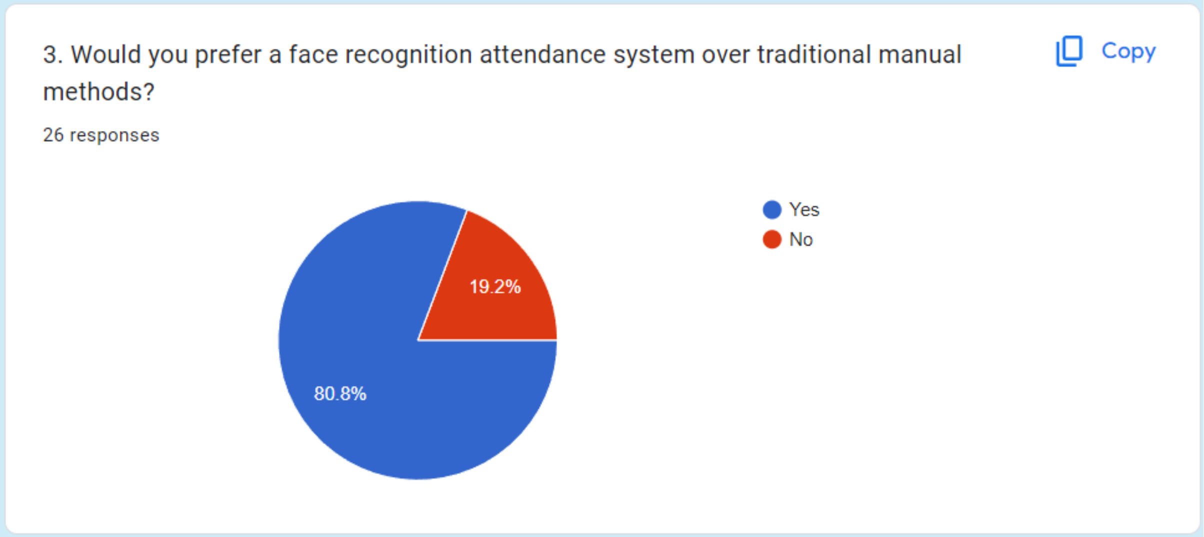


2. Have you ever used a face recognition attendance system before?

26 responses

Figure 6.1.2: Survey Question 2

For the second question, the aim was to ascertain whether respondents had any prior experience with a facial recognition attendance system. The accompanying pie chart illustrates that the majority of respondents, totaling 88.5%, had not encountered or interacted with such a system before. In contrast, a smaller group of respondents, comprising 11.5% of the total, reported that they had indeed tried a facial recognition attendance system at some point.

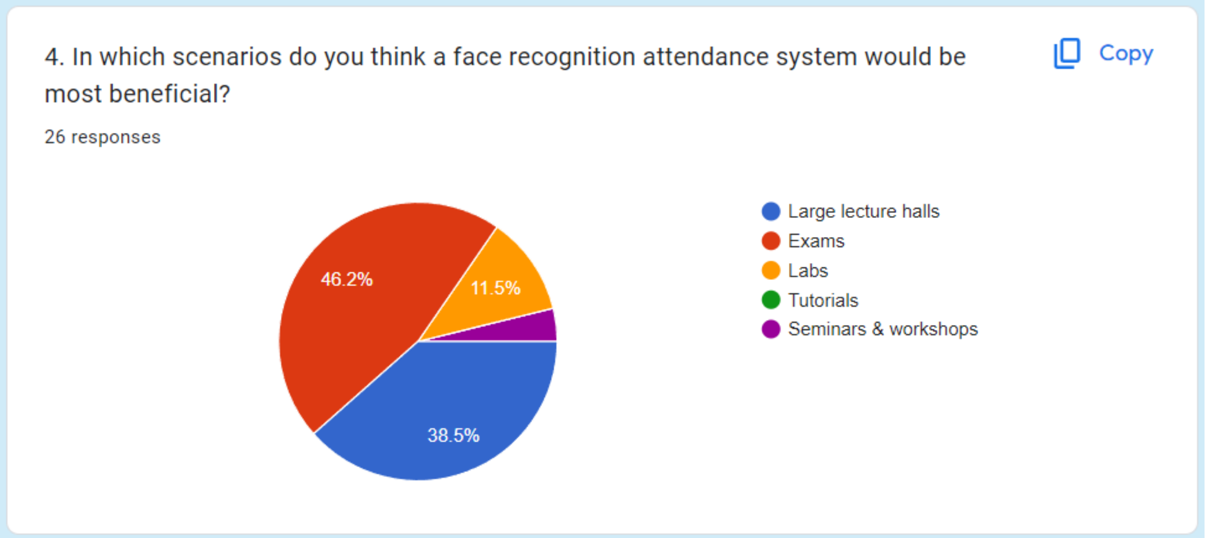


3. Would you prefer a face recognition attendance system over traditional manual methods?

26 responses

Figure 6.1.3: Survey Question 3

The third question aimed to determine respondents' preferences regarding the use of a facial recognition system versus the traditional manual attendance-taking method. Given that a significant portion of the respondents are university students, the pie chart above reveals that the majority, specifically 21 respondents, are open to trying the face recognition attendance system. However, it's worth noting that a smaller segment of the respondents still lean towards the traditional method, indicating that they are more comfortable with it.

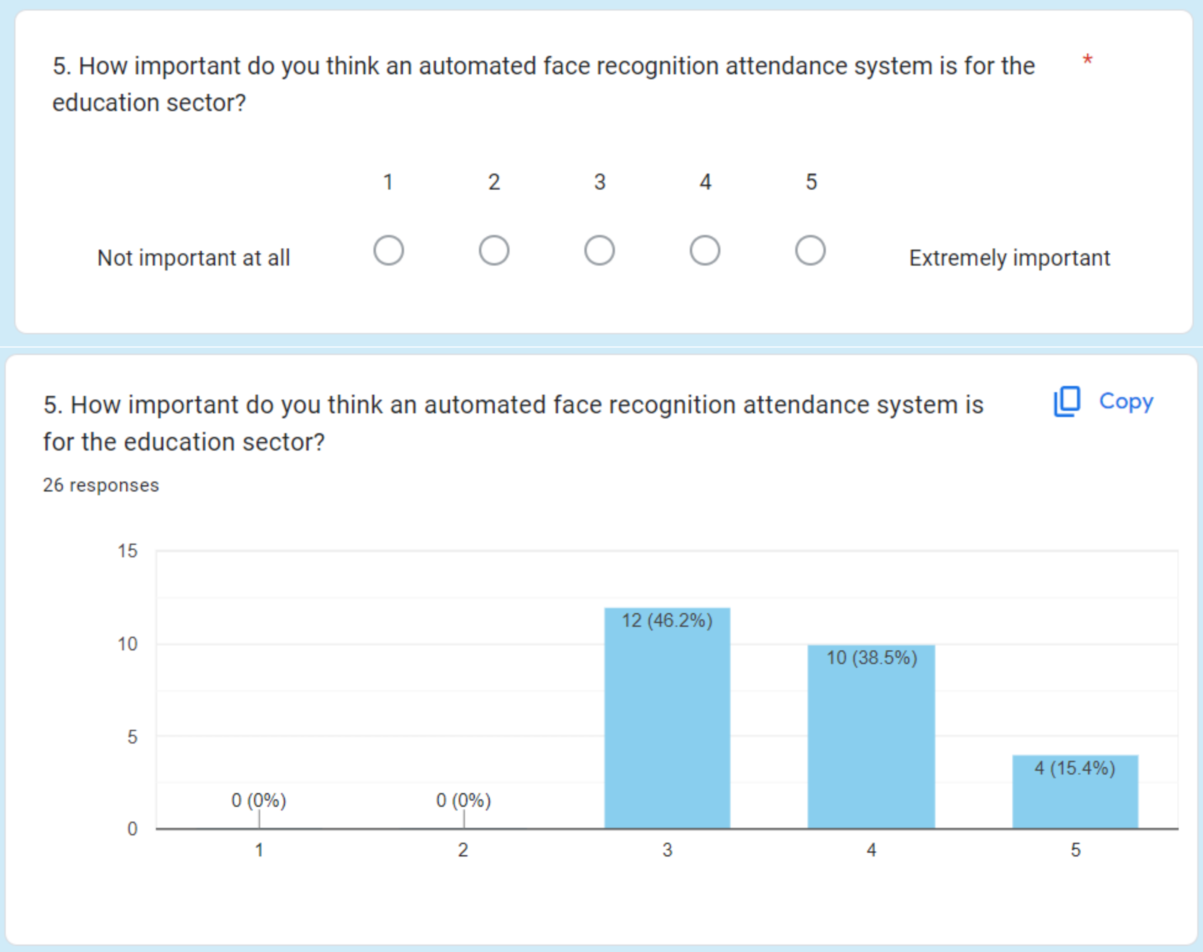


4. In which scenarios do you think a face recognition attendance system would be most beneficial?

26 responses

Figure 6.1.4: Survey Question 4

Next, the question sought to gather respondents' opinions on which scenarios they believe would be the most suitable for implementing a facial recognition attendance system. The results indicate that nearly half of the respondents (46.2%) consider exam scenarios to be the most appropriate for the system's deployment. Following closely, large lecture halls garnered support from 10 respondents, representing 38.5% of the responses. Additionally, 3 respondents expressed the view that the system is well-suited for laboratory environments, primarily due to safety considerations. On the other hand, only 1 respondent selected seminars and workshops as suitable scenarios, while no respondents identified tutorials as ideal settings for the implementation of this technology.



5. How important do you think an automated face recognition attendance system is for the education sector? *

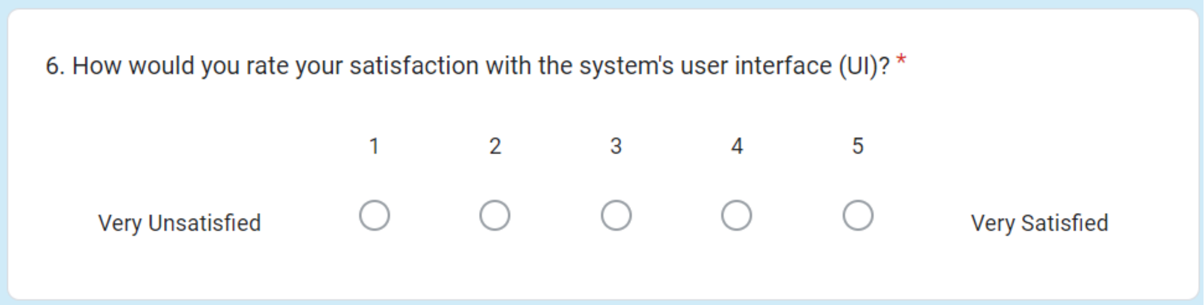
Not important at all ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 Extremely important

5. How important do you think an automated face recognition attendance system is for the education sector?

26 responses

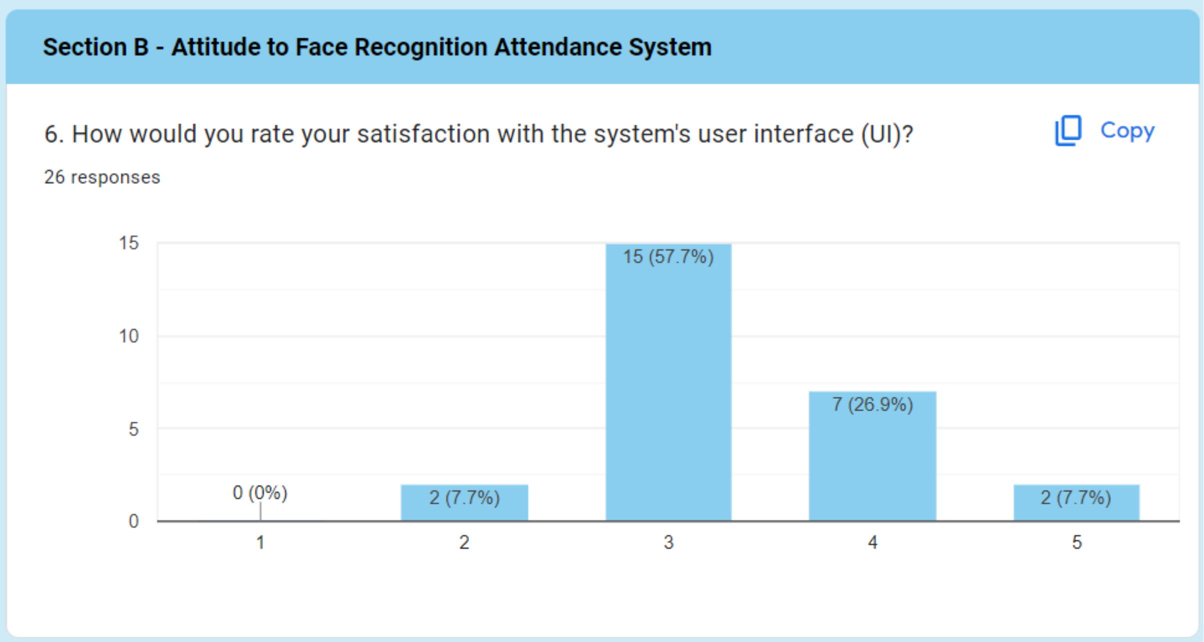
Figure 6.1.5: Survey Question 5

Moving on to question 5, the objective was to assess the perceived importance of an automated face recognition attendance system among respondents in the education sector. The findings reveal that all respondents fell within the spectrum ranging from 'neutral' to 'extremely important.' Notably, none of the respondents indicated that the system was 'not important.' Out of the 26 respondents, 5 of them strongly agreed that it is 'very important' to implement such a system within the educational sector. Furthermore, for those who expressed a 'neutral' stance and those who considered it 'important,' there was only a slight difference, with 10 respondents leaning towards importance and 12 holding a neutral opinion.



6. How would you rate your satisfaction with the system's user interface (UI)? *

	1	2	3	4	5	
Very Unsatisfied	○	○	○	○	○	Very Satisfied



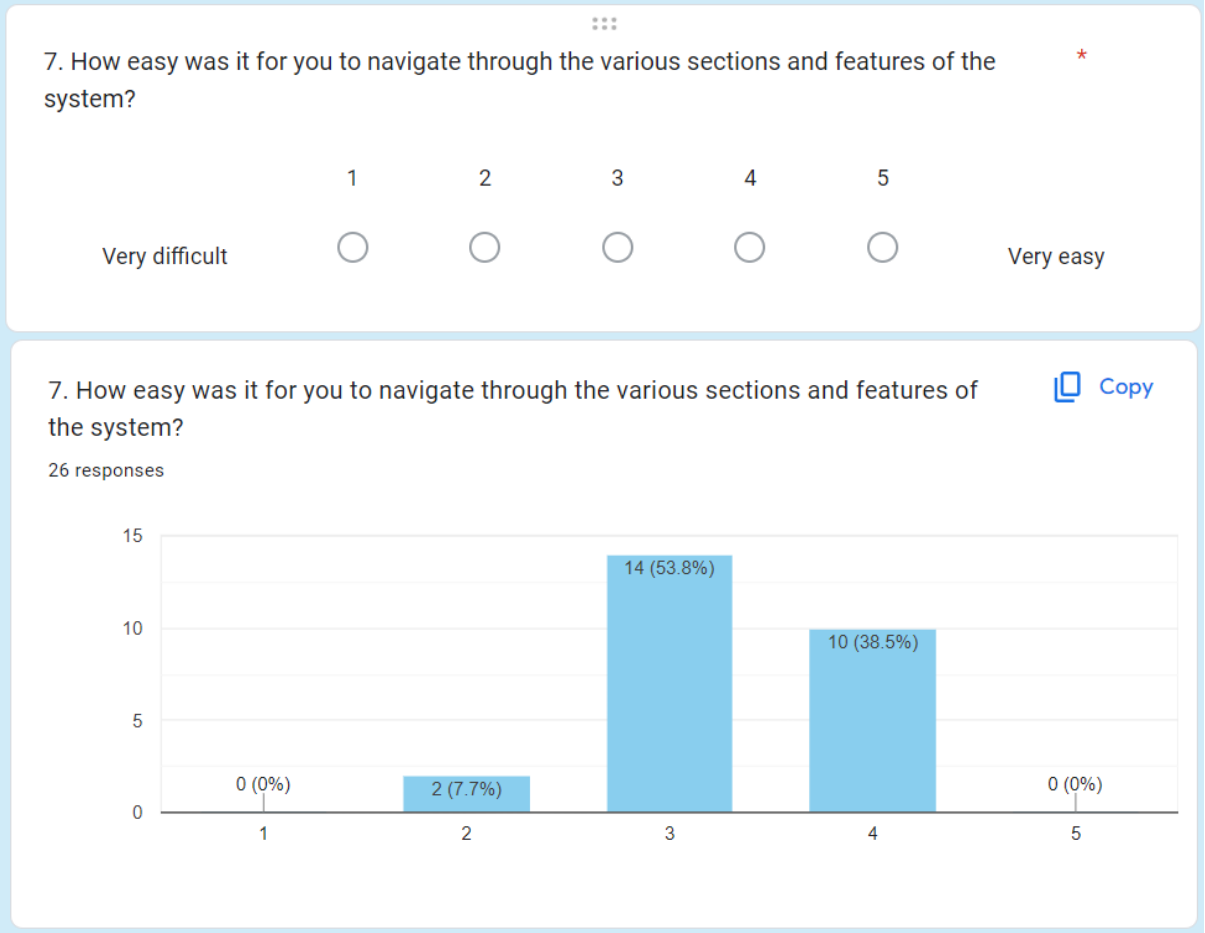
Section B - Attitude to Face Recognition Attendance System

6. How would you rate your satisfaction with the system's user interface (UI)?

26 responses

Figure 6.1.6: Survey Question 6

Now, let's delve into question 6, which initiates section B, aimed at assessing respondents' attitudes toward the developed system. This particular question seeks to gauge the level of satisfaction among respondents regarding the system. The bar chart provides a clear picture of the responses. Interestingly, the majority of respondents, comprising more than half of the total at 15 individuals, expressed a 'neutral' stance regarding the system's User Interface (UI). Following this, 7 respondents conveyed their satisfaction with the system's UI. Additionally, an equal number of respondents, accounting for 7.7% in each category, reported feeling both 'unsatisfied' and 'very satisfied' with the system's UI design.



7. How easy was it for you to navigate through the various sections and features of the system? *

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

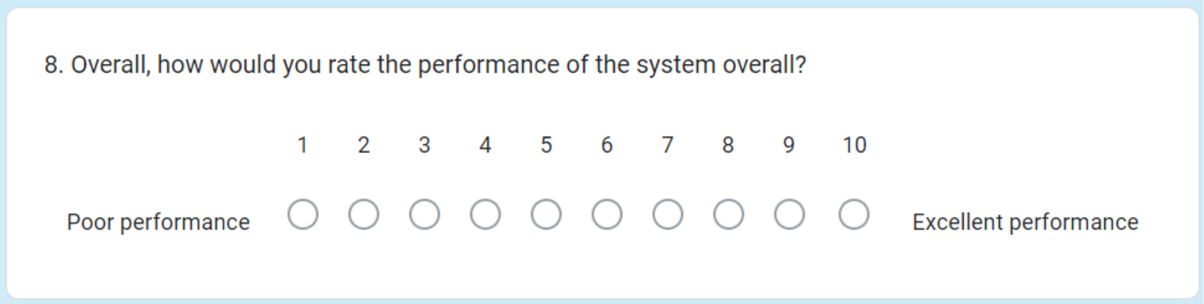
7. How easy was it for you to navigate through the various sections and features of the system?

26 responses

Figure 6.1.7: Survey Question 7

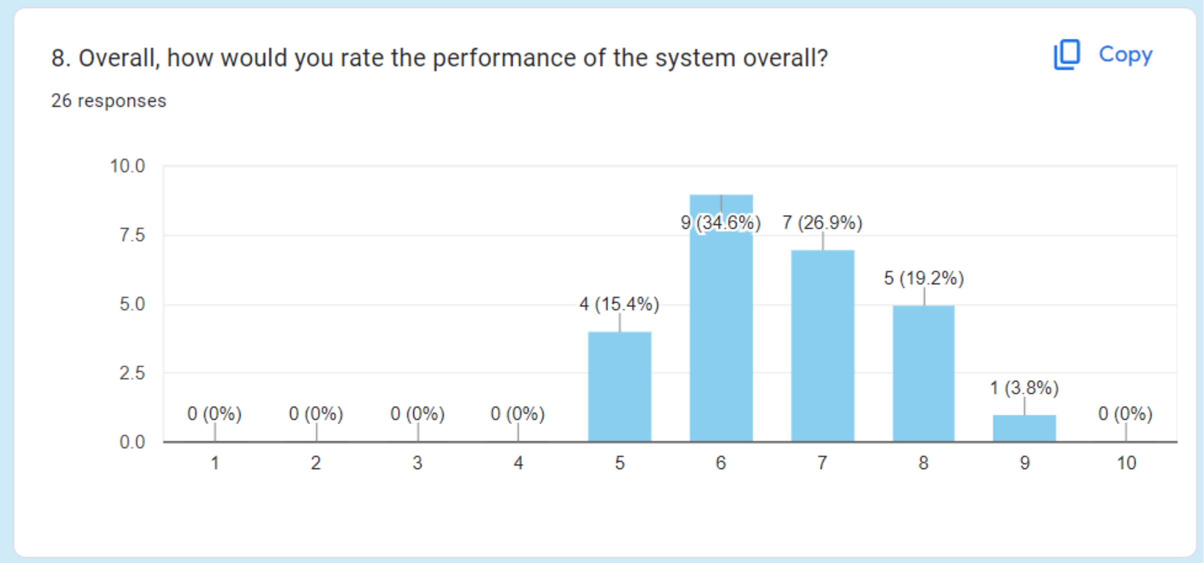
Moving on to the subsequent question in section B, the objective was to assess the ease of navigation within the system as perceived by the respondents. The results depict that a majority of respondents, comprising more than half, expressed a 'neutral' stance regarding the system's navigational ease, indicating that they found it neither very easy nor particularly difficult to operate. In contrast, a notable segment of respondents, totaling 38.5%, which corresponds to 10 individuals, reported that they found the system easy to navigate.

However, it's worth noting that 2 respondents indicated that they found the system difficult to navigate, suggesting some usability challenges for this subset of users. Interestingly, there were no respondents who perceived the system as 'easy' for them, indicating room for potential enhancements in user-friendliness.



8. Overall, how would you rate the performance of the system overall?

	1	2	3	4	5	6	7	8	9	10	
Poor performance	○	○	○	○	○	○	○	○	○	○	Excellent performance

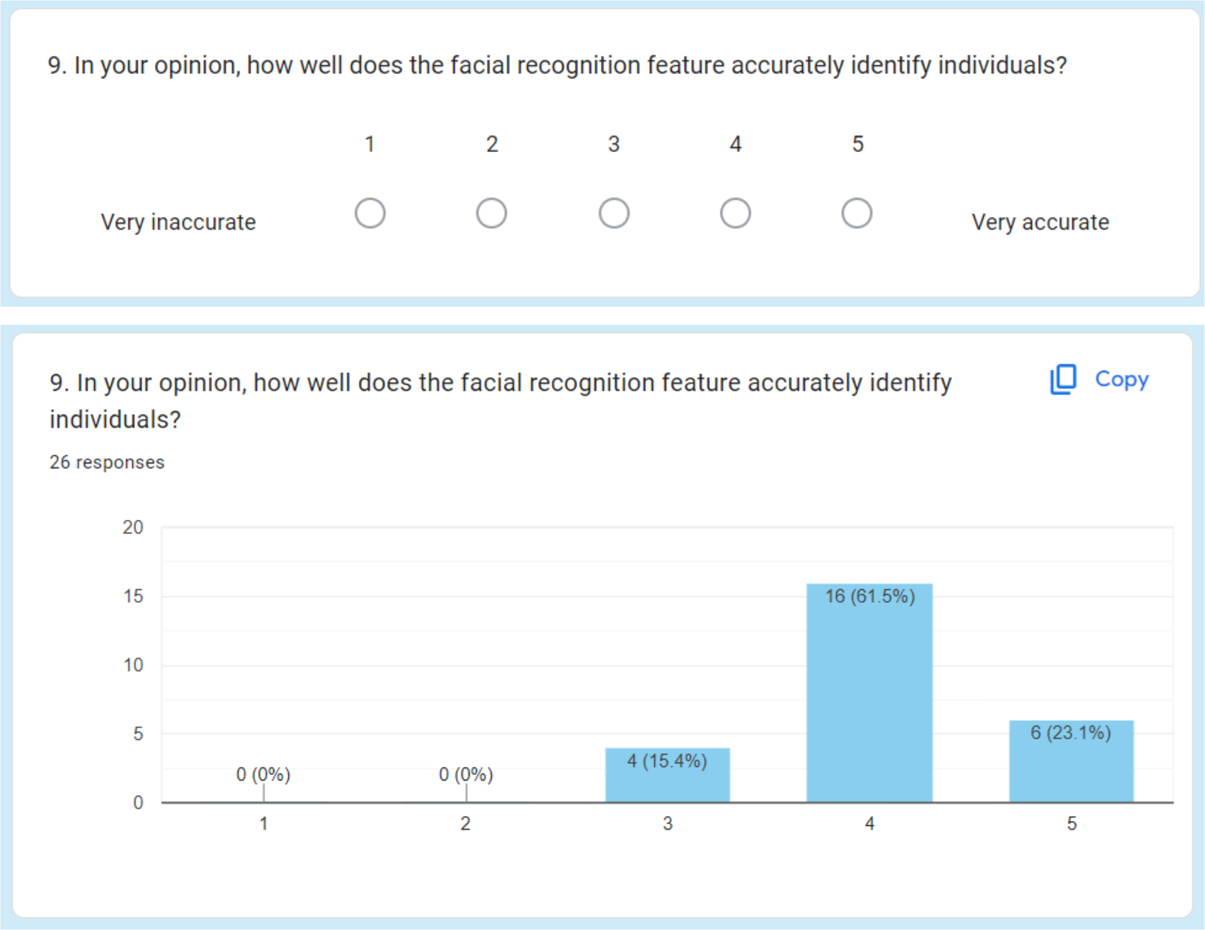


8. Overall, how would you rate the performance of the system overall?

26 responses

Figure 6.1.8: Survey Question 8

Continuing with question 8, the aim was to solicit respondents' evaluations of the overall performance of the system. Respondents were provided with a scale ranging from 1 to 10, where '1' signified poor performance, and '10' represented excellent performance. The highest number of respondents, constituting 34.6%, assigned a grade of '6' to the system, indicating a generally positive but not outstanding performance. This suggests that a significant portion of respondents perceived the system as above average. Following closely, '7' received endorsements from 7 respondents, reflecting a similar sentiment regarding the system's performance. It's noteworthy that the responses are concentrated around these levels, highlighting a consistent perception among respondents. The responses to '5' and '8' were quite similar, differing only by 1 respondent. '5' was chosen by 4 respondents, while '8' was selected by 5. This suggests a moderate spread of opinions regarding the system's performance, with some respondents rating it slightly higher or lower. Remarkably, one respondent bestowed a '9' upon the system, indicating a near-excellent performance. This response suggests a highly positive evaluation by this particular respondent, signifying their satisfaction with the system's overall capabilities.



9. In your opinion, how well does the facial recognition feature accurately identify individuals?

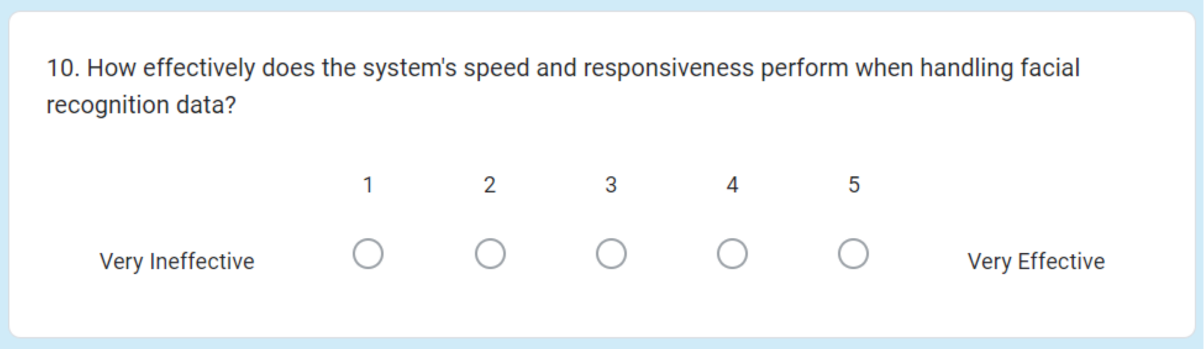
	1	2	3	4	5	
Very inaccurate	○	○	○	○	○	Very accurate

9. In your opinion, how well does the facial recognition feature accurately identify individuals?

26 responses

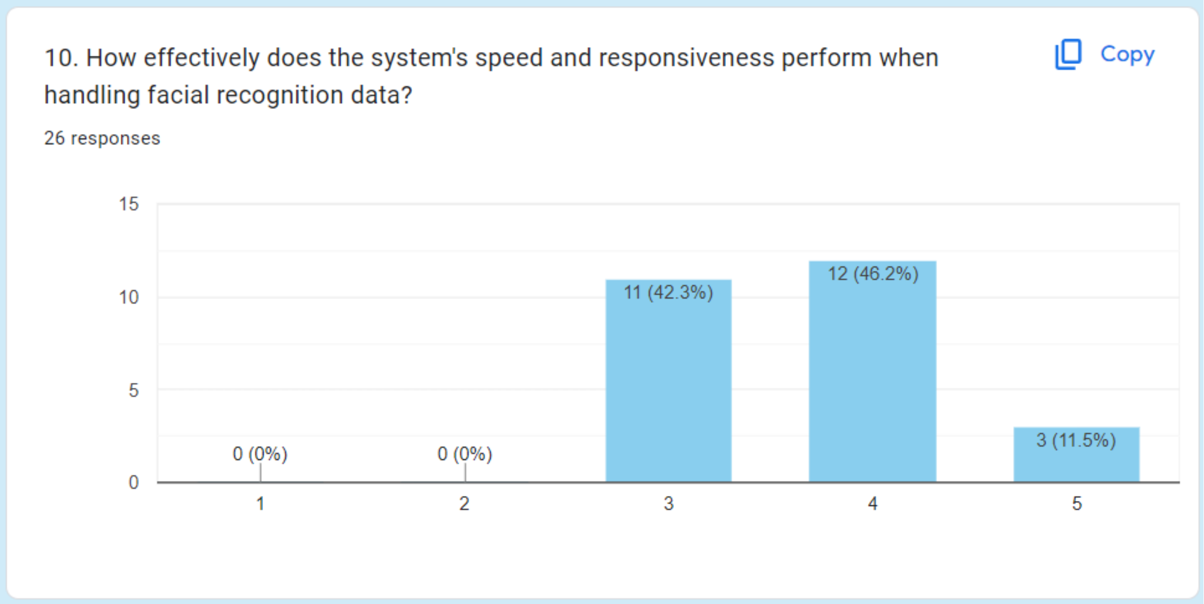
Figure 6.1.9: Survey Question 9

Moving on to question 9, the objective was to assess the system's performance in accurately identifying individuals through face recognition. The responses revealed a notable trend, with a substantial majority of 61.5%, equivalent to 16 respondents, indicating that they found the system to be accurate in its facial recognition capabilities. This suggests a widespread perception among respondents that the algorithm effectively fulfills its primary task of identifying individuals based on facial features. Furthermore, although 4 respondents expressed a 'neutral' stance, it's worth highlighting that a slightly larger group of 6 respondents, representing 23.1%, reported that they found the system to be 'very accurate.' This positive sentiment towards the system's accuracy underscores its effectiveness in the eyes of these users. In summary, the responses to question 9 collectively point to a favorable view among most respondents regarding the system's ability to accurately recognize individuals through facial recognition. The majority either agreed or strongly agreed with the system's accuracy, suggesting a high level of confidence in its performance.



10. How effectively does the system's speed and responsiveness perform when handling facial recognition data?

	1	2	3	4	5	
Very Ineffective	○	○	○	○	○	Very Effective

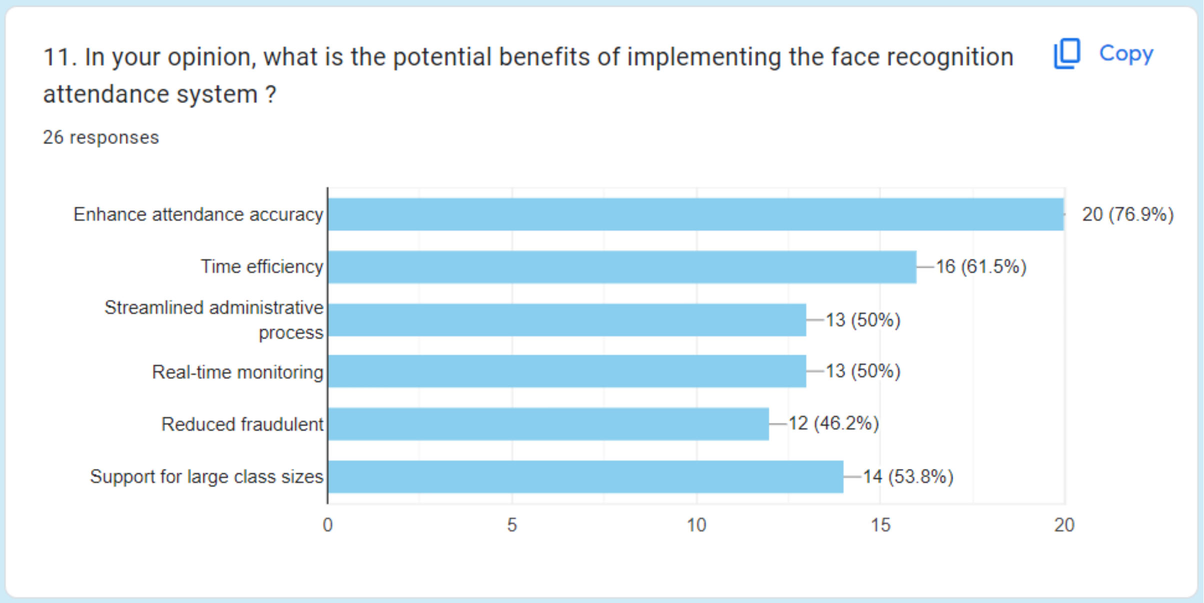


10. How effectively does the system's speed and responsiveness perform when handling facial recognition data?

26 responses

Figure 6.1.10: Survey Question 10

Proceeding to question 10, the objective was to assess the perceived effectiveness of the system in terms of speed and responsiveness when handling facial recognition data, as rated by respondents. The chart illustrates that a significant portion, specifically 46.2% of respondents (12 respondents), found the system to be 'effective' in this regard. This indicates that a substantial majority acknowledged the system's satisfactory performance in terms of speed and responsiveness during data processing. Additionally, a sizable group of respondents, totaling 11 individuals or 42.3%, expressed a 'neutral' stance regarding the system's effectiveness in handling facial recognition data. This suggests that this subset of respondents did not strongly lean towards either a positive or negative evaluation of the system's speed and responsiveness. It's worth noting that there were 3 respondents who expressed a higher level of satisfaction, stating that the system was 'very effective' in this aspect. Their responses indicate a particularly positive perception of the system's performance in terms of speed and responsiveness.

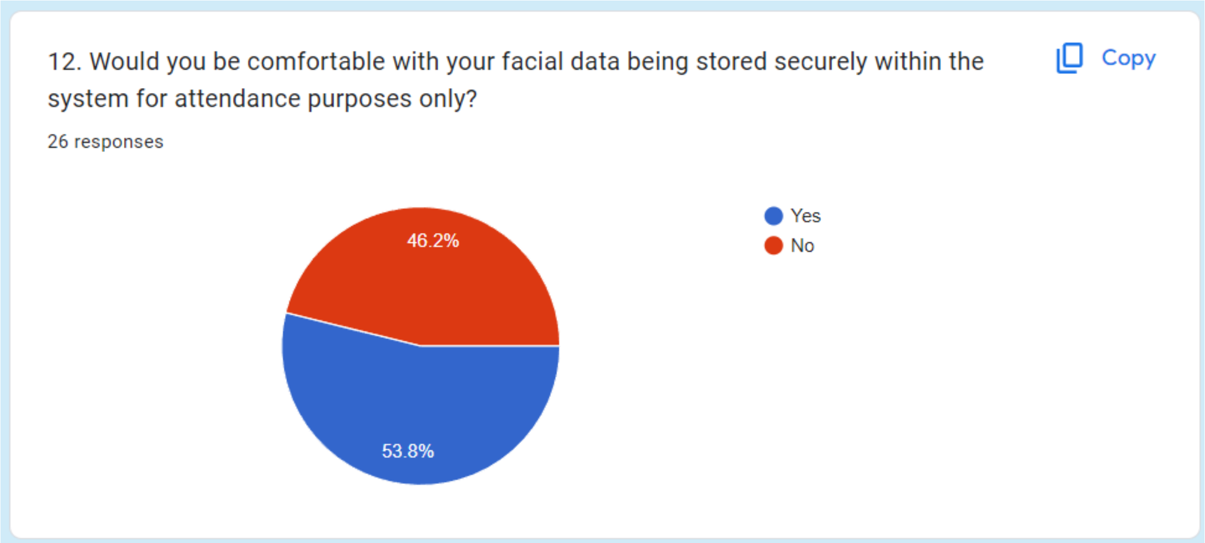

11. In your opinion, what is the potential benefits of implementing the face recognition attendance system ?

26 responses

Figure 6.1.11: Survey Question 11

In question 11, the objective was to gain insights into respondents' opinions regarding the potential benefits of implementing a face recognition attendance system. Respondents were presented with several options, including 'Enhance attendance accuracy,' 'Time efficiency,' 'Streamlined administrative process,' 'Real-time monitoring,' 'Reduced fraudulent,' and 'Support for large class sizes.'

The responses revealed a consensus among respondents, with more than half of them expressing agreement with all of the provided benefits. This widespread agreement underscores the perceived advantages of implementing such a system. Notably, the top three benefits that respondents considered most beneficial were 'Enhance attendance accuracy,' 'Time efficiency,' and 'Support for large class sizes.' 'Enhance attendance accuracy' received substantial support, indicating the importance respondents place on the system's ability to improve attendance tracking precision. 'Time efficiency' was also highly valued, suggesting that respondents recognized the system's potential to expedite attendance-taking processes, optimizing time management in educational or organizational settings. Additionally, 'Support for large class sizes' was acknowledged by many respondents as a valuable aspect of the system, particularly in situations where manual attendance taking for a large number of participants may be cumbersome.

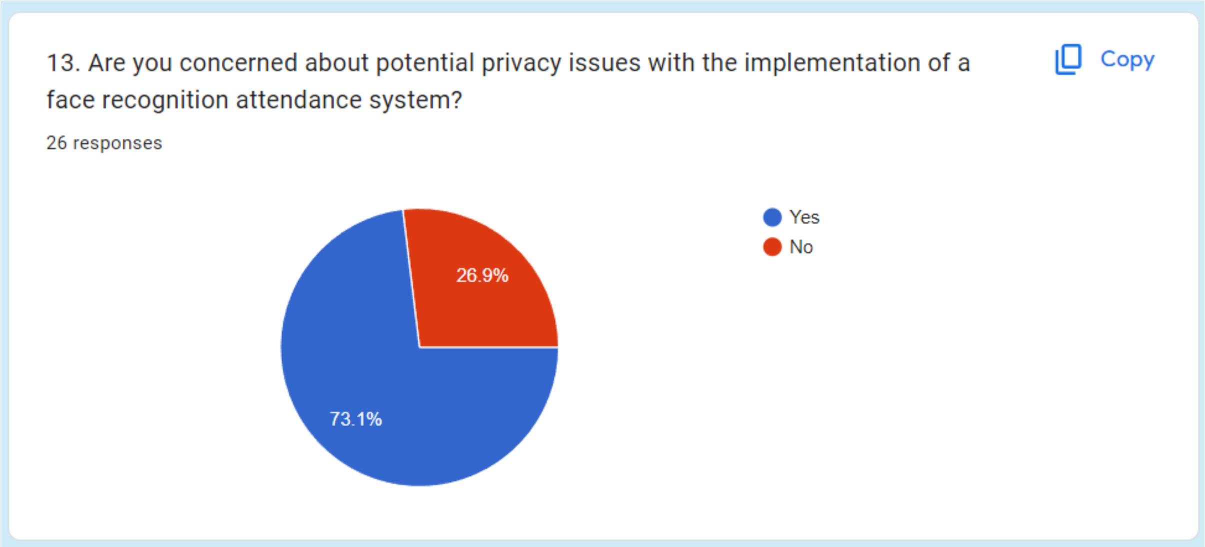


12. Would you be comfortable with your facial data being stored securely within the system for attendance purposes only?

26 responses

Figure 6.1.12: Survey Question 12

In this question, the primary aim was to gauge the comfort level of respondents regarding the secure storage of their facial data within the system, specifically for attendance purposes. The responses yielded a contrasted perspective, with nearly an equal split between those who expressed discomfort and those who were comfortable with this arrangement. It's noteworthy that 12 respondents, constituting a notable portion of the sample, indicated that they were not comfortable with the storage of their facial data for attendance purposes. This suggests that a significant segment of respondents' harbors reservations or concerns about the data storage aspect of the system, possibly due to privacy or security apprehensions. On the other hand, there were 14 respondents who expressed comfort with the idea of their facial data being stored within the system solely for attendance tracking. This group represents individuals who appear to trust in the security measures and purpose-oriented use of their data, thereby perceiving it as an acceptable trade-off for the benefits offered by the system.

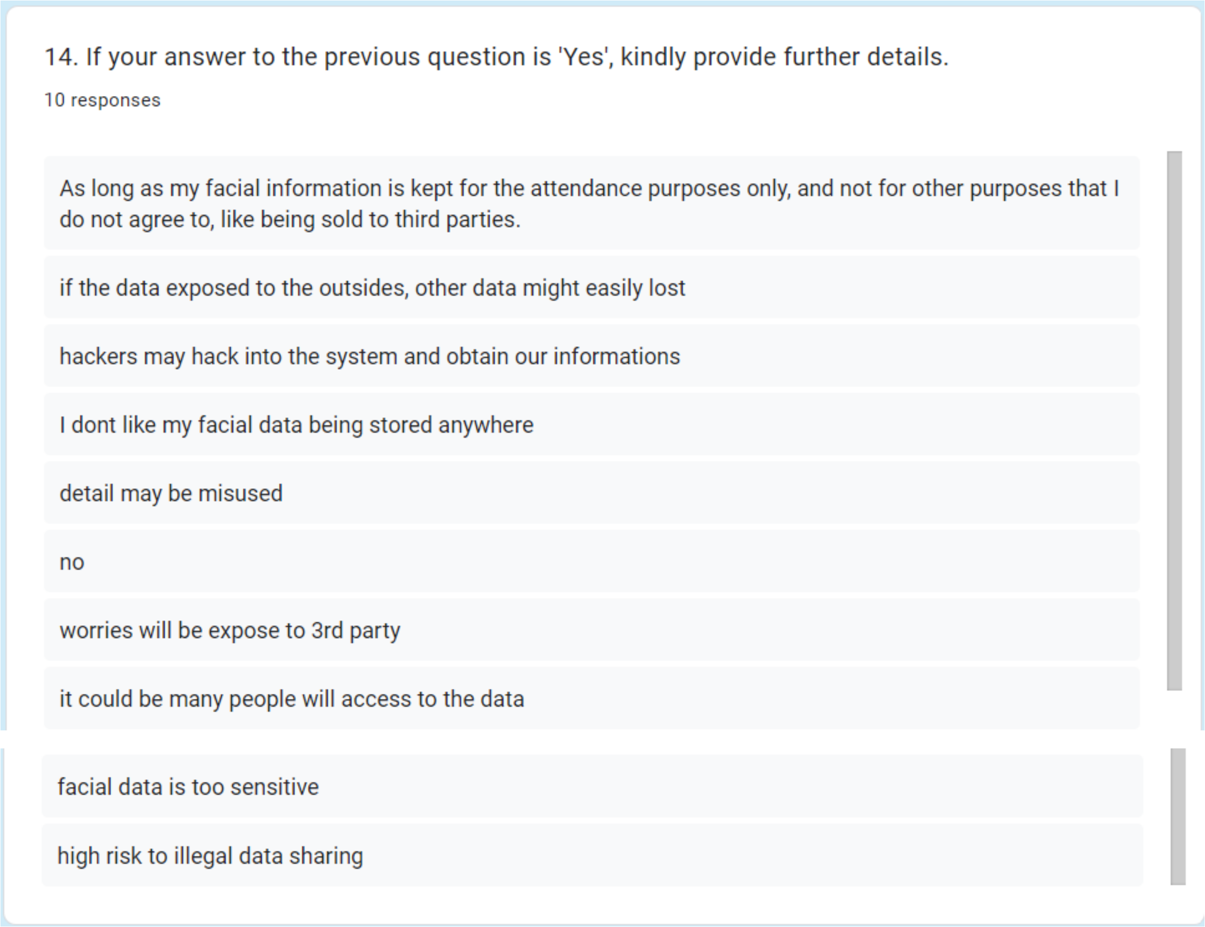


13. Are you concerned about potential privacy issues with the implementation of a face recognition attendance system?

26 responses

Figure 6.1.13: Survey Question 13

The purpose of this question was to assess whether respondents had concerns about potential privacy issues related to the implementation of the face recognition attendance system. The responses are reflected in the pie chart, which clearly illustrates the prevailing sentiment among the respondents. The chart demonstrates that a significant majority of respondents, approximately three-quarters (73.1%), expressed concern about privacy issues associated with the system. This substantial percentage suggests that many respondents are apprehensive about how the technology might impact their privacy, including the handling of their biometric data and the potential for surveillance. Conversely, a smaller but noteworthy portion of respondents, constituting 26.9%, indicated that they did not have concerns about privacy issues arising from the implementation of the face recognition attendance system. This group appears to be more accepting or less worried about potential privacy implications.



14. If your answer to the previous question is 'Yes', kindly provide further details.

10 responses

As long as my facial information is kept for the attendance purposes only, and not for other purposes that I do not agree to, like being sold to third parties.

if the data exposed to the outsides, other data might easily lost

hackers may hack into the system and obtain our informations

I dont like my facial data being stored anywhere

detail may be misused

no

worries will be expose to 3rd party

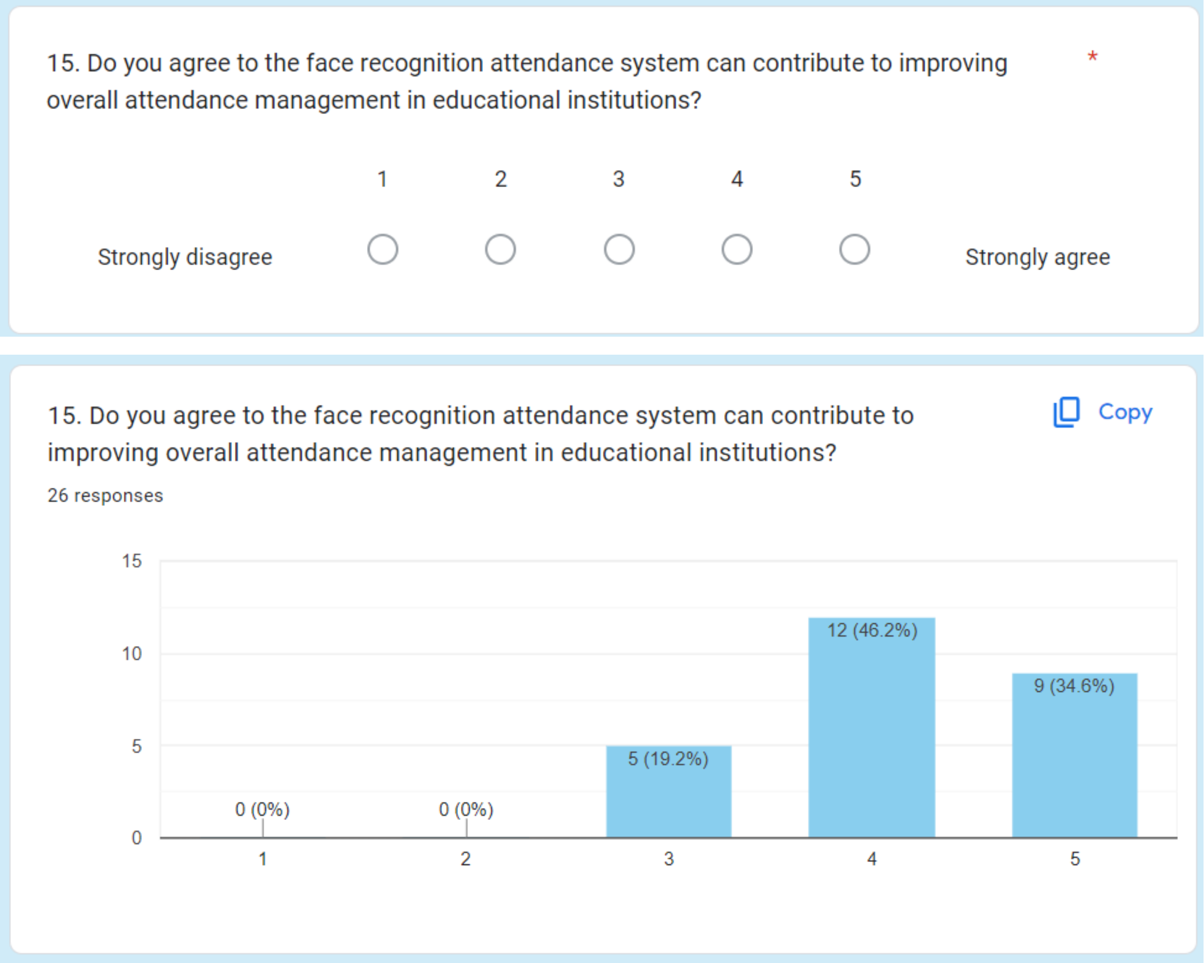
it could be many people will access to the data

facial data is too sensitive

high risk to illegal data sharing

Figure 6.1.14: Survey Question 14

Question 14 serves as a continuation of the exploration of respondents' concerns about privacy issues raised in question 13. However, it's important to note that not all respondents who expressed concerns in question 13 provided further insights or explanations ('Yes') for their concerns. Among those respondents who did elaborate on their concerns, a predominant theme emerged. The majority of respondents who expressed apprehension about privacy issues cited a specific worry: the potential exposure of their facial data to third parties due to security vulnerabilities. This concern reflects a deep-seated apprehension about the security of the facial recognition system and the potential for data breaches or unauthorized access. In essence, respondents who shared this concern are emphasizing the critical importance of safeguarding their biometric data from any form of compromise or unauthorized access. The fear of third-party exposure underscores the significance of robust security measures and data protection protocols in the eyes of these respondents.



15. Do you agree to the face recognition attendance system can contribute to improving overall attendance management in educational institutions? *

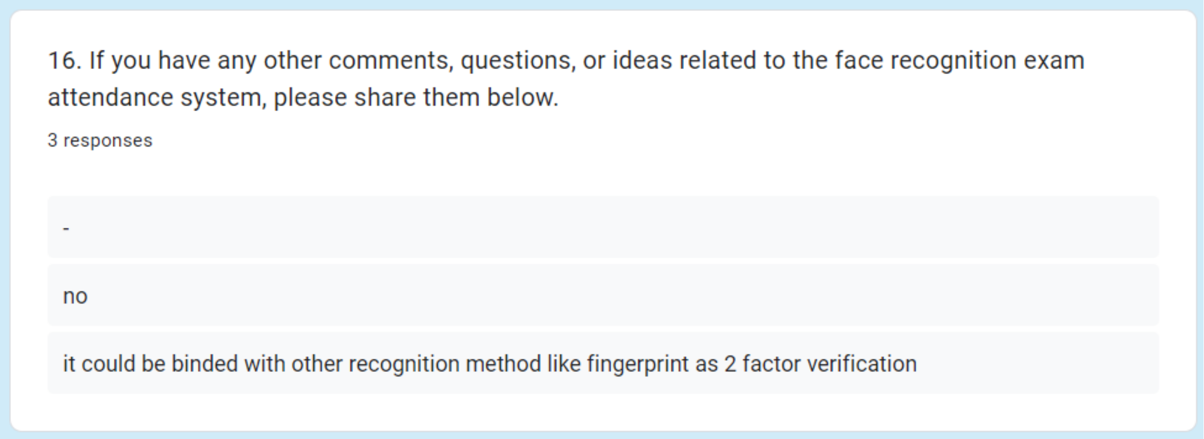
	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

15. Do you agree to the face recognition attendance system can contribute to improving overall attendance management in educational institutions?

26 responses

Figure 6.1.15: Survey Question 15

In question 15, the aim was to assess the level of agreement among respondents regarding the potential contributions of the face recognition attendance system to improving overall attendance management in educational institutions. The responses, as depicted in the chart, reveal a unanimous consensus among the respondents. The chart indicates that all respondents, without exception, expressed a degree of agreement with the notion that the face recognition attendance system can have a positive impact on attendance management. Responses span the spectrum from a neutral stance to strong agreement, with no respondents indicating disagreement or strong disagreement. This unanimous agreement underscores the prevailing belief among respondents that the implementation of a face recognition attendance system holds promise as a beneficial tool for enhancing attendance management within educational institutions. The absence of dissenting opinions suggests a high degree of optimism about the potential benefits of the technology in this context.



16. If you have any other comments, questions, or ideas related to the face recognition exam attendance system, please share them below.

3 responses

-

no

it could be binded with other recognition method like fingerprint as 2 factor verification

Figure 6.1.16: Survey Question 16

In this final question, respondents were provided with an open opportunity to share their opinions and suggestions aimed at enhancing the functionality or performance of the face recognition attendance system. Out of the 26 respondents, it's noteworthy that only one respondent chose to provide feedback in this regard. The single respondent who offered an opinion suggested that the system could be further improved by integrating it with another biometric recognition method, such as fingerprint scanning. This suggestion underscores a key point: the pursuit of enhanced security. By combining facial recognition with fingerprint scanning, the system could potentially offer a more robust and secure means of authentication. While it's important to acknowledge that this was the sole suggestion among the responses, it still holds value as it highlights a potential avenue for system enhancement. Such feedback can serve as a starting point for discussions and considerations about how to augment the system's capabilities, especially in terms of security.

6.2 Testing Setup and Result

6.2.1 System Function Testing Setup and Result

6.2.1.1 Testing Module 1 – Login.py

Objective : To ensure login function is success and will redirect to admin main page.

Input	Expected Output	Actual Output
Login with pair predefined login credential	Success login and redirect to admin main page	User success to login notification and redirect admin main page window
Login with blank input	Login fail and show notification unsuccessful login	Login fail notification pop up and stay at login window
Login with incorrect login credential	Login fail and show notification unsuccessful login	Login fail notification pop up and stay at login window

Table 6.2.1.1: Testing Module 1 – Login.py

6.2.1.2 Testing Module 2 – Init_firebase.py

Objective : To ensure Google Firebase successfully being initialized.

Input	Expected Output	Actual Output
Import firebase admin	Import the firebase token and the system able to link to it.	After executing the authentication process using Firebase Admin, able to initialize the cloud storage and get the credentials.

Table 6.2.1.2: Testing Module 2 – Init_firebase.py

6.2.1.3 Testing Module 3 – AddDatatoDatabase.py

Objective : To ensure module AddDatatoDatabase is able to add candidates information to Google Firebase.

Input	Expected Output	Actual Output
Import and initialize the firebase	Import the firebase token and the system able to link to it	Success to initialize Firebase and able to generate a file/ directory named "Candidates" in Firebase to save candidates information.
Upload candidate information to Firebase in dictionary format	Candidate information will be upload to firebase real-time storage	Successfully upload 15 candidates' information to Firebase real-time storage

Table 6.2.1.3: Testing Module 3 – AddDatatoDatabase.py

6.2.1.4 Testing Module 4 – EncodeGenerator.py

Objective : To encode the facial features of candidate profile image for the recognition process.

Input	**Expected Output**	**Actual Output**
Import and initialize the firebase	Import the firebase token and the system able to link to it	Success to initialize Firebase and able to link with the Firebase Storage that stored candidates image

Encode candidate profile	Face recognition will carry out the process of encoding facial landmarks to recognize the candidate face that is shown in the camera frame	Process each candidate image for face recognition and converts the image to RGB format and extracts facial encodings using face_recognition.face_encodings. These encodings are collected into encodeList
Save the collected facial encodings	Able to save the facials encoding n the local storage.	Save collected facial encodings along with candidate IDs into a Python pickle file (EncodeFile.p).

Table 6.2.1.4: Testing Module 4 – EncodeGenerator.py

6.2.1.5 Testing Module 5 – AdminTab.py

Objective : To carry out the main function of the administrator includes: candidate verification, check candidate list, registration for examination and view report.

Input	Expected Output	Actual Output
Candidate verification, select subject and session then click 'Confirm'	Take candidate attendance and update the attendance time to Firebase.	If the candidate's face is a match, candidate information will be displayed, attendance will be successfully recorded, and the recorded time will be updated in Firebase. If the face is not a match, candidate information will not be displayed.

Click ‘Candidate List’ tab	Direct to the page that include complete candidate name list.	Will direct to the ‘Candidate List’ page and display all the candidates in a list with their information.
Add new candidate to system	Add new candidate to the system and the information will upload to Firebase.	New candidate added to the system with the related information and the record is stored in Firebase real-time storage.
Click ‘Exam Registration’ tab	Direct to the page that allow admin to register for new exam subject.	Will direct to the ‘Exam Registration’ page and display a entry form for the registration.
Register new exam	A new exam will be added to the system, and the related information will be updated in Firebase.	The information about the new exam will be updated in the real-time database file called 'ExamRegistrations.' Under each exam, within the 'sessions' node, it will generate the correct number of sessions based on the session entry number.
Report	Admin can view the report of the system.	Admin can view the report that summarize the candidates and exam information.

Table 6.2.1.5: Testing Module 5 – AdminTab.py

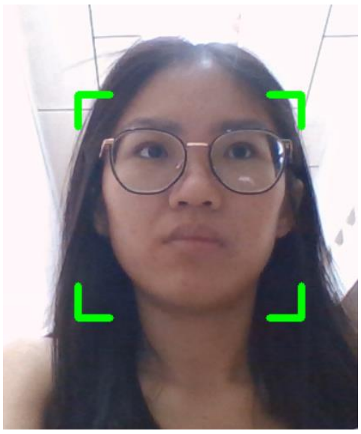
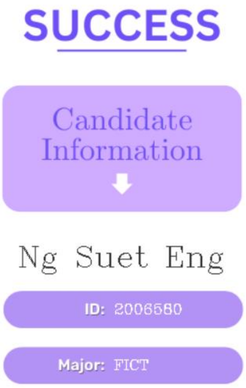
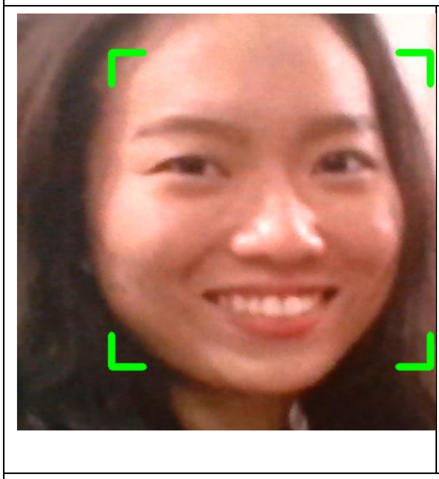
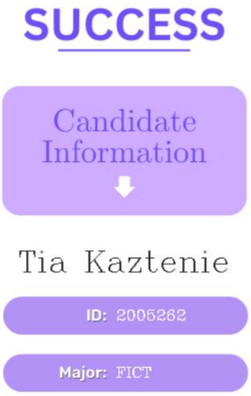
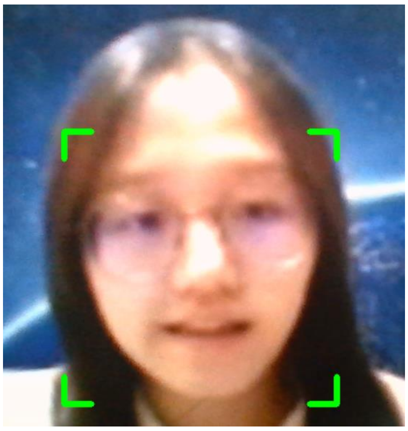
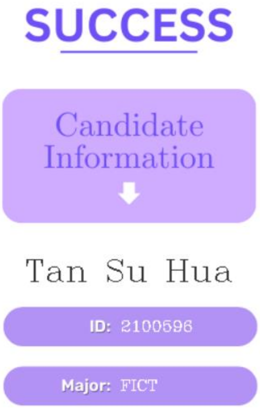
6.2.1.6 Testing Module 6 – Exam_reg_page.py

Objective : To ensure Google Firebase successfully being initialized.

Input	Expected Output	Actual Output
Select certain subject and session and choose candidate and click 'Add Candidate' to register candidates	The candidate Id and name will be added under selected subject and session.	Candidate information, including their candidate ID and name, will be added to the selected subject and the corresponding session child node.

Table 6.2.1.6: Testing Module 6 – Exam_reg_page.py

6.2.2 Face Recognition Testing Setup and Result

Input	Expected Output	Actual Output
		 SUCCESS Candidate Information Ng Suet Eng ID: 2006580 Major: FICT
Result : Matching and display correct candidate information		
		 SUCCESS Candidate Information Tia Kaztenie ID: 2005262 Major: FICT
Result : Matching and display correct candidate information		
		 SUCCESS Candidate Information Tan Su Hua ID: 2100596 Major: FICT
Result : Matching and display correct candidate information		

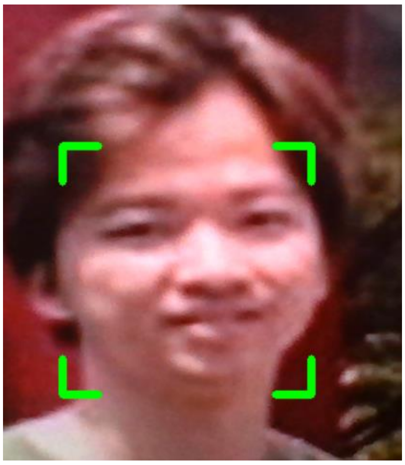
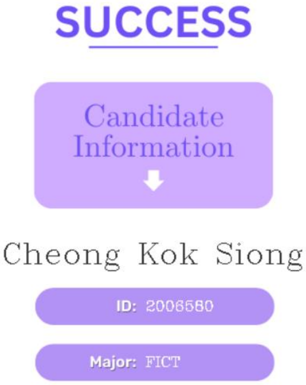
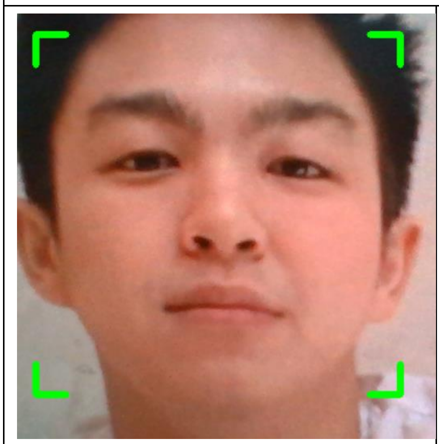
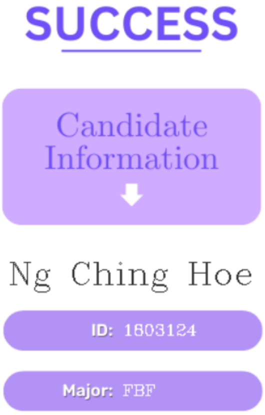
		 SUCCESS Candidate Information Cheong Kok Siong ID: 2006580 Major: FICT
Result : Matching but display wrong candidate ID		
		 SUCCESS Candidate Information Ng Ching Hoe ID: 1803124 Major: FBF
Result : Matching and display correct candidate information		

Table 6.2.2: Face Recognition Testing and Result

Based on the face recognition testing results, all five detections were correct, demonstrating a highly effective detection and recognition algorithm. However, during the fourth recognition, candidate information was displayed incorrectly, and an error occurred during retrieval. The face recognition system demonstrated a high level of accuracy in the majority of cases, underscoring its effectiveness. However, further investigation is needed to address and rectify the issue related to candidate information display during the recognition process.

6.3 Objectives Evaluation

The evaluation of the face recognition attendance system involves a meticulous and comprehensive assessment of its performance against the project's core objectives. The primary objective of reducing fraudulent impersonation cases is gauged by measuring the system's capacity to thwart such attempts. This includes quantifying the actual reduction in successful impersonations, assessing the percentage decrease in impersonation cases compared to previous methods, and gathering valuable feedback from candidates who have experienced the system firsthand.

The second objective, centered around streamlining verification and validation processes, demands a thorough examination of the system's efficiency. Key metrics encompass the average time saved per candidate, resource efficiency gains, and the precision of identity recognition. These factors provide insights into the system's ability to optimize resource allocation while ensuring accurate identity verification.

The third objective pertains to enhancing data integrity and relies on metrics such as the accuracy of facial recognition matches, the precision in facial attribute identification, and the absence of biases or errors. These measures collectively determine the system's reliability and credibility in maintaining data integrity.

In determining the overall evaluation score, each objective can be assigned weights based on its relative significance to the project's success. These scores are then combined, yielding a comprehensive assessment that aids in data-driven decision-making and continual improvement endeavors. Such a detailed evaluation process ensures that the Face Recognition Attendance System effectively fulfills its objectives, ensuring security, efficiency, and data integrity in examination venues and other critical scenarios.

6.4 Implementation Challenges and Issues

Implementing a Face Recognition Attendance System can be a complex undertaking, and various challenges and issues can arise throughout the development process. Here are some of the implementation challenges. First, the "candidate in twins" issue within the system is a specific challenge related to accurately identifying and differentiating between identical twins when using facial recognition technology for attendance tracking. This issue can have significant implications for the system's reliability and effectiveness in recording attendance in scenarios where identical twins are present. The primary reason behind this challenge lies in the inherent limitations of facial recognition algorithms. These algorithms work by extracting unique facial landmarks and features, such as the distance between the eyes, the shape of the nose, and the arrangement of facial landmarks, to create a facial template that represents an individual's identity. However, when dealing with identical twins, these distinctive features are often so similar that the algorithm cannot reliably distinguish between them. Consequently, the system may struggle to assign the correct identity to each twin, leading to potential misidentification. In such scenarios, the inability to differentiate between identical twins can lead to security vulnerabilities and false positives or negatives, potentially compromising the system's reliability. However, it's essential to strike a balance between improving accuracy and respecting privacy and ethical concerns. Implementing measures to handle the "candidate in twins" issue should also consider the legal and ethical implications of facial recognition technology, especially when it comes to data protection and consent.

Moreover, there is an issue that firebase storage read function returning data as array instead of dictionary. This issue within the system can present a significant challenge for developers. In some cases, the Firebase Storage read function may retrieve data in the form of an array, while the system's requirements dictate the need for a dictionary data structure. This misalignment in data format can lead to ongoing complications as developers must repeatedly transform or adapt the data to meet the system's expectations. This not only introduces added complexity and potential for errors but also consumes valuable development time that could be better utilized for system optimization and enhancement. To address this issue effectively, it is crucial to configure the Firebase Storage read function to retrieve data directly in the required dictionary format. By doing so, it could enhance data consistency, and improve the overall efficiency and maintainability of the face recognition attendance system.

The next issue is about the accuracy and reliability, achieving a high level of accuracy and reliability in facial recognition, especially in dynamic, real-world scenarios, is a formidable challenge. The system must contend with a multitude of variables that can impact its performance. Variations in lighting conditions, which can range from harsh sunlight to dimly lit rooms, pose a significant hurdle. Facial expressions, which can change rapidly and widely among individuals, add another layer of complexity. Moreover, the angle at which individuals' faces are captured varies, making it critical for the system to adapt and recognize faces from various perspectives. For instance, consider low-light conditions where ambient illumination is inadequate. In such scenarios, the system may struggle to discern facial features clearly, leading to recognition errors. Similarly, strong shadows cast on the face can obscure critical facial landmarks, impeding the system's ability to accurately identify individuals. These challenges are emblematic of the real-world conditions that a facial recognition system must navigate while striving for precision and dependability in attendance tracking.

Safeguarding biometric data, including facial images, presents paramount privacy and security considerations. The management of this sensitive data demands a multifaceted approach to ensure comprehensive protection. Firstly, all biometric data should be rigorously encrypted using state-of-the-art encryption protocols during both storage and transmission phases. Secondly, it is imperative to establish secure data storage practices, implementing robust access controls, and authentication mechanisms to prevent unauthorized access or breaches. This involves strict access permissions, role-based access controls, and continuous monitoring of data integrity. Moreover, compliance with stringent data protection regulations, such as Personal Data Protection Act 2010 (PDPA), Cybersecurity Act 2018, or local privacy laws, is not only advisable but obligatory in many jurisdictions. Achieving and maintaining compliance often necessitates allocating additional resources, such as legal counsel and cybersecurity experts, to ensure that the facial recognition system operates within legal boundaries, respects individuals' privacy rights, and remains resilient against emerging security threats. The convergence of encryption, secure data handling, and regulatory compliance is indispensable in constructing a facial recognition system that upholds the highest standards of privacy and data security.

6.5 Recommendations

To addressing the ‘candidate in twins’ requires the development of more sophisticated facial recognition algorithms capable of discerning subtle differences between identical twins. Researchers are exploring techniques like 3D facial recognition, which captures depth information in addition to 2D features, to improve accuracy. Additionally, machine learning approaches that take into account contextual information, such as the person's behavior or location, can provide additional cues for distinguishing between twins.

To maximize system performance and overcome the challenge of accuracy and reliability, it is recommended to position the system in areas with optimal lighting conditions. These conditions typically include adequate ambient lighting, minimal shadows, and well-distributed illumination across the faces of individuals being recognized. Good lighting allows the system to capture clear and well-defined facial features, making critical landmarks such as eyes, nose, and mouth easily discernible. This, in turn, facilitates accurate recognition. Moreover, optimal lighting minimizes the presence of strong shadows that can obscure facial features and affect recognition accuracy. By maintaining consistent and uniform lighting conditions, the system's performance remains stable, reducing the likelihood of recognition errors caused by lighting variations. Implementing supplementary lighting sources, educating users about the importance of good lighting, and positioning the hardware thoughtfully all contribute to ensuring optimal lighting conditions for the facial recognition system. Ultimately, this enhances its accuracy and reliability while mitigating potential challenges associated with lighting variations. Elevate the performance of facial recognition system by investing in cutting-edge camera hardware engineered for excellence. Opt for high-quality cameras equipped with advanced features such as exceptional low-light capabilities and an extensive dynamic range. These superior camera systems are designed to excel in challenging environments, ensuring the capture of crystal-clear and precisely exposed facial images, even under diverse and unpredictable lighting conditions. The choice of appropriate camera hardware is not merely an enhancement but a pivotal factor that can profoundly enhance the system's overall performance, reliability, and accuracy. By prioritizing the integration of top-tier camera technology, the face recognition system gains the capability to consistently deliver impeccable results, regardless of the lighting challenges it encounters.

Chapter 7

Conclusion

The adoption of facial recognition technology in various fields presents both opportunities and challenges. This technology, driven by advanced algorithms and deep learning techniques, excels in real-time identification and adapts to dynamic environments. However, challenges such as potential fraud and the time-consuming nature of manual verification processes have emerged. Impersonation fraud is a concern in exams, but using facial recognition for identity checks can solve this issue and ensure accurate records. Manual verification in exams is slow and error-prone, but facial recognition speeds up the process and reduces mistakes, even if verification slips are lost. Human verification can have errors and data problems, but facial recognition is more dependable and accurate. While facial recognition technology helps solve these challenges, we must also consider ethics and privacy when using it.

Setting up a face recognition attendance system entails hardware and software configuration, database creation, security measures, and testing for accuracy. Enrollment, monitoring, and privacy compliance are key components, making it a complex yet valuable solution for attendance tracking. For the methodology chosen, Prototyping Model for this project. It's great for individual programming projects because it allows for flexible adjustments to evolving requirements and encourages rapid progress. The main phases include gathering initial requirements, creating a basic system design, getting user feedback, and refining the prototype, and finally, implementing and testing the system.

Furthermore, the challenges of implementing a face recognition attendances system poses challenges including the "candidate in twins" issue, Firebase Storage data format mismatches, accuracy and reliability concerns, and the need for robust data security and privacy measures. Addressing these challenges requires a balanced approach that integrates technical, ethical, and legal considerations.

To optimize it, consider implementing advanced algorithms, such as 3D facial recognition and contextual machine learning, to address challenges related to identifying identical twins accurately. Ensuring ideal lighting conditions by situating the system in well-lit areas, minimizing shadows, and educating users on lighting importance is crucial for improving accuracy. Additionally, investing in top-tier camera hardware equipped with advanced features like exceptional low-light performance and a wide dynamic range plays a pivotal role in enhancing system performance. These combined efforts lead to improved accuracy, reliability, and overall effectiveness in attendance tracking, mitigating potential challenges associated with lighting variations and similar facial features.

REFERENCES

[1] ALICE J. O'TOOLE, XIAOBO AN, JOSEPH DUNLOP, VAIDEHI NATU, "Comparing Face Recognition Algorithms to Humans on Challenging Tasks". October 2012. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=910638

[2] THALES. "Face recognition history." Digital Identity and Security. https://www.thalesgroup.com/en/markets/digital-identity-andsecurity/government/inspired/history-of-facial-recognition (27November 2022)

[3] Sagar M Rao, Poonam, Bhavana Girish, Shivam Pandey, Dr.Srividya B V, "Fraudulent Detection in Examination Department". 5 May 2020. [Online]. Available: https://www.irjet.net/archives/V7/i5/IRJETV7I5243.pdf

[4] SACBU. "Gaokao, One Article Tell You Everything about China NCEE." SACBU Blog. https://sacbu.com/about-china-ncee (28November 2022)

[5] GLOBAL TIMES. "Record 11.9m students to take gaokao with full preparations under shadow of COVID-19." CHINA SOCIETY. https://www.globaltimes.cn/page/202206/1267415.shtml (28 November 2022)

[6] Henning Lund. "WHAT IS DATA INTEGRITY AND WHY IS IT IMPORTANT FOR YOU?." RAPIDI. https://www.rapidionline.com/blog/data-integrity-what-andwhy#:~:text=Data%20integrity%20is%20important%20as,completeness%20of%20data%20is%20essential (29 November 2022)

[7] Michelle Araujo E Viegas. Richard Simoes. Nikisha Thanekar. Saptak Banerjee. Rahul Dicholkar., "Different Approaches of Face Recognition,". June 2019. [Online]. Available: https://www.ijert.org/different-approaches-of-face-recognition

[8] Oksana Mikhalchuk. "Using AI and biometrics to enhance exam proctoring." BIOMETRIC UPDATE. https://www.biometricupdate.com/202001/using-ai-and-biometrics-toenhance-exam-proctoring (29 November 2022)

[9] Patrick Grother. Mei Ngan. Kayee Hanaoka., "Ongoing Face Recognition Vendor Test (FRVT) Part 2: Identification." WayBack Machine.

https://web.archive.org/web/20200706162214/https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8238.pdf (30 November 2022)

[10] Divyarajsinh N. Parmar. Brijesh B. Mehta., "Face Recognition Methods & Applications." January 2013. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1403/1403.0485.pdf

[11] Girija Shankar Behera. "Face Detection with Haar Cascade." Towards Data Science. https://towardsdatascience.com/face-detection-with-haarcascade-727f68dafd08 (30 November 2022)

[12] Ahmed Fawzy Gad. "Faster R-CNN Explained for Object Detection Tasks." Paperspace Blog. https://blog.paperspace.com/faster-r-cnnexplained-object-detection/#:~:text=Faster%20R%2DCNN%20is%20a,the%20locations%20of%20different%20objects. (1 December 2022)

[13] Chendi Wang., "Human Emotional Facial Expression Recognition." 2018. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1803/1803.10864.pdf

[14] Jian-Gang Wang. "Local Binary Patterns." SCHOLARPEDIA. http://www.scholarpedia.org/article/Local_Binary_Patterns (2 December 2022)

[15] Robert Triggs. "Facial recognition technology explained." Android Authority. https://www.androidauthority.com/facial-recognitiontechnology-explained-800421/. (12 February 2023)

[16] Tech Desk. "Samsung Galaxy S9, S9+ to come with 'Intelligent Scan' unlock feature: Report." The Indian Express. https://indianexpress.com/article/technology/mobile-tabs/samsunggalaxy-s9-s9-plus-may-come-with-intelligent-scan-unlock-feature-appleface-id-5043322/. (12 February 2023)

[17] Shara Tibken. Alfred Ng. "Galaxy S9 Intelligent Scan favors unlocking ease over security." CNET. https://www.cnet.com/tech/mobile/samsunggalaxy-s9-intelligent-scan-unlock-favors-ease-over-security/. (12 February 2023)

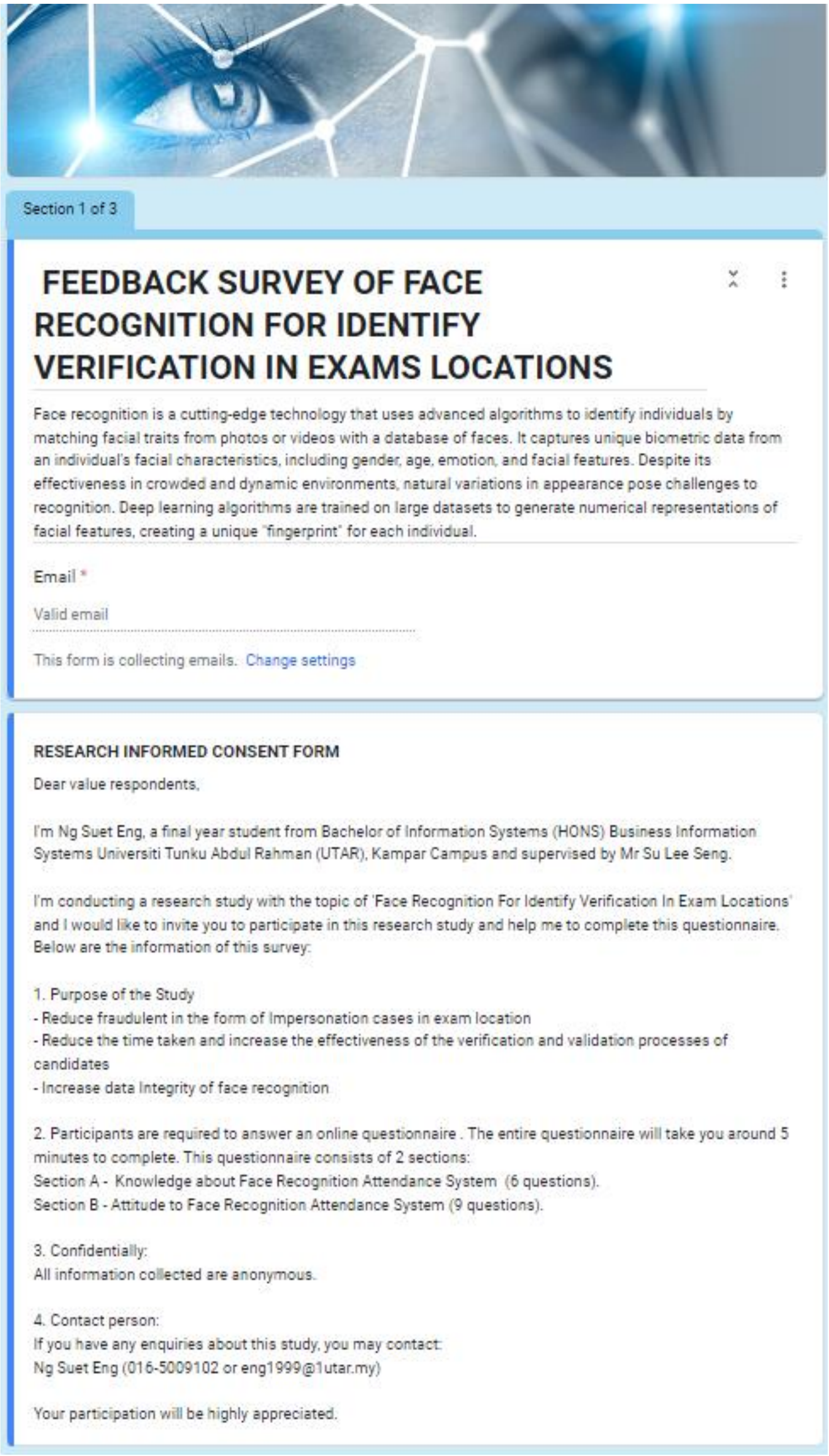
[18] Samsung for Business. "How to Secure Your Phone with Intelligent Scan." Samsung Business Insights. https://insights.samsung.com/2018/04/05/how-to-secure-your-phonewith-intelligent-scan/ (12 February 2023)

[19] Maggie Tillman. “What is Apple Face ID and how does it work?.” PocketLint. https://www.pocket-lint.com/phones/news/apple/142207what-is-apple-face-id-and-how-does-it-work/. (14 February 2023)

[20] Wei-Meng Lee. “Using DeepFace for Face Recognition.” TowardsDataScience. https://towardsdatascience.com/using-deepfacefor-face-recognition-5f8d1e43f2a6. (14 February 2023)

[21] Luka Dulčić. “Face Recognition with FaceNet and MTCNN.” Ars Futura. https://arsfutura.com/magazine/face-recognition-with-facenetand-mtcnn/ (25 February 2023)

APPENDIX



Section 1 of 3

FEEDBACK SURVEY OF FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

Face recognition is a cutting-edge technology that uses advanced algorithms to identify individuals by matching facial traits from photos or videos with a database of faces. It captures unique biometric data from an individual's facial characteristics, including gender, age, emotion, and facial features. Despite its effectiveness in crowded and dynamic environments, natural variations in appearance pose challenges to recognition. Deep learning algorithms are trained on large datasets to generate numerical representations of facial features, creating a unique "fingerprint" for each individual.

Email *

Valid email

This form is collecting emails. Change settings

RESEARCH INFORMED CONSENT FORM

Dear value respondents,

I'm Ng Suet Eng, a final year student from Bachelor of Information Systems (HONS) Business Information Systems Universiti Tunku Abdul Rahman (UTAR), Kampar Campus and supervised by Mr Su Lee Seng.

I'm conducting a research study with the topic of 'Face Recognition For Identify Verification In Exam Locations' and I would like to invite you to participate in this research study and help me to complete this questionnaire. Below are the information of this survey:

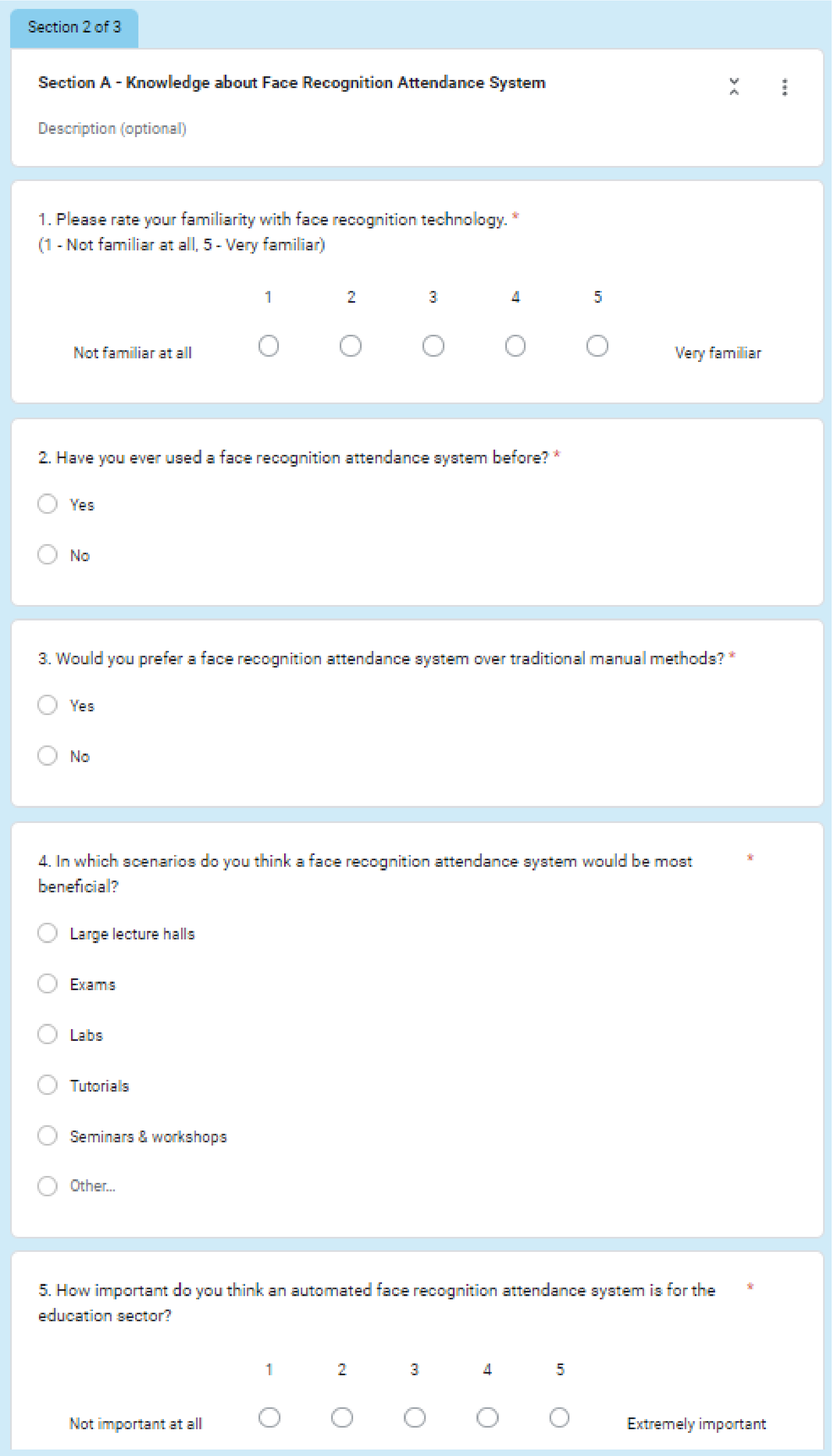
1. Purpose of the Study
- Reduce fraudulent in the form of Impersonation cases in exam location
- Reduce the time taken and increase the effectiveness of the verification and validation processes of candidates
- Increase data Integrity of face recognition

2. Participants are required to answer an online questionnaire . The entire questionnaire will take you around 5 minutes to complete. This questionnaire consists of 2 sections:
Section A - Knowledge about Face Recognition Attendance System (6 questions).
Section B - Attitude to Face Recognition Attendance System (9 questions).

3. Confidentially:
All information collected are anonymous.

4. Contact person:
If you have any enquiries about this study, you may contact:
Ng Suet Eng (016-5009102 or eng1999@1utar.my)

Your participation will be highly appreciated.



Section 2 of 3

Section A - Knowledge about Face Recognition Attendance System

Description (optional)

1. Please rate your familiarity with face recognition technology. *
(1 - Not familiar at all, 5 - Very familiar)

	1	2	3	4	5	
Not familiar at all	○	○	○	○	○	Very familiar

2. Have you ever used a face recognition attendance system before? *

○ Yes

○ No

3. Would you prefer a face recognition attendance system over traditional manual methods? *

○ Yes

○ No

4. In which scenarios do you think a face recognition attendance system would be most beneficial? *

○ Large lecture halls

○ Exams

○ Labs

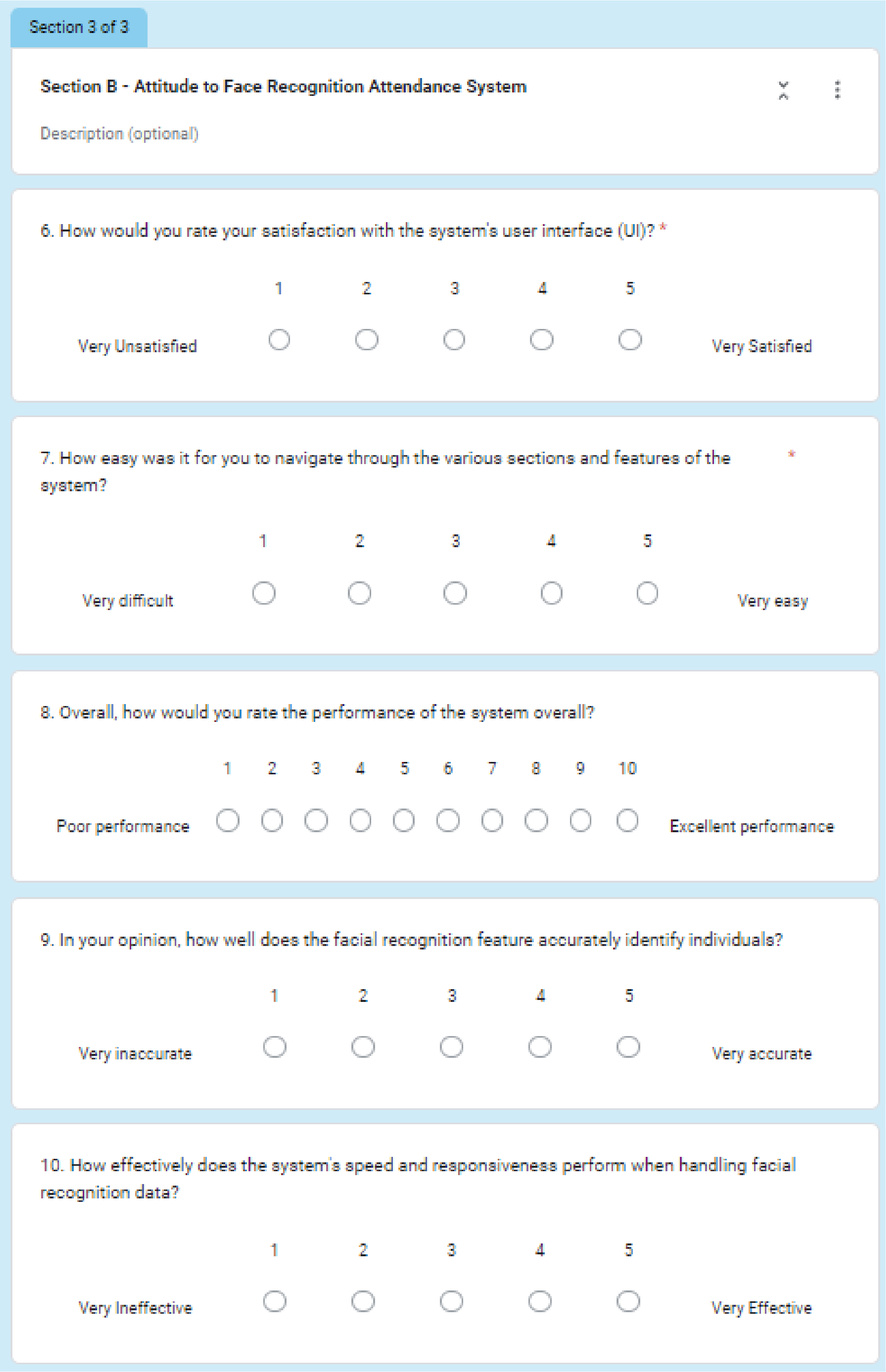
○ Tutorials

○ Seminars & workshops

○ Other...

5. How important do you think an automated face recognition attendance system is for the education sector? *

	1	2	3	4	5	
Not important at all	○	○	○	○	○	Extremely important



Section 3 of 3

Section B - Attitude to Face Recognition Attendance System

Description (optional)

6. How would you rate your satisfaction with the system's user interface (UI)? *

	1	2	3	4	5	
Very Unsatisfied	○	○	○	○	○	Very Satisfied

7. How easy was it for you to navigate through the various sections and features of the system? *

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

8. Overall, how would you rate the performance of the system overall?

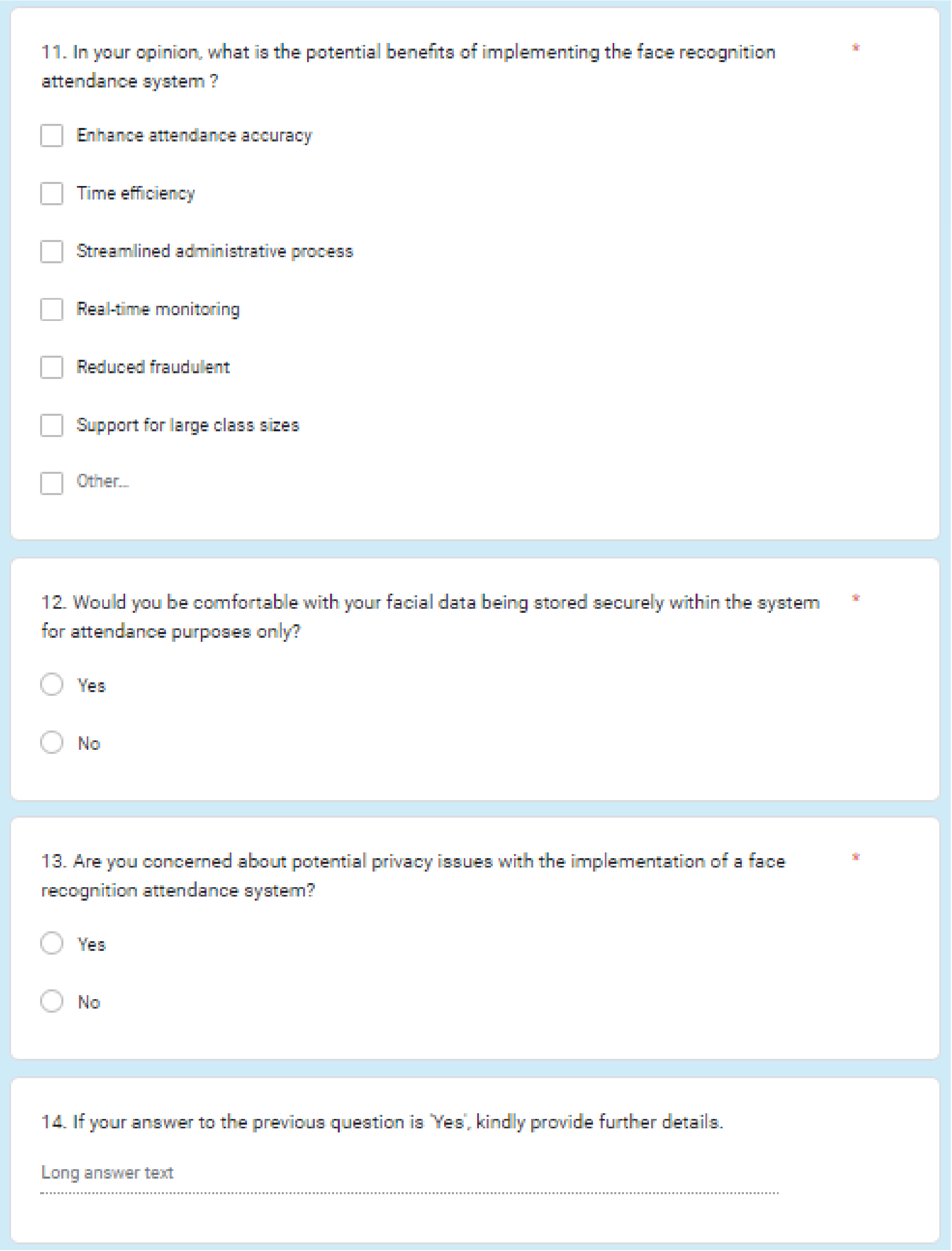
	1	2	3	4	5	6	7	8	9	10	
Poor performance	○	○	○	○	○	○	○	○	○	○	Excellent performance

9. In your opinion, how well does the facial recognition feature accurately identify individuals?

	1	2	3	4	5	
Very inaccurate	○	○	○	○	○	Very accurate

10. How effectively does the system's speed and responsiveness perform when handling facial recognition data?

	1	2	3	4	5	
Very Ineffective	○	○	○	○	○	Very Effective



11. In your opinion, what is the potential benefits of implementing the face recognition attendance system ? *

□ Enhance attendance accuracy

□ Time efficiency

□ Streamlined administrative process

□ Real-time monitoring

□ Reduced fraudulent

□ Support for large class sizes

□ Other...

12. Would you be comfortable with your facial data being stored securely within the system for attendance purposes only? *

○ Yes

○ No

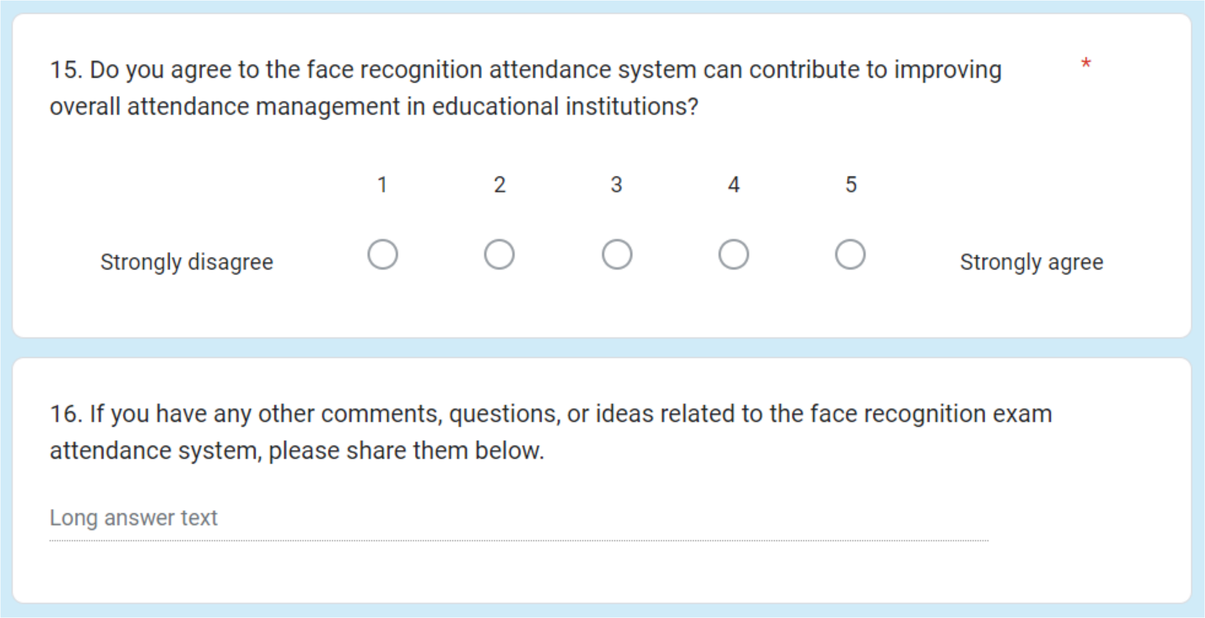
13. Are you concerned about potential privacy issues with the implementation of a face recognition attendance system? *

○ Yes

○ No

14. If your answer to the previous question is 'Yes', kindly provide further details.

Long answer text



15. Do you agree to the face recognition attendance system can contribute to improving overall attendance management in educational institutions? *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

16. If you have any other comments, questions, or ideas related to the face recognition exam attendance system, please share them below.

Long answer text

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 1**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Carried out FYP2 project planning and design the timeline.
- Explore and discuss the system function and feature.

2. WORK TO BE DONE

- Need to continue define the system functions.
- Work on the database linkage

3. PROBLEMS ENCOUNTERED

- Need to confirm the database that can work with the PyCharm Community environment.

4. SELF EVALUATION OF THE PROGRESS

- Have to kick start for the project.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 3**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Refine the project literature review.
- Working on the system function design and the interface design.

2. WORK TO BE DONE

- Develop on the system.
- Continue explore on the system database.

3. PROBLEMS ENCOUNTERED

- Image retrieve problem from Firebase as unable to get the blob.
- Data structure format error.

4. SELF EVALUATION OF THE PROGRESS

- Need more knowledge on database.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 5**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Confirmed to apply Google Firebase as database to the system.
- Able to write data into the Firebase.

2. WORK TO BE DONE

- Continue to work on system development.

3. PROBLEMS ENCOUNTERED

- Too many nested functions to call in the code.

4. SELF EVALUATION OF THE PROGRESS

- Need to debug all the error in the code.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 7**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Completed to define the system interface design.
- Able to store the unique ID into certain session in Firebase.

2. WORK TO BE DONE

- Continue to work on system development.

3. PROBLEMS ENCOUNTERED

- Lack of knowledge to handle Firebase.

4. SELF EVALUATION OF THE PROGRESS

- Progress is a bit slow compared to planning.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 9**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Learned more on the Firebase.
- Developed 70% on the system functionalities.

2. WORK TO BE DONE

- Continue to work on system development.

3. PROBLEMS ENCOUNTERED

- Error to connect between the interface and the data in Firebase due to the data structure stored.

4. SELF EVALUATION OF THE PROGRESS

- Need to put more effort on the system.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 11**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Done most of the system development.
- Conducting system testing.

2. WORK TO BE DONE

- Work on the project report.
- Continue system testing.

3. PROBLEMS ENCOUNTERED

- Some error of the system response and the application crash due to too high computation power.

4. SELF EVALUATION OF THE PROGRESS

- Progress not in track.

SuLee

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 13**
Student Name & ID: Ng Suet Eng & 20ACB06580	
Supervisor: Mr. Su Lee Seng	
Project Title: Face Recognition for Identity Verification in Exams Locations	

1. WORK DONE

- Still working on the report.

2. WORK TO BE DONE

- Complete on the report.

3. PROBLEMS ENCOUNTERED

- No problem.

4. SELF EVALUATION OF THE PROGRESS

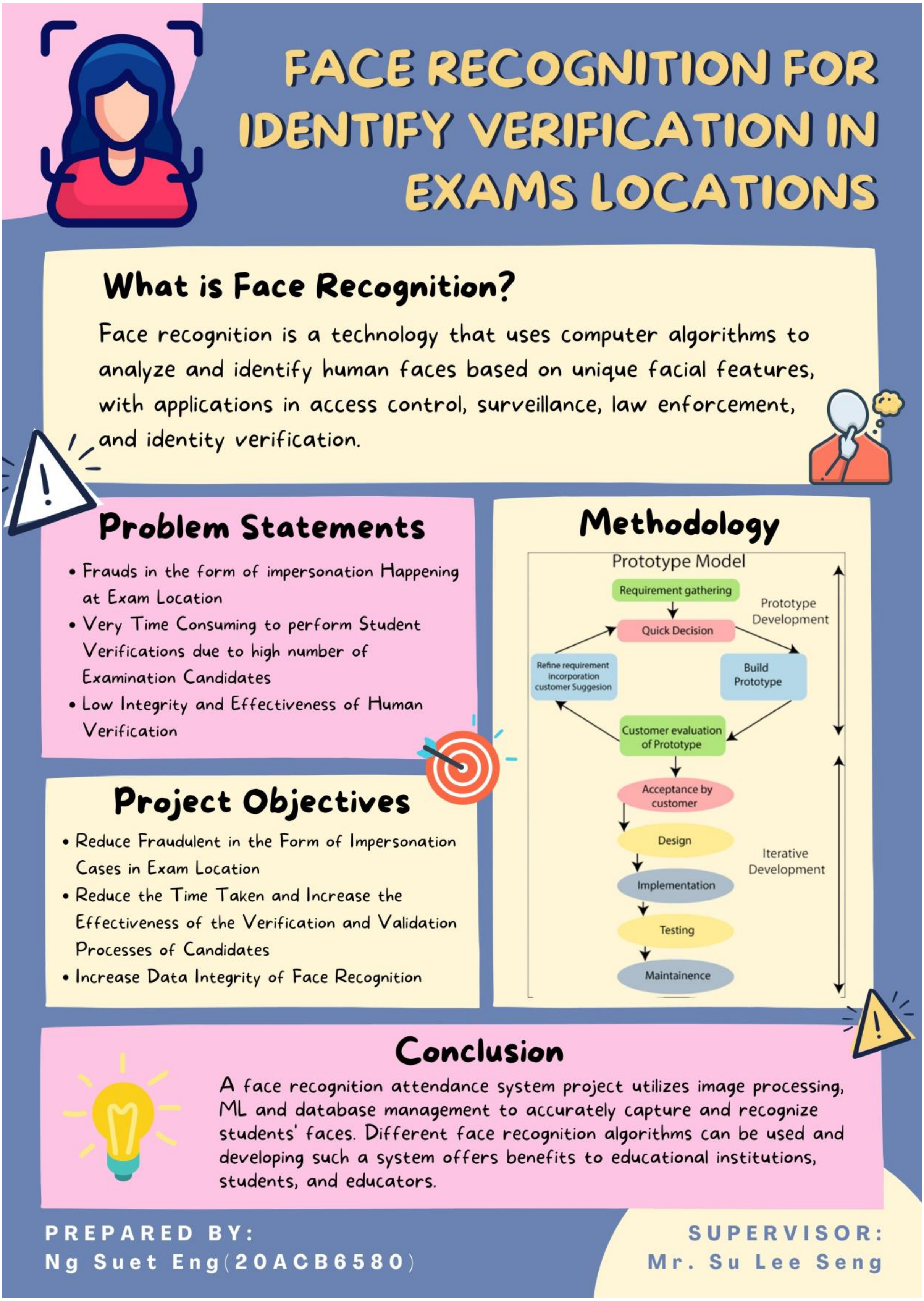
- Able to finish the project in expectation even though it is not perfect.

SuLee

Supervisor's signature

Student's signature

POSTER



FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

What is Face Recognition?

Face recognition is a technology that uses computer algorithms to analyze and identify human faces based on unique facial features, with applications in access control, surveillance, law enforcement, and identity verification.

Problem Statements

- Frauds in the form of impersonation Happening at Exam Location
- Very Time Consuming to perform Student Verifications due to high number of Examination Candidates
- Low Integrity and Effectiveness of Human Verification

Methodology

Prototype Model

Requirement gathering

Quick Decision

Build Prototype

Customer evaluation of Prototype

Refine requirement incorporation customer Suggesion

Prototype Development

Acceptance by customer

Design

Implementation

Testing

Maintainence

Iterative Development

Project Objectives

- Reduce Fraudulent in the Form of Impersonation Cases in Exam Location
- Reduce the Time Taken and Increase the Effectiveness of the Verification and Validation Processes of Candidates
- Increase Data Integrity of Face Recognition

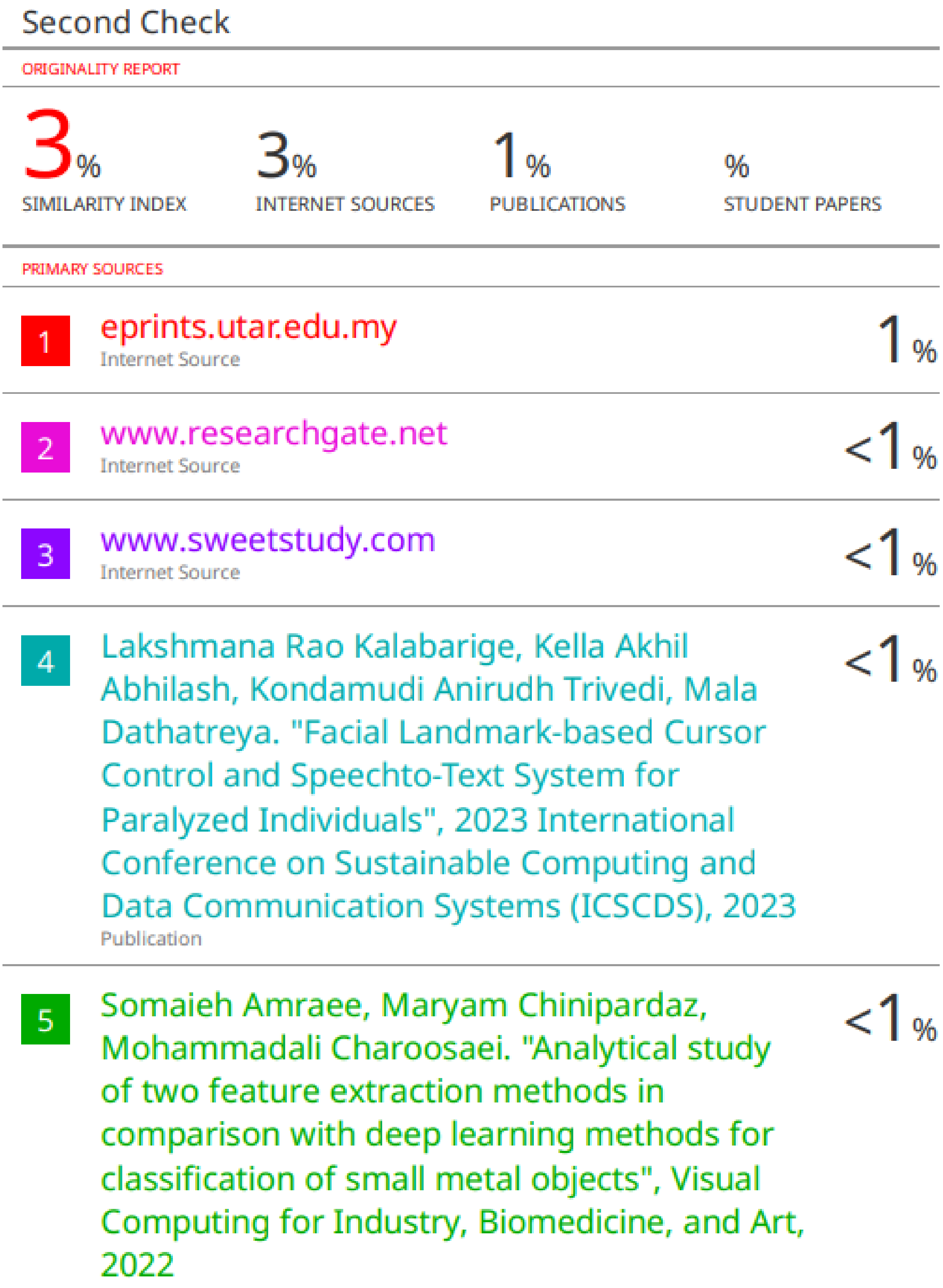
Conclusion

A face recognition attendance system project utilizes image processing, ML and database management to accurately capture and recognize students' faces. Different face recognition algorithms can be used and developing such a system offers benefits to educational institutions, students, and educators.

PREPARED BY:
Ng Suet Eng(20ACB6580)

SUPERVISOR:
Mr. Su Lee Seng

PLAGIARISM CHECK RESULT



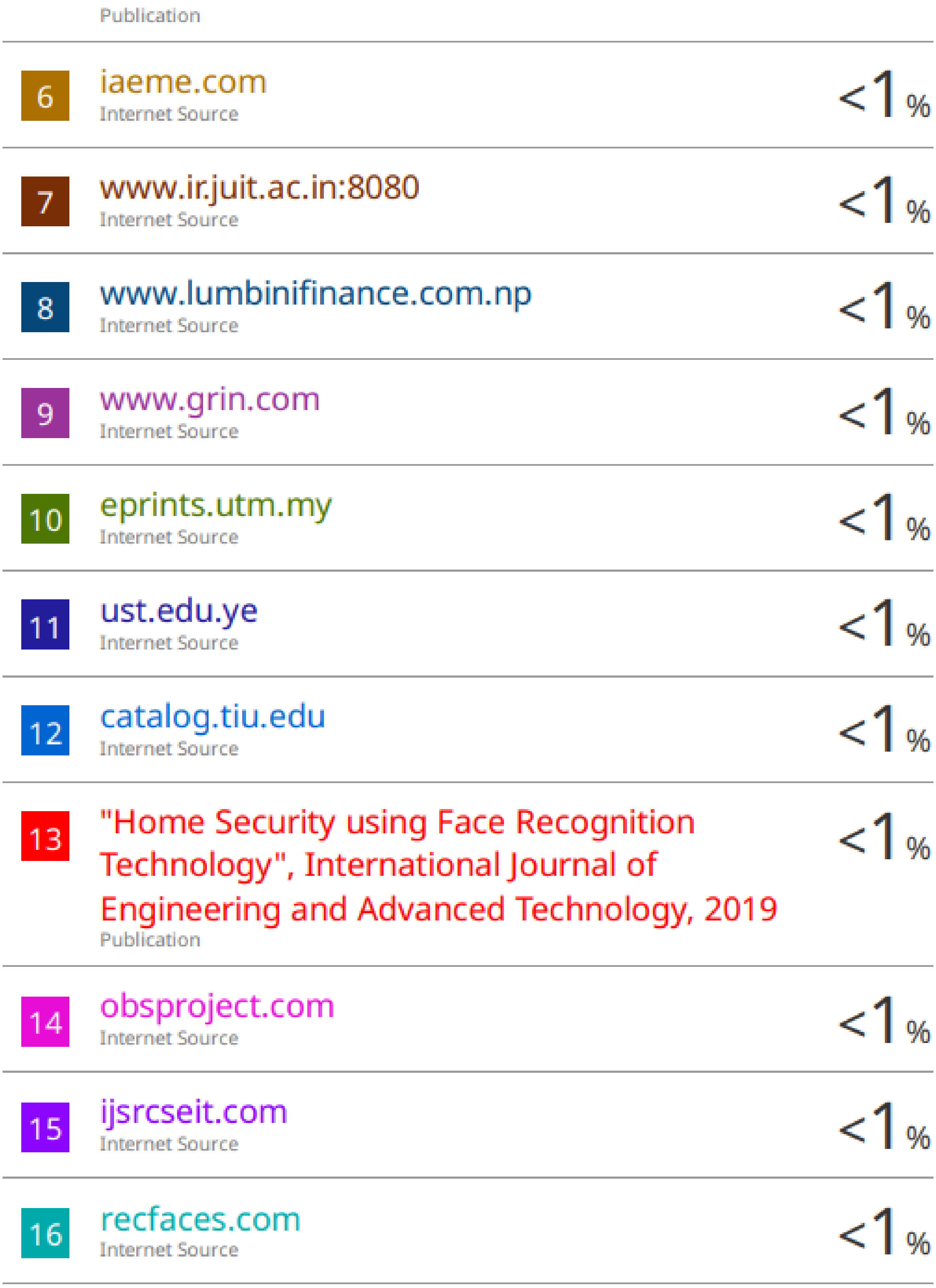
Second Check

ORIGINALITY REPORT

3%	3%	1%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

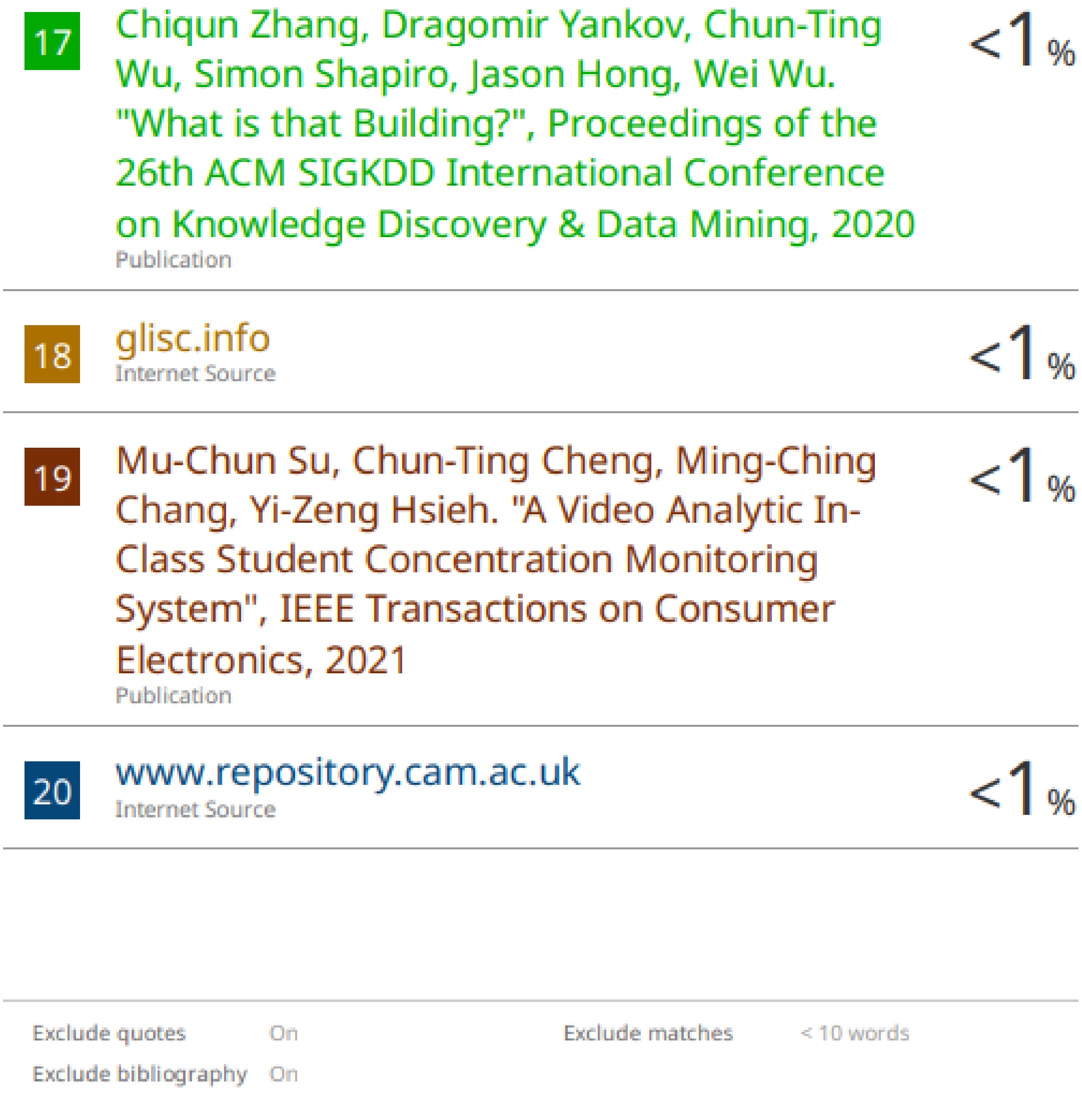
PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	1%
2	www.researchgate.net Internet Source	<1%
3	www.sweetstudy.com Internet Source	<1%
4	Lakshmana Rao Kalabarige, Kella Akhil Abhilash, Kondamudi Anirudh Trivedi, Mala Dathatreya. "Facial Landmark-based Cursor Control and Speechto-Text System for Paralyzed Individuals", 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2023 Publication	<1%
5	Somaieh Amraee, Maryam Chinipardaz, Mohammadali Charoosaei. "Analytical study of two feature extraction methods in comparison with deep learning methods for classification of small metal objects", Visual Computing for Industry, Biomedicine, and Art, 2022	<1%



Publication

6	iaeme.com Internet Source	<1 %
7	www.ir.juit.ac.in:8080 Internet Source	<1 %
8	www.lumbinifinance.com.np Internet Source	<1 %
9	www.grin.com Internet Source	<1 %
10	eprints.utm.my Internet Source	<1 %
11	ust.edu.ye Internet Source	<1 %
12	catalog.tiu.edu Internet Source	<1 %
13	"Home Security using Face Recognition Technology", International Journal of Engineering and Advanced Technology, 2019 Publication	<1 %
14	obsproject.com Internet Source	<1 %
15	ijsrcseit.com Internet Source	<1 %
16	recfaces.com Internet Source	<1 %



17	Chiqun Zhang, Dragomir Yankov, Chun-Ting Wu, Simon Shapiro, Jason Hong, Wei Wu. "What is that Building?", Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020 Publication	<1 %
18	glisc.info Internet Source	<1 %
19	Mu-Chun Su, Chun-Ting Cheng, Ming-Ching Chang, Yi-Zeng Hsieh. "A Video Analytic In-Class Student Concentration Monitoring System", IEEE Transactions on Consumer Electronics, 2021 Publication	<1 %
20	www.repository.cam.ac.uk Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ng Suet Eng
ID Number(s)	20ACB06580
Programme / Course	BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS
Title of Final Year Project	FACE RECOGNITION FOR IDENTIFY VERIFICATION IN EXAMS LOCATIONS

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: __3__ **%** **Similarity by source** Internet Sources: __3__% Publications: __1__ % Student Papers: __0__ %	
Number of individual sources listed of more than 3% similarity: _________	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) **Overall similarity index is 20% and below, and**

(ii) **Matching of individual sources listed must be less than 3% each, and**

(iii) **Matching texts in continuous block must not exceed 8 words**

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

SuLee

____
Signature of Supervisor

Name: Su Lee Seng

Date: 14th September 2023___________________________

____
Signature of Co-Supervisor

Name: ____________________________

Date: ____________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB06580
Student Name	Ng Suet Eng
Supervisor Name	Mr. Su Lee Seng

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date:15th September 2023