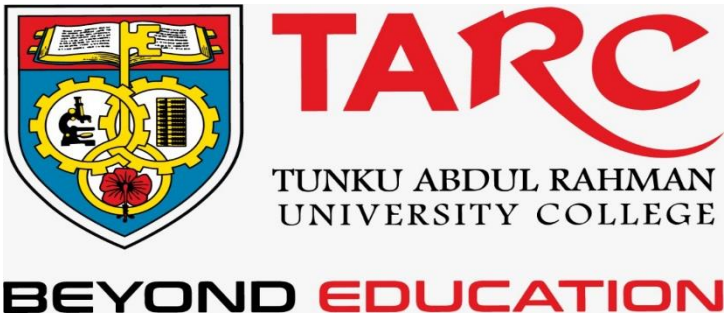
Smart Classroom System – Face Recognition and Attendance Module and Eye Tracking Module

By

Joan Hau



FACULTY OF COMPUTING AND
INFORMATION TECHNOLOGY

TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE
KUALA LUMPUR

ACADEMIC YEAR
2021/22

Smart Classroom System – Face Recognition and Attendance Module and Eye Tracking Module

By

Joan Hau

Supervisor: Ts. Dr Tew Yiqi

A project report submitted to the
Faculty of Computing and Information Technology
in partial fulfillment of the requirement for the
Bachelor of Computer Science (Honours)

Department of Software Engineering and Technology
Faculty of Computing and Information Technology
Tunku Abdul Rahman University College
Kuala Lumpur

Declaration

The project submitted herewith is a result of my own efforts in totality and in every aspect of the project works. All information that has been obtained from other sources had been fully acknowledged. I understand that any plagiarism, cheating or collusion or any sorts constitutes a breach of TAR University College rules and regulations and would be subjected to disciplinary actions.

_____

Joan Hau

Bachelor of Computer Science (Honours) in Software Engineering

ID: 20WMR09465

Abstract

The commencement of a smart classroom is an apparent rapprochement in the current stage of modern teaching. There are many sophisticated technologies in the smart classroom. The root of this research is to implement face detection and recognition algorithms in image processing to build a system that can detect and recognize the frontal face of students in the classroom. Face recognition is a very familiar biometric by all of us as it can be used for authentication, security and identification even in our daily life. The face is a unique representation of one's identity. All humans can recognize individuals from their faces. Face recognition can offer a fast, automatic, and seamless verification services experience as well as security for everyone. Face detection and recognition are believed to provide good analysis and academic services. This will definitely increase the value of a smart classroom in the future. The proposed solution is to develop real-time monitoring techniques and face recognition algorithms for the intelligent classroom system to evaluate students' performance and provide quick feedback for lecturer or tutor. Besides, iris and eyelid tracking are included for iris position estimation, which is used to analyze students' behaviour when communicating in class.

Acknowledgement

We are very grateful and especially thankful to our supervisor, Ts. Tew Yiqi, for frequently following our final project and spending his valuable time to discuss with us. We are grateful to our moderator, Ts. Lim Fung Ji, for guiding our progress and giving constructive feedback during the graduation project.

Table of Contents

Chapter 1

Introduction

1 Introduction

Education plays a paramount role in the modern technological world as it is an essential tool for getting a bright future. Through the rapid development of internet technology and artificial intelligence (AI), in order to provide better teaching services, a smart classroom has been appointed to modern education. In today's society, there are many schools and colleges, and faculty face the difficulties of proxy attendance, maintaining handwritten documentation of all student attendance for each cohort.

In this paper, we are going to use a facial recognition feature as a new technology for tracking the student's attendance and an eye tracking feature for monitoring the student's activities. The proposed system will be able to support online classrooms to keep track of the students' attendance and monitor student's activities in online classroom. In addition, a proper facial detection and recognition algorithm will be applied in order to ensure the efficiency and accuracy of the system proposed. Furthermore, this proposed system will require the students to register their face images in order to allow the system to recognise their faces. The proposed system will be less time consuming and increase productivity for the process of tracking students' attendance compared to the manual attendance system.

1.1 Objectives

The main objective of this paper is to develop a facial recognition attendance system and eye tracking feature in order to keep track of student attendance and monitor student's activities. The face recognition and attendance module and eye tracking module will track the students' attendance and monitor student through the webcam in order to identify the face detected.

There are some other objectives for face recognition and attendance module:

I. To detect the face from video

II. To extract the features of the face for detection

III. To classify the face detected in order to do the face recognition

IV. To track the attendance of the student detected.

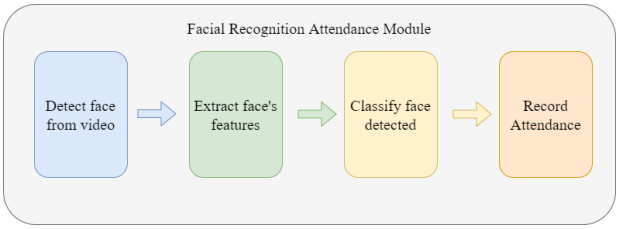


Figure 1.1 Overall Diagram of a General Facial Recognition

1.2 Research Question and Hypothesis

1.2.1 Research Question

The research questions were formulated and studied in the process of achieving the objectives.

1. How to increase the accuracy of face recognition?
2. How to increase the speed of face detection and recognition?

1.2.2 Hypothesis

The hypothesis made based on the question stated above:

1. The more the face images of a person being trained, the higher the accuracy for face recognition.
2. The higher the efficiency of a facial algorithm used, the higher the speed of face detection and recognition.

1.3 Background

1.3.1 Target Market

The target market for this project is the students and the academic staff of the higher education institution. Many kinds of research can prove that face recognition can be used for attendance tracking during the class (Tang, Zhou and Zheng, 2019; Dulyawit, 2019; Chin, 2018; Y. Kawaguchi et al., 2009) to increase the effectiveness of the attendance marking. Higher education institutions can use the intelligent classroom system to improve teaching management and process quality. Besides, facial recognition in the smart classroom system is able to decrease the time consuming and increase the effectiveness for the staff and students to take the attendance (Dulyawit, 2019).

1.3.2 Existing Similar System

1. Radio-frequency identification (RFID)

   Radio-frequency identification (RFID) is used by the event (Figure 1.2) held in Tunku Abdul Rahman University College (TARUC) to track the attendance and the time of the students when the event is carried out. The attendance and time will be recorded when the students run across the starting line of the trek. The attendance is tracked through the card number pin on the clothes of the participants.

Figure 1.2 RFID used for tracking attendance and time during the event

2. Fingerprint system

The fingerprint system is implemented in the University Sains Malaysia (USM) for students' attendance tracking (Figure 1.3). The fingerprint system will need the student to register their fingerprint before taking attendance during the class. The system will compare with the data store in the database in order to recognize the fingerprint captures from the students (Figure 1.4).

Figure 1.3 Fingerprint System used in USM

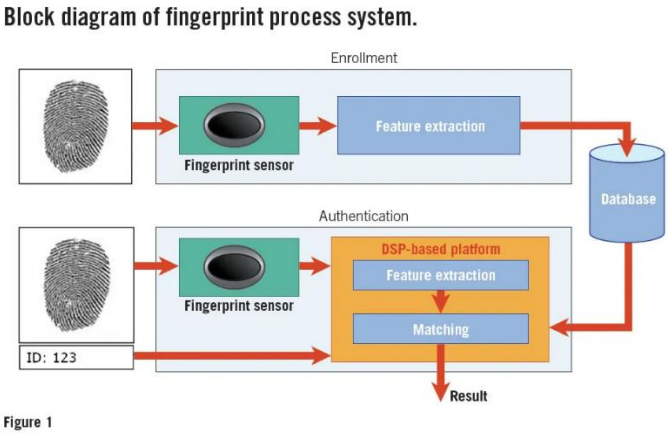


Figure 1.4 Block Diagram of the fingerprint process system (HSU, 2016)

3. Facial Recognition System

The facial recognition system is used by the University Utara Malaysia to track the attendance of the students in class. The faces of the students are captured by the closed-circuit television (CCTV) for further process in order to recognize the students and take attendance. The general process for the facial recognition system using CCTV is stated in Figure 1.5. The system is developed by the lecturer in UUM in 2019 and it only supports face-to-face situations.

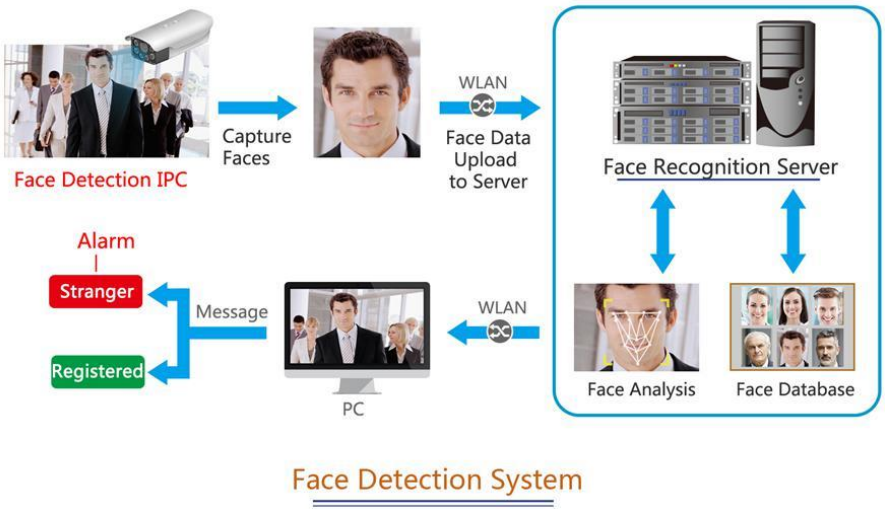


Figure 1.5 Facial Recognition System on CCTV

4. Beacons Sensor

The Bluetooth low-energy (BLE) beacon (UniSas) is proposed and implemented in University Tenaga National (UNITEN) for attendance tracking. A beacon is a small wireless device transmitting a continuous radio signal. The attendance is marked in such a way that the nearby smart devices will detect the signal via Bluetooth and the signal will inform the smart devices by giving a unique ID for attendance tracking (Figure 1.6).

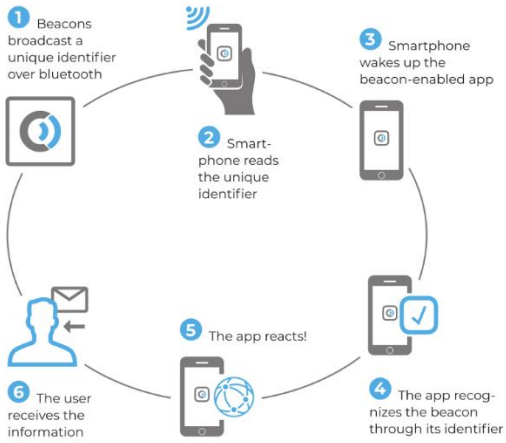


Figure 1.6 General process of a Beacons Sensor

5. QR Code for Attendance Tracking

   University Malaysia Sarawak (UNIMAS) is currently using the QR code for attendance registration. Every student in the class should download an application in order to scan the QR code for them to mark the attendance. Besides, the University Technology PETRONAS (UTP) used the QR code to track the students' attendance since 2018 (Figure 1.7).

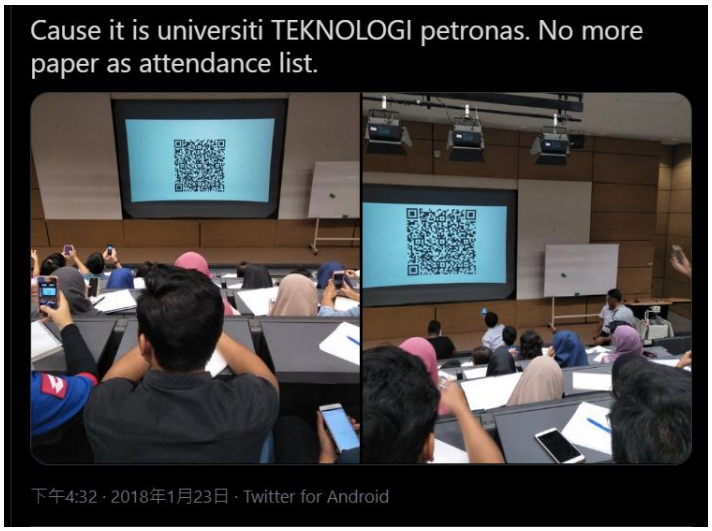


Figure 1.7 QR Code Scanning in Lecture Class

1.3.2.1 Summary of Existing Similar System

Table 1.1 Summary of Existing Similar System

System Type	Pros	Cons	Suitable
RFID	Higher efficiency as it can read multiple RFID tags simultaneously	The transmission of data in an RFID tag may be influenced by certain materials such as heavy metals and sources of radio waves.	Only support face-to-face situations.
Fingerprint	Higher accuracy to identify the students to keep track of attendance during class.	Required unique biometric property, students may face difficulties because of injuries that make the fingerprint become unreadable.	Only support face-to-face situations.

Facial Recognition	Fully automated which is able to track attendance automatically.	The camera angle will affect the accuracy of the recognition as the system might not detect any faces if the angle is not facing the student.	Only support face-to-face situations as it is implemented on CCTV.
Beacons Sensor	The beacon sensors have a range of 30 meters inside buildings which are able to ensure the students do not cheat the attendance.	Less accuracy as the radio signals will be affected by the water, air, human body and even metal.	Support only face-to-face situations within 30 meters.
QR Code	High convenience as the students can scan anywhere at any time.	Everyone is required to download the specific application in order to register the attendance.	Support both face-to-face and online learning, but a high possibility for attendance cheating.
Proposed System	It is less time consuming and can process multiple faces simultaneously.	Required to browse the system in order to track the attendance by the lecturer.	Support both face-to-face and online learning, ensure the reality of attendance as face recognition is carried out.

1.3.3 Potential Work

The proposed system will be using face recognition technology which also includes IoT to capture the students' attendance and carry out monitoring at the same time. Every student will need to register their face before being recognized by the system. The system will train the images captured in order to recognize them in the future. The trained images will generate a data file which will be used for comparing the faces captures in the video recording for comparisons. The system will update the attendance once the students' face is recognized.

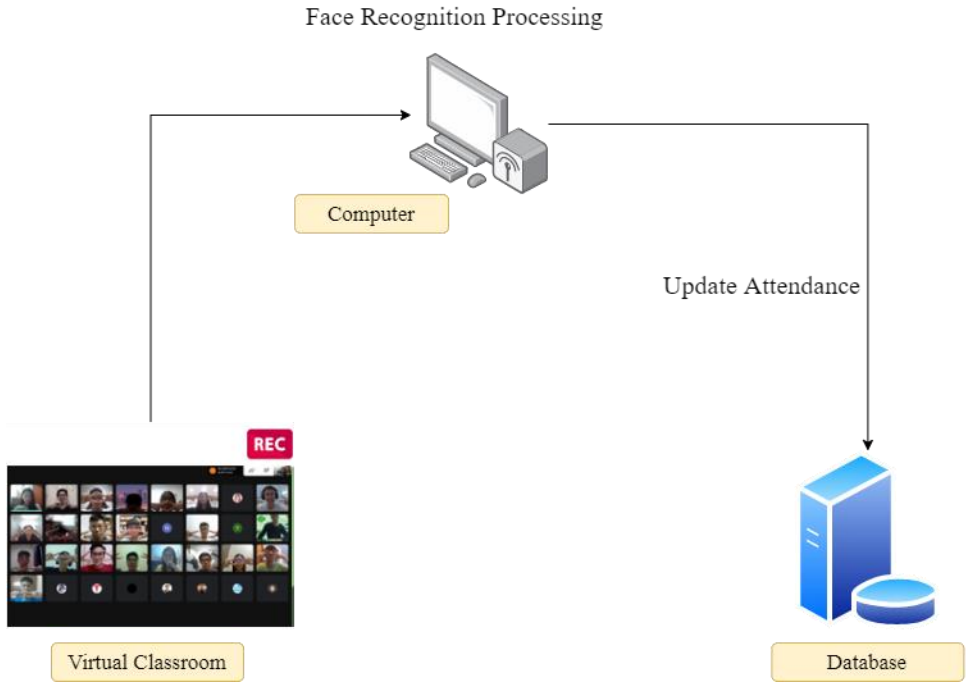


Figure 1.8 Project Structure

1.4 Advantages and Contributions

This intelligent classroom system with face recognition in tracking attendance is more convenient than other monitoring methods such as scanning QR codes, registering class code keys for attendance, or taking the attendance manually. This function is more straightforward for both parties, including teachers and students, to save their time in attendance marking. Facial recognition can avoid attendance cheating compared to other techniques. The teachers are able to ensure the authenticity of class attendance. Other than that, these features also enable the teachers to monitor the students during the teaching session, which can be teaching management. This feature provides better than other techniques using a mobile application or paper-based work. The smart classroom system with face recognition can increase the teachers' productivity and decrease the time consuming during the class session for virtual classrooms. Besides, face recognition is able to keep track of the attendance of the students rapidly.

This system is able to improve the efficiency of attendance tracking and also provide monitoring for the students at the same time. In addition, it is also able to automatically detect multiple faces simultaneously during the attendance tracking process. High accuracy and less time consuming provided by the system. The student's faces will be captured and trained by the machine learning model in order to ensure the accuracy of the face recognition process and also avoid any error detection occurring during the operation. Furthermore, it is able to prevent any attendance cheating among students for virtual class as the students need to open the computer camera for face tracking during the attendance marking process.

1.5 Project Plan

1.5.1 Project Specification

The face recognition attendance module will cover a few functional and non-functional specifications:

a. Functional Specification
   I. The system should be able to capture the students' face
   II. The system should be able to recognise the students' face through a web camera and video capture.
   III. The system should be able to show the unique information (e.g., student id or student name) of the student when the face is being recognized.
   IV. The system should be able to track the students' attendance.

b. Non-functional Specification
   I. The system should be able to train the students' faces captured smoothly without any failure.
   II. The system should be able to recognize the student in a short period of time.

1.5.2 Milestones

Table 1.2 Project Milestone

Task	Remarks	Deadline
Project Proposal	Project Proposal Submission	8 Mach 2021
Chapter 1: Introduction	Identify the objectives, research question and hypothesis, background, advantages and contributions of the project.	29 March 2021
Chapter 2: Literature Review	Done the research and literature review of the project	12 April 2021
Chapter 3: Methodology and Requirements Analysis	Identify the methodology used for the project and analyse the requirements needed for the project.	25 June 2021
Chapter 4: System Design	Based on the requirements analysed, prepare the system design.	5 July 2021
Implementation	Apply the final coding based on the research.	23 August 2021
Integration and Testing	Design the test plan, test data, test cases for the system testing	8 November 2021
Deployment	Submit the Final Report	15 November 2021

Table 1.3 Gantt Chart

ID	Task Mode	Task Name	Duration	Start	Finish
1		**System Planning**	**215 days**	**Mon 18-01-21**	**Fri 12-11-21**
2		Find Supervisor	6 days	Mon 18-01-21	Sat 23-01-21
3		Supervisor Briefing	1 day	Sat 23-01-21	Sat 23-01-21
4		Draft Proposal Preparation	7 days	Sat 23-01-21	Sun 31-01-21
5		Drafy Proposal Checking	1 day	Sun 31-01-21	Sun 31-01-21
6		Finsl Proposal Preparation	27 days	Sun 31-01-21	Mon 08-03-21
7		Final Proposal Submission	1 day	Mon 08-03-21	Mon 08-03-21
8		**Chapter 1: Introduction**	**15 days**	**Tue 09-03-21**	**Sun 28-03-21**
9		Objectives	1 day	Tue 09-03-21	Tue 09-03-21
10		Research Question	1 day	Wed 10-03-21	Wed 10-03-21
11		Hypothesis	1 day	Thu 11-03-21	Thu 11-03-21
12		**Background**	**4 days**	**Fri 12-03-21**	**Wed 17-03-21**
13		Target Market	2 days	Fri 12-03-21	Sat 13-03-21
14		Existing Similar System	3 days	Sun 14-03-21	Tue 16-03-21
15		Potential Work	1 day	Wed 17-03-21	Wed 17-03-21
16		Advantages and Contributions	2 days	Thu 18-03-21	Fri 19-03-21
17		**Project Plan**	**4 days**	**Tue 23-03-21**	**Fri 26-03-21**
18		Project Specification	1 day	Tue 23-03-21	Tue 23-03-21
19		Milestone	1 day	Wed 24-03-21	Wed 24-03-21
20		System Development Model	1 day	Thu 25-03-21	Thu 25-03-21
21		Testing Approach	1 day	Fri 26-03-21	Fri 26-03-21
22		Project Team and Organization	1 day	Sat 27-03-21	Sat 27-03-21
23		Chapter Summary and Evaluation	1 day	Sun 28-03-21	Sun 28-03-21
24		**Chapter 2: Literature Review**	**11 days**	**Mon 29-03-21**	**Mon 12-04-21**
25		**Company Background**	**3 days**	**Mon 29-03-21**	**Wed 31-03-21**
26		Project Background	3 days	Mon 29-03-21	Wed 31-03-21
27		Literature Review	5 days	Thu 01-04-21	Wed 07-04-21
28		**Feasibility Study**	**3 days**	**Thu 08-04-21**	**Sun 11-04-21**
29		Operational Feasibility	1 day	Thu 08-04-21	Thu 08-04-21
30		Technical Feasibility	1 day	Thu 08-04-21	Thu 08-04-21
31		Financial Feasibility	1 day	Fri 09-04-21	Fri 09-04-21
32		Datasets	2 days	Sat 10-04-21	Sun 11-04-21
33		Chapter Summary and Evaluation	1 day	Mon 12-04-21	Mon 12-04-21
34		**Chapter 3: Methodology and Requirements Analysis**	**8 days**	**Tue 13-04-21**	**Thu 22-04-21**
35		**Methodology**	**1 day**	**Tue 13-04-21**	**Tue 13-04-21**
36		Fact Gathering	1 day	Tue 13-04-21	Tue 13-04-21
37		Fact recording	1 day	Tue 13-04-21	Tue 13-04-21
38		**Requirement Analysis**	**4 days**	**Wed 14-04-21**	**Mon 19-04-21**
39		Project Scope	1 day	Wed 14-04-21	Wed 14-04-21
40		Development Environment	2 days	Thu 15-04-21	Fri 16-04-21
41		Operation Environment	2 days	Thu 15-04-21	Fri 16-04-21
42		Non-functional Requirements	1 day	Mon 19-04-21	Mon 19-04-21
43		**Requirement Diagram**	**2 days**	**Tue 20-04-21**	**Wed 21-04-21**
44		Functional Requirements	2 days	Tue 20-04-21	Wed 21-04-21
45		Chapter Summary and Evaluation	1 day	Thu 22-04-21	Thu 22-04-21
46		First Stage Implementation	42 days	Fri 23-04-21	Mon 21-06-21
47		**Chapter 4: System Design**	**9 days**	**Tue 22-06-21**	**Fri 02-07-21**
48		**System Design**	**8 days**	**Tue 22-06-21**	**Thu 01-07-21**
49		Activity Diagram	2 days	Tue 22-06-21	Wed 23-06-21
50		Firebase Data Structure	3 days	Tue 22-06-21	Thu 24-06-21
51		User Interface	5 days	Fri 25-06-21	Thu 01-07-21
52		Chapter Summary and Evaluation	1 day	Fri 02-07-21	Fri 02-07-21
53		Second Stage Implementation	36 days	Mon 05-07-21	Mon 23-08-21
54		**Integration and Testing**	**44 days**	**Tue 24-08-21**	**Fri 22-10-21**
55		Test Plan	10 days	Tue 24-08-21	Mon 06-09-21
56		Test Data	10 days	Tue 07-09-21	Mon 20-09-21
57		Test Cases	10 days	Tue 21-09-21	Mon 04-10-21
58		Fix Defects Found	14 days	Tue 05-10-21	Fri 22-10-21
59		Final Stage Program Compilation	14 days	Mon 25-10-21	Thu 11-11-21
60		**Deployment**	**1 day**	**Fri 12-11-21**	**Fri 12-11-21**
61		Final Report Submission	1 day	Fri 12-11-21	Fri 12-11-21

1.5.3 System Development Model

The software development model that is suitable for this project is an incremental model approach. There are various phases of the life cycle of incremental models, which are requirement analysis, design coding, testing, and implementation phase. Each iteration in the incremental model will pass through these phases stated in Figure 1.9. The face recognition modules will be separated into a few functions in order to ensure the quality of the outcome of the module. The face recognition attendance module will be separated into a few increments:

Increment 1: Face Detection using web camera

At the first incremental of this module, the algorithm will be applied in order to ensure the face can be detected by the web camera accurately. The face that is detected will be covered by a coloured square in order to inform the viewer that the face is being detected. Once the implementation of this function has completed, it will proceed to the testing phase to test and debug each of the functions implemented. If any defects are found, then the defects will be fixed. After all the function is well performed, then will proceed to the following increment.

Increment 2: Captures faces and save them into firebase storage

Besides, after ensuring the face can be detected accurately, the captures face function will be carried out in this incremental. The faces that want to be registered must go through this process in order to let the system capture their faces and save them into firebase storage for further training and references. After the function is being implemented, it will go through all the necessary process in the incremental model in order to ensure the quality of the function.

Increment 3: Train the faces using a machine learning model

In the third incremental, the system will train the faces by using the machine learning model in order to carry out the face recognition function in the following incremental. The faces that are captured will be trained by the system. The system will go through the testing process to minimize the defects of the system.

Increment 4: Face recognition through web camera

In the fourth incremental, the face recognition function will be applied in order to recognize the faces detected. The system will go through the testing process in order to make sure it able to recognize faces through a web camera.

Increment 5: Face Recognition through video capture

In the fifth incremental, the face recognition through the video capture function will be declared in order to capture the face detected and carry out the face recognition at the same

time. The testing process will be carried out once the function is implemented, the system will be tested by multiple videos in order to ensure it able to recognize the faces shown in the videos.

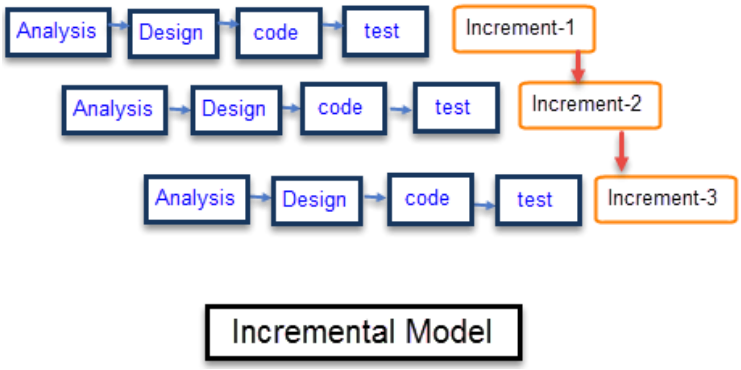


Figure 1.9 Incremental Model Life Cycle

1.5.4 Testing Approach

1. Unit Testing

   Unit testing is used to test each of the components in the module in order to ensure the functions implemented meet the expectations. The components such as face detection functions, face recognitions functions will be tested individually during this testing. For instance, the face recognition function will be tested to ensure it is able to recognise the face that is being trained.

2. Module Testing

   Module testing is to make sure that all the functions are well performed in the module after the combination. The combinations of each of the functions within the same module will be tested in the module testing. For example, the captured faces and train faces within the module will be tested together in order to ensure they perform their own work correctly.

3. Integration Testing

   After all the modules within the system are integrated, this testing will be carried out to ensure the performance and the functionality of every module is well functionally and able to cooperate with each other. The combination of face recognition and gestures detection module will be tested using this testing approach.

4. System Testing

   Once the system is well prepared, the system testing will be performed in order to ensure it meets the expected functional and non-functional requirements.

5. Acceptance Testing

   This testing will be performed by any interested user who will use this system to test whether the system is satisfied and is accepted by the user.

1.6 Project Team and Organization

Table 1.4 Project Team with Module Handled

System and Subsystems	Lim Kah Yee	Joan Hau
Smart Classroom System		
Hand Gesture Recognition Module	✓	
Face Recognition Attendance Module and Eye Tracking Module		✓

1.7 Chapter Summary and Evaluation

This chapter will introduce the project that I wanted to carry out which include the objectives, the background of the project, the advantages and contributions of this project and so on. This chapter will have a simple concept of how the modules will be implemented which also include some other existing similar system as a real-life example. The project is mainly focusing on attendance tracking and monitoring which is able to track the attendance and monitor the user. The system is suitable for all the organizations which need to track the attendance. There are also some of the similar existing attendance systems with the block diagram of the process discussed in this chapter so that one can have a better understanding of the concept of other systems.

Chapter 2

Literature Review

2 Literature Review

This chapter will discuss the background of the project and conduct a literature review by examining relevant work done by others. In addition, it will discuss the feasibility studies that need to be looked at when doing a project. These will help to build on previous work to have a better understanding of the design and implementation of the project.

2.1 Company Background

Company background will discuss the nature of business, services, organization and current system used and other relevant works that can be used to improve understanding of this project.

2.1.1 Nature of Business

Tunku Abdul Rahman University College is the foremost higher education institution established by the MCA in 1969 to provide high quality and an economic higher level of learning to Malaysians. TAR College was named after Allahyarham YTM Tunku Abdul Rahman Putra Al-Haj, Malaysia's first prime minister. It was established to collaborate and supplement Malaysia's efforts to fulfil the country's current and future manpower needs. Thereafter, it successfully ascended to the status of University College on May 2, 2013, which opened up new horizons for further academic exploration and development of University College. The University College has its major campus located in Kuala Lumpur and five branches in Penang, Perak, Johor, Pahang, and Sabah, respectively.

2.1.2 Services

The University College delivers more than 100 pre-university, diploma, and degree level programs. These programs cover a broad field, from specialist courses in such areas as accountancy, the engineering and building and construction environment, as well as economics and management, business and finance, and ICT, to subjects such as the sciences of application, in microelectronics, mass communication, and in creative arts, hospitality management and in social sciences, as well as foundation and A-level courses.

TARUC's university education is accredited both locally and nationally by both academic and specialist institutions. TARUC has produced over 170,000 alumni, many of whom are outstanding leaders in various sectors, thanks partly to the comprehensive education offered by TARUC.

Along with 46 years of outstanding educational achievements, TARUC is well-positioned to become a leading institution recognized nationally and globally for quality education.

2.1.3 Current Systems Used

Usually, University College uses the traditional face-to-face approach to education. Students at University College are required to attend classes in person at a set time and place. The management of the University College arranges the schedule. In addition, University College uses a manual attendance system. As a result, lecturers take attendance by calling students by name one by one or by passing around attendance sheets for students to sign if there are many students in the class. Besides, the TAR mobile application also being used for tracking the student attendance in 2019. The students are able to register the attendance as long as the code is correct and the IP address is in campus range. The attendance registration module in the mobile app will require students to enter a 6-digit code to register for attendance, and the system will track the IP address of the student's phone. The TARUC mobile app allows instructors to register for attendance during face-to-face classes by using the app to generate a random unique code for the student. As a result of the abnormal situation now, university colleges have changed their teaching methods face-to-face to online learning. As a result, instructors needed to manually sign attendance for each student who attended the online classroom. Due to online learning, attendance was again changed to manual. In addition, University College also uses Google Classroom Attendance for courses involving many students, such as lecture class.

2.2 Project Background

2.2.1 Nature of Business

Face recognition and attendance features in a smart classroom are new approaches and solutions for changing traditional teaching methods and environments. The proposed project provides an automated attendance system using face recognition techniques to help the teachers take their class's students attendance. The proposed project also comes with emotion tracking, gender tracking, and face mesh. This proposed project can monitor the students whether they are entering the class and are only allows to take the attendance using facial recognition, especially in an online classroom. For example, when the student enters the class, the student will be required to open the camera to take attendance using the face. This proposed project can improve the productivity of the lecturer in monitoring the students and increase the quality of the online classroom.

2.2.2 Existing Similar Products and Services

There is some existing system that is similar with the proposed project such as iFace II and faceATT. iFace II is an advanced facial recognition system with a built-in fingerprint used for staff attendance and door access. iFace II provides a very accurate facial recognition system by

capturing the position and size of the face, shape of eyes, and features of the nose, cheekbones, and jaw when registering the face. The accuracy of the iFace II will not be affected even the user wears the spec or getting fat. It is able to compare and verify the staff ideally.

Besides, faceATT is an attendance recording system with artificial intelligence-based. It can seamlessly record an individual attendance through the face and geo-fencing tools. FaceATT does not require any additional hardware to operate as it is able to perform with our smartphone. It also provides the attendance history for both management and users to view at any time. FaceATT is designed to thoroughly address time and attendance problems, especially during coronavirus restrictions, and eliminate attendance fraud at the same time.

2.2.3 Target Users

The target market of the proposed project is the students and lecturers of the universities and colleges. It is also able to bring forth to both primary and secondary school some advantages. Fraud attendance is able to eliminate through the facial recognition attendance module. The students and lecturers can avoid any missed attendance during the class, and it also provides us a new standard of a learning environment.

2.3 Literature Review

2.3.1 Face Detection

Face detection and face recognition usually make us confused about the differences between them. Face detection is mainly used to identify the face segments or face regions from an image, while face recognition will identify the individual's face with personal information. Although face detection and face recognition is matured in today's society, they will also face some difficulties during the process (S.Aanjanadevi et al., 2017; Wei-Lun Chao, 2007). The problems are listed in Table 2.1.

Table 2.1 Difficulties of Face Detection (S.Aanjanadev et al., 2017)

Difficulties	Explanation
Background	Changes in the background and surroundings of the person in the image will influence the face recognition accuracy.
Light Level	Various lighting environments reduce the ability to detect facial features.

Pose	The different angles of the captured facial images distort the face recognition process.
Expression	Changes in expressions cause changes in spatial relationships and changes in the shape of facial features.
Occlusion	If there is a part ofthe face that is not observable, it will affect the performance and face recognition due to the not enough information provided.
Rotation, scaling and translation	Transformation of the image may distort the original information of the image.

2.3.2 Image Preprocessing

The face detected from an image is suggested to crop out only the face for further process (Subhi Singh et al., 2015). Any colored image will convert to grayscale for image pre-processing. Also, the face detected will then be aligned based on the eye's position and the scale of the image. Arun Katara et al. (2019), Akshara Jadhav et al. (2017), Shireesha Chintalapati and M.V. Raghunadh (2013), all these three papers proposed to apply histogram equalization to facial images and pre-process the images by scaling.

Pre-processing can improve the performance of the system (Chin, H., 2018). It plays a crucial role in improving the accuracy of face recognition. Scaling is one of the necessary preprocessing steps for processing the size of the image (Chin, H., 2018). Due to the reduced number of pixels, scaling of images can increase the processing speed by reducing the system computation. The image's size and pixels contain its unique spatial information. The spatial information is crucial as it is a measurement of the least recognizable detail in an image (Gonzalez, R. C. and Woods, 2008). Therefore, spatial information must be handled carefully to avoid image distortion and to prevent tessellation effects. For normalization and standardization purposes, the dimensions of all images should be the same. According to Subhi Singh (2015), the length and width of the image are preferred to be the same size based on the proposed PCA (Principal Component Analysis).

Figure 2.1 Checkerboard Effect Increasing from Left to Right (Gonzalez, R. C., & Woods,2008)

In addition to image scaling, colour images are often converted to grayscale images for pre-processing. Grayscale images are considered to be less sensitive to lighting conditions and require less computation time (Chin, H., 2018). A grayscale image is an 8-bit image with pixels ranging from 0 to 255, while a colour image is a 24-bit image with pixels that can have 16 77 7216 values (Chin, H., 2018). Therefore, colour images require more storage space and more computational power than grayscale images. (Kanan and Cottrell, 2012). If the colour image is not necessary for the computation, then it is considered as noise. Moreover, pre-processing is essential to enhance the contrast of the image. In the paper by Pratiksha M. Patel (2016), he mentioned that histogram equalization is one method of pre-processing to improve the image's contrast. It provides a uniform intensity distribution on the horizontal axis of intensity and can reduce the effect of uneven illumination at the same time.

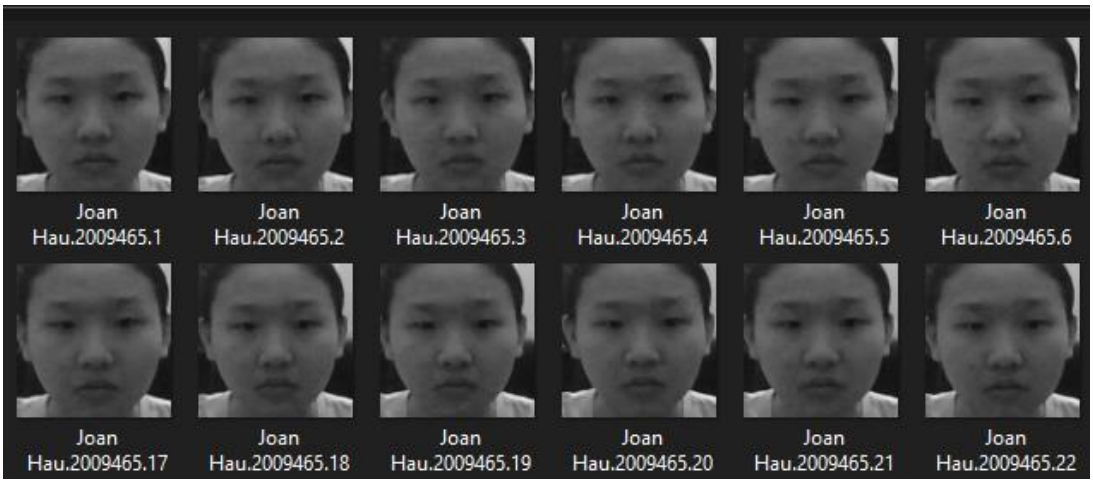


Figure 2.2 Images converted to grayscale and resized

2.3.3 Face Recognition

Facial recognition is a method used to identify or verify an individual from either image, video frame or real-time. It is commonly used as access control in security systems and can be compared to other biometric techniques such as fingerprint or eye iris recognition systems (Virgil Petrescu, 2019). The face recognition will capture the face from a group of faces, and then it will identify the details of the face (Figure 2.3). After that, it will match the faces available in the current storage location, and it will find the exact face that corresponds with the current face in the database.

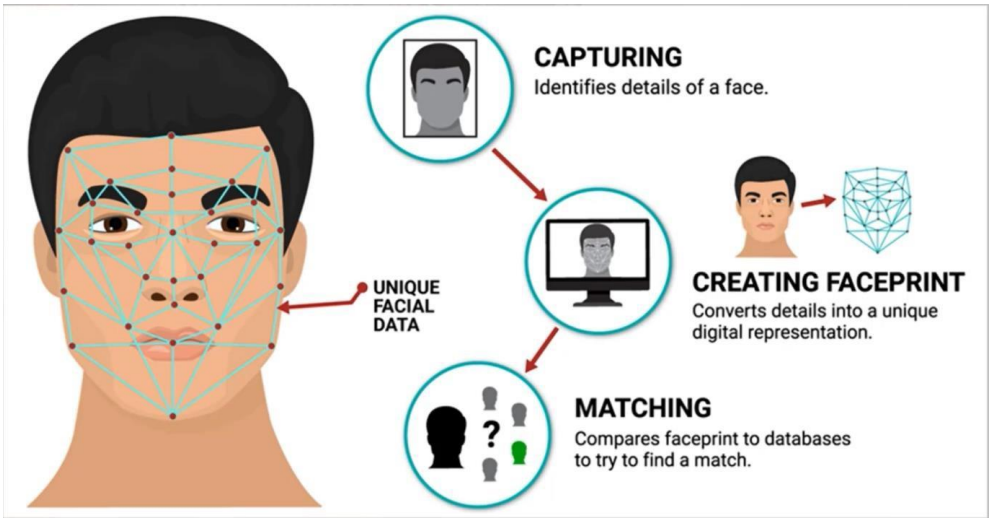


Figure 2.3 Facial Recognition Process

There is a total of four basic steps that involve in the face recognition process:

1. Face Detection

   Face detection is a process of detecting the face that is located in a frame or image by finding the landmarks of the face such as eyes and nose.

2. Face Alignment

   Face alignment plays an important role during the image pre-processing. The face and eye regions are automatically detected and faces are aligned according to translation, scaling and rotation. As shown in Figure 2.4, the person's face is a bit slanted to the right. The eye lens is detected from the face and straightening the eye lens in order to align the face correctly. The face alignment is essential in image pre-processing as it is able to make the face detected more readable for the system to track the features.

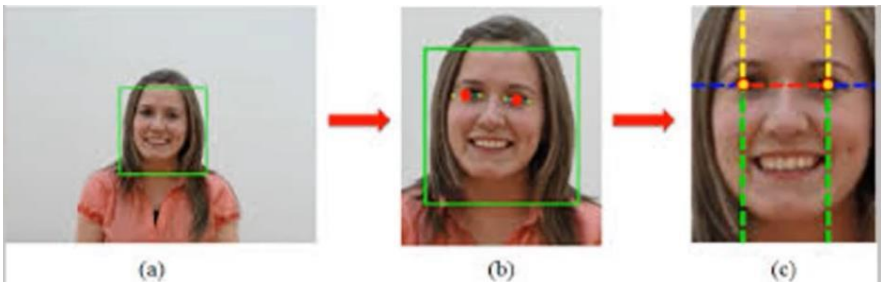


Figure 2.4 Face Alignment Process

3. Features Extraction

After that, once we have the pre-processed face ready to be passed into the system. The facial recognition will be producing 128 dimensions embeddings. The face that we pass into the system, the system will check the face and generate 128 numbered measurements from the image in order to let the computer read the image as it cannot actually observe the complete visual appearance of the image. The number generated is used by the computer for comparison.

```
[-0.16597123  0.11949642  0.10122138 -0.12254387 -0.11838274 -0.02759848
  0.01384261 -0.07288168  0.06527785 -0.13963827  0.17559825 -0.08519795
 -0.25020906 -0.0102928  -0.01079375  0.21948609 -0.21553843 -0.19072869
 -0.13214052 -0.10856642  0.01477439  0.07908501 -0.05431245  0.11583221
 -0.1715932  -0.23868942 -0.10927209 -0.01475592  0.01510147 -0.08837818
 -0.00252565  0.03029939 -0.17696191  0.00717495  0.04937807  0.12862331
 -0.01091681 -0.09640069  0.1934087   0.01389313 -0.23863317  0.01766105
  0.15954995  0.26412117  0.21739453 -0.00336392 -0.03702538 -0.05539587
  0.10163558 -0.28075284  0.05129996  0.20316276  0.02803321  0.16820049
 -0.02042791 -0.12237336 -0.00515839  0.22946236 -0.21929839  0.03012473
  0.10334508 -0.19385433  0.010464   -0.1046315   0.159848    0.14616895
 -0.19181257 -0.22495562  0.19514428 -0.23470432 -0.11755507  0.07461189
 -0.0839375  -0.09592333 -0.35081255  0.03915626  0.32692659  0.20959754
 -0.10889891 -0.01162777 -0.09144997 -0.02747694  0.00839157  0.15324235
 -0.02081387 -0.0425766  -0.06370837  0.0559925   0.19876704  0.04837478
  0.00898859  0.25267416  0.04914283  0.0009048   0.00353496  0.14037418
 -0.15166469 -0.07702    -0.10442869  0.03603458 -0.10455432 -0.05006842
  0.01399595  0.18746354 -0.10759938  0.24044091 -0.03882578 -0.01190661
 -0.12068785  0.03638472 -0.02094681 -0.02857181  0.10984001 -0.22919109
  0.14735889  0.20415246 -0.00772358  0.09734803  0.10852756  0.12523061
 -0.03349054 -0.08732443 -0.17586364 -0.09488015  0.04729999 -0.0654182
  0.10575557  0.07663471]
```

Figure 2.5 128 Measurements Generated from Images

4. Face Recognition or Classification

The system will compare the measurements that we obtained from the input image to what we already have in our database. After that, we will get the score for each and every match, if the score is more than a particular threshold. For instance, if the score is more than 80% or 90%, we can declare that is a match.

2.3.4 Haar Cascade Algorithm

The Haar Cascade Algorithm is used for face detection in this face recognition module. It is an object detection algorithm for recognizing faces in images or live video. The edge or line detection features are used in the haar cascade algorithm. The algorithm is trained by many images with or without human faces. Haar features (Figure 2.7) on the image can easily identify the edges and line found in the image or find out the regions that are abrupt changes in pixel intensity and figure it out.

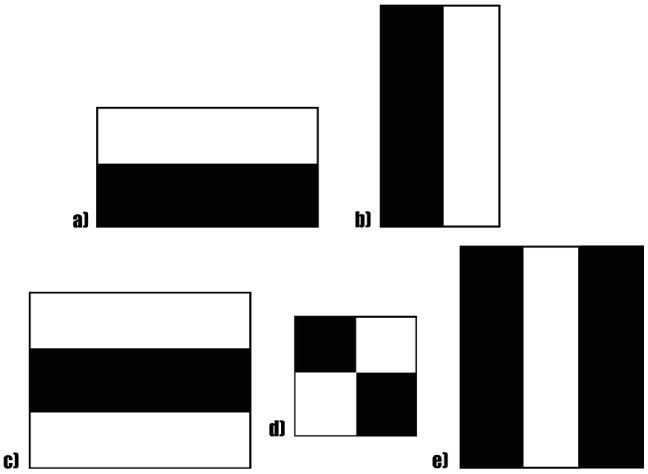


Figure 2.7 Haar Features

An example of computing Haar values from rectangular image sections is shown in figure 2.7. The rectangles located on the left side of figure 2.7 represent an image with 0 to 1 of the pixel value. The haar kernel is shown in the center of figure 2.7, which contains bright pixels on the left and dark pixels on the right. The center rectangle is a haar kernel with all the bright pixels on the left and dark pixels on the right. A haar value is used to calculate the difference between the average pixel value in dark and bright regions. An edge will be detected if the difference between them is close to 1. The value 1 represents darker regions, while the value 0 represents bright regions in Haar features. They are used to find the particular features in the image, such as edge, line, or any element that suddenly changes in intensity.

In addition, the haar features will travel along with the images from the beginning of the image's corner until the end of the image to search for the features inside the image. In order to avoid the time complexity as the traversal of the haar features along the images will involve lots of calculations, therefore an integral image is used to carry out the same process with less time complexity. Figure 2.9 shows the way of making an integral image. Each of the pixels located in the integral image is the same with the total of the pixels located at specific pixel's left and

above that shown in the original image. The time complexity for the calculation can be reduced by using the integral image.

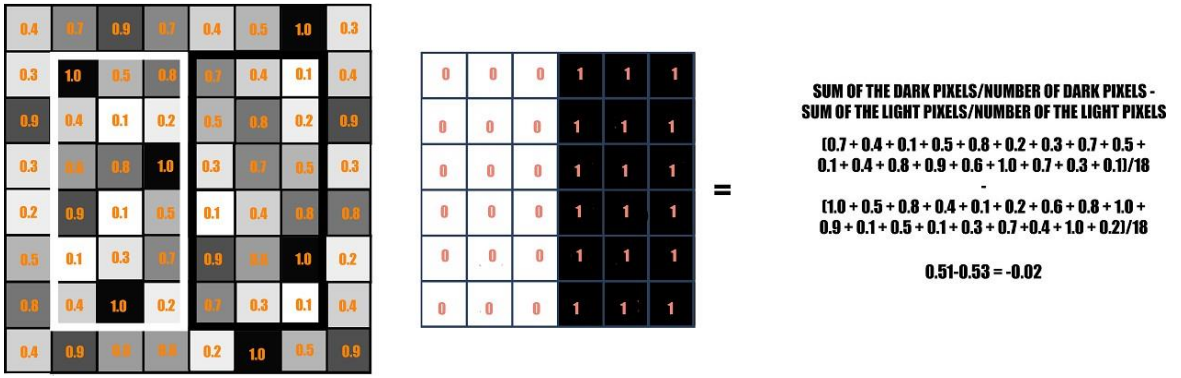


Figure 2.8 Calculation of Haar Value

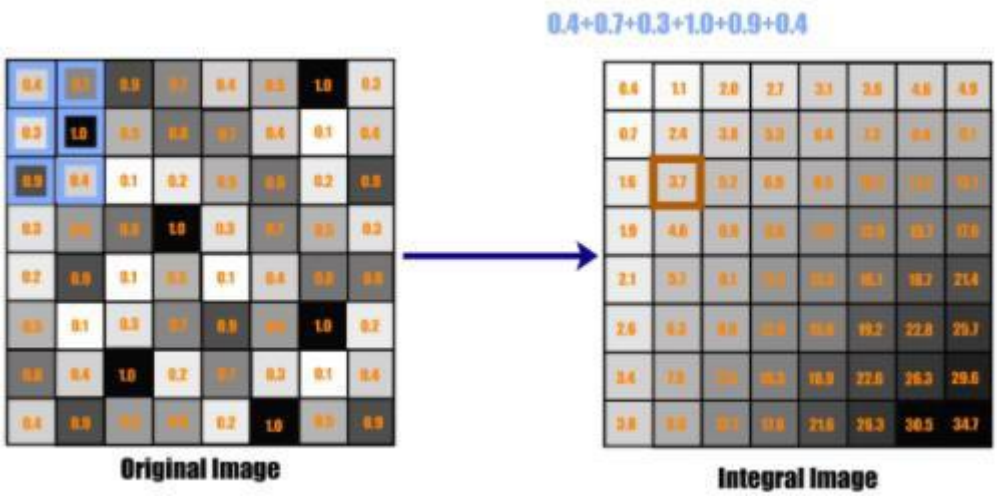


Figure 2.9 Sample Calculation of making integral images

2.3.5 Iris Detection

Biometrics is a reliable method for the authentication of personal identity. Due to the unique, distinctive, and stable of the human's iris pattern throughout human life, this non-invasive verification technique for identifying individuals is more practical (Biswas, R., 2017). The steps involve for iris detection and recognition are segmentation, normalization, feature encoding, feature extraction and training. The main flow of the iris recognition algorithm is depicted in figure 2.10. The picture is first captured using special gear with a built-in megapixel camera. The eye part of the picture is now separated, and advanced image processing techniques are used to segment the inner and outer margins of the iris. The iris in the isolated eye picture is identified and isolated at this stage, and it is encoded using mathematical techniques to generate a code with a distinct iris quality. Although any two pictures taken at various times and under

different conditions will not be precisely the same, this method confirms if the iris belongs to that individual or not (Jauro, S et al., 2018).

.

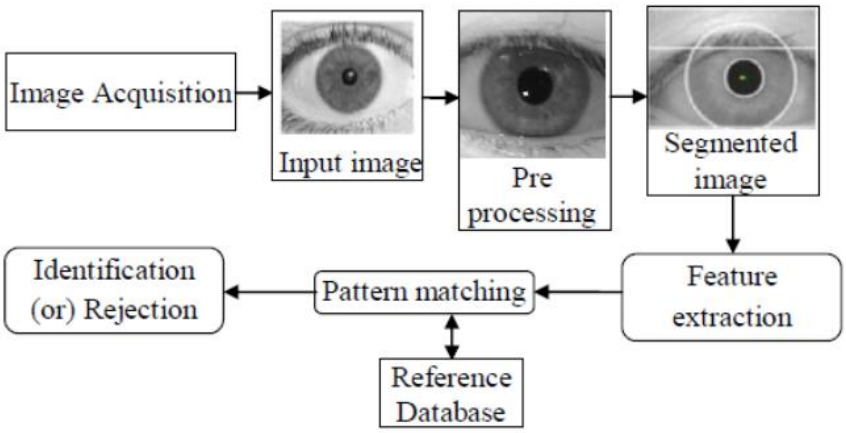


Figure 2.10 General Flow of Iris Detection and Recognition (Jauro, S et al., 2018)

2.3.6 Lie Detection

Psychologists have conducted numerous studies to try to establish the behavioral correlates of lying. According to the Richard Wiseman, Caroline Watt, Leanne ten Brinke, Stephen Porter, Sara-Louise Couper, Calum Rankin (2012), a lie detection experiment is carried out by them and the results indicates that a person who look to the upper right is lying while a person who look to the upper left when talking is telling truth to the listener.

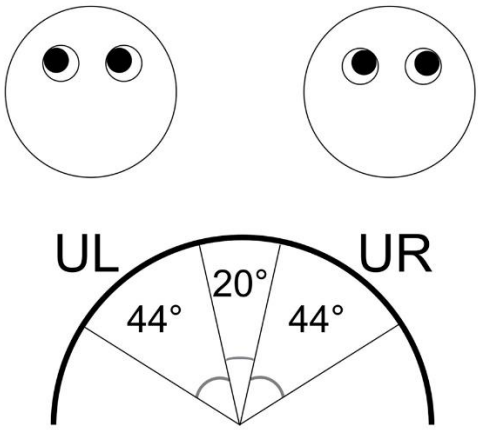


Figure 2.11 Area of Eye Movement classified as Upper Left or Upper Right (Wiseman R et al., 2012)

2.4 Feasibility Study

2.4.1 Operational Feasibility

The face recognition and attendance module can track the students' face and record the attendance. It can increase the convenience of students and lecturers compared to the paperwork attendance-taking method. This can also increase the effectiveness in tracking the student's attendance, especially for a large class with hundreds of students. Preventing fraud attendance and minimizing the workload needed can be achieved through this module.

2.4.2 Technical Feasibility

To capture what is happening in the user's surroundings, webcams or embedded systems with camera modules are needed. They are both capable of capturing video material and stringing it together for post-processing video. In addition, the software resources required to perform video processing, such as OpenCV and TensorFlow, are readily available in the market and are continuously supported by relevant organizations. In addition, all devices connected to the network require a wireless network, especially for online classrooms. Since all hardware and software are commercially available, the project is technically feasible.

2.4.3 Financial Feasibility

The software used in this project is all open-source software and does not require any additional payment for non-commercial use. There is some hardware that being used in this project as stated below:

Hardware	Estimate Price (RM)
Laptop or PC with high specification: • Intel® Core i5-11300H • NVIDIA GeForce GTX 1650 4GB GDDR6 • 8GB 3200Mhz DDR4 RAM • SSD: 512GB PCIe NVMe SSD	3529
Logitech HD 1080 Webcam	489

2.5 Chapter Summary and Evaluation

Throughout this chapter, we have completed a background study of the TAR University College, as well as the project will benefit the TAR University College and will further enhance the learning environment of the university college. A literature review has also been conducted to assess the feasibility of the project. It was concluded that the project is operationally, technically, and financially feasible to implement.

Chapter 3

Methodology and Requirements Analysis

3 Methodology and Requirements Analysis

The methods and requirement analysis utilized in this project are discussed in this chapter. As a result, the development process will be defined. Aside from that, methods for acquiring, recording, and evaluating data will be demonstrated. Project scope, development and operational environment, external interface, functional and non-functional requirements will all be included in the requirement analysis.

3.1 Methodology

The incremental model is the most appropriate methodology for this project (Figure 3.1). There are several steps in the incremental model, including requirements analysis, design, coding, testing, and eventually implementation. As shown in Figure 3.1, each iteration in the incremental model goes through these phases. The requirements analysis phase is the first step in the incremental model. The software's requirements and specifications are gathered. All functional and non-functional criteria will be examined in this phase before moving on to the next. The design phase is the second phase of the incremental model. Before moving on to the next step, the design of the user interface and system interaction will be discussed in this phase. During this stage, some high-end features are created. Coding is the incremental model's third phase. The design and system functionality gathered in the second phase will be turned into observable and testable code in this step. The incremental model's fourth phase is the testing phase. Before moving on to the implementation step and delivering the code to customers, the executable code will go through several tests, such as unit tests, module tests, and so on. The implementation phase is the fifth phase of the incremental model. The runnable code will be implemented into the system and supplied to the user after the test passes. These five phases will also be completed on each iteration.

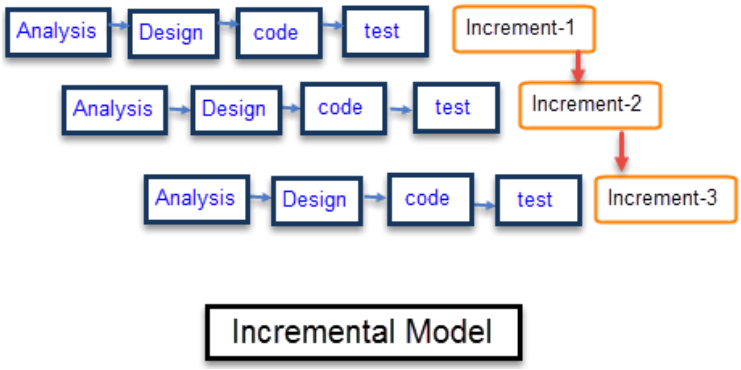


Figure 3.1 Incremental Model Life Cycle

3.1.1 Fact Gathering

Before the development team can determine the system's functional and non-functional needs, all relevant information must be collected during the system development process. Fact gathering strategies are utilized in this scenario to obtain meaningful and necessary data for the system. The information for the system is gathered through observation and research in this project.

1. Observation

   Information is acquired by observing the actual circumstances of how the classroom status and student conduct are observed during the online lesson by using observation techniques. The analysts will gain a better grasp of the present monitoring method in the online class as a result of this. During the online lesson, the analysts can observe the existing monitoring approach and its difficulty through observation. As a result, the system's essential requirements can be identified. It also defined how the suggested system should be integrated into an online course.

   Besides, some of the observation also carried out in order to monitoring and understanding the natural behavior of students at Tunku Abdul Rahman College. The following are examples of observations:

   I. The total number of students entering or exiting the online class at the same time.
   II. The majority of the camera quality used by students.
   III. Under what circumstances, the students will open the camera during the online class.

2. Research

   In addition to observation, research is carried out to obtain data for this project. The data was acquired from other researcher's study papers, articles, and websites through the internet in order to learn how they solved a comparable challenge. In this project, some of the concepts and implementations obtained are used. For example, use the website to determine which model is best for the proposed system and then apply the concept to the project. The analysts are able to locate the optimum answer based on the experience and research of others to incorporate into the suggested system through study.

3. Interview

   It is a scheduled online face-to-face meeting between users and programmers. An interview is used to obtain information from users (including non-functional and functional requirements) (Dr. Tew Yiqi). The reason for conducting an interview with him is that he is the project manager, and he is aware of the shortcomings in the current attendance system

as well as the specifications he desires for the new system. During the interview, both open-ended and closed-ended questions were asked to determine the shortcomings of the present system, as well as the functions and features required for the new system.

4. Performance Testing

   The purpose of a performance test is to collect and aggregate data from many system tests in order to improve or correct the system's flaws and mistakes. The following were included in the performance tests:

   I. Face detection accuracy rate.
   II. The amount of time it takes to detect face.
   III. Percentage of face recorded correctly in the classroom.
   IV. The amount of time it takes to update data in Firebase.
   V. The amount of time it takes to retrieve data from Firebase.
   VI. Time spent comparing data from Firebase.

3.1.2 Fact Recording

Use case diagrams are the method used to record facts in this project. Use case diagrams, also known as behaviour diagrams, are used to describe a series of actions or operations carried out by a certain system. Boundary, actors, use cases, and linkages among and between the actors and use cases should all be included in a use case diagram.

3.2 Requirement Analysis

3.2.1 Project Scope

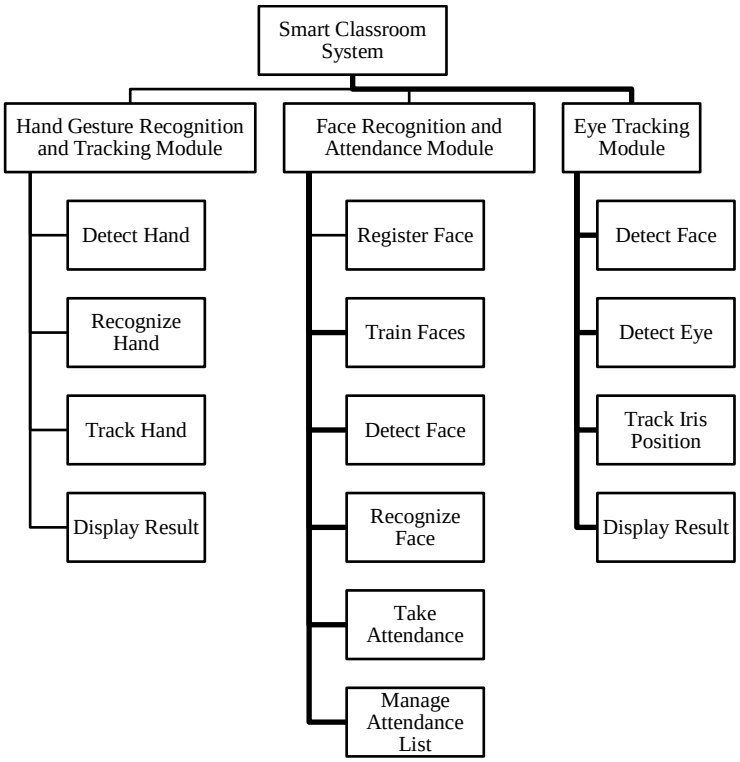


Figure 3.2 Overall Hierarchical Chart of Smart Classroom System

The Smart Classroom System's hierarchical chart, shown in Figure 3.2, depicted the overall system scope of the Smart Classroom System as well as its sub-modules. The hand gesture recognition and tracking module and face recognition and attendance module are all part of the Smart Classroom System. Although all of the modules are created for students and teachers, the user interface and functions for teachers and students varies slightly.

When a hand is spotted, the hand gesture recognition and tracking module will assess the correctness of the recognise. When the accuracy value exceeds the predefined value, the hand is tracked and the result is displayed on the screen.

There are few sub-modules in the face recognition and attendance module. To begin, students and teachers can fulfil the functions of taking attendance. While in the online classroom, students can take their own attendance by face recognition. Teacher will be able to manage the student attendance list in Microsoft Excel after export from the online classroom.

Besides, there is an extra feature that included which is the eye tracking module. This module is used to monitor the student’s iris position during the classroom activities which can help the teacher in monitoring the students.

3.2.2 Development Environment

1. Hardware Requirement

Table 3.1: Hardware Requirement for Smart Classroom System

Hardware	Hardware Specification
Laptop or PC with high specification	• Intel® Core i5-11300H • NVIDIA GeForce GTX 1650 4GB GDDR6 • 8GB 3200Mhz DDR4 RAM • SSD: 512GB PCIe NVMe SSD
Web Camera	Camera with 1080 HD and higher resolution.

2. Software Requirement

Table 3.2: Software Requirement for Smart Classroom System

Hardware	Development Tool and Software
Desktop or Laptop	• Anaconda 3 • PyCharm • Visual Studio Code • TensorFlow • Media pipe • Spyder • Jupyter Notebook • OpenCV

3. Programming Language and Database

Table 3.3: Programming Language and Database for Smart Classroom System

Programming Language	**Database**
• Python • HTML • CSS • JSON • JavaScript	• Google Firebase • Google Sheets

To create the system, we need a desktop or laptop with the specifications stated in Table 3.1, the most important hardware requirement is that the graphic card be a Nvidia Graphic Card with CUDA capability. Visual Studio Code is used to create the system's HTML, CSS and JavaScript code. The libraries utilized to process the photos were OpenCV and

TensorFlow. After that, all of the system's essential data will be uploaded to Google Firebase.

4. Operation Environment

   The suggested system's operational environment necessitated the use of gear such as a laptop, desktop, keyboard, and a web camera. Python, OpenCV, Windows 10 or higher operating system, and web browsers such as Google Chrome, Mozilla FireFox, and Internet Explorer are among the suggested applications for the proposed system to run in the operation environment. The better the performance, the faster the processing speed is. The graphics card must, however, be a Nvidia Graphic Card with CUDA capability.

3.2.3 External Interface Requirements

1. User Interface Consistent

   The system's information is provided to guarantee that it can communicate correctly with external components including hardware, software, and database elements. The best practice in the software industry is to come to an agreement on the external and internal system interfaces that have been discovered. A separate interface specification or system architecture specification should be used for a complicated system with several subcomponents. It could, for example, refer to a distinct application programming interface (API) specification or a hardware device handbook that details the error codes that the device may transmit to the software (Richard, 2015).

2. Provide useful feedback

   When computers are used to aid human communication, the problem of "information overload" arises quickly. Those who have tried to conduct remote education with a large number of pupils via email can attest to the validity of this finding. The minimizing of information overload for both professors and students is a critical feature in the design of software to enable distant education (Murray, 2014). For example, when an issue arises, such as an error message popping up, the format of email and password, and so on, there are simple instructions for students and teachers.

3.2.4 Non-Functional Requirements

1. Efficiency

   Efficiency is the amount of time it takes for a system to fulfil its operations when the user requests it. In a short amount of time, the system should be able to recognise a student's face. For instance, when the student registers the face, the system should be able to track and crop the face within a short period of time in order to ensure the efficiency of the system.

2. Availability

   Availability refers to the likelihood that a system will perform as expected when needed during the course of a task. The system should be able to process video frames without fail at all times. The system must always be ready when the user is browsing.

3. Usability

   Usability refers to the ease with which an end user can use the system's user interface. The system should be able to take user input and provide the findings in a more intelligible manner based on previous experiences. For example, the smart classroom system should be able to track and register the face into the database.

4. Functionality

   The term "functional" refers to the system's ability to accomplish the functions that are needed of it. All of the module's functionalities should be able to perform as expected. When the face detection function receives a video frame it should work properly.

5. Reliability

   The probability that a system will run without failure for a specific number of uses or for a specified amount of time is known as reliability. In order to approximate actual performance, reliability can be measured by the number of failures per transaction in a system test environment. The smart classroom system should be able to perform the task without any failure.

3.3 Requirement Diagrams

3.3.1 Functional Requirements

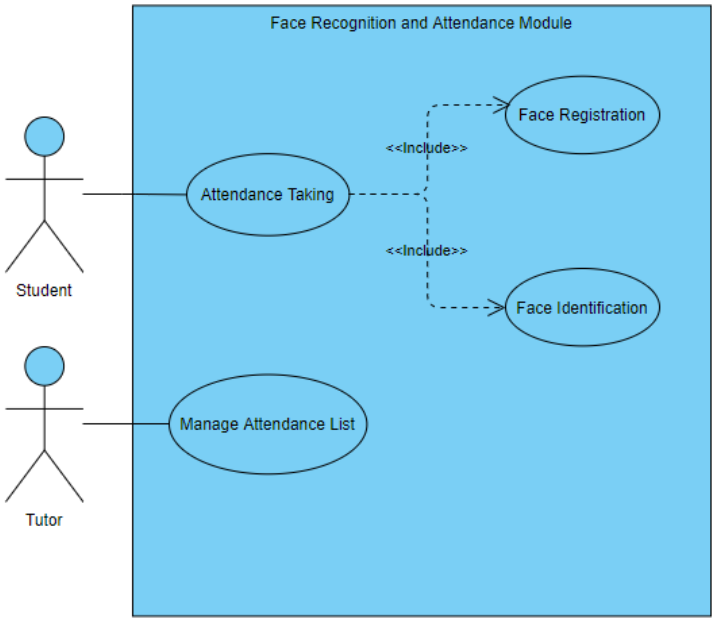


Figure 3.3 Use Case Diagram of Face Recognition and Attendance Module

Figure 3.3 depicts the Face Recognition and Attendance Module's functional requirements. Using the aid of the face recognition function, which is used to determine which student's name corresponds to which student's face, the attendance taking function will take student attendance with their face. Tutors can use the Manage Attendance List tool to keep track of their students' attendance. If there are any incorrect outputs, this is where the instructor may amend the student attendance.

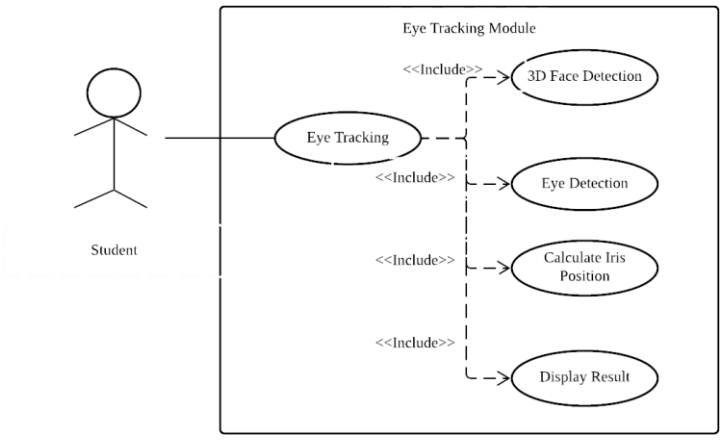


Figure 3.4 Use Case Diagram of Eye Tracking Module

Figure 3.4 shows the function that include in the eye tracking module. In this module a 3D face detection model is used. This module will detect the student eye and then calculate the position of the student's iris in front of the camera in order to estimate the student's looking position. At

the end, the analysed result will be displayed on the google meet screen which can be view by all the participate in the google meet online classroom.

3.4 Chapter Summary and Evaluation

The incremental model was chosen as the software development methodology for this project. In fact, to collect information from users and the implementation environment, approaches such as observation and interview (open and closed-ended questions) are used. To document the information gathered, a Use Case Diagram, a sort of UML Diagram, was employed. HTML and JavaScript will build the system, which will run on desktops or laptops using Nvidia graphics cards that support CUDA.

Chapter 4

System Design

4 System Design

In this chapter, we will discuss the system's overall design, which includes the concept, activity, and process that take place in the system during the operation. Besides, the data structure of the database involved in this project will be mentioned. The flowchart, activity diagram, and user interface design will be prepared to provide an easier understanding of the project's scope.

4.1 Activity Diagram for Face Detection and Registration

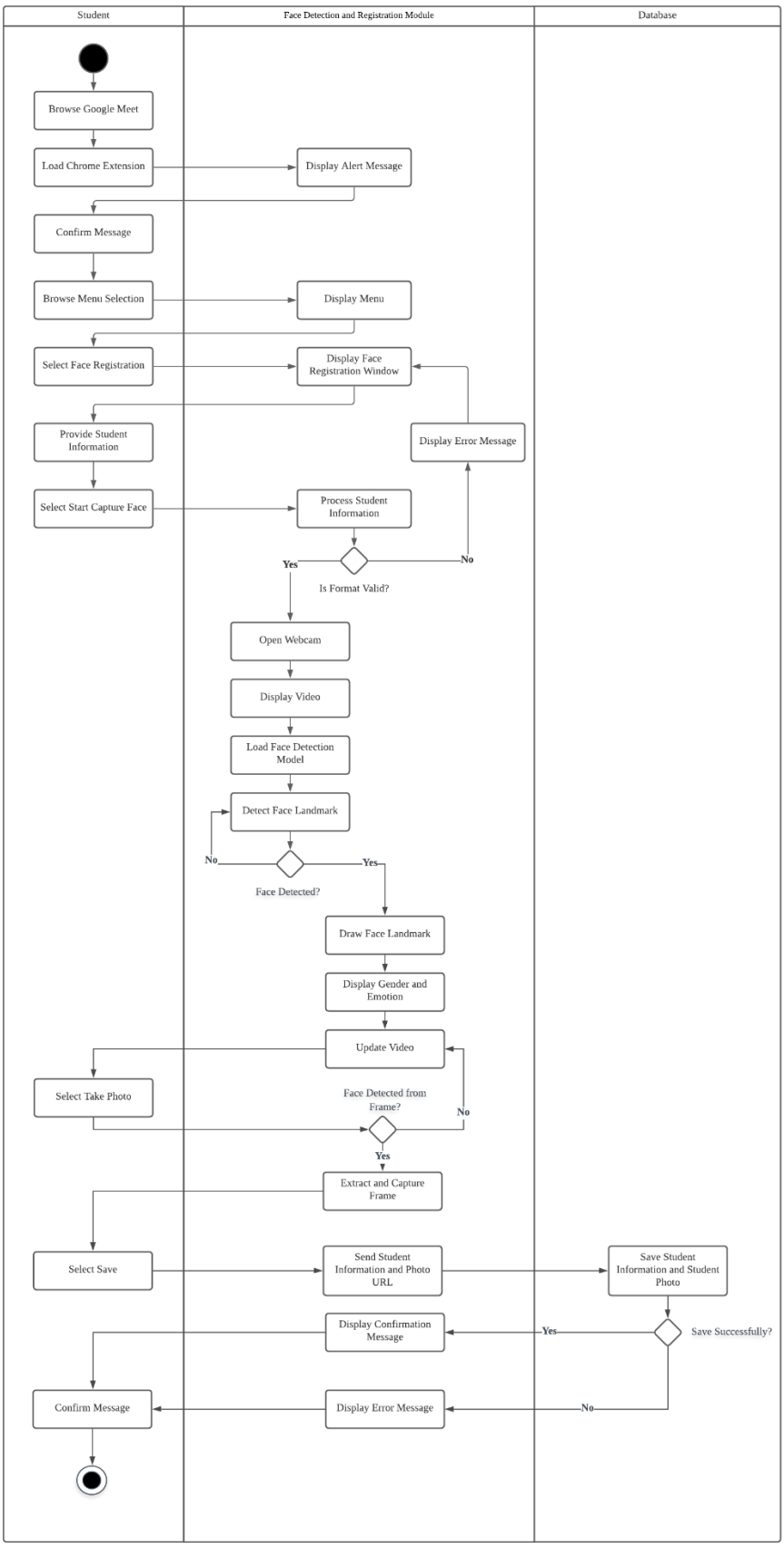


Figure 4.1 Activity Diagram for Face Detection and Registration

4.2 Activity Diagram for Face Recognition and Attendance

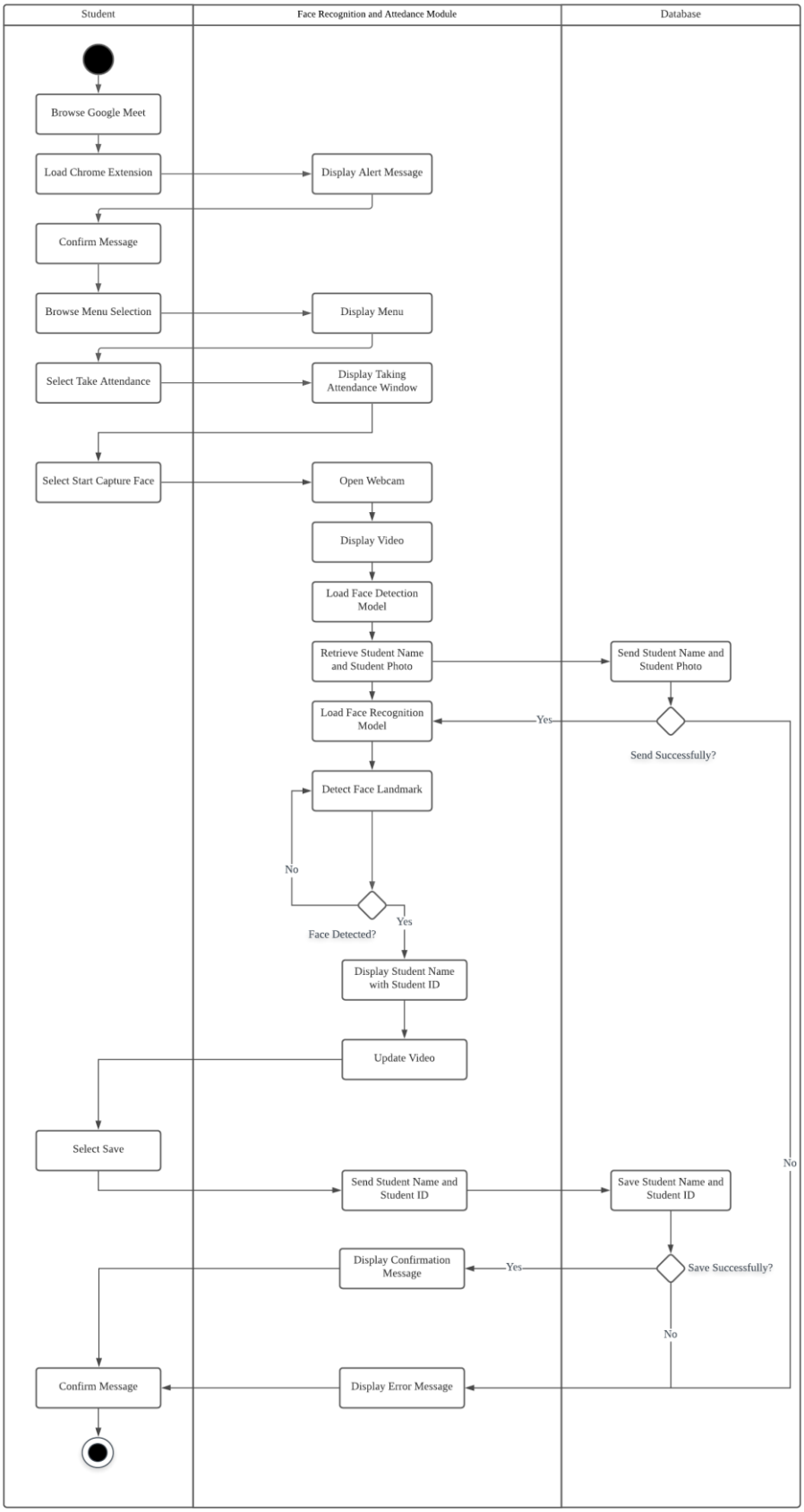


Figure 4.2 Activity Diagram for Face Recognition and Attendance

4.3 Activity Diagram for Eye Tracking Module

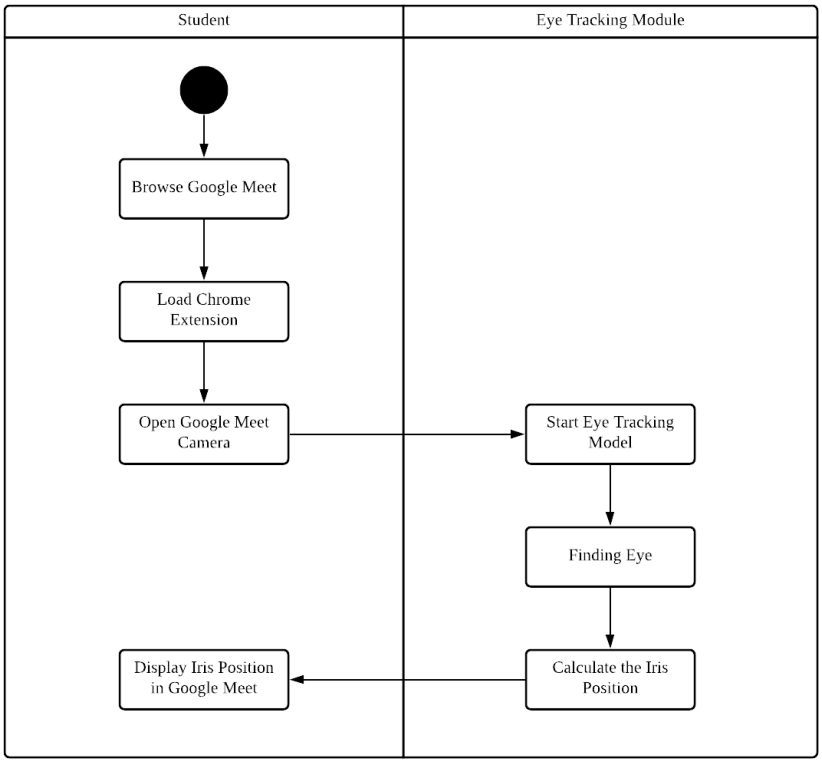


Figure 4.3 Activity Diagram for Eye Tracking

4.4 Firebase Data Structure

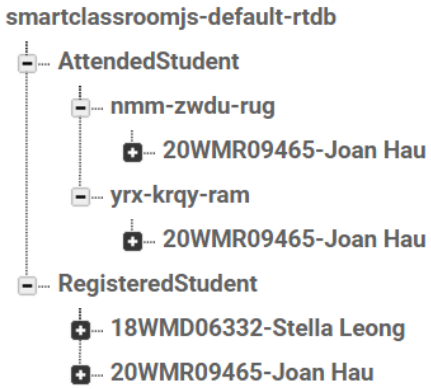


Figure 4.4 Overall Data Structure of Smart Classroom

Figure 4.4 shows the data structure of this project. There is a total of 2 tables in charge in this module which involve AttendedStudent and RegisteredStudent. The tables are used to store the student who attends the specific class by using the google meet link as a representation and to record the students who registered their faces in the database, respectively.

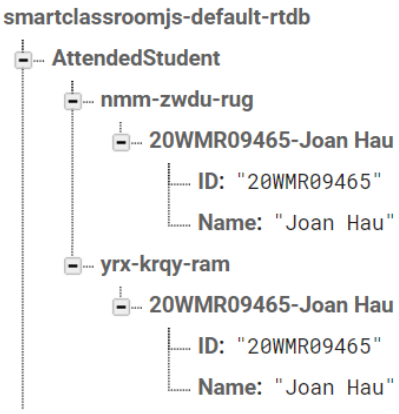


Figure 4.5 Data Structure for Student Attendance Table

Figure 4.5 shows the detailed element of the student attendance table. After the student takes the attendance, the name and student id displayed in the video frame will be saved into the database. The google meet link will act as a unique key to separate each of the classes.

Node: AttendedStudent Description: To store the attendance details of each class **AttendedStudent**: Parent Node **MeetLink**: The unique ID of the google meet link. E.g. “nmm-zwdu-rug” **StudentLabel**: The label display on the student video. E.g. “20WMR09465-Joan Hau” **ID**: The id of the student who attend. E.g. “20WMR09465” **Name**: The name of the student who attend. E.g. “Joan Hau”

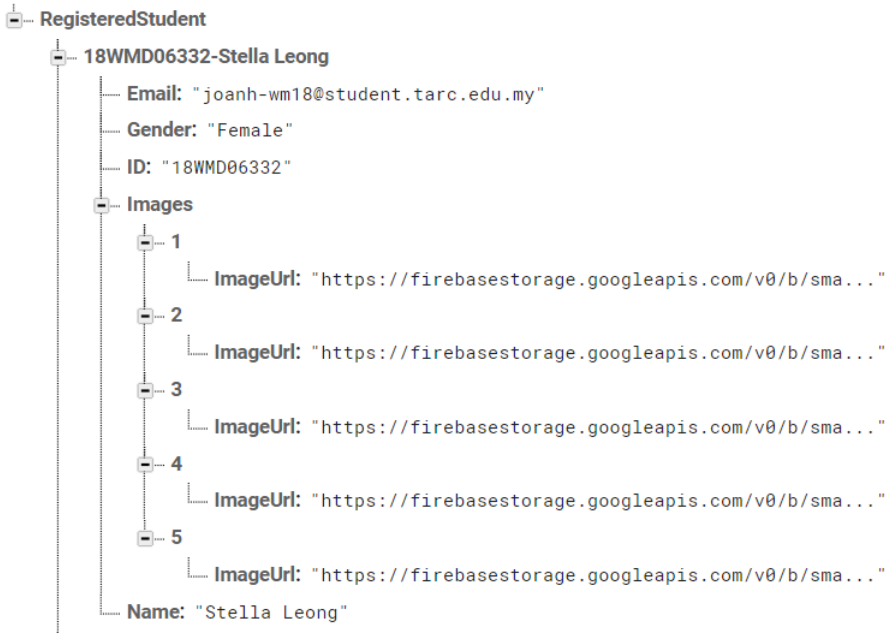


Figure 4.6 Data structure for Student Registration Table

Figure 4.6 shows the detailed element for the student registration table. This table is used to store the detailed information for every student who registered their face into the database. There are a total of 5 faces will be capture and store in the database for further training.

Node: RegisteredStudent

Description: To store the student list who registered the face

RegisteredStudent: Parent Node

StudentLabel: The unique ID of the student. E.g. “20WMR09465-Joan Hau”

Email: The current email used by the student. E.g. “joanh-wm18@student.tarc.edu.my”

ID: The id of the student who registered. E.g. “20WMR09465”

Images: Images node that contain multiple image

ImageID: An auto count number to represent the unique of the image

ImageUrl: The image URL which convert from the firebase storage.

Name: The name of the student who registered. E.g. “Joan Hau”

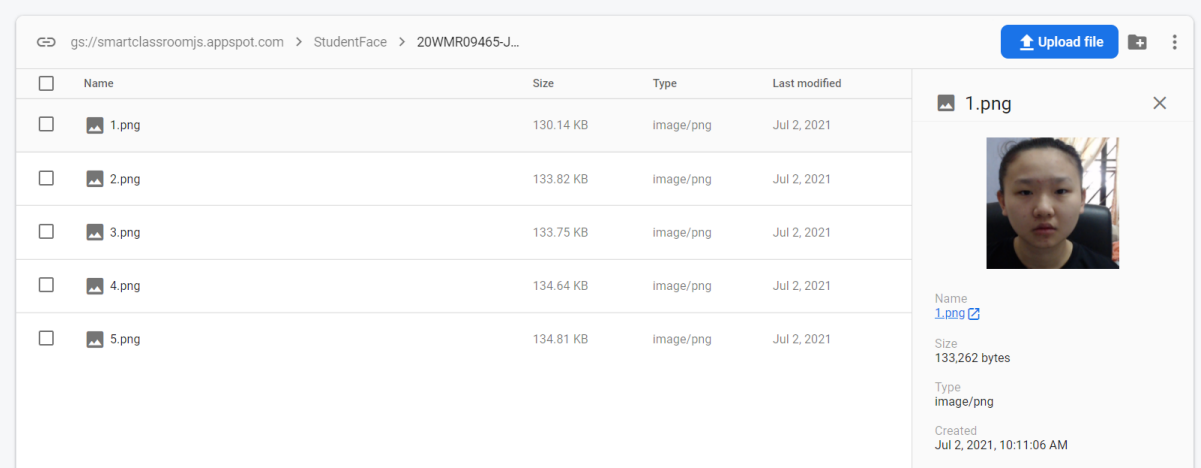


Figure 4.7 Image Save in Firebase Storage

Figure 4.7 shows the firebase storage used by this project to save the student's image for further reference.

4.5 User Interface Design

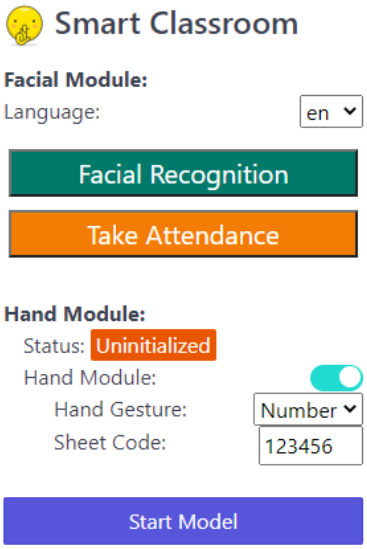


Figure 4.8 User Interface of Chrome Extension Menu

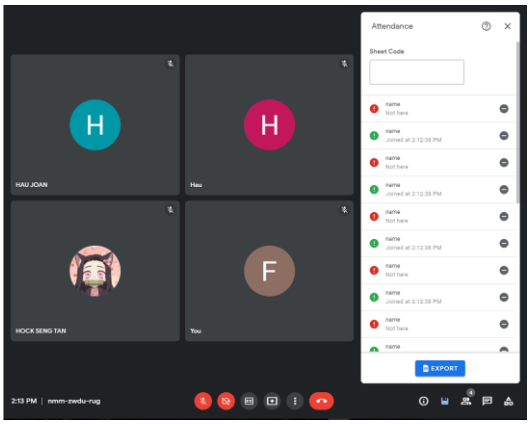


Figure 4.9 Google Meet for Attendance Exporting

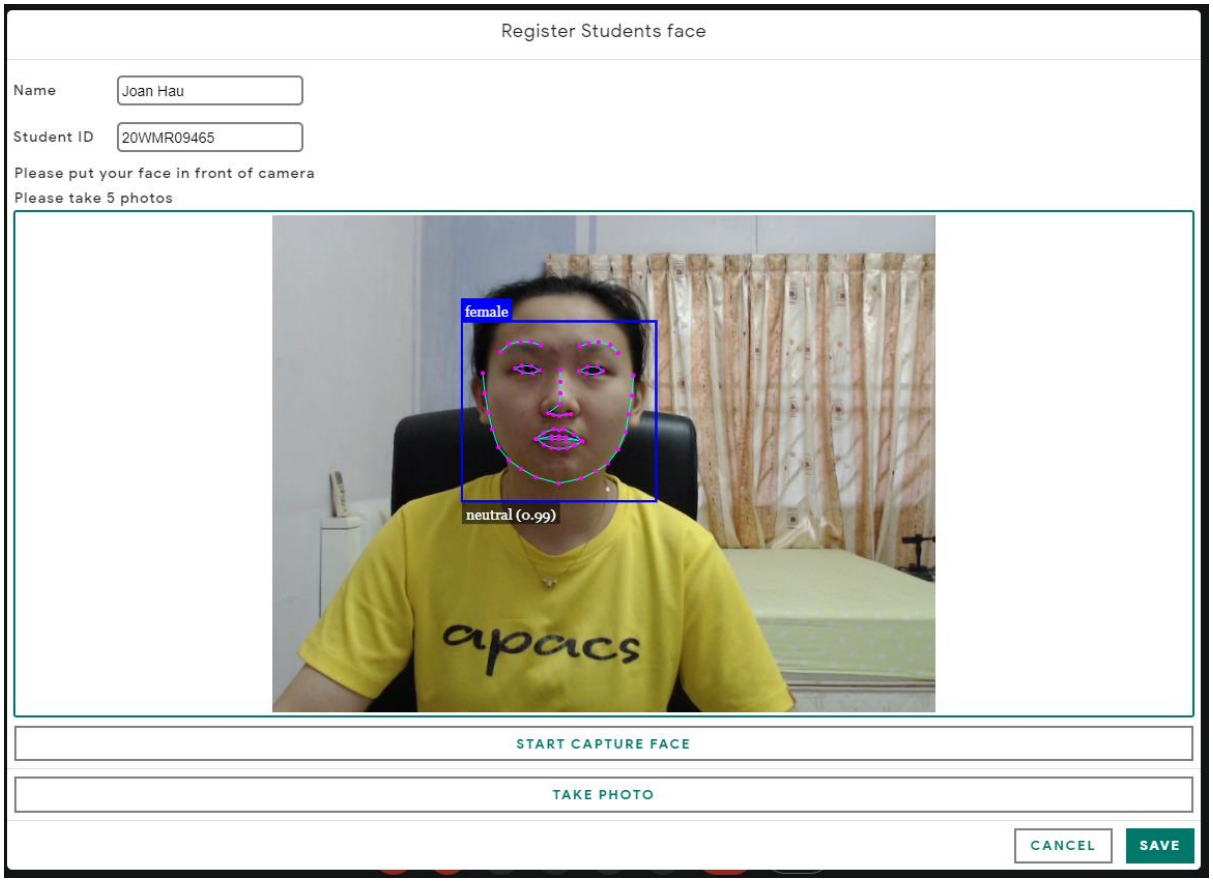


Figure 4.10 Face Registration Window

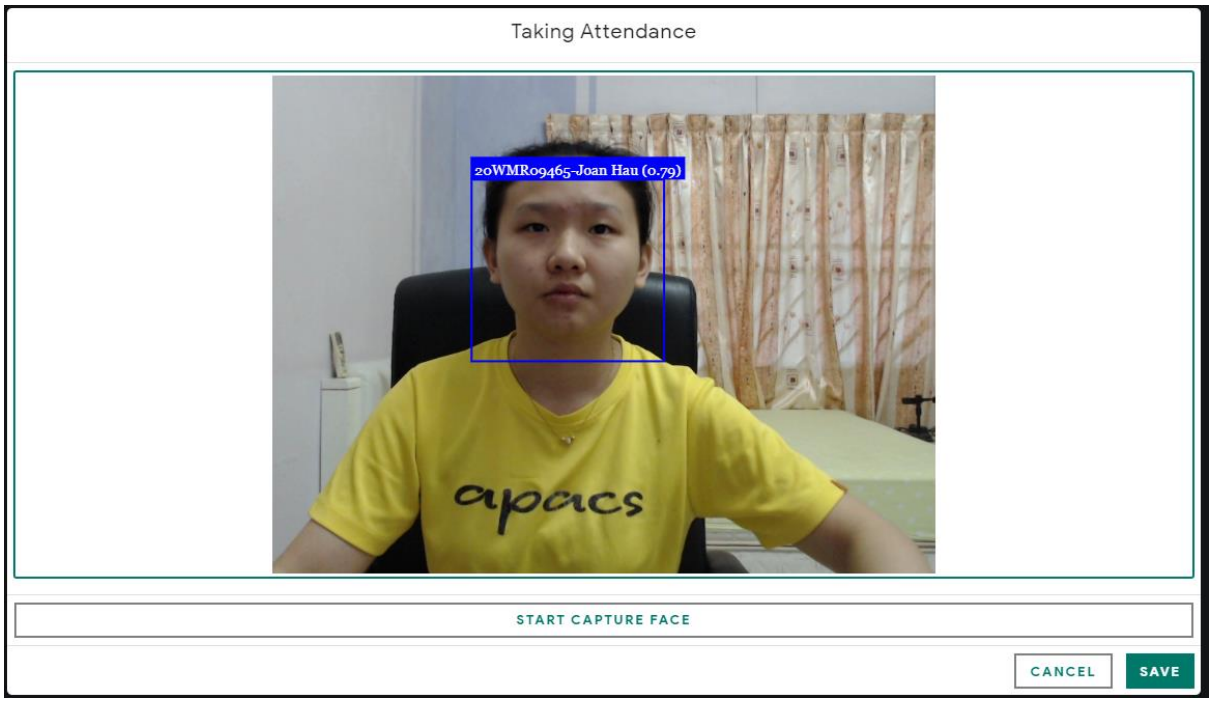


Figure 4.11 Face Attendance Window

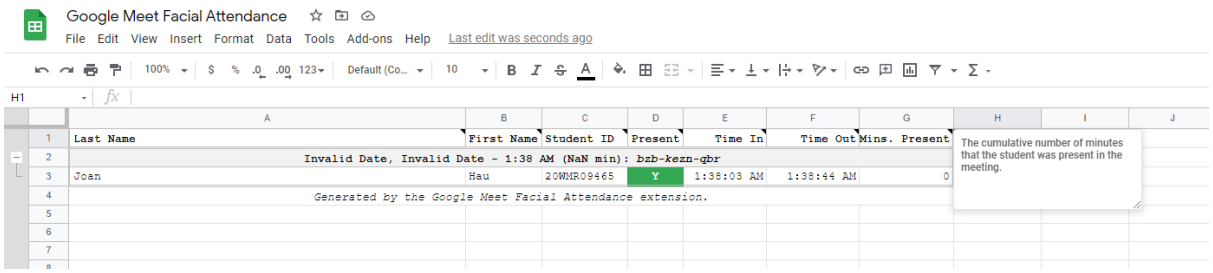


Figure 4.12 Google Meet Attendance in Google Sheet

4.6 Chapter Summary and Evaluation

The content of this chapter provides a more precise concept about the system, including the system's data flow and the structure and process of the system during a regular operation. Various diagrams and interfaces are shown in this chapter to gives an easier way for understanding.

Chapter 5

Implementation

5 Implementation

This chapter will carry out the discussion of implementation and testing of the system in detail. The way of implementing the smart classroom system with facial recognition and attendance module and eye tracking module will be fully covered in this section. Besides, the test cases will be constructed to gather the results of testing.

5.1 Implementation Description

Smart Classroom system with Face Recognition and Attendance Module and Eye Tracking Module are developed using the JavaScript and Hypertext Markup Language (HTML) programming language in Window 10 Operating System. There are also some of the requirements that need to be downloaded and installed to enable the module running in the chrome extension. Requirements such as TensorFlow, MediaPipe, Face-api.js, Firebase, Google Sheet API (GAPI) are needed.

5.2 Code Snippets

5.2.1 Chrome Extension Configuration

In order to load the system in the chrome extension, there are some of the configurations that need to be done in the program code. In this project, all chrome extension configurations are combined and structured in the **manifest.json** file.

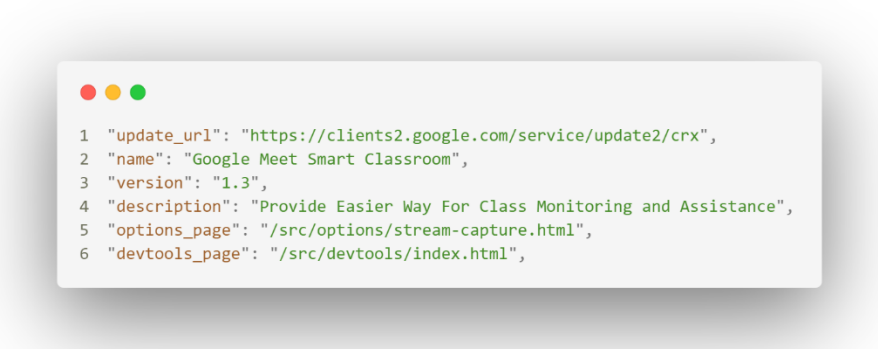


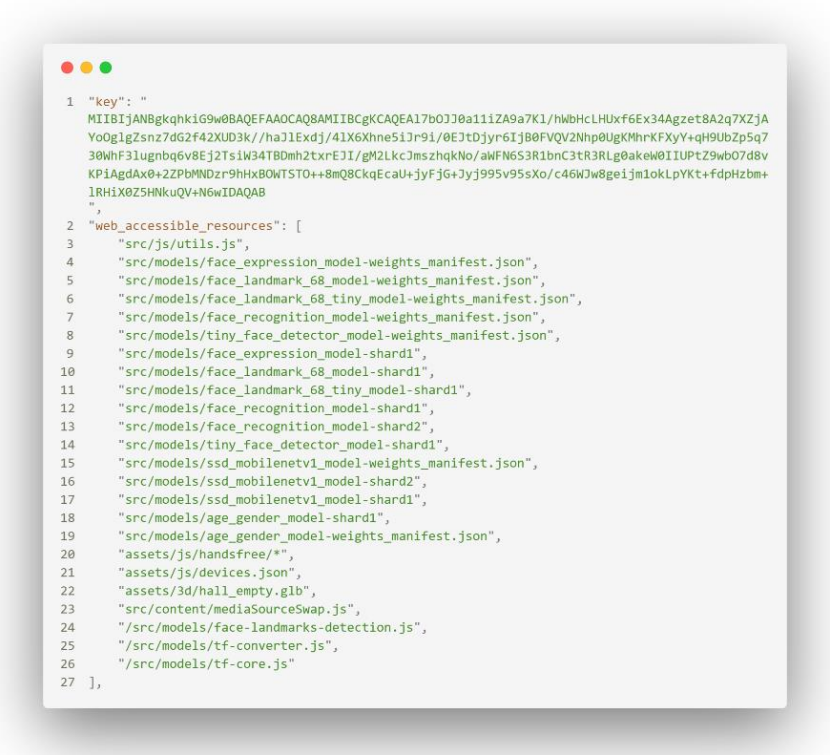
Figure 5.1 Chrome Extension Detail

The detail of the chrome extension can be set in the manifest.json as shown in figure 5.1. The *update url* indicates the location for doing update checks. Then, the *name* mentioned about the project name of the extension when uploading to the google chrome extension. The *version* is for the project team to classified the version of the project being uploaded to the google chrome extension. Next, the description is set in order for the user to understand the purpose of the project. Other than that, the *options_page* and *devtools_page* is pointing to the html files which used to display the user interface when user launch the extension in the google chrome browser.

```
"background": {
    "scripts": [
        "/src/firebase/firebase-app.js",
        "/src/firebase/firebase-auth.js",
        "/src/firebase/firebase-database.js",
        "/src/firebase/firebase-storage.js",
        "/src/js/firebase.js",
        "/src/js/request.js",
        "/src/js/utils.js",
        "/assets/js/handsfree/handsfree.js",
        "/src/background/handsfree.js",
        "/src/background/webxr.js",
        "src/background/jsrsasign-all-min.js",
        "/src/background/background.js"
    ],
    "persistent": false
},
```

Figure 5.2 Chrome Extension Background Script

In the chrome extension there are some of the files that need to be execute at the project background throughout the running progress. Therefore, a *background* is used to indicate some of the JS file that need to be run during the program execution. In this project, the JS file for firebase database, database update and retrieve, detection model etc is run in the background of the project.



```
"key": "
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA17bOJJ0a11iZA9a7K1/hWbHcLHUxf6Ex34Agzet8A2q7XZjA
YoOglgZsnz7dG2f42XUD3k//haJlExdj/4lX6Xhne5iJr9i/0EJtDjyr6IjB0FVQV2Nhp0UgKMhrKFXyY+qH9UbZp5q7
30WhF3lugnbq6v8Ej2TsiW34TBDmh2txrEJI/gM2LkcJmszhqkNo/aWFN6S3R1bnC3tR3RLg0akeW0IIUPtZ9wbO7d8v
KPiAgdAx0+2ZPbMNDzr9hHxBOWTSTO++8mQ8CkqEcaU+jyFjG+Jyj995v95sXo/c46WJw8geijm1okLpYKt+fdpHzbm+
lRHiX0Z5HNkuQV+N6wIDAQAB
",
"web_accessible_resources": [
    "src/js/utils.js",
    "src/models/face_expression_model-weights_manifest.json",
    "src/models/face_landmark_68_model-weights_manifest.json",
    "src/models/face_landmark_68_tiny_model-weights_manifest.json",
    "src/models/face_recognition_model-weights_manifest.json",
    "src/models/tiny_face_detector_model-weights_manifest.json",
    "src/models/face_expression_model-shard1",
    "src/models/face_landmark_68_model-shard1",
    "src/models/face_landmark_68_tiny_model-shard1",
    "src/models/face_recognition_model-shard1",
    "src/models/face_recognition_model-shard2",
    "src/models/tiny_face_detector_model-shard1",
    "src/models/ssd_mobilenetv1_model-weights_manifest.json",
    "src/models/ssd_mobilenetv1_model-shard2",
    "src/models/ssd_mobilenetv1_model-shard1",
    "src/models/age_gender_model-shard1",
    "src/models/age_gender_model-weights_manifest.json",
    "assets/js/handsfree/*",
    "assets/js/devices.json",
    "assets/3d/hall_empty.glb",
    "src/content/mediaSourceSwap.js",
    "/src/models/face-landmarks-detection.js",
    "/src/models/tf-converter.js",
    "/src/models/tf-core.js"
],
```

Figure 5.3 Chrome Extension Public Key and Web Accessible Resources

The *key* is the public key which auto generated by the chrome web store once the project being uploaded in the Chrome Web Store Developer Dashboard. By having the unique public key, the project will have the permission to work with the chrome storage, tabs, identity, identity.email and host permission which is very useful for keeping track the information in the google chrome. Additionally, the

web_accessible_resources are all the models that need to be used for the facial detection and recognition and hand gesture detection and recognition.

```
"content_scripts": [
    {
        "matches": [
            "*://*/*"
        ],
        "exclude_matches": [
            "*://localhost/*"
        ],
        "js": [
            "/assets/js/polyfill/webxr-polyfill.js",
            "/assets/js/ConfigurationManager.js"
        ],
        "run_at": "document_start",
        "all_frames": true
    },
```

Figure 5.4 Chrome Extension Content Scripts (A)

The *content_scripts* is used to manage the execution of JS file. Based on the figure 5.4, the `webxr-polyfill.js` and `ConfigurationManager.js` will execute when the chrome extension is launch in the browser with URL format `*://*/*` (* can be any alphabet). These JS file will not execute if the chrome extension is launch in the browser with URL format `*://localhost/*`.

```
"content_scripts": [
    {
        "matches": [
            "*://*/*"
        ],
        "exclude_matches": [
            "*://localhost/*"
        ],
        "js": [
            "/assets/js/polyfill/webxr-polyfill.js",
            "/assets/js/ConfigurationManager.js"
        ],
        "run_at": "document_start",
        "all_frames": true
    },
    {
        "matches": [
            "*://meet.google.com/*"
        ],
        "js": [
            "src/js/utils.js",
            "src/js/material-components.min.js",
            "src/js/lib/face-api.min.js",
            "src/js/html-templates.js",
            "src/js/contentScript.js",
            "src/js/classButtons.js",
            "src/js/storeFace.js",
            "src/js/takeAttendance.js",
            "src/js/tagbox-logic.js",
            "src/js/translations.js",
            "src/js/videoConfiguration.js",
            "/assets/js/handsfree/handsfree.js",
            "/src/content/browser-polyfill.js",
            "/src/content/drawing_utils.js",
            "/src/content/hands.js",
            "/src/content/pose.js",
            "/src/content/jquery.min.js",
            "/src/content/content.js"
        ],
        "css": [
            "assets/css/material-components.min.css",
            "assets/css/style.css",
            "assets/css/btn.css",
            "/assets/js/handsfree/assets/handsfree.css"
        ],
        "all_frames": true,
        "run_at": "document_end"
    },
    {
        "matches": [
            "*://*/*"
        ],
        "exclude_matches": [
            "*://localhost/*",
            "*://meet.google.com/*"
        ],
        "css": [
            "/assets/js/handsfree/assets/handsfree.css"
        ],
        "js": [
            "/src/content/browser-polyfill.js",
            "/assets/js/handsfree/handsfree.js",
            "/src/content/mouse.js"
        ],
        "all_frames": true,
        "run_at": "document_end"
    }
],
```

Figure 5.5 Chrome Extension Content Scripts (B)

All the JavaScript file mentioned under the *js* will only execute if the URL matches the format `://meet.google.com/*`. The css files which used for defining the styles of the webpage is categorised in *css* which only be execute when the user browses the chrome extension in google meet.

```
"oauth2": {
    "client_id": "816803315444-q2f7l0nmlvie8pirqhu17d0r5iqv3kbk.apps.googleusercontent.com",
    "scopes": [
        "https://www.googleapis.com/auth/drive.file"
    ]
},
```

Figure 5.5 Chrome Extension OAuth

OAuth2 is a standard authorization protocol that used in the industry. It allows users to grant access to confidential information to online and desktop programmes without having to divulge their usernames, passwords, or other sensitive credentials. *client_id* and *scope* are important Google API which used to connect to the Google Service such as Google Sheets. The *client_id* is generated by the Google Cloud Platform once the project finished setup in the Google Cloud Platform.

```
"icons": {
    "16": "./src/img/Attendance16.png",
    "32": "./src/img/Attendance32.png",
    "64": "./src/img/Attendance64.png",
    "128": "./src/img/Attendance128.png"
},
"browser_action": {
    "default_icon": {
        "16": "./src/img/Attendance16.png",
        "32": "./src/img/Attendance32.png",
        "64": "./src/img/Attendance64.png",
        "128": "./src/img/Attendance128.png"
    },
    "default_popup": "src/html/popup.html"
},
```

Figure 5.6 Chrome Extension Icons

The icons that used to represent the smart classroom system in the chrome extension is set in the **manifest.json**. The *icon* and *default_icon* is used to represent the image of the icon while the *16*,*32*,*64* and *128* is the size of the icons displayed in the google chrome.

```
    "manifest_version": 2,
    "content_security_policy": "
script-src 'self' https://www.gstatic.com/ https://*.firebaseio.com https://
www.googleapis.com https://cdn.firebase.com https://*.firebaseio.com http
s://apis.google.com/ 'unsafe-eval'; object-src 'self';
",
    "permissions": [
        "*://*/*",
        "storage",
        "tabs",
        "activeTab",
        "identity",
        "identity.email",
        "notifications",
        "*://meet.google.com/**-**-**"
    ]
```

Figure 5.7 Chrome Extension Security Policy and Permission

Other that setup the authorization ID and Google API in the chrome extension, the services that need to be launch in the chrome extension will need to be mentioned in the *content_security_policy* in order to allow the user to access to the Google Services such as Google Sheets, Firebase, etc in the extension. The *permission* is help to limit the damage, if the chrome extension is corrupted by malware.

5.2.2 User Interface

This section will cover several the code snippets for the implementation of the user interface and the buttons' function carry out in the user interface in the smart classroom system with face recognition and attendance module.

5.2.2.1 Popup Screen

```
<section class="container grid-lg">
    <h1 style="font-size: 1.5em"><img src="/src/img/Attendance128.png"
        style="vertical-align: middle;" height="32px"> Smart Classroom</h1>
    <b>Facial Module:</b><br>

    <div class="columns">
        <div class="column popup-lang" id="language">
            Language:
        </div>
        <div class="column" style="text-align: end;" id="language">
            <select class="language-choice" style="margin-left: auto">
                <option>en</option>
                <option>bm</option>
                <option>cn</option>
                </select>
        </div>
    </div>

    <div class="buttons">
        <button class="popup-face-register hover-state">Facial Recognition</button>
        <button class="popup-take-attendance hover-state">Take Attendance</button>
        <button class="popup-overall-analysis hover-state">Eye Tracking</button>
    </div>

</section>
```

Figure 5.8 Popup Screen in Chrome Extension for Face Recognition and Attendance Module Section

The popup screen of the face recognition and attendance module enables the users to change the displayed language. Besides, there are also 3 buttons prepared in the *buttons* class to allow the user to launch the specific function int eh google meet.

5.2.2.1.1 Change Language

```
/**
 * Initialize the startup language
 */
chrome.storage.sync.get(null, function(result) {
    if (!result['lang']) {
        chrome.storage.sync.set({'lang': 'en'}, null);
    }
});
```

Figure 5.9 Default Language of the Popup Screen

```javascript
let langChoice = document.querySelector('.language-choice');
langChoice.onchange = function() {
    let currentLang = langChoice.options[langChoice.selectedIndex].text;
    chrome.storage.sync.set({'lang': currentLang}, null);
    updatePopup(currentLang);
    chrome.tabs.sendMessage(tabs[0].id, {data: 'translate'});
};

chrome.storage.sync.get(['lang'], function(request) {
    for (let i=0; i<langChoice.length; i++) {
        if (langChoice.options[i].text == request.lang) {
            langChoice.options[i].selected = true;
            break;
        }
    }
    updatePopup(request.lang);
});
```

Figure 5.10 Change Language for the Option in Popup Screen

A `querySelector` is used to point to the name of the class in the popup screen as shown in figure 5.8. The language of the popup screen will be updated based on the user's selection by calling the `updatePopup()` as shown in figure 5.11.

```javascript
function updatePopup(currentLang) {
    document.querySelector('.popup-lang').innerText = meetLanguage[currentLang]['popup']['popup_lang'];
    document.querySelector('.popup-face-register').innerText = meetLanguage[currentLang]['popup']['popup_face_register'];
    document.querySelector('.popup-take-attendance').innerText = meetLanguage[currentLang]['popup']['popup_take_attendance'];
    document.querySelector('.popup-overall-analysis').innerText = meetLanguage[currentLang]['popup']['popup_overall_analysis'];

    document.querySelector('#language').innerText = meetLanguage[currentLang]['popup']['language'];
}
```

Figure 5.11 Update Popup Screen Language

This `updatePopup()` function not only updating the language in the popup screen but also others user interface which used for face recognition and attendance module.

```
const meetLanguage = {
    en: {
        popup: {
            popup_lang: 'Smart Classroom',
            popup_face_register: 'Facial Registration',
            popup_take_attendance: 'Take Attendance',
            popup_overall_analysis: 'Eye Tracking',
            language: 'Language',
        },
        classHTML: {
            new_class_name: 'Register Students face',
            input_name_student_name: 'Name',
            input_name_student_id: 'Student ID',
            student_names: 'Please put your face in front of camera',
            SRF: 'Start Capture Face', // SRF - "Start Register Face"
            TP: 'Take Photo',
            Cancel: 'Cancel',
            Ok: 'Save',

        },
        updateHTML: {
            name_2: 'Taking Attendance',
            Ok: 'Save',
            Cancel: 'Cancel',
            SCF: 'Start Capture Face'
        }
    },
    bm: {
        popup: {
            popup_lang: 'Bilik Darjah Pintar',
            popup_face_register: 'Pendaftaran Muka',
            popup_take_attendance: 'Kehadiran',
            popup_overall_analysis: 'pengesanan mata',
            language: 'Bahasa',
        },
        classHTML: {
            new_class_name: 'Mendaftar Muka Pelajar',
            input_name_student_name: 'Nama',
            input_name_student_id: 'ID Pelajar',
            student_names: 'Sila letakkan Muka anda di hadapan kamera',
            SRF: 'Mula Tangkap Muka',
            TP: 'Ambil Gambar',
            Cancel: 'Batal',
            Ok: 'Simpan',
        },
        updateHTML: {
            name_2: 'Ambil Kehadiran',
            Ok: 'Simpan',
            Cancel: 'Batal',
            SCF: 'Mula Tangkap Muka'
        }
    },
    cn: {
        popup: {
            popup_lang: '智慧课堂',
            popup_face_register: '人脸登记',
            popup_take_attendance: '出勤登记',
            popup_overall_analysis: '眼球追踪',
            language: '语言',
        },
        classHTML: {
            new_class_name: '学生人脸注册',
            input_name_student_name: '学生名字',
            input_name_student_id: '学生卡号',
            student_names: '请把你的脸放在镜头前',
            SRF: '开始捕捉人脸',
            TP: '拍照',
            Cancel: '取消',
            Ok: '保存',
        },
        updateHTML: {
            name_2: '报到',
            Ok: '保存',
            SCF: '开始捕捉人脸',
            Cancel: '取消'
        }
    }
}
```

Figure 5.12 English, Malay and Chinese Translations in Chrome Extension Popup Screen

The translation of the words with English, Malay and Chinese is specified in the **translation.js** with specific key in order to access the specific word translation based on the user's selection in **popup.js**.

5.2.2.1.2 Launch Function

```
document.addEventListener('DOMContentLoaded', function() {
    chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {
        if (!tabs[0].url.match(/https:\/\/meet.google.com\/.+-.+-.+/)) return;
        document.querySelector('.popup-face-register').addEventListener('click', function() {
            chrome.tabs.sendMessage(tabs[0].id, {data: 'register-face'});
            window.close();
        }, false);
        document.querySelector('.popup-take-attendance').addEventListener('click', function() {
            chrome.tabs.sendMessage(tabs[0].id, {data: 'take-attendance'});
            window.close();
        }, false);
```

Figure 5.13 Launch Specific Function in Popup Screen

The user is able to launch the function of the chrome extension by clicking on the button shown on the popup screen. Once the user makes the decision, then the system will check which button is being click by the user and it will send the message with the unique name of the selected screen to the background in order to display the specific screen for the user.

```
chrome.runtime.onMessage.addListener(function(request, sender, sendResponse) {
    Utils.log (request);
    switch(request.data) {
        case 'translate': {
            updateCards();
            break;
        }
        case 'register-face': {
            /**
             * Assume the student name entered in follow the google meet registered name
             */
            alert("Please Enter Full Name and Student ID as Registered \nFormat of Input: \r\n
Name: XXXX XXXX XXXX \r\nStudent ID: XXWMRXXXXX \r\nClick the take photo button once you are ready")
            document.getElementById('registerFaceCard').style.visibility = 'visible';
            break;
        }
        case 'take-attendance': {
            alert("Please Show your face in front of camera with enough light source")
            document.getElementById('takeAttendanceCard').style.visibility = 'visible';
            break;
        }
    }
});
```

Figure 5.14 Display User Interface in Google Meet

The `chrome.runtime.onMessage.addListener()` is to check whether there is any message send from the chrome during the runtime. Then, the system will display the screen in the google meet based on the decision made by the user.

5.2.2.2 Google Meet Side Bar

```
/**
 * Initialize the interface
 */
function initialize() {
    chrome.runtime.sendMessage(
        {
            data: 'starting...',
        },
        function (response) {
            if (response.ready) {
                /**
                 * Initialize the start time of the user joined and send to bakcground
                 */
                const startTime = new Date().toLocaleTimeString();
                chrome.storage.sync.set({start: startTime}, function() {
                    console.log('Start Time is set to ' + startTime);
                });
                /**
                 * To draw and Initialize the side bar in the google meet
                 */
                const panelContainer = document.querySelector('.R3Gmyc.qwU8Me')
                document
                    .querySelector('[jsname="HlFzId"]')
                    .insertAdjacentHTML('afterend', panelHTML)
                const attendancePanel = document.getElementById('panel')

                const ariaPressedObserver = new MutationObserver(
                    (mutations, me) => {
                        mutations[0].target.setAttribute('aria-pressed', false)
                        me.disconnect()
                    }
                )
                const ariaPressedObserverOptions = {
                    attributes: true,
                    attributeFilter: ['aria-pressed'],
                    attributeOldValue: true,
                }
                const panelUnhiddenObserver = new MutationObserver(
                    (mutations, me) => {
                        const mutation = mutations[0]
                        const target = mutation.target
                        if (
                            mutation.oldValue.includes('qdulke') &&
                            !target.classList.contains('qdulke')
                        ) {
                            target.classList.add('qdulke')
                            attendancePanel.classList.remove('qdulke')
                            me.disconnect()
                        }
                    }
                )
                const panelSpawnedObserver = new MutationObserver(
                    (mutations, me) => {
                        const mutation = mutations[0]
                        if (mutation.addedNodes.length > 0) {
                            const addedNode = mutation.addedNodes[0]
                            if (addedNode.getAttribute('data-tab-id') === '5') {
                                addedNode.classList.add('qdulke')
                                attendancePanel.classList.remove('qdulke')
                                me.disconnect()
                            }
                        }
                    }
                )

                document
                    .querySelector('.r6xAKc')
                    .insertAdjacentHTML('afterend', buttonHTML)
```

Figure 5.19 Initialize Google Meet Attendance Side Bar

The attendance sidebar is created once the user joins the meeting with the chrome extension added. The system will automatically insert the attendance bar into the right side of the google meet screen.

```
document
    .querySelector('.r6xAKc')
    .insertAdjacentHTML('afterend', buttonHTML)

const infoButton = document.querySelector('.r6xAKc button')
definePressedProperty(infoButton)

const attendanceButton = document.getElementById('attendance')
definePressedProperty(attendanceButton)

infoButton.addEventListener('click', (event) => {
    if (!infoButton.pressed) {
        if (!attendanceButton.pressed) {
            ariaPressedObserver.observe(
                attendanceButton,
                ariaPressedObserverOptions
            )
        } else {
            event.stopPropagation()
            infoButton.pressed = true
            document
                .querySelector('[data-tab-id="5"]')
                .classList.remove('qdulke')
            attendanceButton.pressed = false
            attendancePanel.classList.add('qdulke')
        }
    }
})

attendanceButton.addEventListener('click', (event) => {
    if (!attendanceButton.pressed) {
        const infoPanel =
            document.querySelector('[data-tab-id="5"]')
        if (infoPanel === null) {
            panelSpawnedObserver.observe(panelContainer, {
                childList: true,
            })
            attendanceTracker();
        } else {

            attendanceTracker();
            panelUnhiddenObserver.observe(
                document.querySelector('[data-tab-id="5"]'),
                {
                    attributes: true,
                    attributeFilter: ['class'],
                    attributeOldValue: true,
                }
            )
        }
        if (!infoButton.pressed) {
            ariaPressedObserver.observe(
                infoButton,
                ariaPressedObserverOptions
            )
        } else {
            event.stopPropagation()
            infoButton.pressed = false
            document
                .querySelector('[data-tab-id="5"]')
                .classList.add('qdulke')
            attendanceButton.pressed = true
            attendancePanel.classList.remove('qdulke')
        }
    }
})
```

Figure 5.20 Adding Attendance Button in Google Meet

Once the sidebar is initialized, the system will create an attendance button which enable the user to show the side bar at the right side of the google meet screen. Therefore, a clickable attendance button is generated in order to allow the user to access to the side bar created.

```
const exportButton = document.getElementById('export')
exportButton.addEventListener('click', () => {
    const sheetCode = document.getElementsByClassName("mdc-text-field__input")[0].value
    const msg = document.getElementById("errMsg");
    if(sheetCode !== ""){
        if(/^\d{6}$/.test(sheetCode)){
            chrome.runtime.sendMessage({command: "checkSheetCode", data:sheetCode}, (response) => {
                msg.innerHTML = "connecting...";
                Utils.log("Check and Assign Spreadsheet ID");
                var code = getMeetCode();
                chrome.runtime.sendMessage({
                    command: 'RetrieveStudentAttend',
                    data: code,
                    sheet: response.response,
                    meetCode: getMeetCode(),
                    currentTime: getCurrentTime()
                })

            });
            msg.innerHTML = "invalid sheet code...";
        }
        else{
            msg.innerHTML = "6-digits Sheet Code Required";
        }
    }
    else{
        msg.innerHTML = "Please Enter the Sheet Code";
    }
})
```

Figure 5.21 Google Meet Side Bar Export Button

An export button for the system to store the data in the google spreadsheet is created in order to keep track the student attendance. A validation of the sheet code is being check together when the user clicks on the export button.

5.2.3 Main Functions

There are a few main functions that involved in the face recognition and attendance module. First of all, **facial registration** is one of the functions that allow the student to capture their face and store into the firebase database for further recognition in the future. Next, **facial recognition** is the function that allow the system to track the student's face and recognise the name and ID of the student that registered in the database. **Attendance taking** is used to keep track of the student attendance after the student being recognise by the system. Last but not least, **eye tracking** is used to analyse the percentage of truthy when the student being ask by the teacher during the class. The system will start working only if the user activates the functions.

5.2.3.1 Facial Registration

```javascript
/**
 * Capturing face when user click facial registration
 */
function facialRegistration() {

    if(checkValidity()){
        var canvas = document.getElementById('canvas'),
        context = canvas.getContext('2d'),
        video = document.getElementById('video'),
        vendorUrl = window.URL || window.webkitURL;

        navigator.getUserMedia(
            { video: {} },
            stream => video.srcObject = stream,
            err => console.error(err)
        )

        Promise.all([
            faceapi.nets.tinyFaceDetector.loadFromUri(chrome.runtime.getURL('src/models/')),
            faceapi.nets.faceLandmark68Net.loadFromUri(chrome.runtime.getURL('src/models/')),
            faceapi.nets.faceRecognitionNet.loadFromUri(chrome.runtime.getURL('src/models/')),
            faceapi.nets.faceExpressionNet.loadFromUri(chrome.runtime.getURL('src/models/')),
            faceapi.nets.ssdMobilenetv1.loadFromUri(chrome.runtime.getURL('src/models/')),
            faceapi.nets.ageGenderNet.loadFromUri(chrome.runtime.getURL('src/models/')),
        ])

        video.addEventListener('play', function() {
            draw(this, context, 640, 480);
        }, false);

        async function draw(video, context, width, height) {

            context.drawImage(video, 0, 0, width, height);


            const displaySize = { width: width, height: height }
            faceapi.matchDimensions(context, displaySize)

            detectionsStore = await faceapi.detectAllFaces(video, new faceapi.TinyFaceDetectorOptions())
            .withFaceLandmarks()
            .withFaceExpressions()
            .withFaceDescriptors()
            .withAgeAndGender()

            if(detectionsStore.length > 0){
                const resizedDetections = faceapi.resizeResults(detectionsStore, displaySize)
                faceapi.draw.drawDetections(context, resizedDetections)
                faceapi.draw.drawFaceLandmarks(context, resizedDetections)
                faceapi.draw.drawFaceExpressions(context, resizedDetections)

                resizedDetections.forEach( detection => {
                    const box = detection.detection.box
                    const drawBox = new faceapi.draw.DrawBox(box, { label: detection.gender })
                    studGender.push({
                        'gender':detection.gender
                    })
                    drawBox.draw(canvas)
                })
            }
            setTimeout(draw, 100, video, context, width, height);
        }
    }
    else{
        console.log("Error Detected")
    }
}
```

Figure 5.22 Facial Registration Function

In the facial registration function, the system will check whether the face is valid for further extract for recognition. The system will draw the landmark on the face if there is any face being capture by the webcam of the computer.

```
/**
 * Add the face being capture into an Array
 */
function addFaceCaptured(){
    var studName = document.querySelector('.student-name').value;
    var studID = document.querySelector('.student-id').value;
    var studFace = dataURL;
    var malePredict = [];
    var femalePredict = [];

    studGender.forEach( gender => {

        if(gender.gender == "male"){
            malePredict.push({
                'gender':gender.gender
            })
        }
        else if (gender.gender == "female"){
            femalePredict.push({
                'gender':gender.gender
            })
        }
    })

    if(facesCapture.length > 5 ){
        alert("Congratulations, Your Face is Successfully Captured")
    }
    else{


        if(studName != "" && studFace != "" && studID != ""){

            if(malePredict.length > femalePredict.length){
                facesCapture.push({
                    'name': studName.toUpperCase(),
                    'id': studID,
                    'gender': "Male",
                    'face': studFace
                });
            }
            else if(femalePredict.length > malePredict.length){
                facesCapture.push({
                    'name': studName.toUpperCase(),
                    'id': studID,
                    'gender': "Female",
                    'face': studFace
                });
            }

        }
    }
}
```

Figure 5.22 Store Face for Facial Registration

Once the user captures the face, the system will store the image into an array. The system will store the image into the database only if the user clicks the save button. The student's name, student's id, gender and student's face will be store into an array before storing it into the database.

5.2.3.2 Facial Recognition

```
function loadLabeledImages() {

    Promise.all([
        faceapi.nets.tinyFaceDetector.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.faceLandmark68Net.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.faceRecognitionNet.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.faceExpressionNet.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.ssdMobilenetv1.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.ageGenderNet.loadFromUri(chrome.runtime.getURL('src/models/')),
    ])


    studentList.forEach((name, i) =>{
        labels[i] = name.name
    })

    console.log(labels)
    var x = 0;

    if(labels.length > 0){
        return Promise.all(

            labels.map(async label => {
                const descriptions = []

                for(let i = 1; i < 6; i++){
                    const img = await faceapi.fetchImage(studentList[x].image[i].DataUrl)
                    const detections = await faceapi.detectSingleFace(img).withFaceLandmarks().withFaceDescriptor()

                    descriptions.push(detections.descriptor)
                }

                if(x < studentList.length){
                    x++;
                }
                return new faceapi.LabeledFaceDescriptors(label, descriptions);

            })


        )
    }
}
```

Figure 5.23 Load Facial Image Stored in Database

The load facial image function is used to read all the face being stored in the database and the face will be further extract by the face-api.js library for further recognition. Different face will have different descriptions as it belongs to different persons. Therefore, the face-api.js will use the descriptions of the stored face that generated to carry out the facial recognition.

```
function facialAttendance(){
    getNameList();
    var canvas = document.getElementById('canvasAttd'),
    context = canvas.getContext('2d'),
    video = document.getElementById('videoAttd'),
    vendorUrl = window.URL || window.webkitURL;

    navigator.getUserMedia(
        { video: {} },
        stream => video.srcObject = stream,
        err => console.error(err)
    )

    Promise.all([
        faceapi.nets.tinyFaceDetector.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.faceLandmark68Net.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.faceRecognitionNet.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.faceExpressionNet.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.ssdMobilenetv1.loadFromUri(chrome.runtime.getURL('src/models/')),
        faceapi.nets.ageGenderNet.loadFromUri(chrome.runtime.getURL('src/models/')),
    ])

    video.addEventListener('play', function() {
        draw(this, context, 640, 480);
    }, false);


    async function draw(video, context, width, height) {

            context.drawImage(video, 0, 0, width, height);
            const displaySize = { width: width, height: height }
            faceapi.matchDimensions(context, displaySize)

            if(studentList.length > 0){
                const labeledFaceDescriptors = await loadLabeledImages()
                const faceMatcher = new faceapi.FaceMatcher(labeledFaceDescriptors, 0.9)
                const detections = await faceapi.detectAllFaces(video, new faceapi.TinyFaceDetectorOptions())
                .withFaceLandmarks()
                .withFaceDescriptors()

                if(detections.length > 0){
                    const resizedDetections = faceapi.resizeResults(detections, displaySize)
                    const results = resizedDetections.map(d => faceMatcher.findBestMatch(d.descriptor))

                    results.forEach((result, i) => {
                        const box = resizedDetections[i].detection.box
                        const drawBox = new faceapi.draw.DrawBox(box, { label: result.toString() })
                        drawBox.draw(context)
                        attendedStud = result.label
                    })
                }
                setTimeout(draw, 500, video, context, width, height);
            }
    }


}
```

Figure 5.24 Facial Recognition Function

The facial recognition function will load all the necessary API that need to be used for face detection and recognition in order to obtain the accurate result in recognize the face detected. The facial landmark and face's information will be printed out on the google meet screen for the user.

5.2.3.3 Attendance Taking

```
function saveAttendance(){

    if(attendedStud != null){
        alert("Attendance " + attendedStud + " is taken.");
        clearVideo();
        stopWebCamera();
        document.getElementById('takeAttendanceCard').style.visibility = 'hidden';
        let meetingCode = getMeetCode();
        chrome.runtime.sendMessage({command: "SaveStudentAttendance", data: attendedStud, type: meetingCode, email: currentEmail, time: Date.now()}, (response)
=> {
            Utils.log("Take Student Attendance");
        });

        studentList = [];
        labels = [];
    }else{
        alert("Sorry, there is an error occured. Please try again");
    }

}
```

Figure 5.25 Save Attendance

The student's attendance will being store in the database once the user clicks the save button. A command will be send using the `chrome.runtime.sendMessage()` in order to carry out the process at the background.

5.2.3.4 Eye Tracking

```
function faceMeshInRealTime() {

    async function faceMeshFrame() {
        var ctx = state.canvas.getContext("2d");
        ctx.drawImage(state.video, 0, 0);

        //Detect Face Mesh in Video
        var face = await detectFaceMesh();

        if (face.length > 0)
        {
            face.forEach((prediction) => {
                const keypoints = prediction.scaledMesh;

                state.leftEyeAnalysis.innerHTML = "Truth Teller: 0 %";
                state.eyeBox.innerHTML = "DETECTING...";

                if(Math.abs(keypoints[466][0] - keypoints[468][0]) < 5){
                    state.eyeBox.innerHTML = "LOOKING LEFT";
                    left.push("left");
                }
                else if(Math.abs(keypoints[263][0] - keypoints[468][0]) < 5){
                    state.eyeBox.innerHTML = "LOOKING LEFT";
                    left.push("left");
                }
                else if (Math.abs(keypoints[468][0] - keypoints[398][0]) < 5){
                    state.eyeBox.innerHTML = "LOOKING RIGHT";
                    right.push("right");
                }
                else if (Math.abs(keypoints[468][0] - keypoints[364][0]) < 5){
                    state.eyeBox.innerHTML = "LOOKING RIGHT";
                    right.push("right");
                }

                state.percentTruth = (left.length / (left.length + right.length)) * 100;
                state.leftEyeAnalysis.innerHTML = "Truth Teller: " + state.percentTruth + "%";

            });
        }
        else{
            state.eyeBox.innerHTML = "";
            state.leftEyeAnalysis.innerHTML = "";
        }

        //Refresh
        requestAnimationFrame(faceMeshFrame);
    }

    faceMeshFrame();
}
```

Figure 5.26 Analyze and Display Eye Tracking Result

The eye tracking result will be draw on the google meet screen in order for the student and tutor to view the result. The estimation of iris position is calculate based on the position of the landmark draw on the iris and eyelid. A percentage of truth teller will be display on the screen based on the student's iris position during the online class activities.

5.2.4 Database

There is a total of 2 database being used in the face recognition and attendance module which are firebase and google spreadsheet. The setup and implementation of the database will be discussed.

5.2.4.1 Firebase

```
/**
 * Chrome Extension Firebase Configuration
 */
var firebaseConfig = {
    apiKey: "AIzaSyCFe9RV9CHaR-yqzCwe65K8K8iQKxozLNY",
    authDomain: "smartclassroomjs.firebaseapp.com",
    projectId: "smartclassroomjs",
    storageBucket: "smartclassroomjs.appspot.com",
    messagingSenderId: "356809279066",
    appId: "1:356809279066:web:bf986b4e7cb3c5b2dd3a63"
};

/**
 * Initialize Firebase
 */
firebase.initializeApp(firebaseConfig);
```

Figure 5.27 Firebase Configuration

The *apiKey*, *authDomain*, *projectId*, *storageBucket*, *messegingSenderId and appId* will need to specified before using the firebase in the chrome extension. The firebase real time databse is used in the system in order to store the student information and the image URL of face captured while the firebase storage is used to store the image of the face captured. The management is able to view the face captured in the firebase storage.

```
case "saveFaceToFirebase": {
    chrome.identity.getAuthToken({'interactive': true}, function(token) {
            if(request.data.length > 0) {

                for(let i = 1; i < request.data.length; i++){

                        var studFace = request.data[i].face;
                        var studName = request.data[0].name;
                        var studID = request.data[0].id;
                        var gender = request.data[0].gender

                        const dbRealTime = firebase.database().ref('RegisteredStudent/' + studID + "-" + studName);
                        dbRealTime.set({
                            Name: studName,
                            ID: studID,
                            Email: request.email,
                            Gender: gender
                        });

                        //To split out the format of image and get only the face image data
                        const image = studFace.split(',')[1];
                        const metadata = {
                            contentType: 'image/png',
                        };

                        const dbStorage = firebase.storage().ref('StudentFace/' + studID + "-" + studName + "/" + i + ".png");
                        dbStorage.putString(image, 'base64', metadata).then(snapshot => {
                            console.log('snapShot',snapshot);
                            snapshot.ref.getDownloadURL().then(function(downloadUrl) {
                                const dbFaceRealTime = firebase.database().ref('RegisteredStudent/' + studID + "-" + studName + "/Images/" + i);

                                dbFaceRealTime.set({
                                    ImageUrl: downloadUrl,
                                    DataUrl: studFace
                                });
                            })

                        }).catch(error => {
                            console.log(error);
                        });
                }
            }
    });


}
```

Figure 5.28 Storing Face into Firebase Storage

To store the face into the Firebase, the system will store the student information and image URL into the real time database and image into the firebase storage. A sample of how to store the student information and face into both firebase database in shown in figure 5.28.

```
case "RetrieveStudentName":{
    var studList = [];

    const dbStud = firebase.database().ref('RegisteredStudent/').on('value', (snapshot) => {
        console.log(snapshot.val());
        snapshot.forEach(function(childSnapshot) {
            studList.push({
                'name': childSnapshot.key,
                'image': childSnapshot.val().Images
            })
        });

        sendResponse({
            response: studList
        });

        }, (errorObject) => {
        console.log('The read failed: ' + errorObject.name);
    });

    return true;
}
```

Figure 5.29 Retrieve Student Information from Firebase

The student information is retrieved from the firebase for the student attendance checking and management purpose. The information retrieve from the database will store into the array in order to allow the system in read the data at the front end.

```
        case "SaveStudentAttendance":{
            var curData = request.data
            var fields = curData.split('-')
            var studID = fields[0];
            var studName = fields[1].toUpperCase();
            const attend = firebase.database().ref('AttendedStudent/' + request.type + '/' + request.data);

            chrome.storage.sync.get(['start'], function(result) {
                attend.set({
                    Name: studName,
                    ID: studID,
                    Email: request.email,
                    TakeTime: new Date().toLocaleTimeString(),
                    StartTime: result.start,
                    EndTime: ""
                });
            });
        }
        case "RetrieveStudentAttend":{
            chrome.identity.getAuthToken({ interactive: true }, (token) => {
                var attendList = [];
                firebase.database().ref('AttendedStudent/' + request.data).on('value', (snapshot) => {
                    snapshot.forEach(function(childSnapshot) {
                        attendList.push({
                            'id': childSnapshot.val().ID,
                            'name': childSnapshot.val().Name,
                            'email': childSnapshot.val().Email,
                            'start': childSnapshot.val().StartTime,
                            'take': childSnapshot.val().TakeTime,
                            'end': childSnapshot.val().EndTime
                        })
                    });

                  }, (errorObject) => {
                    console.log('The read failed: ' + errorObject.name);
                });

            })
        }
        case "RetrieveStudentAttdDetail":{
            var attedListDetail = [];
            firebase.database().ref('AttendedStudent/' + request.data).on('value', (snapshot) => {
                snapshot.forEach(function(childSnapshot) {
                    attedListDetail.push({
                        'name': childSnapshot.val().Name,
                        'time': childSnapshot.val().StartTime
                    })
                });
                sendResponse({
                    response: attedListDetail
                });

                }, (errorObject) => {
                    console.log('The read failed: ' + errorObject.name);
            });
        }
        case "user-leave":{
            if(request.active){
                const dbAttd = firebase.database().ref('AttendedStudent/' + request.type + '/' + request.active);
                dbAttd.update({
                    'EndTime': request.data
                })
            }

        }
```

Figure 5.30 Firebase Update for Attendance Function

Figure 5.30 shows the way to add, update and retrieve the student attendance into the firebase database. The student's name, student's id, student's email, student taking attendance time, student join the google meet time and student leave the google meet time will be updated into the database by the system. The end time will only be updated once the user leaves the google meet by end the google meet call.

5.2.4.2 Google Spreadsheet

```
let clientToken;

postJWT(getJWT(), function (response) {
    clientToken = JSON.parse(response).access_token;
});

const API_KEY = 'AIzaSyCzT1Ltha85DX-xRemUx1b9JkipCdPEgiU';
const DISCOVERY_DOCS = ["https://sheets.googleapis.com/$discovery/rest?version=v4"];
const MANAGEMENT_SPREADSHEET_ID = '1Oa6kkkpFVh7o98gEmoNgnAsQmvNeRmzu8awyNrp3wWI';
const MANAGEMENT_SPREADSHEET_TAB_NAME = 'main';
let range = null;

// Initialize gapi
function onGAPILoad() {

    gapi.client.init({
        // Don't pass client nor scope as these will init auth2, which we don't want
        apiKey: API_KEY,
        discoveryDocs: DISCOVERY_DOCS,

    }).then(function () {
        // Set service account token
        gapi.auth.setToken({
            'access_token': clientToken,
        });

        console.log('gapi initialized');

    }, function (error) {
        console.log('Error = ', error);
    });
}
```

Figure 5.31 Initialize Google Sheets API

The system will need to initialize the Google Sheets API in order to used the google spreadsheet in the system. The `onGAPILoad()` is used to check the validity of the API key bu using the apiKey and discoveryDocs. A confirmation message will be displayed once the API key is valid. This function plays an important role for the system to store the information into the google sheets.

```
// Check whether sheet code in the management spreadsheet
function checkSheetCode(programmeAvailable, sheetCode, meetName) {

    for (let i = 0; i < programmeAvailable.length; i++) {
        console.log(programmeAvailable[i][0]);

        if (sheetCode === programmeAvailable[i][0]) {

            SPREADSHEET_ID = programmeAvailable[i][2];

            data.sheetCode = programmeAvailable[i][0];
            data.detail = programmeAvailable[i][1];
            data.sheetID = programmeAvailable[i][2];

            checkSheetTab(data.sheetID, meetName);

            break;
        }
    }

    // Send data to popup.js to process
    chrome.runtime.sendMessage(data);
}
```

Figure 5.32 Verify Sheet Code Validity

The system required the user to enter the sheet code before storing the data into the google sheets in order to verify the correctness of the sheet code. Therefore, a `checkSheetCode()` function is used to retrieve the valid sheet code from the main google sheets and compare with the current sheet code entered by the user.

Figure 5.33 Create Header in Google Sheet

The table header that wants to display in the google sheet is set in the system as shown in figure 5.33. Once the user exports the student attendance to the google sheets, the system will check whether the sheets contain any header. An empty sheet with proper header will be created, when the user export the data to the google sheets.

```
function addSheet(sheetName) {
    const requests = [
        {
            addSheet: {
                properties: {
                    title: sheetName,
                    gridProperties: {
                        rowCount: 2,
                        columnCount: 8,
                        frozenRowCount: 1,
                    },
                },
            },
        },
    ]
    // requests.push(createSheetMetadata(className, sheetId))
    return requests
}
```

Figure 5.34 Add sheet into the current sheets

In order to have a proper management for the data enter the google sheets, a new sheet will be added into the current spreadsheet every time when the user export the data.

```
async function createSpreadsheet(token, code, attend, sheet, meetCode, currentTime) {
   if(attend.length > 0){
       const body = {
           properties: {
               title: 'Google Meet Facial Attendance',
               spreadsheetTheme: getSpreadsheetTheme(),
           },
       }
       const init = {
           method: 'POST',
           async: true,
           headers: {
               Authorization: 'Bearer ' + token,
               'Content-Type': 'application/json',
           },
           body: JSON.stringify(body),
       }
       let spreadsheetId = sheet;
       let sheetId = 0;
       let requests = []
       requests = requests.concat(createHeaders(sheetId))
       const icReqs = await initializeCells(code, sheetId,attend)
       requests = requests.concat(icReqs)
       const data = await batchUpdate(token, requests, spreadsheetId, sheetId)
       console.log('Initialize spreadsheet response:')
       console.log(data)
       if(data == null){
           updateSpreadsheet(token, code, attend, sheet, meetCode, currentTime)
       }

   }

}

async function updateSpreadsheet(token, code, attend, spreadsheetId, meetCode, currentTime
) {
    let requests = [];
    let sheetName = `${meetCode}$${new Date().toLocaleTimeString()}`;
    Utils.log('Updating spreadsheet...')

        let sheetId
        const spreadsheet = await getSpreadsheet(token, spreadsheetId)
        sheetId =
            spreadsheet.sheets.reduce(
                (acc, sheet) => Math.max(acc, sheet.properties.sheetId),
                0
            ) + 1
        requests = requests.concat(addSheet(sheetName))
        // requests = requests.concat(createHeaders(sheetId))
        Utils.log(`Creating new sheet for class ${code}, ID ${sheetId}`)

    let data = await batchUpdate(token, requests, spreadsheetId, 0);

    let spreadsheetProperties = data.updatedSpreadsheet.sheets;
    let maxlength = data.updatedSpreadsheet.sheets.length
    let sheetID = spreadsheetProperties[maxlength - 1].properties.sheetId;

    editSpreadSheet(token, code, attend, spreadsheetId, sheetID);
}
```

Figure 5.35 Create new Spreadsheet and Update Current Spreadsheet

Figure 5.35 shows the way on how to implement the spreadsheet function in the system. The system will create a new spreadsheet and header in the google sheet for the first time. Then, the system will allow the user to update the spreadsheet after the first-time update if there is any updated information need to store. For creating the new spreadsheet, the system will create a new header in the sheets.

5.3 Results

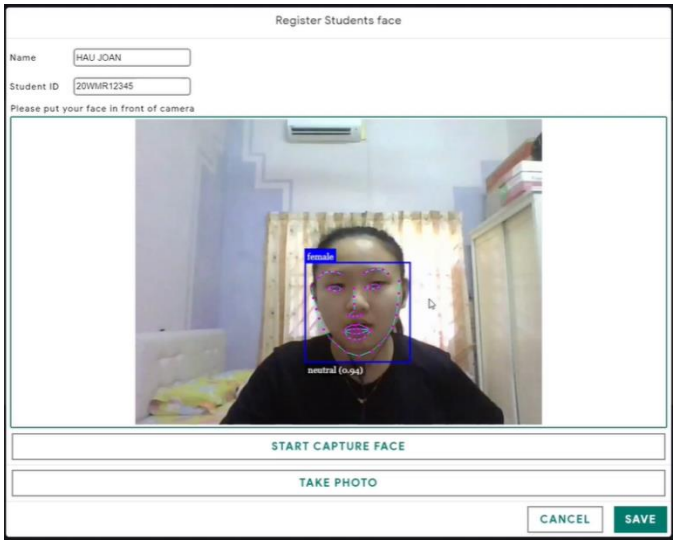


Figure 5.36 Facial Detection and Registration

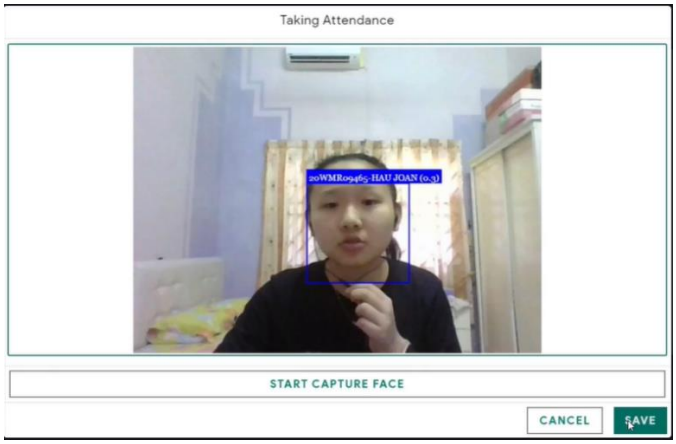


Figure 5.37 Facial Recognition and Attendance Taking

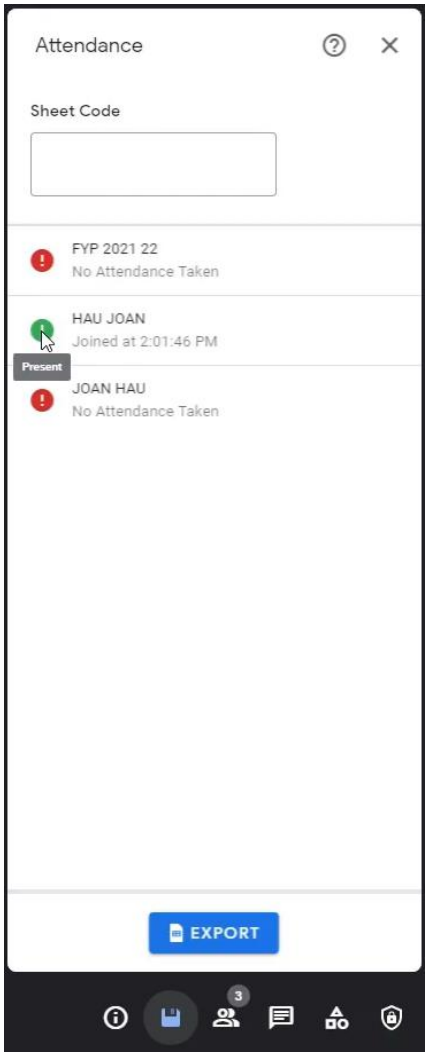


Figure 5.38 Google Meet Side Bar for Attendance Tracking

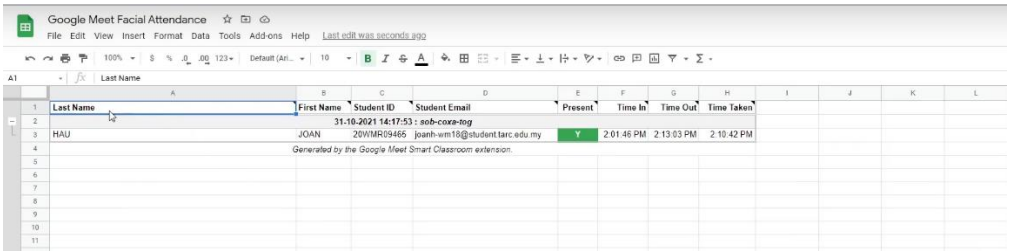


Figure 5.39 Google Sheets Attendance

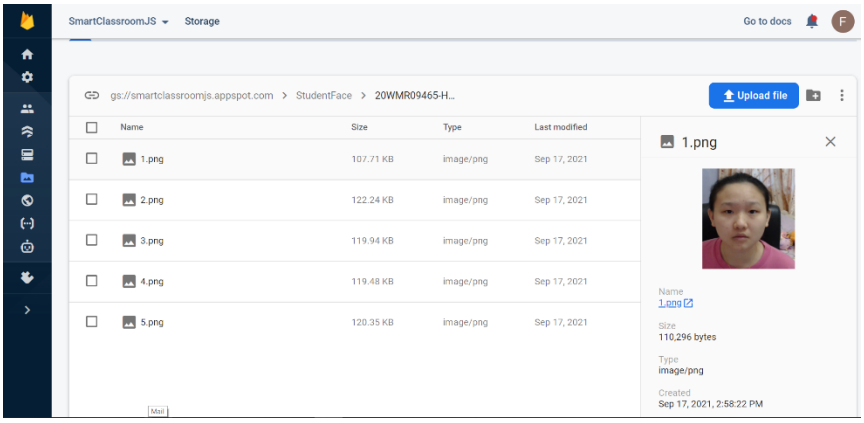


Figure 5.40 Face Captured and Stored in the Firebase Storage

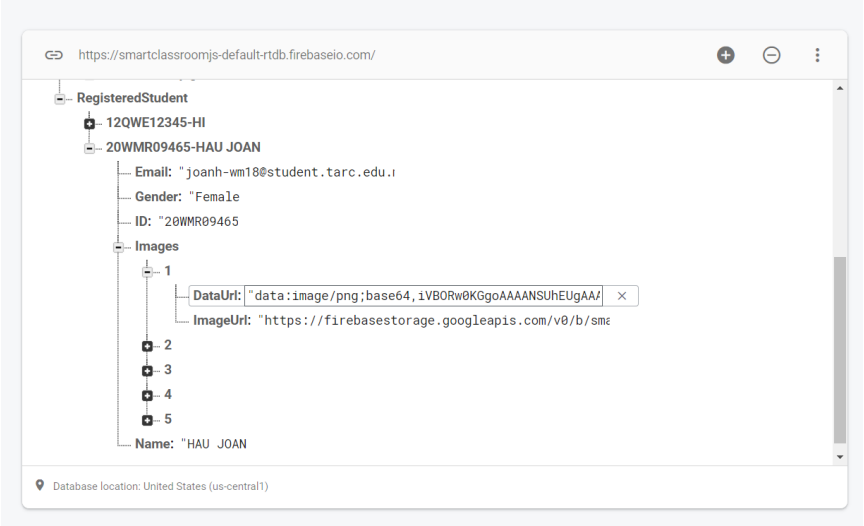


Figure 5.41 Student Information Stored in Firebase Database

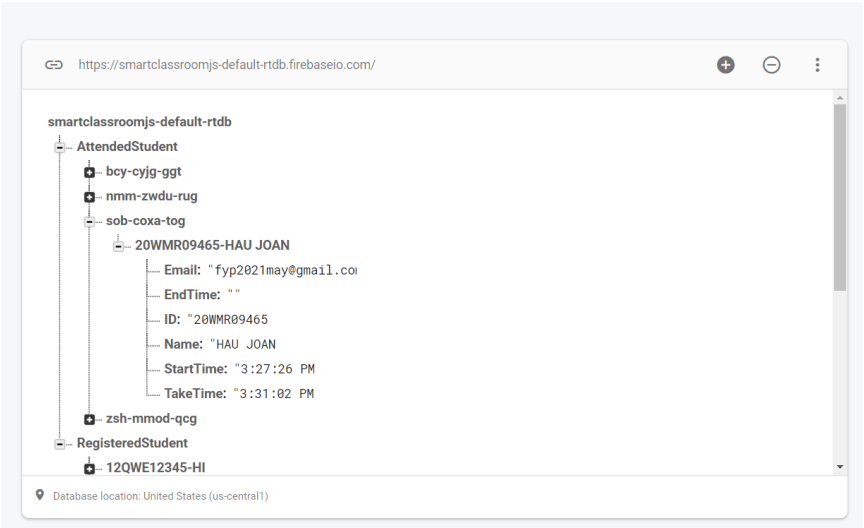


Figure 5.42 Attendance Tracking in Firebase Database

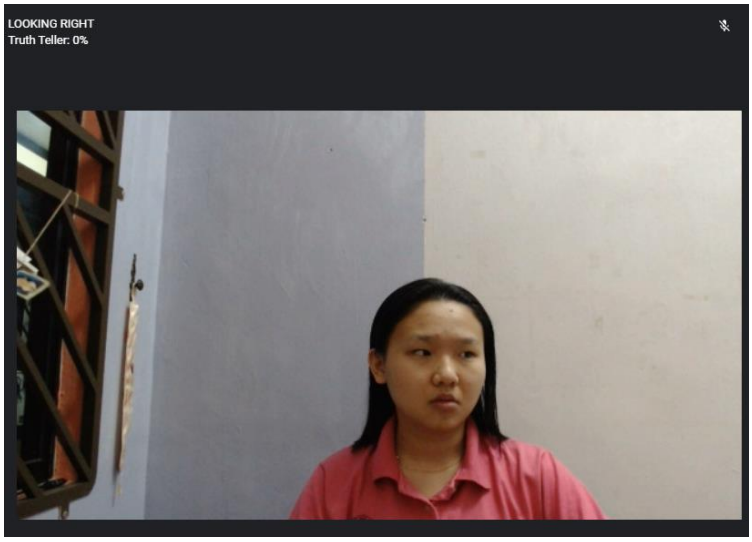


Figure 5.43 Eye Tracking in Google Meet

5.4 Chapter Summary and Evaluation

In Chapter 5, the way of implementing the function of the face recognition and attendance module is discussed. Besides, this chapter also covered the way to setup the system such as the chrome extension configuration, face-api.js initialization, load the face detection and recognition API, Google Sheets API, etc. Majority of the complicated function which need to setup before enable it is covered in this section in order to provide the reader a clear concept of how the system is being implemented. Other than that, the results of the face detection and recognition module is shown. In conclusion, this section covered the discussion about the main functions of the face recognition and attendance module which include register face, detect face, recognize face, store face, store student attendance, manage student attendance, etc.

Chapter 6

Testing

6 Testing

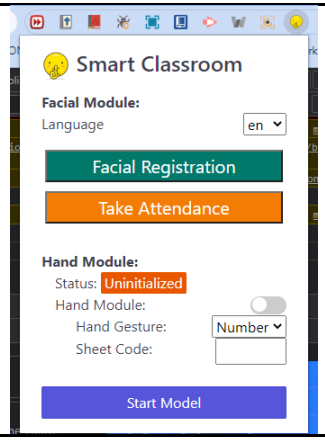
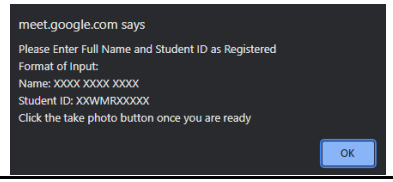
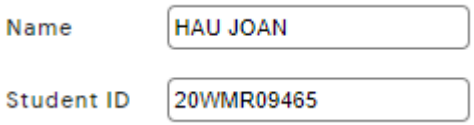
Testing is one of the important phases of the Software Development Life Cycle (SDLC). Face Recognition and Attendance Module will be tested in order to ensure that all the requirements met the expectations. In this stage, there are several cases will be created to test the module as shown below.

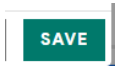
6.1 Test Case 1

Test Case #: TC1	**Test Case Name: Face Registration**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Register the face into the firebase database.	

Pre-conditions: The chrome extension is added in the google chrome before browsing the google meet.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Click the system icon in the top right of google chrome.	Display the popup screen of the system which include the face and hand module.	Display the popup screen of the system which include the face and hand module.	Pass	

					
2	Click the *Facial Registration* button. 	Display a message which contain the instruction of how to register the face using the system.	Display a message which contain the instruction of how to register the face using the system.	Pass	
3	Click *OK*. 	Display a facial registration screen.	Display a facial registration screen.	Pass	
4	Enter a valid name and valid student id into the column provided and click the *START CAPTURE FACE*. Name: HAU JOAN Student ID: 20WMR09465 	No error message pop out, the webcam is opened and able to detect the face shown in front of the camera with gender detection, facial landmark and emotion detection.	No error message pop out, the webcam is opened and able to detect the face shown in front of the camera with gender detection, facial landmark and emotion detection.	Pass	

5	Click *TAKE PHOTO* 6 times. 	Display a successful message. *Congratulations, Your Face is Successfully Captured*	Display a successful message. *Congratulations, Your Face is Successfully Captured*	Pass	
6	Click *SAVE.* 	Display a successful message. *Congratulations, Faces are successfully saved.*	Display a successful message. *Congratulations, Faces are successfully saved.*	Pass	

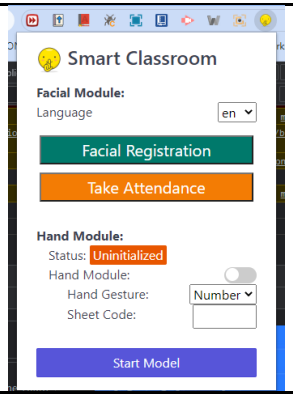
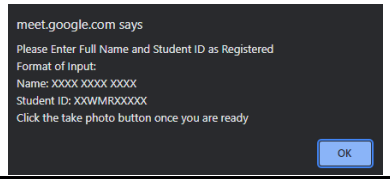
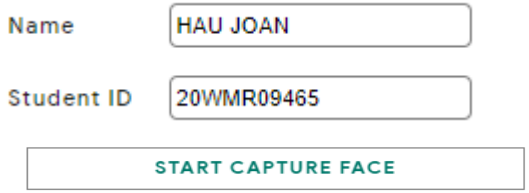
Post-conditions: User successfully store the user information and faces information in the firebase Realtime database and storage.

6.2 Test Case 2

Test Case #: TC2	**Test Case Name: Face Registration with valid input**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Test the face registration function with valid data.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Click the system icon in the top right of google chrome.	Display the popup screen of the system which include the face and hand module.	Display the popup screen of the system which include the face and hand module.	Pass	

					
2	Click the *Facial Registration* button. 	Display a message which contain the instruction of how to register the face using the system.	Display a message which contain the instruction of how to register the face using the system.	Pass	
3	Click *OK*. 	Display a facial registration screen.	Display a facial registration screen.	Pass	
4	Enter a valid name and valid student id into the column provided and click the *START CAPTURE FACE*. Name: HAU JOAN Student ID: 20WMR09465 	No error message pop out, the webcam is opened and able to detect the face shown in front of the camera with gender detection, facial landmark and emotion detection.	No error message pop out, the webcam is opened and able to detect the face shown in front of the camera with gender detection, facial landmark and emotion detection.	Pass	

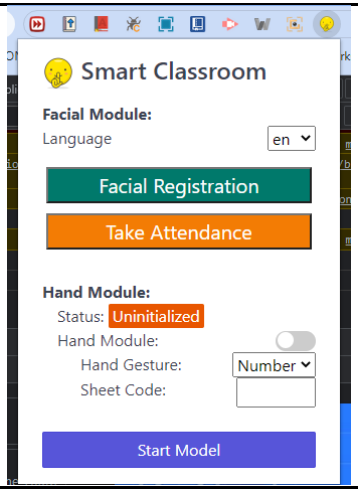
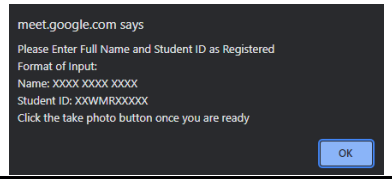
Post-conditions: User successfully enable the face detection function.

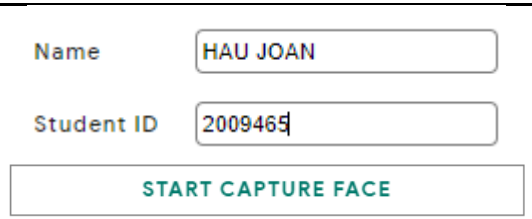
6.3 Test Case 3

Test Case #: TC3	**Test Case Name: Face Registration with invalid student id**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Test the face registration input with invalid student id.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Click the system icon in the top right of google chrome.	Display the popup screen of the system which include the face and hand module.	Display the popup screen of the system which include the face and hand module.	Pass	

					
2	Click the *Facial Registration* button. 	Display a message which contain the instruction of how to register the face using the system.	Display a message which contain the instruction of how to register the face using the system.	Pass	
3	Click *OK*. 	Display a facial registration screen.	Display a facial registration screen.	Pass	
4	Enter a valid name and invalid student id into the column provided and click the *START CAPTURE FACE*. Name: HAU JOAN Student ID: 2009465	Display an error message. *Wrong Format Detected for Student ID* *Student ID Format: XXWMRXXXXX*	Display an error message. *Wrong Format Detected for Student ID* *Student ID Format: XXWMRXXXXX*	Pass	

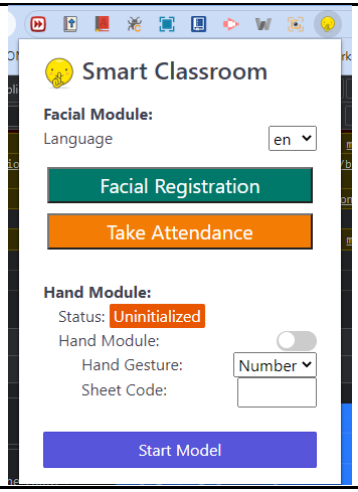
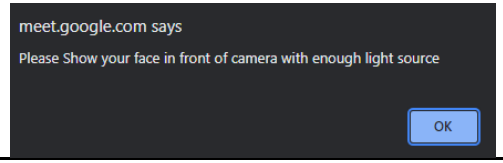
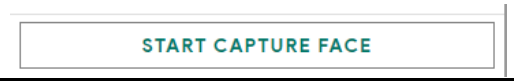
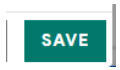
Post-conditions: User fail to enable the face detection function.

6.4 Test Case 4

Test Case #: TC4	**Test Case Name: Attendance Taking with Face Recognition**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Take the student attendance using the face recognition function.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Click the system icon in the top right of google chrome.	Display the popup screen of the system which include the face and hand module.	Display the popup screen of the system which include the face and hand module.	Pass	

					
2	Click the *Take Attendance* button. 	Display a message which contain the instruction of how to take the attendance.	Display a message which contain the instruction of how to take the attendance.	Pass	
3	Click *OK*. 	Display Taking Attendance Screen.	Display Taking Attendance Screen.	Pass	
4	Click *START CAPTURE FACE*. 	Webcam open automatically and the video is display in the Taking Attendance Screen.	Webcam open automatically and the video is display in the Taking Attendance Screen.	Pass	
5	Click *SAVE*. 	Display a confirmation message with the student information detected. *Attendance 20WMR09465-HAU JOAN is taken.*	Display a confirmation message with the student information detected. *Attendance 20WMR09465-HAU JOAN is taken.*	Pass	

Post-conditions: User successfully save the attendance in firebase Realtime database.

6.5 Test Case 5

Test Case #: TC5	**Test Case Name: Google Meet Sidebar**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Test the Google Meet Sidebar.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

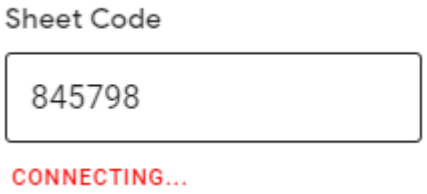
Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Browse the Google Meet with Google Meet Smart Classroom Chrome Extension.	Display an attendance icon at the bottom right of the google meet screen once loaded.	Display an attendance icon at the bottom right of the google meet screen once loaded.	Pass	
2	Click the *Attendance* button. 	Display a sidebar with all the current participate in the current google meet link.	Display a sidebar with all the current participate in the current google meet link.	Pass	

Post-conditions: User successfully view the current participate in the sidebar.

6.6 Test Case 6

Test Case #: TC6	**Test Case Name: Export the attendance to the Google Sheets**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Export the current student attendance to the Google Sheets.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Browse the Google Meet with Google Meet Smart Classroom Chrome Extension.	Display an attendance icon at the bottom right of the google meet screen once loaded.	Display an attendance icon at the bottom right of the google meet screen once loaded.	Pass	
2	Click the *Attendance* button. 	Display a sidebar with all the current participate in the current google meet link.	Display a sidebar with all the current participate in the current google meet link.	Pass	
3	Enter a valid sheet code and Click *Export* button. Sheet Code: 845798 	Display *CONNECTING* and Export a Google Sheets with current student attendance.	Display *CONNECTING* and Export a Google Sheets with current student attendance.	Pass	

Post-conditions: User successfully export a Google Sheets with students' attendance.

6.7 Test Case 7

Test Case #: TC7	**Test Case Name: Export the attendance to the Google Sheets with invalid Sheet Code.**
System: Smart Classroom System	**Module: Face Recognition and Attendance Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Export the current student attendance to the Google Sheets with invalid Sheet Code.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Browse the Google Meet with Google Meet Smart Classroom Chrome Extension.	Display an attendance icon at the bottom right of the google meet screen once loaded.	Display an attendance icon at the bottom right of the google meet screen once loaded.	Pass	
2	Click the *Attendance* button. 	Display a sidebar with all the current participate in the current google meet link.	Display a sidebar with all the current participate in the current google meet link.	Pass	
3	Enter a valid sheet code and Click *Export* button. Sheet Code: 111111	Display *INVALID SHEET CODE* and export a Google Sheets with current student attendance.	Display *INVALID SHEET CODE* and export a Google Sheets with current student attendance.	Pass	

Post-conditions: User fail to export a Google Sheets with students' attendance.

6.8 Test Case 8

Test Case #: TC8	**Test Case Name: Eye Tracking**
System: Smart Classroom System	**Module: Eye Tracking Module**
Design By: Joan Hau	**Design Date: 20 October 2021**
Executed By: Joan Hau	**Execution Date: 20 October 2021**
Short Description: Test the eye tracking function.	

Pre-conditions: The user have loaded the Google Meet Smart Classroom Chrome extension.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Browse the Google Meet with Eye Monitoring Chrome Extension and open the Google Meet Camera.	Display a eye tracking status at the top left of the google meet screen.	Display a eye tracking status at the top left of the google meet screen.	Pass	

					
2	Eye look at the left of the screen in front of the camera. 	Display the *LOOKING LEFT* status with percentage of truth teller.	Display the *LOOKING LEFT* status with percentage of truth teller.	Pass	
3	Eye look at the right of the screen in front of the camera. 	Display the *LOOKING RIGHT* status with percentage of truth teller.	Display the *LOOKING RIGHT* status with percentage of truth teller.	Pass	

Post-conditions: User successfully enable the eye tracking function.

6.9 Chapter Summary and Evaluation

This chapter discussed all the necessary test cases for the functions of the face recognition and attendance module and eye tracking module. This section is important to ensure the module developed meet the module requirements and expectations. Last but not least, all test cases are carried out and all the requirements are met for this module.

Chapter 7

Discussions and Conclusion

7 Discussions and Conclusion

This chapter will cover the project summary, the achievement made at the end of this project, the contributions, limitations and future improvements of the implemented system as well as the issues and solutions involved during the process of the project development.

7.1 Summary

As the global pandemic that we face today, online learning has become one of the popular teaching methods not only for higher institutions but also secondary and primary schools. Google Meet, Zoom, Microsoft Teams, etc., are examples of online learning platforms used by institutions nowadays. Human resources' manual monitoring of the students' activities is getting outcasted from society. Therefore, the implementation of Artificial Intelligent in the e-learning platform has become popular. Technological advancements have brought forth to us some advantages. As these technologies save more resources while providing superior performance, they gradually replace manpower. Due to the rising trends of the students in the higher institutions, the needs for the lecturer or tutor to ensure the students' concentration is increased. Therefore, this proposed module was designed and implemented to increase the productivity of the lecturer or tutors in marking the student attendance by using facial recognition and increase the efficiency in monitoring the students' behaviour during the class activities by using eye-tracking.

Additionally, the incremental model is applied to develop this system's module as it can separate the whole software development process into multiple tasks. Some tools are being applied to the face recognition and attendance module and eye-tracking module in the smart classroom system, including face-api.js, MediaPipe face landmarks detection, Google Sheets and firebase database. Face-api.js is being used because it consumes the least memory than MediaPipe and TensorFlow. Besides, face-api.js is also built using the TensorFlow library. Therefore, it provides all the functions provided by the TensorFlow face detection but with better performance and low memory consumption, making it more affordable for most users' computers. It is essential to use a standard face detection library as this chrome extension will need to use for many students from different places with different computer specifications. Therefore, I will need to ensure that this library can run smoothly on most laptop specifications. Other than that, MediaPipe is used for eye tracking as MediaPipe provides a default eye movement tracking that the landmark will display based on the user movement. This function enables me to track the user's iris position more accessible and obtain a more accurate result from the user's iris movement. There is a total of 2 databases being applied in the face recognition and attendance module, which are the firebase database and Google Sheets. Having

two databases is for the staff to have an effective way to manage the information. The firebase database is prepared for the staff with IT-related backgrounds and needs to track all the student information in the database. In contrast, Google Sheets is prepared for the lecturer and tutor who does not have an IT-related background and can adequately manage by exporting Google sheets. In a nutshell, a Google Chrome Browser is a must for this project execution as it is a chrome extension that must be run on google chrome.

7.2 Achievements

At first, the objective of this project is to provide a system that can keep track of student attendance using facial recognition. The objective has been met by the current implemented system. The face recognition and attendance module is developed with facial registration, take attendance, management, etc. Currently, the students are able to register their faces into the firebase database through the Google Meet Smart Classroom System. Besides, an attendance taking function is implemented in order for the students to take the attendance through the facial recognition function. The system will automatically update the student attendance once the student information is detected by the system's model. A firebase database and firebase storage are used for the face recognition and attendance module to store the student's information and the student's image. Besides, a management screen is also provided for the lecturer and tutor to view the current participants with attendance status at the sidebar of the google meet. Other than that, the system is able to export the attendance sheets for the lecturer to make a copy of the attendance for future references in order to avoid any attendance cheat among the students.

98% of the proposed functionality and requirements has been done for the face recognition and attendance module. The remaining percentage indicates the bugs and defects that did not found during the system testing. In addition, an extra eye-tracking module is implemented separately for the lecturer and tutor to monitor the students' behaviours. The eye-tracking module will be tracking the position of the iris. The module uses a calculation to identify the current user's iris position. Last but not least, this project can only support the Google Meet Desktop site.

7.3 Contributions

The proposed smart classroom system helps overcome the inconveniences and problems of traditional student attendance systems currently on the market, such as the use of paper, fingerprints and QR codes. This proposed system will address issues such as attendance fraud and attendance taking a long time to complete by using the student's face as an input to take attendance in each class, which is the fastest and most accurate method compared to other

attendance systems. Therefore, the proposed system will increase the efficiency and effectiveness of student attendance and will be able to improve the quality of education at TARUC as attendance time is greatly reduced. In addition, we use an online database that allows instructors and lecturers to view student attendance anytime and anywhere. In addition, the eye-tracking module, which helps in monitoring the students' iris position, is implemented. The system will analyze the percentage of truth-tellers based on the student's eye, which can help the lecturer and tutor filter the truth of the talk given by the students during the classes.

7.4 Limitations and Future Improvements

The most worrisome limitation is the facial detection speed in the facial registration and facial recognition section. The facial detection library used in this proposed system is face-api.js, one of the most popular face detection APIs used for JavaScript programming. Due to the low hardware specifications, the system will delay in displaying the detection result in the google meet. For example, the user will need to wait for a few seconds to let the webcam open automatically and display the result on the popup screen, which will influence user satisfaction. The detection speed is slow as the laptop GPU which execute this system is low. Therefore in order to increase user satisfaction, a high-performance computer is needed. The GPU is required in order to launch the face-api.js API in the google meet. Therefore, the computer's GPU quality will highly affect the performance of the face recognition and attendance module in the proposed system.

Additionally, the limitation of the eye-tracking module is the lack of an available eye-tracking API that can be used to develop the eye-tracking function in the proposed system. Many libraries can be used to detect the iris position in Python, such as OpenCV, Dlib, etc., but not in JavaScript. Therefore, facial landmark detection in MediaPipe is used in the eye-tracking module in order to track the iris position. A simple calculation based on the face landmark is carried out to get the iris's distance and get the estimated position of the iris. For future improvements, the system can add the proper eye tracking library to keep track of the iris position instead of manually calculating based on the face landmarks.

7.5 Issues and Solutions

The most problematic issue I encountered in developing the proposed system is the tool and technique used during the development process, as they are new for us. For instance, how to develop the system as a chrome extension to execute in Google Meet, use the Firebase database at the background of the system when executing in Google meet, and apply the Google Sheets API in chrome extension and connecting to Google Services, etc. Fortunately, we successfully

overcame all the issues we faced by referring to lots of documents and community groups such as GitHub, stack overflow, etc.

Additionally, due to the lack of performance research on face detection and recognition and hand detection API, it is difficult to decide the better API to be used in the chrome extension. Therefore, we have researched in order to figure out the available API for face and hand detection. For instance, MediaPipe, TensorFlow and face-api.js are selected to explore further the most suitable face detection API in this proposed system. The research results show that face-api.js need the least time to carry out face detection in both high and low specifications of the computer. In the end, face-api.js are selected and implemented into the proposed face detection and recognition system.

Moreover, the video stream of Google Meet cannot be input into the system for face and gesture detection, which makes the system inoperable. So to solve this problem, we also did a lot of research and replaced the Google Meet video stream with our video stream and fed our video stream into the system for face and gesture detection. By using our video stream, the system can perform the detection and draw face and hand landmarks in the video stream.

References

1. 'Automatic Attendance System Using Face Recognition Technique' 2020, *International Journal of Recent Technology and Engineering*, vol. 9, no. 1, pp. 2134–2138.
2. Adouani, A, Ben Henia, WM & Lachiri, Z 2019, 'Comparison of Haar-like, HOG and LBP approaches for face detection in video sequences', *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, viewed 19 June 2021, <https://ieeexplore.ieee.org/document/8893214>.
3. Akmal Hakim 2020, *Universiti Utara Malaysia - TRP*, TRP, viewed 19 June 2021, <https://www.therakyatpost.com/tag/universiti-utara-malaysia/>.
4. Arun Katara, Mr. Sudesh V. Kolhe, Mr. Amar P. Zilpe, Mr. Nikhil D. Bhele, Mr. Chetan J. Bele., 2019, 'ATTENDANCE MONITORING SYSTEM USING FACE DETECTION & FACE RECOGNITION', *International Research Journal of Engineering and Technology (IRJET)*, vol. 06, no. 3, pp. 6629 – 6632.
5. Azlina 2019, *UNIMAS : Penggunaan Teknologi QR Untuk Kehadiran Kakitangan Dan Pelajar Serta Strategi Komersialisasi*, Upm.edu.my, viewed 19 June 2021, <https://idec.upm.edu.my/article/unimas_attendance_for_staff_and_students_using_qr_tech_and_our_strategy_to_commercialize-51327>.
6. Biswas, R. & Uddin, Jia & Hasan, Md Junayed, 2017, A New Approach of Iris Detection and Recognition. International Journal of Electrical and Computer Engineering (IJECE). 7. 2530. 10.11591/ijece.v7i5.pp2530-2536.
7. Chang-yeon, J., 2008, *Face Detection using LBP features*, viewed 3 April 2021, <http://cs229.stanford.edu/proj2008/Jo-FaceDetectionUsingLBPfeatures.pdf>.
8. Chen, J., & Lin, T. 2008. Class attendance and exam performance: A randomized experiment, *Journal of Economic Education.* vol. 39, no. 3, pp. 213-227.
9. Chin, H., 2018. *Face Recognition Based Automated Student Attendance System*, viewed 31 March 2021, <http://eprints.utar.edu.my/2832/1/EE-2018-1303261-1.pdf>.
10. Chong, T., Cheung, K., and Hui, P. 2009, Skipping economics classes: a case study from Hong Kong, *Journal of Higher Education Policy and Management*, vol. 31, no. 3, pp. 37 – 42.
11. Crede M., RochS. and Kieszczynka, 2010, Class Attendance in College: A Meta-Analytic Review of the Relationship of Class Attendance with Grades and Student Characteristics, *Review of Educational Research*. vol. 80, no. 2, pp. 272 – 295.
12. Dinesh, K., 2017, *Facial recognition: Finding faces in a crowd*. Patriot One Technologies Inc., viewed 3 April 2021, <https://patriot1tech.com/magazine/facial-recognition-finding-faces-crowd/>.
13. Dulyawit, 2019, *Face Recognition for Attendance Management System Using Neural Networks Technique*, viewed 28 January 2021, <https://dl-acm-org.tarcez.tarc.edu.my/doi/10.1145/3386164.3389102>.
14. Elankovan, V 2018, *Malaysian University Revealed to Have Epic New Way Of Taking Attendance*, WORLD OF BUZZ, viewed 19 June 2021, <https://worldofbuzz.com/malaysian-university-revealed-epic-new-way-taking-attendance/>.
15. Gonzalez, R. C., & Woods, R. E., 2008, Digital image processing. Upper Saddle River, N.J., Prentice Hall.
16. Goutam Bhadani 2020, *Haar Cascade Classifier vs Histogram of Oriented Gradients(HOG)*, Medium, Medium, viewed 19 June 2021, <https://medium.com/@goutam0157/haar-cascade-classifier-vs-histogram-of-oriented-gradients-hog-6f4373ca239b#:~:text=Note%20that%20HOG%20has%20higher>.
17. Guennouni, S., Ahaitouf, A., & Mansouri, A., 2015, *A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms*. Modelling and Simulation in Engineering, viewed 5 April 2021, <https://www.hindawi.com/journals/mse/2015/948960/#related-work>.
18. HSU, DS 2016, *Fingerprint Sensor Technology And Security Requirements*, Semiconductor Engineering, viewed 19 June 2021, <https://semiengineering.com/fingerprint-senor-technology-and-security-requirements/>.
19. Idris, F, Hassan, Z, Ya'acob, A, Gill, SK & Awal, NAM 2012, 'The Role of Education in Shaping Youth's National Identity', *Procedia - Social and Behavioral Sciences*, vol. 59, pp. 443–450.
20. J. Tang, X. Zhou, J.Zheng, 2019, *Design of Intelligent classroom facial recognition based on Deep Learning*, viewed 28 January 2021, <https://www.researchgate.net/publication/331681396_Design_of_Intelligent_classroom_facial_recognition_based_on_Deep_Learning>.
21. Jadhav, Akshara and Jadhav, Akshay and Ladhe, Tushar and Yeolekar, Krishna 2017, 'AUTOMATED ATTENDANCE SYSTEM USING FACE RECOGNITION', *International Research Journal of Engineering and Technology (IRJET)*, vol. 04, no. 01, pp. 1468 – 1471.

22. Jauro, S & Yadav, R, 2018, 'Review on Iris Recognition Research Directions-A Brief Study', *International Journal of Applied Engineering Research*, vol. 13, pp. 8728–8735.

23. Kanan, C. and Cottrell, G., 2012. Color-to-Grayscale: Does the Method Matter in Image Recognition? *PLoS ONE*, vol. 7, no. 1, p. e29740.

24. Li, Y. (n.d.). *FACE RECOGNITION SYSTEM*, viewed 5 April 2021, <https://arxiv.org/ftp/arxiv/papers/1901/1901.02452.pdf>.

25. Lino, A. F. S., Silva, B. C. R., Rocha, D. P. C., Furriel, G. P., & Calixto, W. P., 2017, *Performance of haar and LBP features in cascade classifiers to whiteflies detection and counting*. IEEE Xplore, viewed 5 April 2021, <https://doi.org/10.1109/CHILECON.2017.8229737>.

26. *Local binary patterns*, 2020, Wikipedia, viewed on 6 April 2021, <https://en.wikipedia.org/wiki/Local_binary_patterns#Implementations>.

27. Maheshkar, V, Agarwal, S, Srivastava, VKr & Maheshkar, S 2012, 'Face Recognition using Geometric Measurements, Directional Edges and Directional Multiresolution Information', *Procedia Technology*, vol. 6, pp. 939–946.

28. Mittal, K., 2020, *A Gentle Introduction into The Histogram of Oriented Gradients*. Medium, viewed 7 April 2021, <https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa>.

29. Mohd Azmi, MS, Mohamed Zabil, MH, Lim, KC, Raja Azman, RF, Noor Adnan, NA & Mohamed Azman, MAS 2018, 'UNITEN Smart Attendance System (UniSas) Using Beacons Sensor', *2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, viewed 19 June 2021, <https://ieeexplore.ieee.org/document/8632631/authors#authors>.

30. Moore, R. 2003, Attendance and Performance: How important is it for students to attend class? *Journal of College Science Teaching*, vol. 32, pp. 367 – 371.

31. Nefian, A. V., & Hayes, M. H., 1998, *Face detection and recognition using hidden Markov models*. IEEE Xplore, viewed 5 April 2021, <https://doi.org/10.1109/ICIP.1998.723445>.

32. Newman-Ford, L, Fitzgibbon, K., Lloyd, S., and Thomas, S. 2008, A large scale investigation into the relationship between attendance and attainment: a study using an innovation, electronic attendance monitoring system, *Studies in Higher Education*, vol. 33, no. 6, pp. 699 – 717.

33. OBE, P. B. B., 2019. *Facial Recognition: You Cannot Hide from The Machine*. Medium, viewed 6 April 2021, <https://billatnapier.medium.com/facial-recognition-you-cannot-hide-from-the-machine-bc148519901b>.

34. Ojala, T., Pietikainen, M., & Maenpaa, T., 2002, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987. <https://doi.org/10.1109/tpami.2002.1017623>.

35. *OpenCV: Cascade Classifier*. (n.d.). Docs.opencv.org., viewed 7 April 2021, <https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html>.

36. *OpenCV: Face Detection using Haar Cascades*., 2017, Opencv.org., viewed 7 April 2021, <https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html>.

37. Paisey, C. and Paisey, N. 2004, Student attendance in an accounting module – reasons for non-attendance and the effect on academic performance at a Scottish University, *Accounting Education*, vol. 13 (Supplement 1), pp. 39- 53.

38. Pratiksha M. Patel, 2016, Contrast Enhancement of Images and videos using Histogram Equalization. *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 4, no. 11.

39. Python, R. (n.d.). *Face Recognition with Python, in Under 25 Lines of Code – Real Python*. Realpython.com, viewed 10 April 2021, <https://realpython.com/face-recognition-with-python/>.

40. Raja, R. & Nagasubramani P.C., 2018, *Impact of Modern Technology in Education*, viewed 28 January 2021, <www.researchgate.net/publication/325086709_Impact_of_modern_technology_in_education>.

41. Rosebrock, A 2018, OpenCV Face Recognition, viewed 5 February 2021, <https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>.

42. Rosebrock, A., 2017, *Histogram of Oriented Gradients and Object Detection - PyImageSearch*. PyImageSearch., viewed 8 April 2021, <https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>.

43. Rosebrock, A., 2018, *Local Binary Patterns with Python & OpenCV - PyImageSearch*. PyImageSearch., viewed 9 April 2021, <https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>.

44. Rungta, K 2020, *Incremental Model in SDLC: Use, Advantage & Disadvantage*, Guru99.com, Guru99, viewed 19 June 2021, <https://www.guru99.com/what-is-incremental-model-in-sdlc-advantages-disadvantages.html>.

45. S.Aanjanadevi, Dr.V.Palanisamy,R.Anandha Jothi., 2017, A Study on Secure Online Voting System using Biometrics Face Detection and Recognition Algorithms. International Journal for Modern Trends in Science and Technology, V3(8).

46. Satya, M., 2016, *Histogram of Oriented Gradients explained using OpenCV*. Learn OpenCV., viewed 1 April 2021, < https://learnopencv.com/histogram-of-oriented-gradients/>.

47. Schalkwyk, S. Menkveld H. and Ruiters, J. 2010, What's the story with class attendance? First-year students: Statistics and Perspectives. South African, *Journal of Higher Education*, vol. 24, no. 4, pp. 630 – 645.

48. Shireesha Chintalapati, M.V. Raghunadh., 2014, "Automated Attendance Management System Based on Face Recognition Algorithms", IEEE International Conference on Computational Intelligence and Computing Research.

49. Smriti Tikoo., 2016, *Detection and recognition of face using neural network*. Slideshare, viewed 5 April 2021, <https://www.slideshare.net/SmritiTikoo/detection-and-recognition-of-face-using-neural-network?next_slideshow=1>.

50. Sobral Andrews, C., 2019, *Lbp-library*, viewed on 2 April 2021, <https://github.com/carolinepacheco/lbp-library>.

51. Subhi Singh., 2015, Automatic Lecture Attendance System Using Face Reorganizationmatrix. Academic International Online Journal of Engineering and Technology, vol. 3, no. 1.

52. Sunkari P, P., 2020, *HOG*, viewed on 1 April 2021, <https://github.com/preethampaul/HOG>.

53. T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, 2002.

54. Tang, J, Zhou, X & Zheng, J 2019, 'Design of Intelligent classroom facial recognition based on Deep Learning', *Journal of Physics: Conference Series*, vol. 1168, p. 022043, viewed 19 June 2021, <https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022043>.

55. V. Nefian and M. H. Hayes, 1998, "Face detection and recognition using hidden Markov models," *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, Chicago, IL, USA, 1998, pp. 141-145 vol.1, doi: 10.1109/ICIP.1998.723445.

56. Virgil Petrescu, R. V., 2019, Face Recognition as a Biometric Application. *Journal of Mechatronics and Robotics*, vol. 3, no. 1, pp. 237–257. <https://doi.org/10.3844/jmrsp.2019.237.257>.

57. Wei-Lun Chao., 2007, Face Recognition, GICE, National Taiwan University.

58. Wikipedia Contributors., 2019, *Histogram of oriented gradients*. Wikipedia; Wikimedia Foundation, viewed 1 April 2021, <https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients>.

59. Wiseman, R, Watt, C, ten Brinke, L, Porter, S, Couper, S-L & Rankin, C, 2012, 'The Eyes Don't Have It: Lie Detection and Neuro-Linguistic Programming', in M Lappe (ed.), *PLoS ONE*, vol. 7, no. 7, p. e40259, viewed 2 December 2021, <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0040259>.

60. Wójcik, W., Gromaszek, K., & Junisbekov, M., 2016, Face Recognition: Issues, Methods and Alternative Applications. *Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods, DOI: 10.5772/62950*. <https://doi.org/10.5772/62950>.

61. Y. Kawaguchi, T. Shoji, W. Lin, K. Kakusho, and M. Minoh, 2009, *Face Recognition-based Lecture Attendance system*, viewed 28 January 2021, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.560.7003&rep=rep1&type=pdf>.

62. Zhao, W., & Chellappa, R. (n.d.). *Image-based Face Recognition: Issues and Methods*, viewed 2 April 2021, <https://www.face-rec.org/interesting-papers/General/Chapter_figure.pdf>.

Appendices

Appendix 1 User Guide

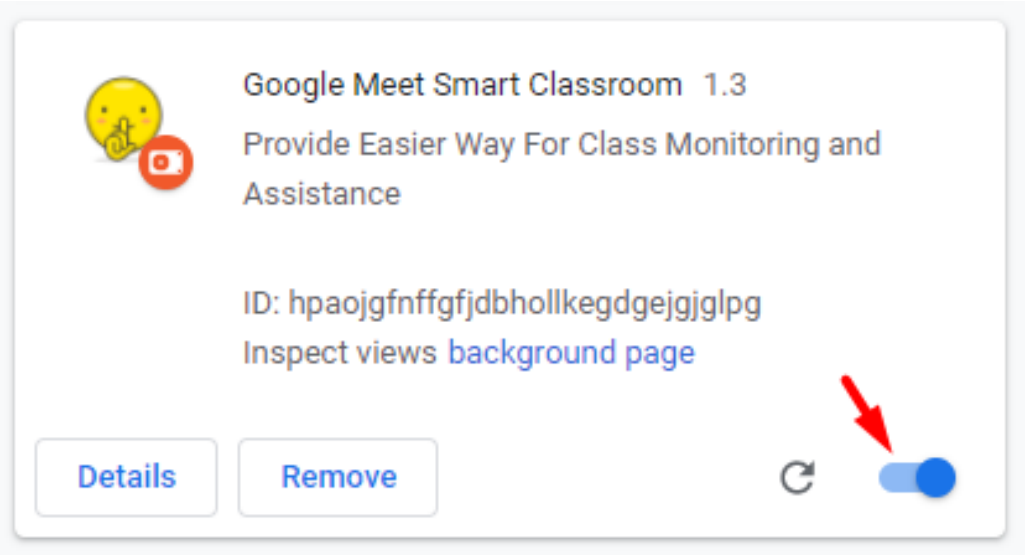
Requirements: Google Chrome Browser, Internet Connection

Appendix 1.1 Load Chrome Extension

1. Visit https://github.com/Joan0018/SmartClassroom to download the system.
2. Login account with test account for Face Recognition and Attendance Module (Google Chrome Browser).
    a. Email: fyp2021may@gmail.com
    b. Password: SmartClassroom2021
3. Visit **chrome://extensions/.**
4. Enable **Developer Mode.**

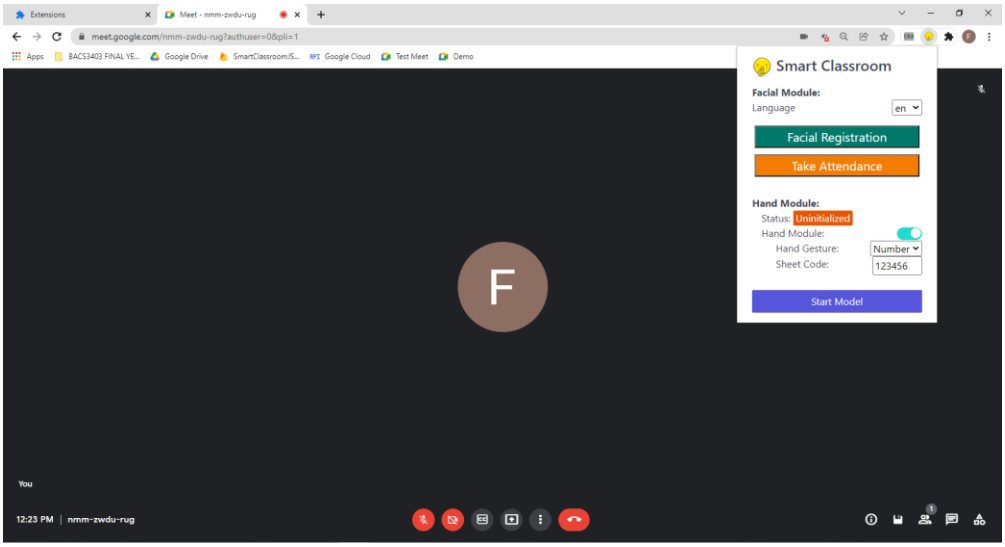
5. Click **Load unpacked** and select the project downloaded.

6. Enable Chrome Extension.

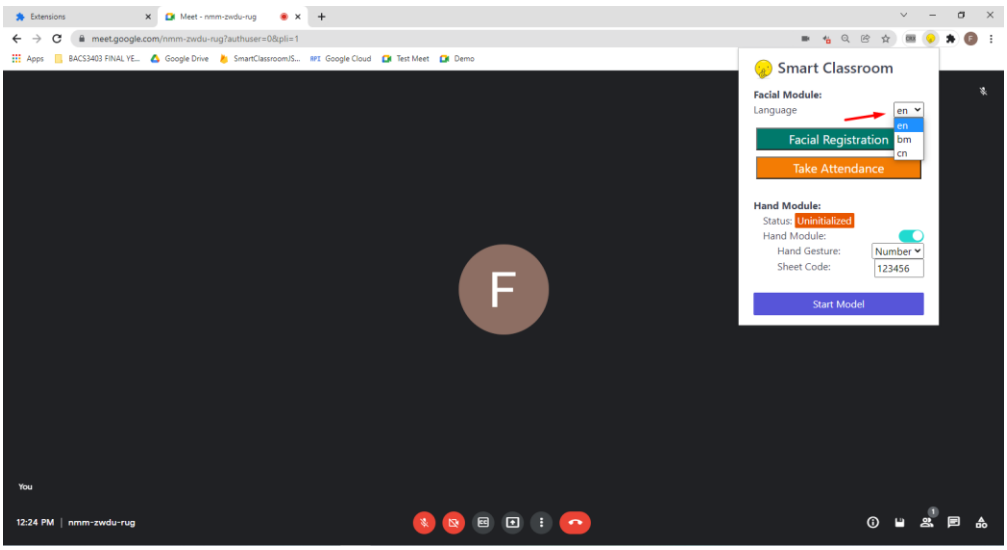


Appendix 1.2 Facial Registration

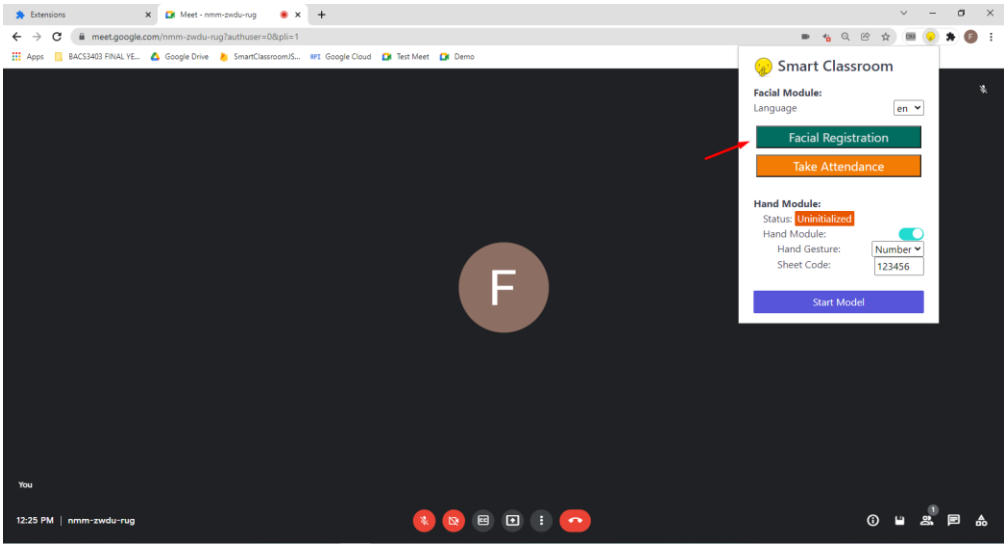
1. Join Google Meet and open the system user interface.



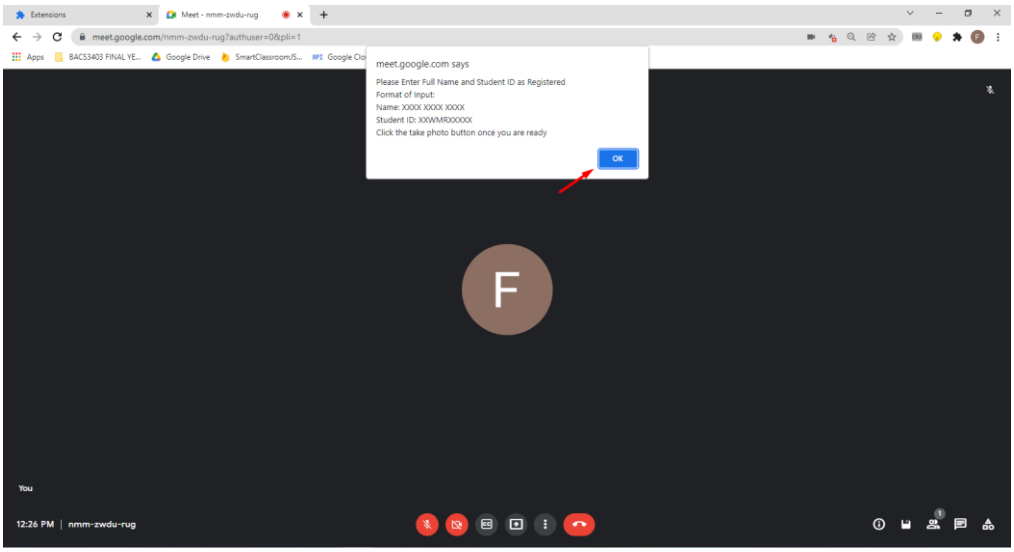
2. Change Language.



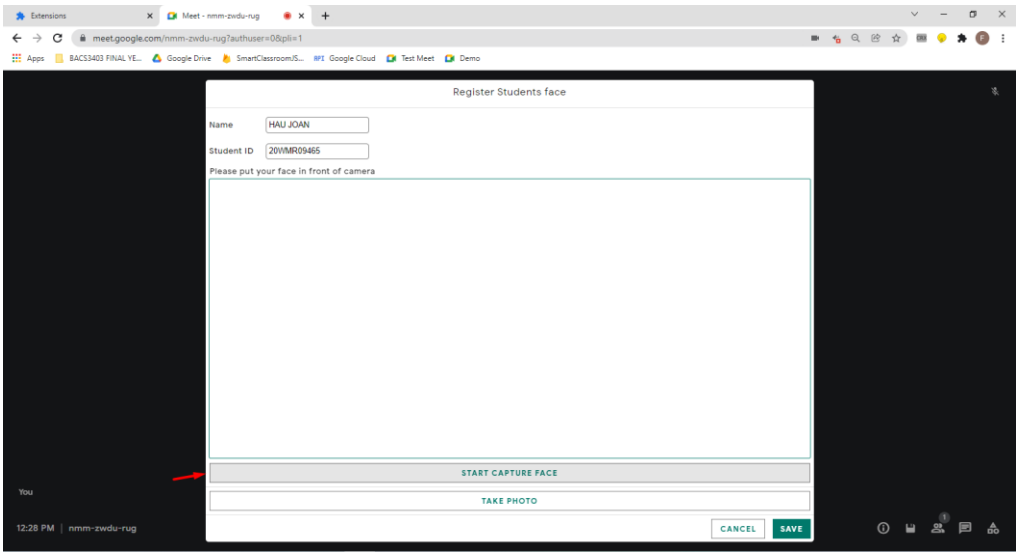
3. Click *Facial Registration* to display the user interface.



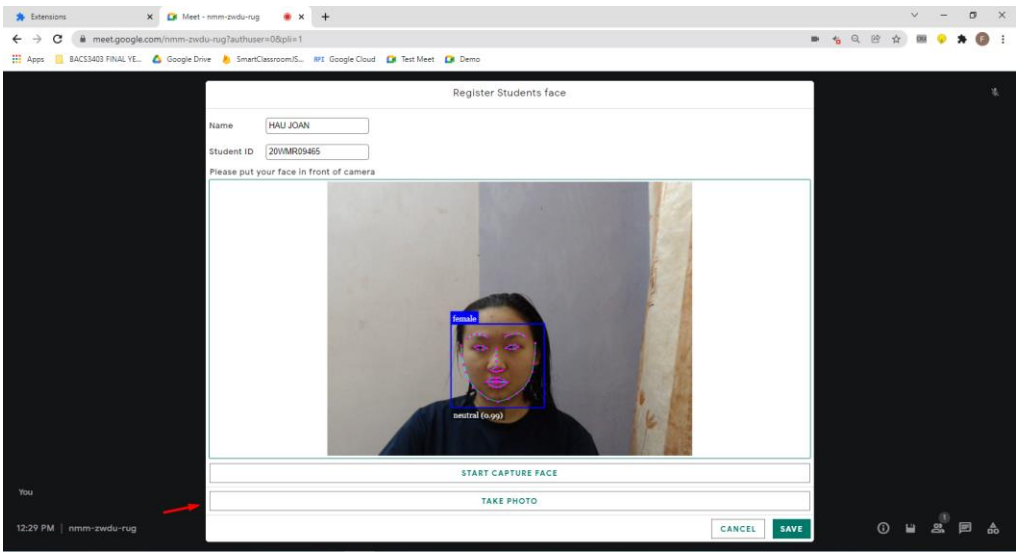
4. Click *OK*.



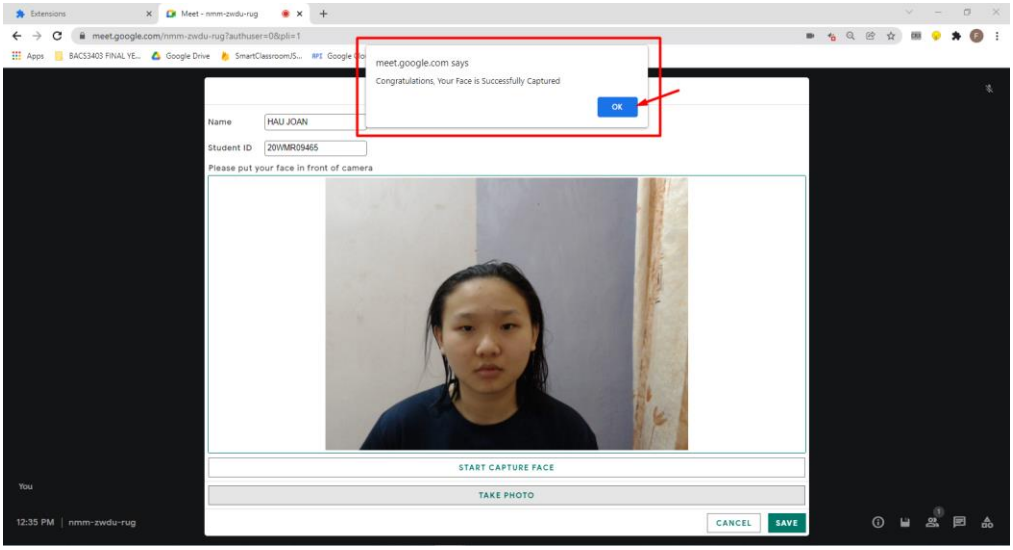
5. Input Name (follow Google Account Name (e.g., HAU JOAN)) and Student ID (e.g., 20WMR09465). Then, click *START CAPTURE FACE*.



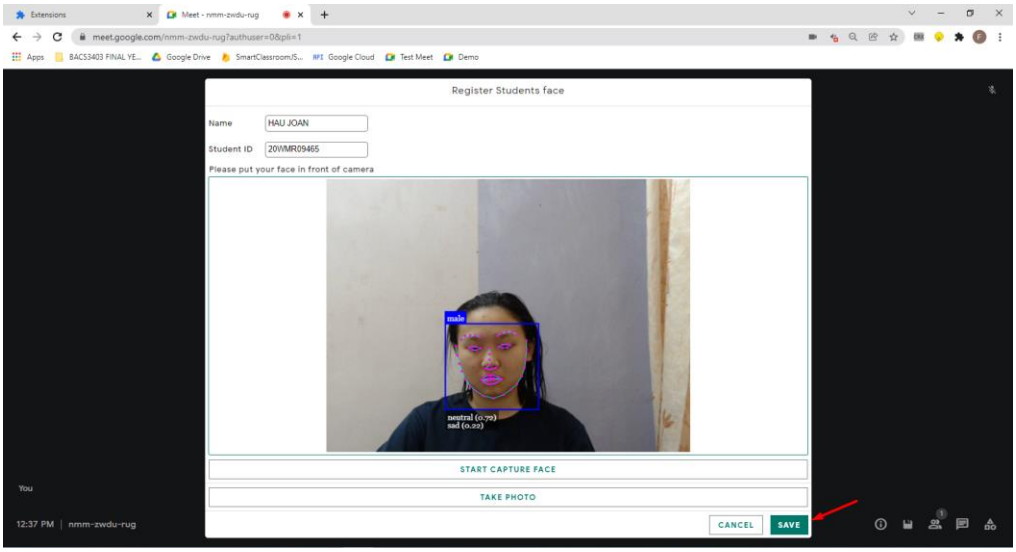
6. Click *TAKE PHOTO* 7 times.



7. A confirmation message will be prompt out and click *OK*.

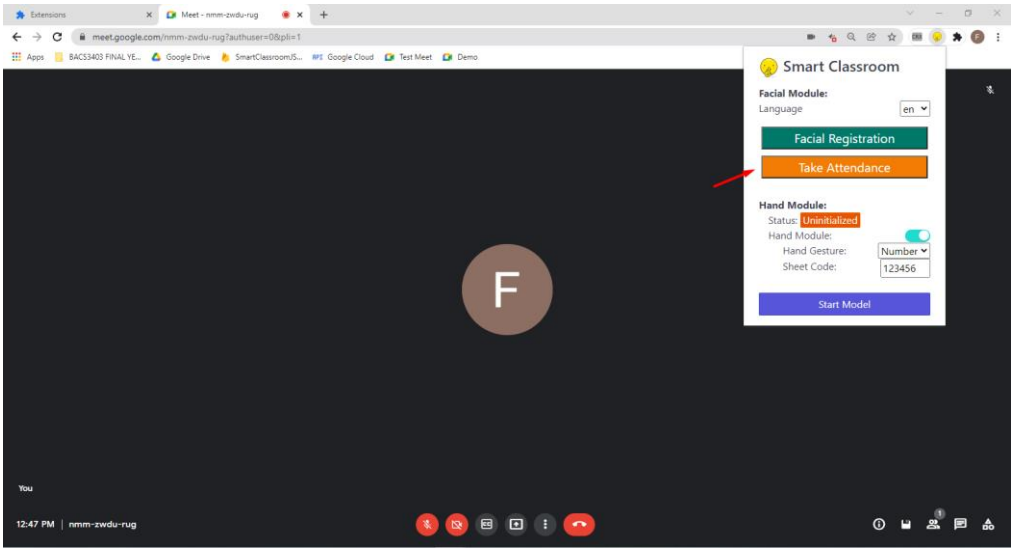


8. Click *SAVE* to save the relevant data to database or *CANCEL* to clear all the data.

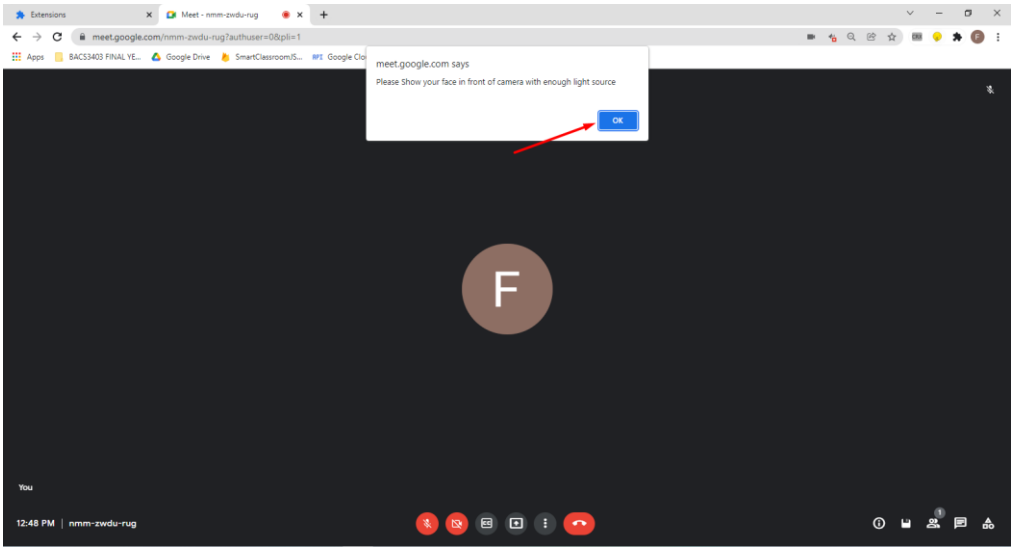


Appendix 1.3 Take Attendance

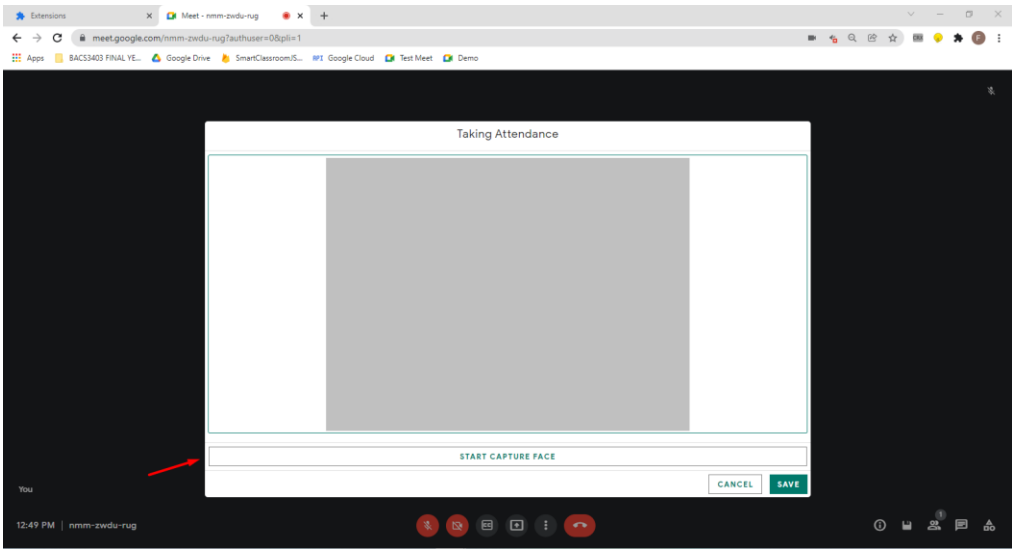
1. Click *TAKE ATTENDANCE*.



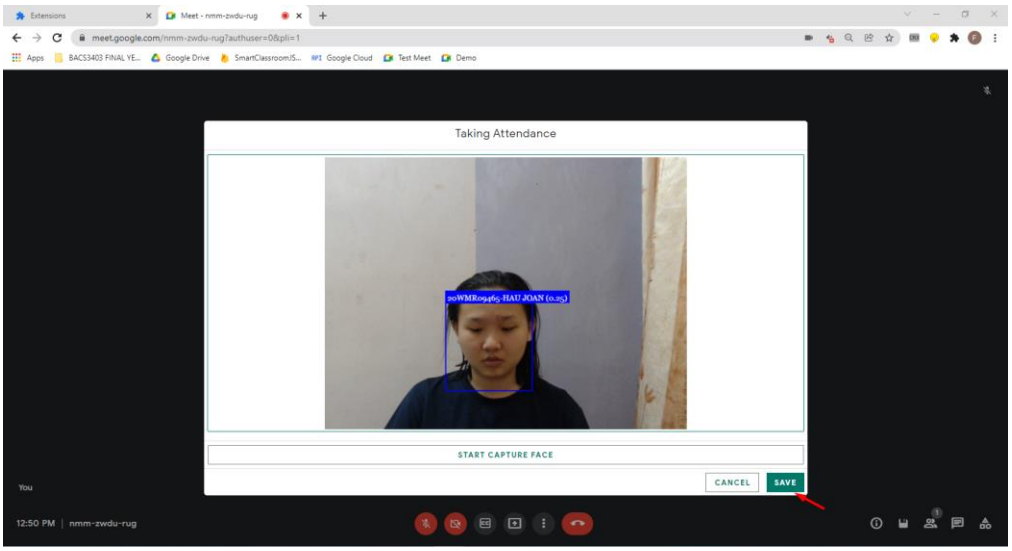
2. Click *OK*.



3. Click *START CAPTURE FACE*.

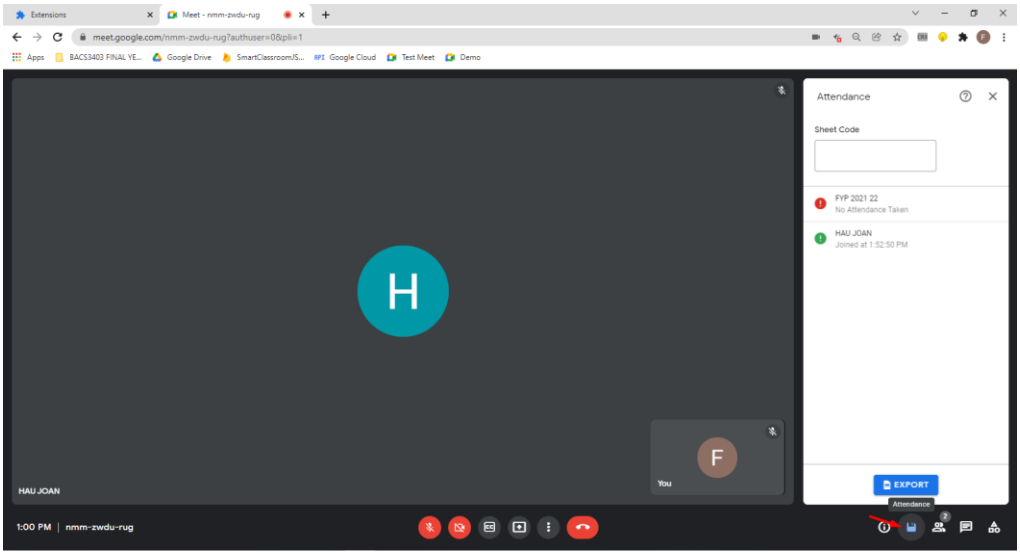


4. Click *SAVE* to take the attendance or *CANCEL* to clear all the data.

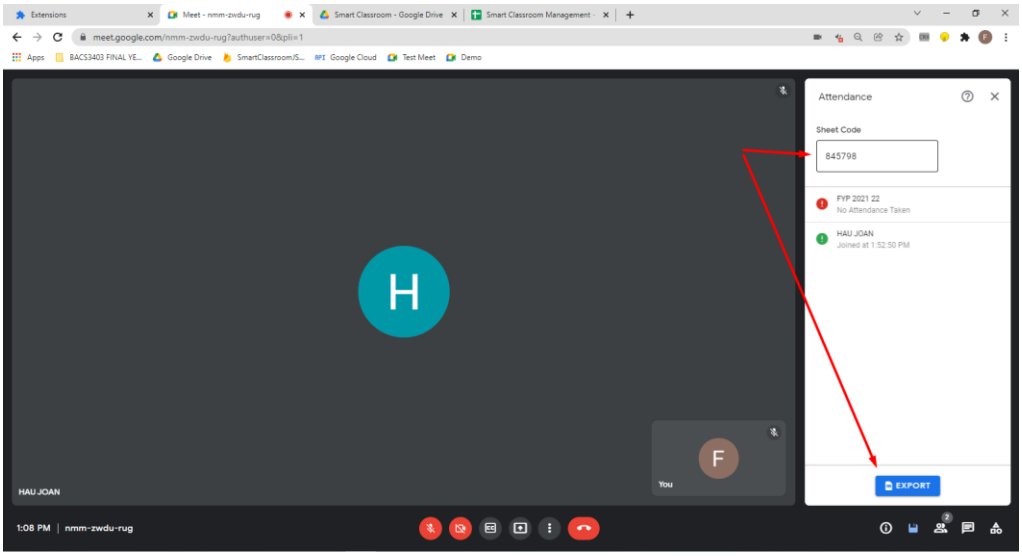


Appendix 1.4 Attendance Management

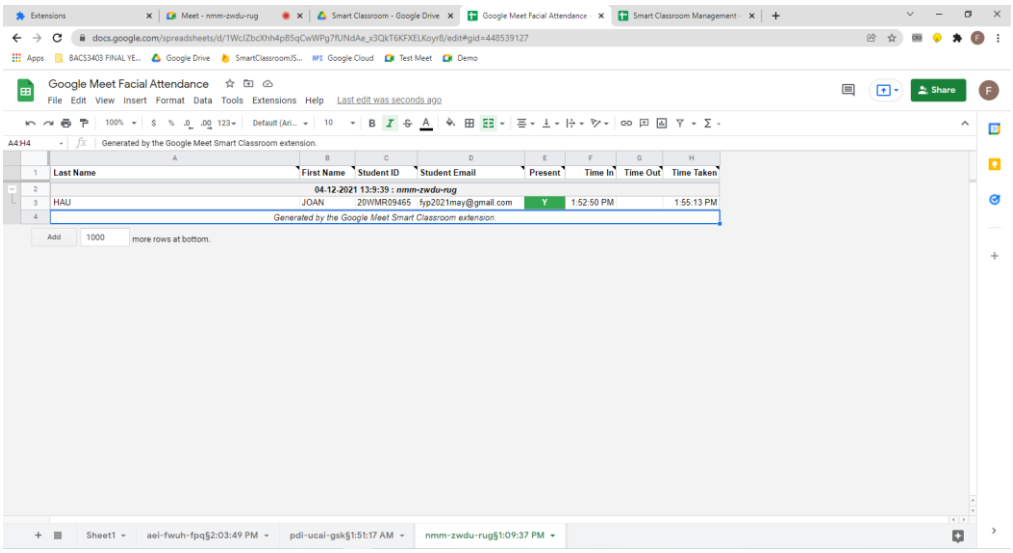
1. Click *Attendance* icon to view the current attendance take by students.



2. Enter *845798* for Sheet Code and click *Export*.



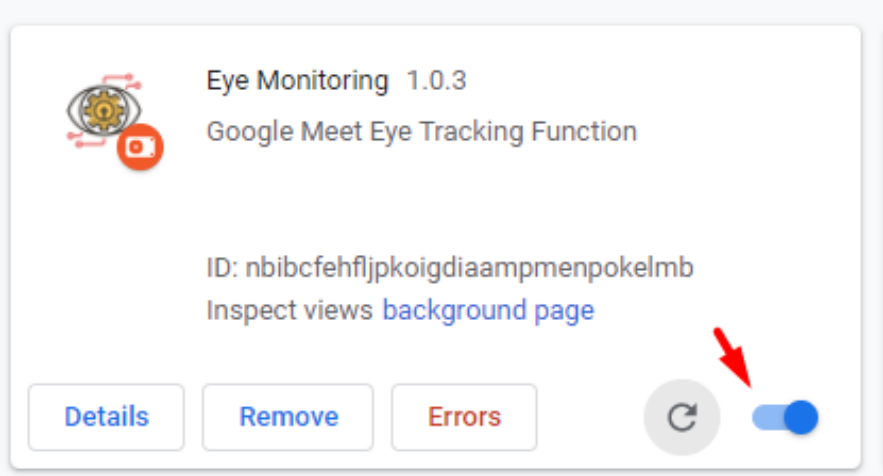
3. Attendance List Export.



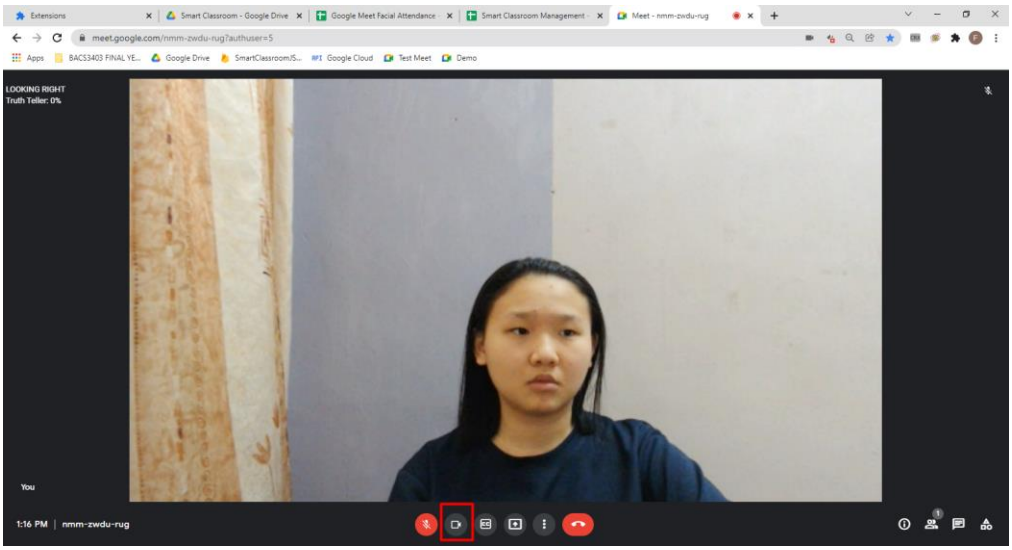
Appendix 1.5 Eye Tracking

1. Visit https://github.com/Joan0018/EyeTrackingExtension to download the system and follow the step in Appendix 1.1.

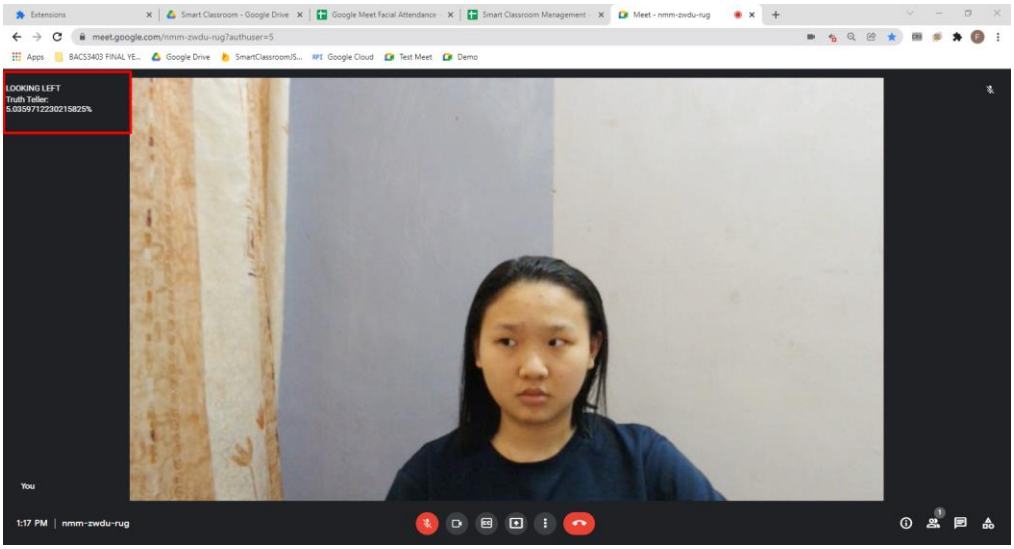
2. Enable Eye Monitoring Chrome Extension.



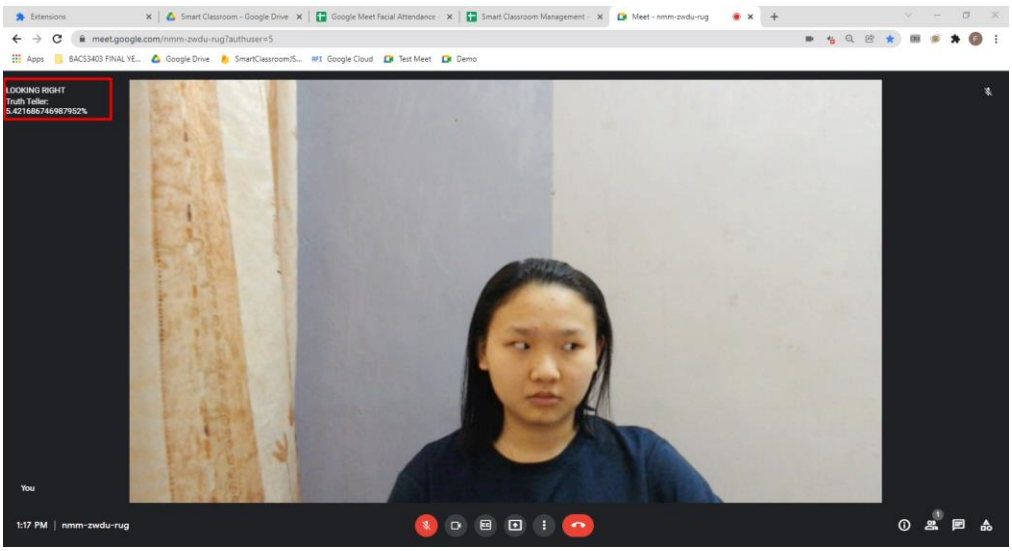
3. Join Google Meet and Open Camera.



4. Result looks at left.



5. Result looks at right.



Appendix 2 Developer Guide

Requirements:

Software

1. Google Chrome Browser.
2. Face-api.js (API).
3. Mediapipe Facial Landmark Detection (API).
4. Visual Studio Code (Coding Tool).
5. Webcam with at least 1080p for facial detection and recognition.
6. Access Firebase Database via https://console.firebase.google.com/u/0/project/smartclassroomjs/overview with google account provided in Appendix 1.1.

This page is intentionally left blank to indicate the back cover. Ensure that the back cover is black in color.