Smart Classroom System – Hand Gesture Recognition and Tracking Module

By

Lim Kah Yee

TARC
TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE
BEYOND EDUCATION

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE
KUALA LUMPUR

ACADEMIC YEAR
2021/22

Smart Classroom System – Hand Gesture Recognition and Tracking Module

By

Lim Kah Yee

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.

Lim Kah Yee

Bachelor of Computer Science (Honours) in Software Engineering

ID: 20WMR09487

Abstract

The advancement of computer technology allows students to interact with Artificial Intelligence (AI) through smart classrooms. Smart classroom is one of the latest technology-enhanced learning (TEL) which allows the classroom and students to interact during the learning process. Currently, smart classrooms are believed to change current dull teaching methods and enhance the students' learning experience. Therefore, the proposed project is an intelligent classroom system which offers real-time monitoring techniques and hand gesture detection and recognition algorithms to monitor student activities and recognize student behaviour (Hand Gesture Recognition and Tracking) through the e-learning classroom. This intelligent classroom system also aims to recognize and learn the student behaviours from time to time.

Acknowledgement

We are very grateful and especially thankful to our supervisor, Ts. Dr Tew Yiqi, for frequently following our final project and spending his valuable time to discuss with us. We are also 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

Interactive learning between teachers and students is one of the main concerns of every educational institution. In today's society, most of the education institutions are still using the traditional teaching methods to impart knowledge to students. For example, Tunku Abdul Rahman University College (TARUC) is still using the traditional teaching methods such as reading through the lecture notes (Lecturer), asking students to answer the question (Tutor), etc.

In this project, we are going to use hand gesture recognition and tracking as a new method for monitoring student's activities and their hand gestures. In addition, the hand gesture recognition and tracking feature will also act as a new communication tool between the teachers and students during the online class. The proposed system will be able to support the online classrooms (e-learning Classroom) to monitor student's activities and their hand gestures. Besides, proper hand gesture recognition and tracking algorithms will be applied into the system in order to ensure the efficiency and accuracy of the proposed system. Last but not least, pose detection algorithms will also be implemented into the proposed system to monitor the presence of students in front of their devices.

1.1 Objectives

The Smart Classroom is designed to monitor the student behaviour when they are in the e-learning Classroom. The objective of this system is to improve the efficiency of lecturer and tutor in monitoring student behaviour during the class. This system can reduce the work of the teacher during the online class as it can detect the student's activities and hand gestures automatically. The system enables teachers to know the activities of students without having to pin students on the screen during online class. For example, determine whether student attending the class.

The second objective of this system is to allow e-learning classrooms to interact with students' hand gestures and movement. In addition to learning progress, the interaction between students and teachers is also an important issue, especially in online classrooms, because some students may just enter the online class and then do other things. This system will capture what is done by the students and display to the teacher. Other than that, this system also can assist students during the physical and online class. The system will use the student's pc webcam to capture the hand gesture and movement of the student and response immediately. For example, when the student intends to ask questions, the system will capture the raised hand of the student and respond to it.

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 ensure a person's palm is detected for the hand gesture recognition and tracking?
2. How to increase the accuracy of hand gesture recognition and tracking?
3. How to increase the speed of hand gesture recognition and tracking in real time?

1.2.2 Hypothesis

The hypothesis made based on the question stated above:

1. The better the hand landmark detection model, the higher the credibility of the detected hands.
2. The more gesture's hand landmarks coordinates been trained, the higher the accuracy for hand gesture recognition and tracking.
3. The higher the efficiency of a gesture algorithm used, the higher the speed of hand gesture recognition and tracking.

1.3 Background

1.3.1 Target Market

The system targets education institutions. According to Potnis and Jahagirdar (2014), capturing a student's hand gesture is particularly important, especially in a smart classroom environment. From the research of Chen and Fang (2014), hand gestures are able to enhance the memory of learners and also be able to understand the subject easily. In the study, researchers have used Microsoft's motion-sensing sensor, Kinect, to conduct an experiment. In the experiment, the researchers found that the body-based method was more successful than the mouse-based method (traditional input method) in helping learners keep what they learned. Therefore, the emergence of this system will benefit education institutions, especially when it provides more interactive interactions between teachers and students and replaces the dull traditional teaching methods in e-learning classrooms.

In addition to education institutions, this system can also be applied to companies that may need to have meetings or events with many people online. With the outbreak of the Covid-19 pandemic, online virtual seminars may become more and more popular in the future. The system will be able to benefit the organizers of virtual seminars because they can monitor the activities of participants, such as monitoring which participants have questions.

1.3.2 Existing Similar System

1. Huawei’s AI Gesture Control

   Huawei has introduced AI hand gesture control technology in its P and Mate series of smartphones, which allows users to use gestures to perform some operation on the phone by using hand gestures. For example, users can scroll the article up and down, grabshot the screen, navigate to another screen, etc.

Figure 1.1: Huawei P40 Pro’s AI Hand Gesture Control to take a grabshot. (Consumer.huawei.com, 2021)

2. BMW Gesture Control

   BMW has also implemented hand gesture recognition and tracking in some of its series of cars. The BMW Gesture Control allows the driver to use hand gestures to perform some of the operation on its car. For example, the BMW Gesture Control allows the driver to change music volume, pick up a call, end a call, etc.

Figure 1.2: BMW Gesture Control that allow user swipe hand to right to end a call (BMW, 2021)

3. Samsung Smart TV

Samsung Smart TV has also implemented hand gesture control. The Samsung Smart TV allows the user to control the TV's mouse to perform some actions. For example, it allows the user to control the TV's mouse to adjust the volume of the TV.

Figure 1.3: Samsung Smart TV that allows the user to control TV's mouse top perform actions (Samsung ph, 2021)

4. Kinect Drawing

Kinect has also implemented hand gesture control. The Kinect allows the user to use their hand to draw something on the screen. For example, users can use Kinect to graffiti screens.

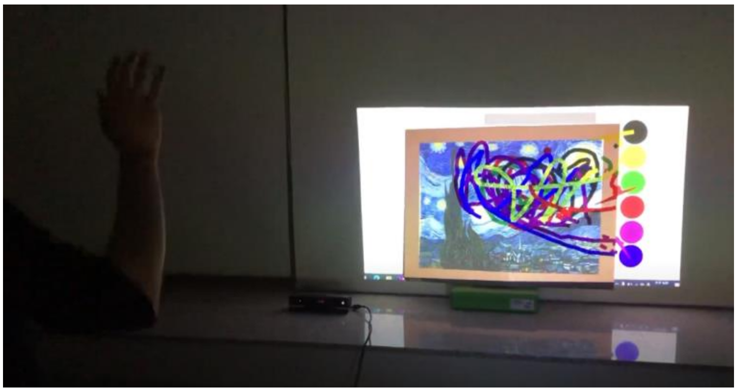


Figure 1.4: Kinect that allow user to graffiti the screen (WER On Wave, 2020)

1.3.3 Potential Work

The proposed system will be using hand gesture recognition and tracking technology which also includes the Internet of Things (IoT) to capture the webcam of students and carry out monitoring at the same time. This project focuses on capturing the video feed from the students' webcam, and then performing image processing on the video frames to identify the hand gestures. If a hand gesture has been detected and recognized, the system will respond to the student's hand gesture.

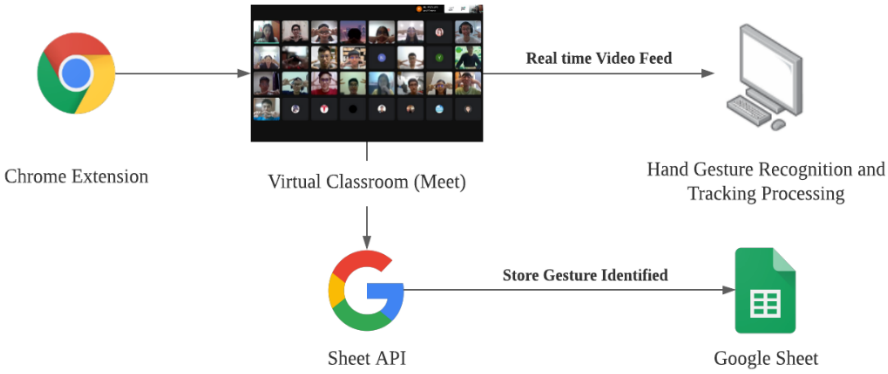


Figure 1.5: Project Structure of Hand Gesture Recognition and Tracking Module

1.4 Advantages and Contributions

The smart classroom system with hand gesture recognition and tracking in monitoring the student's activities is more convenient and efficient than other monitoring methods especially in the class that has a lot of students. For example, in the physical classroom, teachers need to look at students frequently to monitor students' activities or whether any student has question(s). In addition, in the e-learning classroom, there has a lot of student will join the online class, this also cause inefficient to the teachers to monitor students' activities as may has more than 100 students joined. With this system, the inconvenience and inefficient issues mentioned above can be avoided. This is because the system will implement hand gesture recognition and tracking that can monitor the activities of students through the student's hand gesture and movement. Other than that, the hand gesture recognition and tracking module will allow students to use hand gestures to perform actions. For example, when a student raises his hand to ask a question(s) to the webcam, the system will notify the teacher by calling Google Meet's raise hand function.

In addition, this system is able to replace the dull traditional teaching methods. This is because the system has a function of detecting a student's hand gestures. This feature allows teachers to conduct lessons with students in a more interesting way. For example, the tutor can conduct a quiz with students, then students can use their hand gestures that represent the answer to answer the quiz. After that, the smart classroom system will automatically identify the user's hand gesture and send it to Google Meet Chatbox. In addition, the smart classroom system with hand gesture detection and recognition can improve the tutors' productivity and replace dull traditional teaching methods.

1.5 Project Plan

1.5.1 Project Specification

The hand gesture recognition and tracking module will cover a few functional and non-functional specifications as below:

a) Functional Specification

   i) The system should be able to capture the students' hand gesture and movement through a web camera.

   ii) The system should be able to recognize the student's hand gesture.

   iii) The system should be able to display the detected hand gesture text when the hand gesture of the student is being recognized.

   iv) The system should be able to call Google Meet's functions when then hand gesture is detected.

   v) The system should be able to call Chatbot to send gesture message in Google Meet's Chatbox.

   vi) The system should be able to store the hand gesture detected into Google Sheet.

b) Non-functional Specification

   i) The system should be able to recognize the student's hand gesture in a short period of time (within 1 – 2 seconds).

   ii) The system should be able to display the detected hand gesture text in the correct position and readable.

1.5.2 Milestones

Table 1.1: Project Milestones

Task	Remarks	Deadline
Project Proposal	Project Proposal Submission	8 March 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 analyze 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 1.2: 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			User Interface	5 days	Fri 25-06-21	Thu 01-07-21
51			Chapter Summary and Evaluation	1 day	Fri 02-07-21	Fri 02-07-21
52			Second Stage Implementation	36 days	Mon 05-07-21	Mon 23-08-21
53			**Integration and Testing**	**44 days**	**Tue 24-08-21**	**Fri 22-10-21**
54			Test Plan	10 days	Tue 24-08-21	Mon 06-09-21
55			Test Data	10 days	Tue 07-09-21	Mon 20-09-21
56			Test Cases	10 days	Tue 21-09-21	Mon 04-10-21
57			Fix Defects Found	14 days	Tue 05-10-21	Fri 22-10-21
58			Final Stage Program Compilation	14 days	Mon 25-10-21	Thu 11-11-21
59			**Deployment**	**1 day**	**Fri 12-11-21**	**Fri 12-11-21**
60			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 will be the incremental model approach. There are various phases of the life cycle of incremental models, which are the requirement analysis, design, coding, testing, and lastly the implementation phase. Each iteration in the incremental model will go through these phases as shown in Figure 1.7. The hand gesture recognition module will be separated into a few functions at different iterations in order to ensure the quality of the module guaranteed. The hand gesture recognition will be separated into a few incremental:

Incremental 1: Implement Pose and Hand Landmarks Detection

At the first increment of this module, the algorithm to detect the pose and hand landmarks (MediaPipe Pose & Hands) will be applied into the module in order to enhance the accuracy of the web camera to detect the presence and hand of the user. The detected hand landmark will be displayed with 21 points and these points will be joined to form the hand landmarks as shown in Figure 1.6. In addition, if the user not presence in front their device, the system will display popup message to inform user to attend the class. If the user not presence for 10 minutes, the system will use Google Meet's Chatbox to inform the organizer. Once the implementation of this function has completed, it will proceed to the testing phase to perform the testing. If any defects are found in the function, then the defects will be fixed before proceeding to the next incremental. Will proceed to the next incremental once incremental 1 completed.

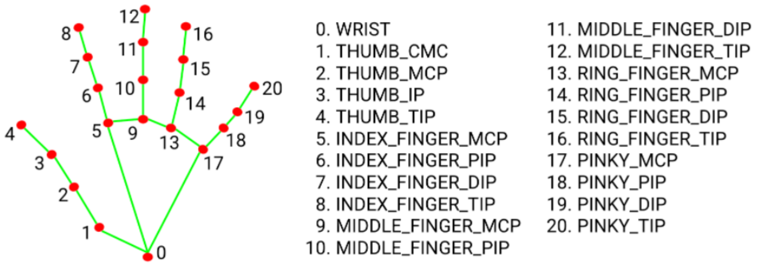


Figure 1.6: MediaPipe Hand Landmarks (MediaPipe, 2020)

Incremental 2: Hand Gesture Recognition using Webcam

At the second increment of this module, the algorithm and trained hand gestures model (i.e. Fingerpose) will be applied into the module in order to ensure the module is able to respond to the hand gestures of the user. For example, the users would like to choose 1 – 4, then the user can just show its index finger to indicate choice 1. Once the implementation phase of this function has completed, it will proceed to the testing phase to perform the testing. If any defects are found in the function, then the defects will be fixed before proceeding to the next incremental. Will proceed to the next incremental once incremental 2 completed.

Incremental 3: Perform action based on Hand Gesture Recognized

At the third increment of this module, a hand gesture function will be added into the system. For example, when the system detects a user shown "Help" hand gesture, the system will use JavaScript code to perform a click action on the Google Meet's Raise Hand function. The system will go through the testing process to ensure it can recognize hand gestures that are trained.

Incremental 4: Store Hand Gesture in Google Sheet

At the fourth increment of this module, Google Sheet API will be added into the module in order to allow the module to store the hand gesture data in Google Sheets through Google Sheet API. The system will go through the testing process to ensure it store recognized hand gestures detected into Google Sheet.

Incremental 5: Hand Gesture Recognition through Webcam

At the fifth increment of this module, more test data will be added into the system and the hand gesture recognition will be applied in order to allow the system to recognize the more hand gestures of the user. The system will go through the testing process to ensure it can recognize hand gestures that are trained.

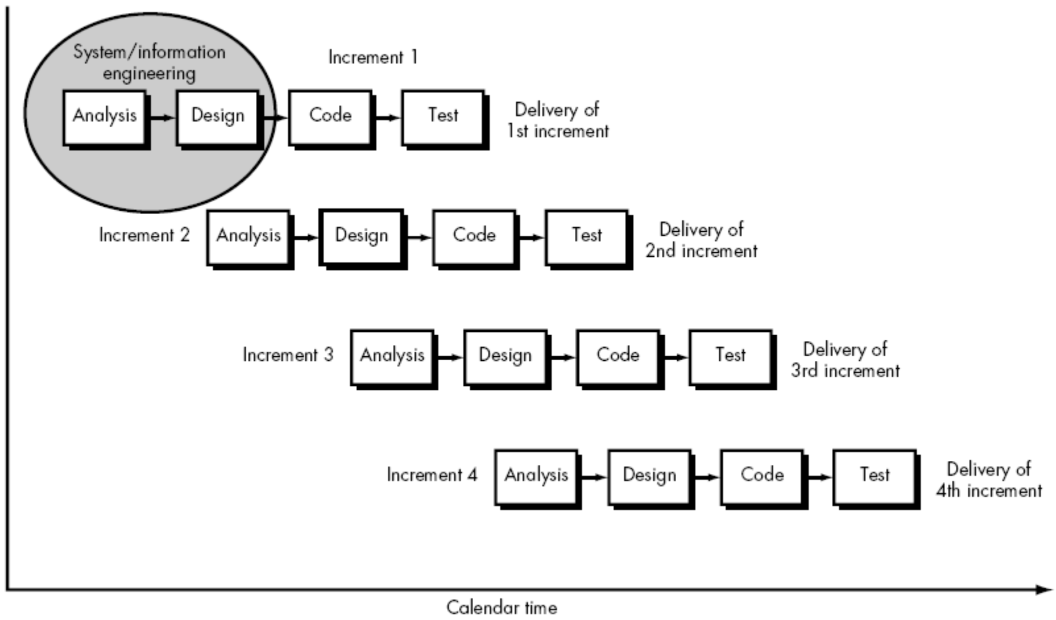


Figure 1.7: Incremental Model Life Cycle (Olusola Olajide, A., 2016)

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 into the system meet the expectations. The components such as hand gesture detection, hand gesture recognition, etc will be tested individually during this testing. For instance, the hand gesture recognition will be tested to ensure it is able to recognize the hand gesture that is being trained.

2. Module Testing

    Module testing is used to test if all the functions in the module are well performed after combining. The combinations of each of the functions within the same module which is Hand Gesture Recognition and Tracking Module will be tested. For example, the hand gesture detection function and hand gesture recognition function will be tested together to ensure the system can detect and recognize the hand gesture of the user.

3. Integration Testing

    After all the modules within the system are integrated, integration testing will be carried out to ensure the performance and functionality of each module is able to perform well and cooperate with each other. For example, integration testing will be carried out after the Face Recognition Attendance Module and Hand Gesture Recognition and Tracking Module have been integrated.

4. System Testing

    Once the integration testing is completed, the system testing will be carried out in order to ensure the system is able to meet the expected functional and non-functional requirements.

5. Acceptance Testing

    Before the system has been released, acceptance testing will be performed by any interested users who will use this system. This testing is to test whether the system is satisfied and is accepted by the users.

1.6 Project Team & Organization

Table 1.3 Project Team with Module Handled

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

1.7 Chapter Summary and Evaluation

In this chapter, it introduces the project to be implemented, including the system's objectives, background, advantages as long as the contributions, and the project plan of developing the Hand Gesture Recognition and Tracking Module of the Smart Classroom System. The main objective of this project is to allow the e-learning classroom to interact with students' hand gestures. For the project background, it will include the target market, existing similar system, and potential work of the system. Moreover, this chapter also included the advantages and contributions of the system in monitoring students' hand gestures and movements. Lastly, Gantt Chart and the milestone of this project will be also included.

Chapter 2

Literature Review

2 Literature Review

In Chapter 2, it will discuss the company and project background, which will include the business nature, current system used, potential market and target users. Moreover, it will also include literature review which discuss supporting details to propose ideas of the system (Hand Gesture Recognition Module integration in Smart Classroom System). In this research, discussed topics and resources found are documented and used as a guidance and support

2.1 Company Background

2.1.1 Nature of Business

Tunku Abdul Rahman University College ("TAR UC") is Malaysia's leading higher education institution established by Malaysia Chinese Association (MCA) in 1969. Currently, TAR UC has a main campus in Setapak, Kuala Lumpur and 5 branch campuses in Malaysia. TAR UC was also established to meet Malaysia's current and future manpower needs (TARUC, 2021).

2.1.2 Services

TAR UC offers more than 120 programmes at Pre-University, Diploma, Bachelor's Degree and Postgraduate Levels, ranging from foundation and A Level to accountancy, finance, business, economics, engineering, applied sciences, hospitality management, etc. Currently, approximately 28,000 students have enrolled in TAR UC (TARUC, 2021).

2.2 Project Background

This project is to propose a new system for hand gesture recognition and tracking module in the smart classroom system to assist students during their learning process and also replace dull teaching methods. This proposed system is used to improve the lecturers' and tutors' productivity in monitoring students' activities especially during the online class. Besides, this proposed system also intends to add on new communication tools between lecturers, tutors and students during the online class and replace dull teaching methods. For example, if lecturers and tutors want to conduct a quiz with students, then students can use hand gestures to answer the quiz. The main objective of this proposed module is to improve the efficiency of the lecturer and tutor in monitoring student's behaviour during the class and implement hand gesture recognition and tracking features into the e-learning classroom to improve e-learning classroom.

In the market, there are some existing systems that have similar features with proposed modules such as BMW Gesture Control that allow users to control BMW car's monitor with hand gestures, Samsung Smart TV that allow users to control TV's mouse, etc. The target market that is suitable for the proposed module is for educational institutions and companies that will conduct classes, events, conferences, etc online. The Hand Gesture Recognition and Tracking Module can bring a lot of benefits such as monitor users' activities in a more efficient way, act as a new communication way, replace dull traditional teaching methods, etc.

2.3 Literature Review

2.3.1 Hand Gesture Recognition and Tracking using Handsfree

Handsfree is a JavaScript client-side library that was released by Oz Ramos in 2018 (Oz Ramos, 2021). In the Handsfree, it provided a lot of models such as Hands (2D), Handpose (3D), FaceMesh, etc. In this project, the Hands Model and Pose Model is taken for the proposed module. In the Hands Model, it is using the MediaPipe for detecting hand landmarks and Fingerpose for recognizing hand gestures. In the Pose Model, it is using the MediaPipe for detecting pose landmarks of the user. By implementing Handsfree in the Hand Gesture Recognition and Tracking Module, the proposed module is able to use MediaPipe and Fingerpose in the Chrome Extension.

2.3.2 Hand Landmarks Detection using MediaPipe

MediaPipe is a framework for building multimodal, cross platform applied Machine Learning (ML) pipelines, which was released by Google in June 2019. (Google Open Source, 2021; MediaPipe, 2019). One of its ML solutions, which is Multi-Hand Tracking, is used for this project in order to identify users' hand landmarks. In the MediaPipe, it will use the Palm Detection Model that implements a single shot detector to detect hands in the image. After the palm is detected, the Hand Landmark Model will accurately locate the key points of the 21 3D hand joint coordinates in the detected hand area (Figure 2.1). By implementing MediaPipe in the Hand Gesture Recognition and Tracking Module, the proposed module is able to detect the hand landmarks of the user.

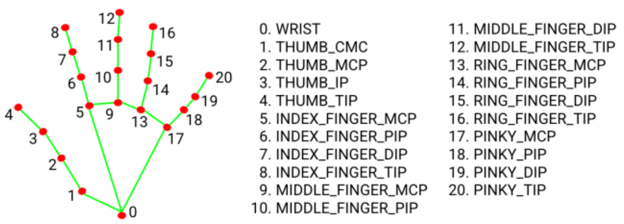


Figure 2.1: MediaPipe Hand Landmark Model (MediaPipe, 2020)

2.3.3 Hand Gesture Recognition using Fingerpose

In this project, Fingerpose that developed by Andreas Schallwig in 2020 will be used for hand gesture recognition for Hand Gesture Recognition and Tracking Module. In Fingerpose, it is estimating the direction and curl of each individual finger, and then compare the result to a set of gesture descriptions in order to recognize hand gesture. For each individual finger, it will categorize it into 3 categories which is no curl, half curl and full curl (Figure 2.2). In addition, Fingerpose also provided 8 direction for each of the finger (Figure 2.3). After Fingerpose finish setup, it will be added into Handsfree in order to recognize hand gesture.

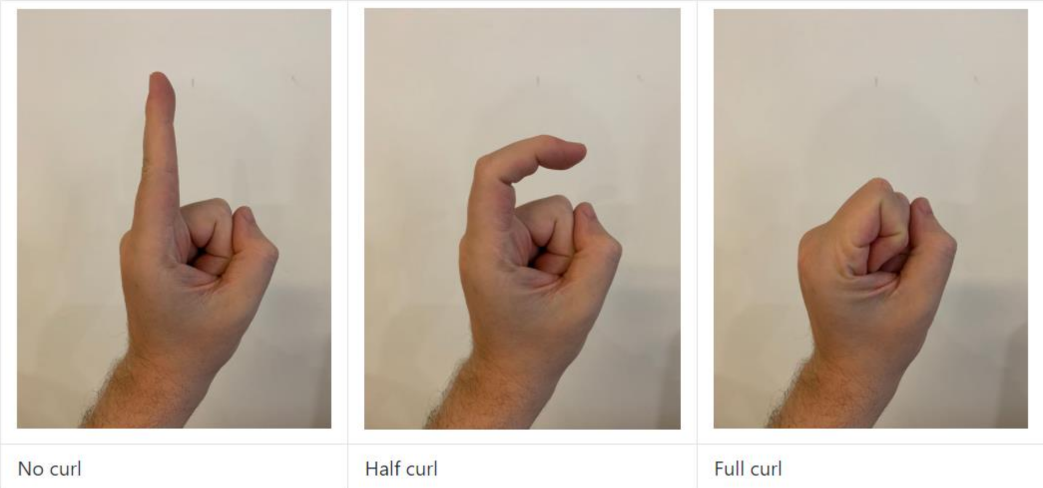


Figure 2.2: Fingerpose Curl (Andreas Schallwig, 2020)

Direction	Name
0	Vertical Up 👆
1	Vertical Down 👇
2	Horizontal Left 👈
3	Horizontal Right 👉
4	Diagonal Up Right ↗️
5	Diagonal Up Left ↖️
6	Diagonal Down Right ↘️
7	Diagonal Down Left ↙️

Figure 2.3: Fingerpose Direction (Andreas Schallwig, 2020)

2.3.4 Data Storing using Google Sheet

In this project, Google Sheet will be used as a database to store the hand gesture detected by Hand Gesture Recognition and Tracking Module. In order to establish connection with Google Sheet, Google's client library for browser-side JavaScript (GAPI) is used. Once hand gesture detected, the module will call GAPI to append data into the Google Sheet. In the Google Sheet, it will contain data as shown in Figure 2.4. By using Google Sheet and GAPI in the Hand Gesture Recognition and Tracking Module, the proposed module is able to store the hand gesture detected in Google Sheet.

Username	Email	Hand Gesture	Date	System Start Time	Gesture Time	Time Interval
FYP 2021 22	fyp2021may@gmail.com	Five	Sat, 30-10-2021	8:25:56 AM	8:26:12 AM	0:00:16
FYP 2021 22	fyp2021may@gmail.com	Five	Sat, 30-10-2021	8:25:56 AM	8:26:34 AM	0:00:38
FYP 2021 22	fyp2021may@gmail.com	Three	Sat, 30-10-2021	8:25:56 AM	8:27:07 AM	0:01:11
FYP 2021 22	fyp2021may@gmail.com	Nice,I'm_Good	Sat, 30-10-2021	8:49:02 AM	8:50:11 AM	0:01:09

Figure 2.4: Sample Data Stored in Google Sheet

2.4 Feasibility Study

2.4.1 Operational Feasibility

The proposed Hand Gesture Recognition and Tracking Module will be able to monitor student's activities and their hand gestures. This is used to improve tutors' productivity in monitoring students' activities especially during the online class that may contain more than hundreds of students. In addition, the proposed module is able to act as a new communication tool between tutors and students during the online class and is able to replace the dull teaching methods.

2.4.2 Technical Feasibility

Webcam or embedded system with camera is required in order to capture the situation in the user's environment as both of them are capable of capturing the video feed and stream it for post video processing. Besides, the software resources such as Mediapipe and Tensorflow needed to perform the video processing are also readily available on the market with continuous support provided by respective organizations. In addition, a wireless network is needed for all the devices connected to the network especially for the online class. As all the hardware and software is available in the market, the project is technically feasible to be carried out.

2.4.3 Financial Feasibility

There is no software cost in this project as all the software libraries that are used in this project are all open sources and are free to use as long as not for commercial use. In addition, there are hardware costs needed for the project (Table 2.1).

Table 2.1: Hardware Price List

Hardware		Approximate Price (RM)
Motherboard	LGA 1151 Socket Motherboard	265.00
CPU	Intel Core i5-8 Gen	960.00
Memory	8GB RAM	215.00
Storage	128GB Solid State Drive (SSD)	120.00
Computer Case	Depends on Motherboard Size	125.00
WIFI Card	IEEE 802.11ac	150.00
Power Supply	400W Power Supply	148.00
Monitor	Full HD Monitor	160.00
Mice & Keyboard	Wireless	109.00
	Total	**2252.00**

2.5 Chapter Summary and Evaluation

In this chapter, the company background and project background where the problem occurred, and the proposed module to solve it is discussed. In addition, the literature review discusses an overview of the methods and algorithms used in the project. At the end, the feasibility study such as operational, technical, and financial is discussed.

Chapter 3

Methodology and Requirements Analysis

3 Methodology and Requirements Analysis

In Chapter 3, all the methodology used and requirements analysis done in this project will be listed. Then, it discusses what are the fact gathering techniques and fact recording methods that are carried out to analyze the system requirements. After that, the list of requirements in the perspective of hardware, software, services, programming language will be discussed as well as the functional requirements and non-function requirements.

3.1 Methodology

The most suitable methodology to use in this project is the incremental model (Figure 3.1). In the incremental model, there are various stages such as requirements analysis, design, coding, testing, and finally the implemental stage. Each iteration in the incremental model goes through these phases, as shown in Figure 3.1. The first phase of the incremental model is the requirements analysis phase. At this phase, before we move on to the next phase, all functional and non-functional requirements will be analysed. The second phase of the incremental model is the design phase. In this phase, the design of the user interface and interaction of the system will be discussed first before proceeding to the next phase. The third phase of the incremental model is coding (a.k.a Development phase). At this phase, the design and system functions collected from the second phase will be converted into observable and runnable code for testing. The fourth phase of the incremental model is the testing phase. At this phase, the runnable code will go through some tests, such as unit tests, module tests, etc., before it proceeds to the implementation phase and is delivered to users. The fifth phase of the incremental model is the implementation phase. At this phase, after the test passes, the runnable code will be implemented into the system and delivered to the user. In addition, these five phases will be carried out for each iteration.

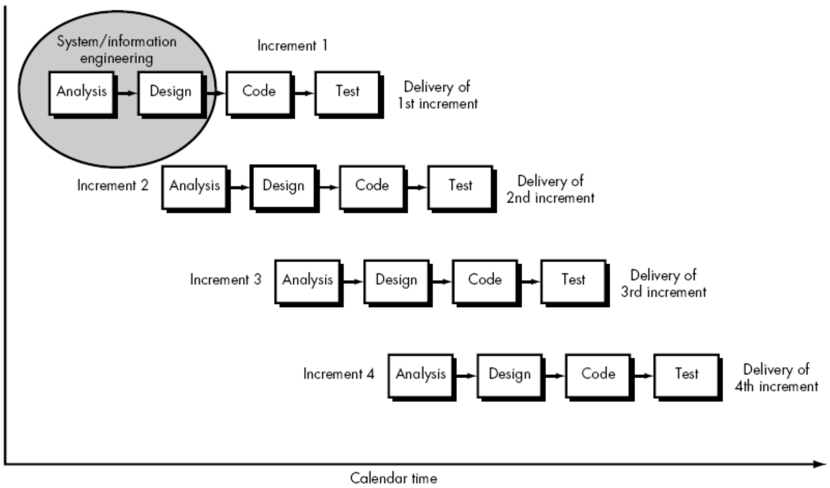


Figure 3.1: Incremental Model Life Cycle (Olusola Olajide, A., 2016)

3.1.1 Fact Gathering

In the process of system development, all necessary information must be collected before the development team can identify the functional and non-functional requirements of the system. In this case, fact gathering techniques are used to gather useful and necessary information for the system. In this project, observation and research are used to gather information for the system.

Observation

Using observation techniques, information is gathered by observing the actual situation of how the classroom status and student behaviour is being monitored during the online class. This will help the analysts to have a better understanding of the current monitoring method in the online class. Through observation, the analyst is able to observe the current monitoring method and the difficulty of the method during the online class. This leads to necessary requirements of the system that can be identified. In addition, it also determined how the proposed system should be implemented into online classes.

Research

Other than observation, research also is conducted to gather information for this project. The information is gathered from others' research papers, articles, or websites on the Internet to understand how others use to solve the similar problem. Some of the ideas and implementations gathered are taken to suit this project. For example, find out from the website which model is more suitable for the proposed system and apply the idea to the project. Through research, the analyst is able to find the best solution based on the experience and research of others to adopt into the proposed system.

3.1.2 Fact Recording

After all the related information has been gathered through fact recording techniques above, the analysts need to identify and record the suitable fact into a documentation so that the system structure can be referred to during the subsequent phases. In the documentation, it may contain several diagrams that are used in fact recording such as Use Case Diagram, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), Structure Chart, etc. For example, figure 3.2 will contain a Data Flow Diagram (DFD) that shows how the data will flow in the system for hand gestures.

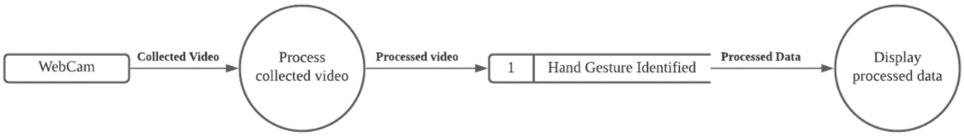


Figure 3.2: Data Flow Diagram (DFD) for hand gesture

3.2 Requirement Analysis

3.2.1 Project Scope

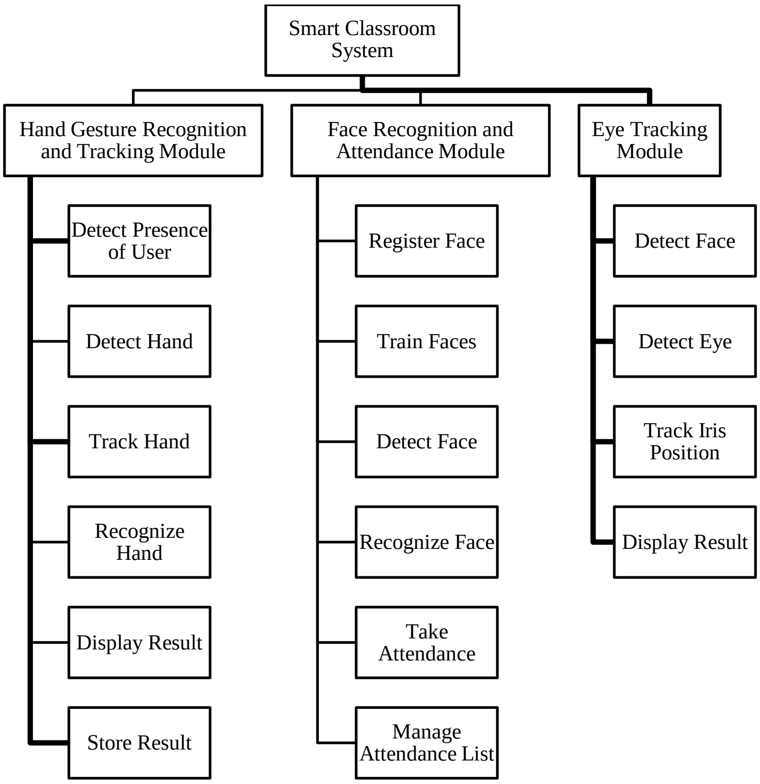


Figure 3.3: Structure chart of Smart Classroom System

Figure 3.3 above shows the scope of the proposed system. It includes 2 major modules which are hand gesture recognition and tracking module, and face recognition and attendance module. For the hand gesture recognition and tracking module, it will check the accuracy of the recognition when the hand is detected. When the accuracy value is higher than the predefined value, the hand will be tracked and display the result on the screen. After the results are displayed, the detailed information of the user's gestures will also be stored in Google Sheet. For the face recognition and attendance module, students can take their own attendance by using the face recognition. Besides, the teacher will be able to manage the student attendance list in Microsoft Excel after exporting from the online classroom. For 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

In this subchapter, the hardware, software, database, and programming languages that are used to implement the system will be discussed.

<u>Hardware Requirement</u>

Table 3.1: Hardware Requirement for Smart Classroom System

Hardware	Hardware Specification
Desktop or Laptop	• Processor: Intel Core i5 processor or above • Operating System: Windows 10 64-bits • Memory: 8GB RAM or above • Graphic Card: GTX 1050 or above • Storage: 128GB and above
IP Camera	Camera with 1080 HD and higher resolution

<u>Software Requirement</u>

Table 3.2: Software Requirement for Smart Classroom System

Hardware	**Development Tool and Software**
Desktop or Laptop	• Anaconda 3 • PyCharm • Visual Studio Code • Tensorflow • MediaPipe • Spyder • Jupyter Notebook • OpenCV

<u>Programming Language and Database</u>

Table 3.3: Programming Language and Database for Smart Classroom System

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

To create the system, we need a desktop or laptop with the specifications stated in Table 3.1. In addition, because the hand gesture recognition and tracking module is designed to use only the processing power of the central processing unit (CPU), it does not require the processing power of any advanced graphics card such as the Nvidia GTX series, etc. However, in order to speed up the performance of the module, advanced graphics cards can still be used. Visual Studio

Code is used to develop the system's HTML, CSS, JavaScript and JSON code. The libraries utilized to recognize and track hand is MediaPipe Hands.

3.2.3 Operation Environment

The suggested system's operational environment necessitated the use of gear such as a laptop, desktop, keyboard, and a web camera, Windows 10 or higher operating system, and web browsers such as Google Chrome are among the suggested applications for the proposed module to run in the operation environment.

3.3 Requirement Overview

3.3.1 Functional Requirements

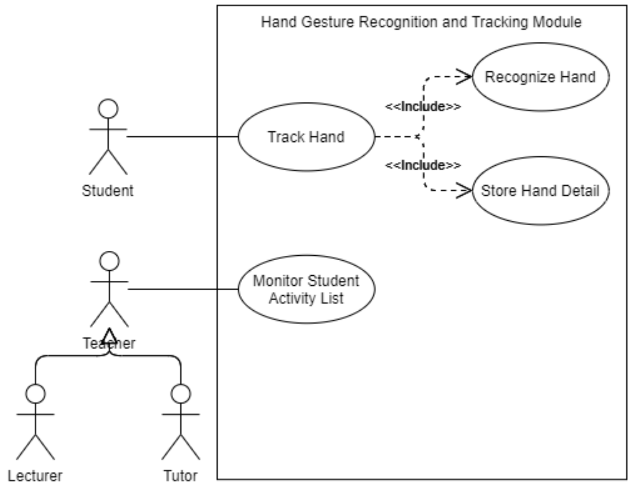


Figure 3.4: Use Case Diagram of Hand Gesture Recognition and Tracking Module

Figure 3.4 shows the functional requirements of Hand Gesture Recognition and Tracking Module. The hand tracking function will track the student's hand movement with the help of a hand recognition function which is to recognize a student's hand. Once the student's hand gesture been detected, the module will store it into the Google Sheet. Lecturer and Tutor that inherited from Teacher is able to use the module to monitor student activity with monitor student activity list function (Google Sheet). The list will contain the hand gesture that was done by students during the online class.

3.3.2 Non-Functional Requirements

Efficiency

Efficiency is the time that a system is required to perform its functions when it is required by the user. The system should be able to recognize a student's hand gesture in a short period of time. For example, when a student shows his/her hand on the Webcam, the system shall be able to recognize it within a few seconds.

Functional

Functional is the frequency of the system is able to do its jobs and satisfy the user. The system should be able to perform all its functions all the time when the function is requested by the user. For example, when students show his/her hand on the Webcam, the system shall be able to process the hand gesture and show the result to the student or teacher.

Usability

Usability refers to the simplicity of using the user interface of the system that is provided to the end user. The system should be able to receive user input and display the results to the user in a more understandable form that is drawn from past experiences. For example, the graphical user interface of the system should provide a button or switch to the user in order to activate the modules in the system.

3.4 Chapter Summary and Evaluation

In this chapter, the methodology used in this project which is an incremental model is discussed. The fact gathering techniques such as observation and research has been carried out to analyse the system requirements. After fact has been gathered, a Data Flow Diagram (DFD) is used for the fact recording. In addition, the hardware requirements, software requirements, database, and programming language requirements were also listed in this chapter. Finally, functional requirements and non-functional requirements of the system have also been listed.

Chapter 4

System Design

4 System Design

In Chapter 4, it will discuss the system design on how the data flows in the module and how the end user accesses the application with the aid of diagrams. In addition, the overall system architecture will also be discussed in this chapter. Lastly, the user interface of the module also will be discussed.

4.1 System Design

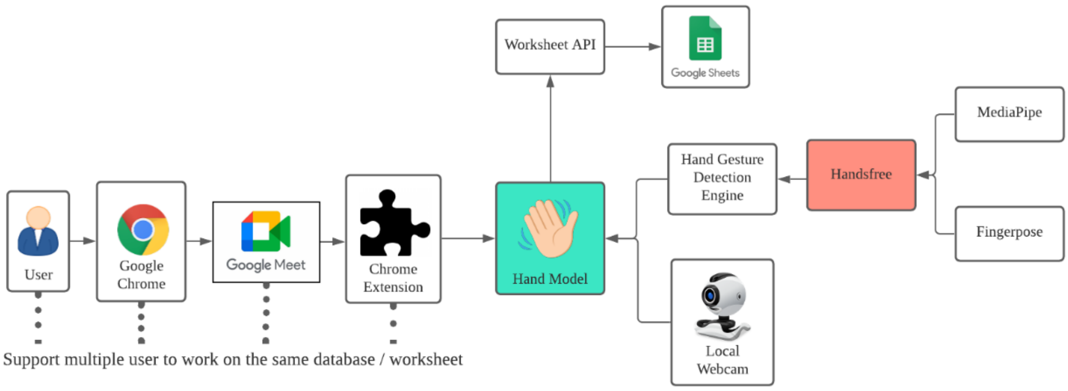


Figure 4.1: System Design Diagram for Hand Gesture Recognition and Tracking Module

Figure 4.1 above shows the system design diagram for the smart classroom system. In order to use the Hand Model (Hand Gesture Recognition and Tracking Module), the user needs to open the Google Chrome Web Browser, and use the Google Meet and chrome extension to access it. When the user starts the model, then the Hand Model will start the user local webcam and call the Application Programming Interface (API) from Handsfree (Oz Ramos, 2021). In the Handsfree, it will use MediaPipe to detect the user's presence in front of the webcam, the user's hand and draw hand landmarks, and Fingerpose (Andreas Schallwig, 2020) will be used for detecting the hand gesture. Once the gesture is detected, it will be saved to Google Sheet via the Worksheet API (a.k.a Sheet API).

4.2 Activity Diagram

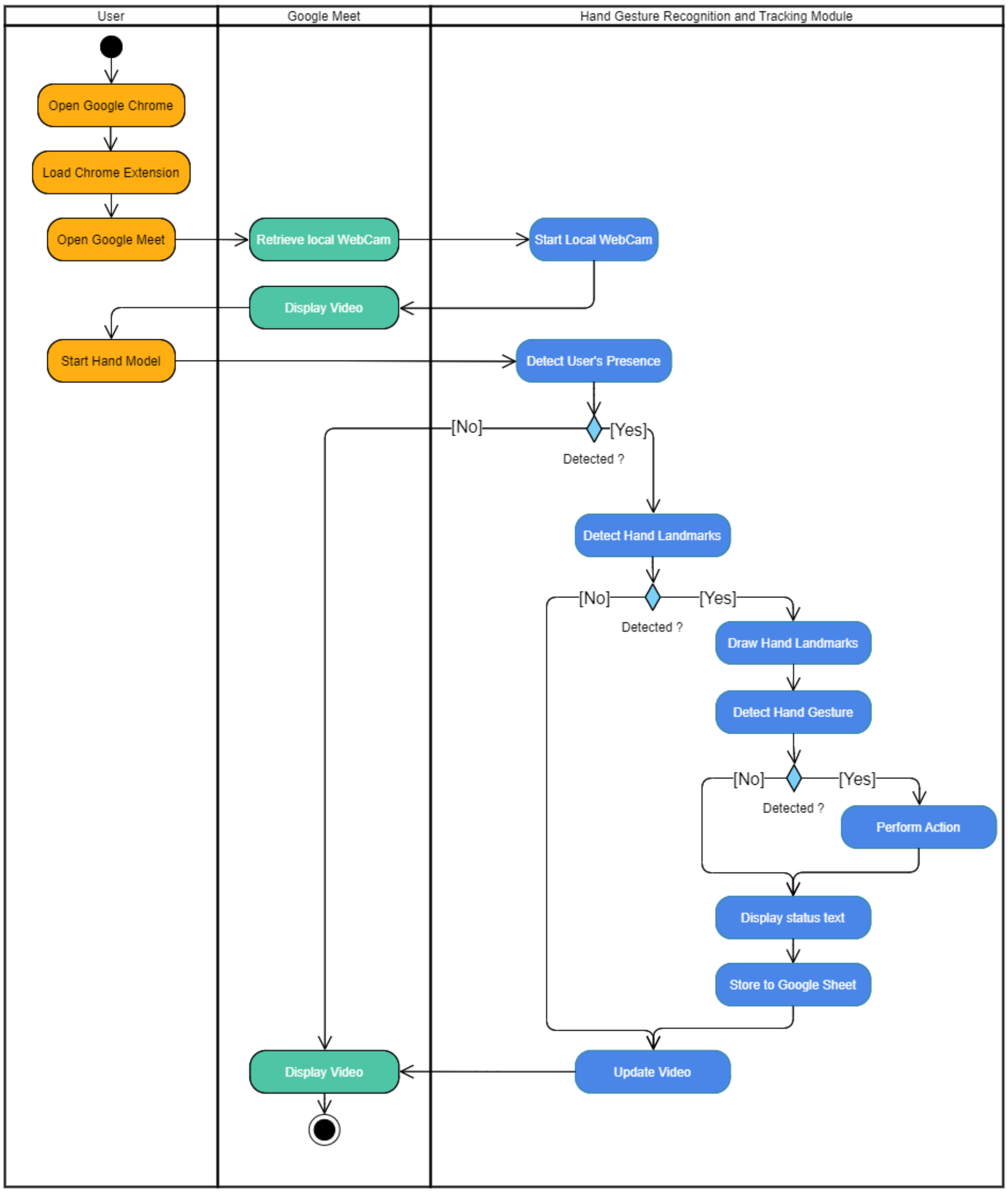


Figure 4.2: Activity Diagram for Hand Gesture Recognition and Tracking Module

Figure 4.2 shows the activity diagram of the Hand Gesture Recognition and Tracking Module. After the user opens Google Chrome, loads the chrome extension, and opens Google Meet, the Module will start the user's local webcam instead of taking Google Meet's own video stream. After the module successfully start the user's local webcam, it will replace the video stream of Google Meet. When the user starts the model, the module will first detect the user's presence and then the hand landmark. Once hand landmarks are detected, then the module will draw the hand landmarks on the video and then start to detect the hand gesture. If a hand gesture is detected, the module will send instructions to Google Meet to perform action. In addition, the module will also store the detected Hand Gesture into Google Sheet. At the end, the module will update the video and pass to Google Meet for display.

4.3 User Interface

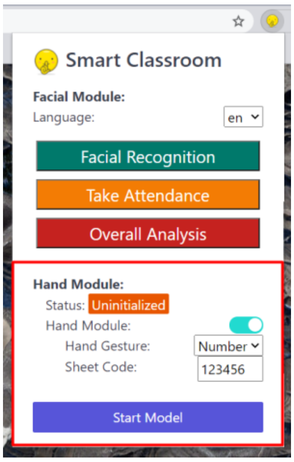


Figure 4.3: UI for Hand Gesture Recognition and Tracking Module

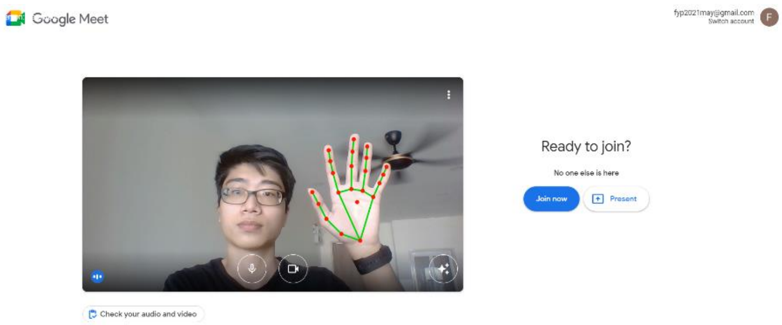


Figure 4.4: Design in Google Meet Main Page

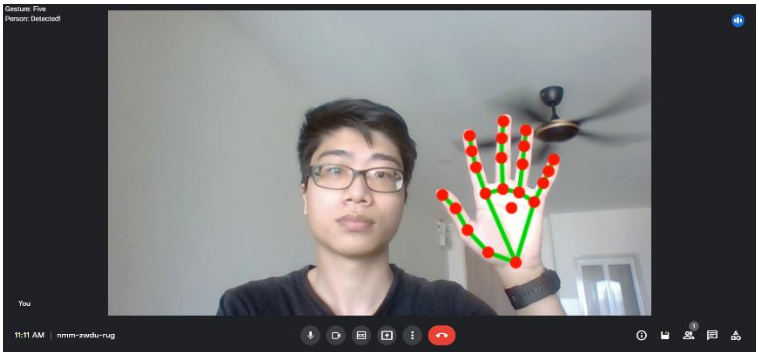


Figure 4.5: Design in Google Meet

Figure 4.3 shows the user interface for chrome extension of the module. In the UI, the user can see the status of the hand module, on/off hand module, change hand gesture set, and start the Handsfree Model. Then, Figure 4.4 and Figure 4.5 show the design when the model is started, and user hand landmarks detected. In addition, in Figure 4.5, it also will show the text of the hand gesture detected on the top left of Google Meet.

4.4 Chapter Summary and Evaluation

In this chapter, the data flows in the module and how the user accesses the system is discussed. In addition, the activity diagram shows how the system will operate. Lastly, the user interface of the module has also been discussed in this chapter.

Chapter 5

Implementation and Testing

5 Implementation and Testing

In Chapter 5, it will discuss the implementation and testing of the system. The implementation of the hand gesture recognition and tracking module will be discussed in detail. In addition, the test case of each requirement that identifies and analyses before will be tested and discussed in this chapter.

5.1 Implementation

5.1.1 Implementation Description

The Smart Classroom System consists of two different modules. One of the modules is the Hand Gesture Recognition and Tracking module and this module will be discussed in the following section. This module is developed in Windows 10 Operating System and programming languages such as HyperText Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript (JS), JSON, and other requirements. Requirements such as MediaPipe Hands, Google Sheet API (GAPI), Handsfree and FingerPose.

5.1.2 Code Snippets

Sample Code Snippets taken from manifest.json

```
{
    "update_url": "https://clients2.google.com/service/update2/crx",
    "name": "Google Meet Smart Classroom",
    "version": "1.3",
    "description": "Provide Easier Way For Class Monitoring and Assistance",
    "options_page": "/src/options/stream-capture.html",
    "devtools_page": "/src/devtools/index.html",
    "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 5.1: Sample code of Chrome Extension Detail and Background

Figure 5.1 above shows the code to set up the basic details of the Chrome Extension such as the name, version, description, etc. In addition, in the “background” key, it shows the import file to be run in the background of the Chrome extension. All the JS file that required to run in background will be import to “background” key.

```
    "key":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAl7bOJJ0a11iZA9a7Kl/hWbHcLHUxf6Ex34Agzet8A2q7XZjAYoO
glgZsnz7dG2f42XUD3k//haJlExdj/4lX6Xhne5iJr9i/0EJtDjyr6IjB0FVQV2Nhp0UgKMhrKFXyY+qH9UbZp5q730WhF3l
ugnbq6v8Ej2TsiW34TBDmh2txrEJI/gM2LkcJmszhqkNo/aWFN6S3R1bnC3tR3RLg0akeW0IIUPtZ9wbO7d8vKPiAgdAx0+2
ZPbMNDzr9hHxBOWTSTO++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"
    ],
```

Figure 5.2: Sample code of Chrome Extension Key and Web Accessible Resources

Figure 5.2 above shows the extension key to use Google API (GAPI) in the Chrome Extension in the “key” key. Besides, the “web_accessible_resources” key is to allow the file to be accessed by other JS file. For example, “src/content/mediaSourceSwap.js” are used to change the source of Google Meet’s Video Element with our Video Element that will draw with Face and Hand landmarks. Therefore, the JS files that need to be used by other JS files in the web page will be placed in “web_accessible_resources” key.

```
"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",
                "/src/content/idleBox.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 5.3: Sample code of Chrome Extension Content Scripts

Figure 5.3 above shows the JavaScript (JS) and CSS file that will be run when the user browse their website to Google Meet (“*://meet.google.com/*”).

```
"oauth2": {
        "client_id": "816803315444-q2f7l0nmlvie8pirqhu17d0r5iqv3kbk.apps.googleusercontent.com",
        "scopes": [
            "https://www.googleapis.com/auth/drive.file"
        ]
    },
    "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"
    },
    "manifest_version": 2,
    "content_security_policy":
"script-src 'self' https://www.gstatic.com/ https://*.firebaseio.com https://www.googleapis.co
m https://cdn.firebase.com https://*.firebaseio.com https://apis.google.com/ 'unsafe-eval'; obje
ct-src 'self';"
,
    "permissions": [
        "*://*/*",
        "storage",
        "tabs",
        "activeTab",
        "identity",
        "identity.email",
        "notifications",
        "*://meet.google.com/**-**-**"
    ]
}
```

Figure 5.4: Sample code of Chrome Extension Security Policy

Figure 5.4 above shows the security policy's code that is required to be inserted in order to allow the Chrome Extension to have the permission to access Google Services such as Firebase, Google API, Local Storage, etc. These permissions configurations are required by the Chrome Extension in order to allow the system to access Google Services in Google Meet.

Sample Code Snippets taken from background.js

```
var head = document.getElementsByTagName('head')[0];

// Call  onGAPILoad
var script = document.createElement('script');
script.type = 'text/javascript';
script.src = "https://apis.google.com/js/client.js?onload=onGAPILoad";
head.appendChild(script);

// Check whether gapi token get
chrome.identity.getAuthToken({ interactive: true }, function (token) {
    console.log('got the token', token);
})

// Set Up Service Account Token (To prevent Google Sheet give access denied issue)
function postJWT(jwt, callback) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4) {
            if (this.status == 200 && callback) {
                callback(this.responseText);
                return;
            }
            if (console) console.log(this.responseText);
        }
    };
    var parameters = "grant_type=" + encodeURIComponent(
"urn:ietf:params:oauth:grant-type:jwt-bearer") + "&assertion=" + encodeURIComponent(jwt);
    xhttp.open("POST", "https://www.googleapis.com/oauth2/v4/token", true);
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhttp.send(parameters);
}
```

Figure 5.5: Sample Code to Load Google API

Figure 5.5 above shows the sample code to load the Google API. postJWT function are used to set up a Service Account Token for all the users of our system to prevent Google Sheet show access denied. For example, if Service Account Token was not used, every time a new student wants to use the system, the lecturer or tutor may need to manually share Google Sheets permission to the new student. Therefore, Service Account Token are required in order to prevent the complicated processes such as share permission in using the system by new students.

```
function getCert() {
    var cert = //your json key (downloaded earlier) goes here
    {
        "type": "service_account",
        "project_id": "tidy-rainfall-319700",
        "private_key_id": "32fa7038d4277f57aef9f006633dac2eeec7aaec",
        "private_key": "-----BEGIN PRIVATE KEY-----\n
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC2NvKVVlYqLGVx\n
hklkWMPET7ag7gv+daIAjTWR4q7knrwQea+j5GD/gva1v+jsDH+IHiTIpL9ks3Yn\n
BkhihBkY25ztgGZAPiuUkB9rwHex+wGb4HEOSzJNUzdI+tsLxWxyRMi6uABjVSdp\n
uZQ8axjdqt66f7lVDC3gj+5K7AKcMfMNmSHZhFl41BbZHyJJM4LMvN2dd98vzrFW\n
wJxJiqN7m2Caf9FVcdoIB9DOEGx0b4AG0UrVj9MBaL3nLctBC4pWZuXAGbR/PBJ4\n
MMh2F484vrBF2J5/sIG3aqAgvG2ZT2rP+MeGzzfNxGTWxQGhJ7/WWCeZ2KEz1bU/\n
zHKlzOfDAgMBAAECggEAAUqdaIqOnMjoxSe3AHOSm6MW/8gAJMMPrWJS1WBebfdk\n
DNfX81jsiRMUDZC5JWcnRj80comcY6nT9g3Y2H3oMaFo2NQWpXDrXSeyWVCpaIUH\n
a2PpWTkWN68VzroccFZrYNn09EHWmt2hJGIw10bJC9Uos/iwxxYJgzRx82z9zb6X\n
0cBxFwMnlUp2brsRqBKWONCtI/how8KemKx/6LGUaeq9+zzLAJ9igLt+i48+wUbR\n
6LQYXdkQJPGm9P5AvEcK3ZLcFbmvXVy4LuneiOLuoDA3QHrYjiY84+d5e+l1PVWu\n
In/SRrOP0ETsyv9oeMR+bZyDI3w4B8DAJkK5ADZVhQKBgQDlQp3smCrvjFh7SLHJ\n
6Serz2gFj8QPvUmoLHxZKIMJOjRlxYhUrUEnEP0sO8wWccwthRng4bnGyIAluKBO\n
UyKQpyD40ffOhi+OBIt9UZouyDGvy8ZhihOKgt8VX52Dfq2kLIeGEdDAbbUkcg8J\n
A3MS5/iippERbG+3ZId7MAgKfQKBgQDLd53g9SiWC1ImV6jt7dQDaNuep7dDXeqF\n
1Z1/7CCeqpUOIflInyijeAHyhs8MJZfr+8odGecjFyBQLeLJhQrKf8B28QPP5tCx\n
2L/GuvdyAml8phuWhUvte3CZJXVbwEpwSaHg/yQ14kU6X6PM+t2geQPk2S3s73ni\n
2E8EuRkPPwKBgHUoVPH6VT6zdugRnyFyr95CEIN/E4+DiOeTTDx30nbdsrcred7j\n
5v6i1O5M5J+3qcK9WRst3zUxcyNTtUAxheFsNxJQopV8x0KJGguBrwwnLBUNSqS/\n
3fqTpJMUmLcA/pV+WAdwhUBBECe+zeYqSB0qREpCWrIPuwt6KS4Pvpw1AoGAR2M2\n
n4EktlBOiGllOeWh53hY5feOUzKWYFKsqf9dCo5rMcQ1xIcjKQeoF0G5T1e5T/DN\n
3xwBfsxUaA1RbschHB9qldDGN6GSQ3P7BWz/+J5kCHn3X/5wv2f5/6Gm5YQ4GeGb\n
zGd4evditrJrtDWooO8UsKyKcoErUK34O33+Dt0CgYAWWoUUzMM3HWFfVOVsrW0P\n
C/DNyq64KDF/MTZO8QPtBzfYhcUKi11vaHY63zTMGatc9AM59noyMOrXvKo5t30m\n
eujGfiGyFkKHLcEwwqHzBPdDcWitXqY+E1Azp/N7iYOOkeM9WS8dnaOFxkwouzaa\nlOg9sWNBRe3WY8moaOrwTg==\n
-----END PRIVATE KEY-----\n",
        "client_email":
"smartclassroomserviceaccount@tidy-rainfall-319700.iam.gserviceaccount.com",
        "client_id": "101198967386035525868",
        "auth_uri": "https://accounts.google.com/o/oauth2/auth",
        "token_uri": "https://oauth2.googleapis.com/token",
        "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
        "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/smartclassroomserviceaccount@tidy-rainfall-31
9700.iam.gserviceaccount.com"
    }

    return cert;
}

function getJWT() {
    var cert = getCert();
    var key = KEYUTIL.getKey(cert.private_key);
    var headers = { "alg": "RS256", "typ": "JWT" };
    var issued = Math.floor(new Date().getTime() / 1000);

    var claims = {
        "iss": cert.client_email,
        "scope": "https://www.googleapis.com/auth/spreadsheets",
        "aud": "https://www.googleapis.com/oauth2/v4/token",
        "exp": issued + 3600,
        "iat": issued
    };

    var jwt = KJUR.jws.JWS.sign(headers.alg, headers, JSON.stringify(claims), key);
    return jwt;
}
```

Figure 5.6: Sample Code to Obtain Service Account Token Configuration

Figure 5.6 above shows the sample code to obtain the configuration of Service Account Token in order to use the Service Account in the Chrome Extension.

```javascript
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';

// 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 5.7: Sample Code to Initialize GAPI

Figure 5.7 above shows the sample code to initialize the Google API (GAPI). The onGAPILoad function will use the apiKey and discoveryDocs to check whether the API key is valid or not. If valid, the system will print a success message, else will print an error message with detail. The onGAPILoad function is very important because we need to initialize it before we can use the system.

```
function checkSheetCode(programmeAvailable, sheetCode) {

    var data = {
        name: 'sheetCodeIsOn',
        sheetCode: null,
        detail: null,
        sheetID: null
    }

    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];

            // Create header for Sheet if Sheet empty
            checkEmptySheet(data.sheetID);

            break;
        }
    }

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

Figure 5.8: Sample Code to Check Valid Sheet Code

```javascript
function checkEmptySheet(sheetID) {

    gapi.client.sheets.spreadsheets.values.get({
        spreadsheetId: sheetID,
        range: 'Sheet1',
    }).then(async function (response) {

        if(response.result.values === undefined){
            //Create header
            let requests = []
            requests = requests.concat(createHandSheetHeaders(0));
            console.log(requests);

            const body = {
                requests: requests,
                includeSpreadsheetInResponse: true,
            }

            const init = {
                method: 'POST',
                async: true,
                headers: {
                    Authorization: 'Bearer ' + clientToken,
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify(body),
            }

            const response = await fetch(

`https://sheets.googleapis.com/v4/spreadsheets/1sRoB8cDP6Zi3jdmAYAyEpMStV3dGBu3FtW1KLPAKi78:batc
hUpdate`
,
                init
            )

            if (response.ok) {
                console.log('Successfully create Sheet\'s header!');
            } else {
                console.log('Fail to create Sheet\'s header! Please contact TARUC Management');
            }

        }

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

Figure 5.9: Sample Code to Create Header for Google Sheet

Figure 5.8 above shows the sample code to check the validity of sheet code entered by the user into the Chrome Extension. The system can only continue if the sheet code is valid. Besides, if the Google Sheet is empty, the system will create the header for the Google Sheet by using checkEmptySheet as shown in Figure 5.9 in order to ensure that a consistent header is created.

```javascript
function formatDate(d){
    const weekday = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];
    var day = weekday[d.getDay()];
    var date = ('0' + d.getDate()).slice(-2);
    var month = ('0' + (d.getMonth() + 1)).slice(-2);
    var year = ('0' + d.getFullYear()).slice(-4);

    return day + ', ' + date + '-' + month + '-' + year;
}

function calculateInterval(currentTime, startTime){

    var timeInterval = Math.abs(currentTime - startTime) / 1000;

    // calculate hours
    const hours = Math.floor(timeInterval / 3600) % 24;
    timeInterval -= hours * 3600;

    // calculate minutes
    var minutes = Math.floor(timeInterval / 60) % 60;
    timeInterval -= minutes * 60;

    // calculate seconds
    var seconds = timeInterval % 60;

    return  hours + ':' + ('0' + Math.trunc(minutes)).slice(-2) + ':' + ('0' + Math.trunc
(seconds)).slice(-2);
}
```

Figure 5.10: Sample Code to Format Date and Calculate Time Interval

Figure 5.10 above shows the sample code to format the date to be inserted into Google Sheet in order to standardize the data in Google Sheet. In addition, the sample code also shows the calculation of the time interval for the user to perform gestures. In the calculation of the time interval, it will use the current time minus the system start time. The time interval is used to analyze how often students use the hand module to perform hand gestures after starting the system.

```javascript
chrome.extension.onMessage.addListener(
    function (request, sender, sendResponse) {

        // User need to authorize their own account first to use the service account
        chrome.identity.getAuthToken({ interactive: true }, function (token) {

            if (request.name === 'sheetCode') {
                gapi.client.sheets.spreadsheets.values.get({
                    spreadsheetId: MANAGEMENT_SPREADSHEET_ID,
                    range: MANAGEMENT_SPREADSHEET_TAB_NAME,
                }).then(function (response) {

                    console.log(`Got ${response.result.values.length} rows back`);

                    const programmeAvailable = (response.result.values);
                    programmeAvailable.shift(); // Remove title row in Sheet from the array
                    // console.log(programmeAvailable);

                    checkSheetCode(programmeAvailable, request.code, request.meetName);

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

            } else if (request.name === 'meetAction') {

                // To prevent model start will append data to Google Sheet
                if (request.name !== undefined && request.gesture !== undefined) {

                    const d = new Date();
                    var currentDate = formatDate(d);
// Use different format instead of toLocaleDateString()
                    var currentTime = new Date();
                    var startTime = new Date(request.startTime);

                    var interval = calculateInterval(currentTime, startTime);

                    const body = {
                        values: [[
                            request.username,
                            request.email,
                            request.gesture,
                            currentDate,
                            startTime.toLocaleTimeString(),
                            currentTime.toLocaleTimeString(),
                            interval
                        ]]
                    };

                    // Append values to the spreadsheet
                    gapi.client.sheets.spreadsheets.values.append({
                        spreadsheetId: request.sheetID,
                        range: range,
                        valueInputOption: 'USER_ENTERED',
                        resource: body
                    }).then((response) => {
                        // On success
                        console.log(`${response.result.updates.updatedCells} cells appended.`)
                        sendResponse({ success: true });

                    }, function (error) {

                        var status = error.result.error.status;
                        alert('Error occur in Google SpreadSheet: ' + status + '\n
Please contact TARUC Management');

                    });
                }
            }
        })

        // Wait for response
        return true;
    }
);
```

Figure 5.11: Sample Code to Insert Hand Gesture Info to Google Sheet

Figure 5.11 above shows the sample code for inserting hand gesture information into Google Sheet after the user performs a gesture in the system. The system will append the hand gesture information into Google Sheets that entered by the user through sheetCode. In the sample code, it is using a listener to listen to messages passing in the Google Chrome. If the system detected a 'meetAction' message (POST Request), it would append the hand gesture information to the Google Sheet.

Sample Code Snippets taken from mediaSourceSwap.js

```javascript
var realUserMediaCall = window.navigator.mediaDevices.getUserMedia;

window.navigator.mediaDevices.getUserMedia = async function (constraints) {

    if (constraints.video.deviceId) {
        var canvas = document.getElementById("sourceCanvas");
        var stream = await realUserMediaCall.call(
            navigator.mediaDevices,
            constraints
        );

        var res = canvas.captureStream(10);
        var videoTrack = res.getVideoTracks()[0];
        var videoTrackStop = videoTrack.stop;
        videoTrack.stop = function () {
            stream.getVideoTracks()[0].stop();
            videoTrackStop.call(videoTrack);
        };

        var videoElement = tryGetVideoElement();
        videoElement.height = 400;
        videoElement.width = 400;
        videoElement.srcObject = stream;

        return res;
    } else {
        return await realUserMediaCall.call(navigator.mediaDevices, constraints);
    }
};

function tryGetVideoElement() {
    var existingElement = document.getElementById("realVideo");
    if (existingElement) {
        return existingElement;
    }
    var realVideo = document.createElement("video");
    realVideo.setAttribute("id", "realVideo");
    realVideo.setAttribute("style", "display:none");
    document.documentElement.appendChild(realVideo);
    return realVideo;
}
```

Figure 5.12: Sample Code to Change Google Meet Video Source

Figure 5.12 above shows the sample code to change the Google Meet Video Source with another Video Stream to ease the process of drawing Hand landmarks in the Google Meet. In this sample code, it will use Canvas Element and Video Element to replace the video stream of Google Meet. Firstly, the sample code will use Canvas Element to capture the stream and then get the video stream from the Canvas Element. After that, tryGetVideoElement function will be used to created new Video Element and append it to the document element. Lastly, our own video stream will replace Google Meet's Video Source.

Sample Code Snippets taken from content.js

```
const handsfree = new Handsfree({
    isClient: true,
    hands: {
        enabled: true,
        maxNumHands: 1,
        minDetectionConfidence: 0.85,
        minTrackingConfidence: 0.5
    },
    pose: {
        enabled: true,

        // Outputs only the top 25 pose landmarks if true,
        // otherwise shows all 33 full body pose landmarks
        // - Note: Setting this to true may result in better accuracy
        upperBodyOnly: true,

        // Helps reduce jitter over multiple frames if true
        smoothLandmarks: true,

        // Minimum confidence [0 - 1] for a person detection to be considered detected
        minDetectionConfidence: 0.5,

        // Minimum confidence [0 - 1] for the pose tracker to be considered detected
        // Higher values are more robust at the expense of higher latency
        minTrackingConfidence: 0.5
      }
})

var totalConfidence = 0;
var totalConfidenceCount = 0;

const handState = {
    video: null,
    canvas: null,
    username: null, // current username
    statusBox: null, // use to display the gesture identified
    poseBox: null, // use to display the pose identified
    chatbotText: null, // use to store message to put in Google Meet's Chatbox
    chatbotEnable: false, // use to determine whether send the chatbotText (Spacebar Key)
    handModuleIsOn: true, // whether start the hand gesture tracking and recognition module
    handActiveTimeStamp: null, // use to store the start time for interval calculation
    handTimeInterval: 0,
//use to store the time for interval calculation (prevent same hand appear more than one in speci
fic time)
    idleTimeInterval: 0,
//use to store the time for interval calculation (send warning when person not detected)
    idleBoxOn: false, // use to indicate whether idlebox popup is on or off
    handGesture: null, // store current hand gesture module (Number, Sign, Mouse)
    handStatusDrawing: null, // whether can draw hand landmark on the canvas
    previousGesture: null, // used to prevent gesture perfrom too many time
    sheetCode: null // use to store gapi spreadsheet detail get from popup.js
};
```

Figure 5.13: Sample Code to run handsfree (API)

Figure 5.13 above shows the sample code to run the handsfree API through new Handsfree(). Besides, in order to have more efficient control on the Hand Gesture Recognition and Tracking Module, a handState variable been created in order to manage the state in the module.

```
function keydown(evt) {
    if (!evt) {
        evt = event;
    }

    // Only available for number && sign (Mouse excluded)
    if (evt.ctrlKey && evt.shiftKey && evt.keyCode == 49) { // CTRL + Shift + 1 (number)
        if (handState.handStatusDrawing === 'start') {

            handState.handGesture = "number";
            chrome.storage.sync.set({ "handGesture": "number" });

        } else {

            alert("Please start the model first!");

        }
    }
    else if (evt.ctrlKey && evt.shiftKey && evt.keyCode == 50) { // CTRL + Shift + 2 (sign)
        if (handState.handStatusDrawing === 'start') {

            handState.handGesture = "sign";
            chrome.storage.sync.set({ "handGesture": "sign" });

        } else {

            alert("Please start the model first!");

        }
    } else if (evt.keyCode == 32) {
// Spacebar key (To activate Chatbot to send message in Google Meet)

        if (evt.path[0].tagName === "TEXTAREA") {
            if (evt.path[0].value.trim().length === 0) {
                handState.chatbotEnable = true;
            }
        } else {
            handState.chatbotEnable = true;
        }
    }
}
```

Figure 5.14: Sample Code for Shortcut Key

Figure 5.14 above shows the sample code in handling the shortcut key of the user while using the module in the Google Meet. CTRL + Shift + 1 shortcut key will be used for the number hand gesture, while CTRL + Shift + 2 shortcut key will be used for the sign hand gesture. This keydown function allows the system to change the gesture set based on the shortcut key. In addition, spacebar key with keycode 32 also provided to the user in order to send message during emergency situation. For example, if the system detected wrong hand gesture, then the user can press spacebar key to send a correct message immediately.

```
async function overrideGetUserMedia() {
    // Create canvas use for replace the Google Meet Video
    var canvas = document.createElement("canvas");
    canvas.setAttribute("id", "sourceCanvas");
    canvas.setAttribute("style", "display:none");
    document.documentElement.appendChild(canvas);
    handState.canvas = canvas;

    // Create status box for display message
    var statusBox = document.createElement("div");
    statusBox.setAttribute("id", "statusBox");
    statusBox.setAttribute("style",
"width: 175px; height: 25px; color: white; position: absolute; z-index: 999; padding: 5px 5px 0px
5px"
);
    document.body.appendChild(statusBox);
    handState.statusBox = statusBox;

    // Create status box for display message
    var poseBox = document.createElement("div");
    poseBox.setAttribute("id", "poseBox");
    poseBox.setAttribute("style",
"width: 175px; height: 25px; color: white; position: absolute; z-index: 999; padding: 5px 5px 0px
5px; top: 20px"
);
    document.body.appendChild(poseBox);
    handState.poseBox = poseBox;

    //Get user local video to replace the video in Google Meet
    injectMediaSourceSwap();

    // set up the mutation observer
    var observer = new MutationObserver(function (mutations, me) {
        // `mutations` is an array of mutations that occurred
        // `me` is the MutationObserver instance

        var canvas = document.getElementById("realVideo");
        if (canvas) {
            realVideoAdded(canvas);
            me.disconnect(); // stop observing
            return;
        }
    });

    // start observing
    observer.observe(document, {
        childList: true,
        subtree: true,
    });
}
```

Figure 5.15: Sample Code for Replace Google Meet Video Stream

Figure 5.15 above shows the sample code that used to replace the Video Stream in Google Meet with our own video stream. Besides, the canvas can also be used to draw the hand landmark in the own created video stream. In addition, the status box and pose box is created to show the result text in the Google Meet.

```javascript
function handGestureChatBox(chatboxAction) {
    // Index 2 == meet chatbox
    var chatbox = document.querySelector('[aria-label="Chat with everyone"]');

    if (handState.username != null) {

        if (chatbox.ariaPressed === "false") {
            chatbox.click();
        }

        // Wait for Google Meet to open the Chat Box
        var delayInMilliseconds = 500; //0.5 second

        setTimeout(function () {
            var textarea = document.getElementsByTagName("textarea");;

            if (textarea != null && handState.chatbotText != "") {
                // console.log(textarea[0]);
                textarea[0].click();

                if(chatboxAction != 'idle'){
                    textarea[0].value = 'ChatBot: \n' + handState.username + ' ' + handState.
chatbotText;
                }else{
                    textarea[0].value = 'ChatBot: \n' + handState.username +
' has idle for 10 Minutes!';
                }


                // Assign Enter key to Google Meet's chatbox to send message
                const keyboardEvent = new KeyboardEvent('keydown', {
                    code: 'Enter',
                    key: 'Enter',
                    charCode: 13,
                    keyCode: 13,
                    view: window,
                    bubbles: true,
                });

                textarea[0].dispatchEvent(keyboardEvent);


// Append the gesture detected to Gogole Sheet (Based on handState.sheetCode.sheetID)
                if (handState.sheetCode.sheetID !== null) {
                    var data = {
                        name: "meetAction",
                        username: handState.username,
                        email: currentEmail,
                        gesture: chatboxAction,
                        sheetID: handState.sheetCode.sheetID,
                        startTime: handState.handActiveTimeStamp
                    }

                    // Update last active timestamp
                    //chrome.storage.sync.set({ "handActiveTimeStamp": new Date().toString() });
                    handState.handTimeInterval = new Date();

                    chrome.runtime.sendMessage(data);
                }
            }
        }, delayInMilliseconds);
    }
}
```

Figure 5.16: Sample Code to Handle Hand Gesture

Figure 5.16 above shows the sample code to handle the hand gesture of the user. Once the user has performed some hand gestures, the module will send the user's hand gesture to others in the Google Meet's Chatbox. Besides, if the user idle for 10 minutes, the module will send the message to others in the Google Meet's Chatbox. In addition, once the message is sent in the chatbox, the module will also store the user's hand gesture information into the Google Sheet through background.js. Finally, the time interval of the module will be updated to the current date to prevent repeated hand gestures from appearing many times at once.

```
if(handsfree.data.pose.poseLandmarks !== undefined){
                        // Draw pose landmarks on the canvas
                        if (handsfree.data.pose.poseLandmarks) {
                            handState.poseBox.innerHTML = "Person: Detected!";
                            handState.idleTimeInterval = 0; // update idle time interval
                            handState.idleBoxOn = false;

                            if(document.getElementById("idleBoxPopup") != null){
                                document.getElementById("idleBoxPopup").outerHTML='';
                            }

                            if (handsfree.data.hands.multiHandLandmarks !== undefined &&
 handsfree.data.hands.multiHandedness != undefined) {

                                totalConfidence += handsfree.data.hands.multiHandedness[0].score;
                                totalConfidenceCount += 1;

// console.log("Confidence: " + (handsfree.data.hands.multiHandedness[0].score * 100).toFixed(2)
 + "%");
                                const gesture = handsfree.model.hands.getGesture();

                                var handGesture;

                                if (gesture[0] != null) {
                                    handGesture = gesture[0];

                                } else {
                                    handGesture = gesture[1];
                                }

                                if (handState.handGesture !== 'mouse') {
                                    if (handGesture.name !== "") {

                                        handGestureAction(handGesture.name);


// Backup plan (Spacebar key to send correct message immediately)
                                        if (handState.chatbotText != null && handState.
chatbotEnable == true) {

                                            // Let Google Meet send message
                                            handGestureChatBox(handGesture.name);
                                            handState.chatbotEnable = false;
                                            handState.handTimeInterval = new Date();
                                        }

                                        handState.statusBox.innerHTML = "Gesture: " +
 handGesture.name;

                                    } else {
                                        handState.previousGesture = "undefined";
                                        handState.statusBox.innerHTML = "Gesture: Undefined";
                                        handState.chatbotEnable = false;
                                    }
                                }

                                // Draw hand landmarks on the canvas
                                if (handsfree.data.hands.multiHandLandmarks) {
                                    for (const landmarks of handsfree.data.hands.
multiHandLandmarks) {

                                        drawConnectors(ctx, landmarks, HAND_CONNECTIONS, {
                                            color: '#00FF00',
                                            lineWidth: 5
                                        });

                                        drawLandmarks(ctx, landmarks, {
                                            color: '#FF0000',
                                            lineWidth: 2
                                        });
                                    }
                                }

                            } else {
                                handState.statusBox.innerHTML = "Finding Hands...";
                            }

                        }
                    }else{
                        handState.poseBox.innerHTML = "Finding Person...";

                        if (handState.idleTimeInterval == 0){
                            handState.idleTimeInterval = new Date(); // user start idle time
                        }
                    }
```

Figure 5.17: Main Code for Hand Gesture Recognition and Tracking Module

Figure 5.17 above shows the main code for Hand Gesture Recognition and Tracking Module, which will detect the presence of user, user's hand gestures and draw the user's hand landmark in Google Meet for visualization purposes. In addition, the sample code also shows a backup plan, if the module detects the wrong hand gesture, the user can immediately send the correct message by using ‘Spacebar’ key.

```javascript
// Action when gesture detected
function handGestureAction(gestureName) {

    if (gestureName !== handState.previousGesture) {
        var currentTime = new Date();

        // Calculate time interval to prevent duplicate hand gesture
        var interval = Math.abs(currentTime - handState.handTimeInterval) / 1000;
        // console.log(interval);

        // Update lastest hand gesture
        handState.previousGesture = gestureName;

        // Perform action based on current hand gesture
        switch (gestureName) {
            case "One":
                handState.chatbotText = "chosen first options!";
                break;

            case "Two":
                handState.chatbotText = "chosen second options!";
                break;

            case "Three":
                handState.chatbotText = "chosen third options!";
                break;

            case "Four":
                handState.chatbotText = "chosen forth options!";
                break;

            case "Five":
                handState.chatbotText = ""; // Will do nothing
                return;

            case "Help":
                //Click on Google Meet Raise Hand Button
                var help = document.querySelector('[jsname="SqzZRd"]');
                if (help != null) {
                    help.click();
                }

                handState.chatbotText = "";
                return

            case "Thank_You":
                handState.chatbotText = "saying Thank You!"
                break;

            case "Nice,I'm_Good":
                handState.chatbotText = "currently nice and good!"
                break;

            case "No_Question":
                handState.chatbotText = "no question for now!"
                break;

            case "Webcam_Microphone":
                // Click on Google Meet Microphone and Webcam button
                var webcam_microphone = document.querySelectorAll('[jsname="BOHaEe"]');
                webcam_microphone[0].click();
                webcam_microphone[1].click();

                handState.chatbotText = "";
                return

            case "Stick_Captions":
                // Click on Google Meet Caption button
                var cap = document.querySelector('[jsname="r8qRAd"]');
                if (cap !== null) {
                    cap.click();
                }

                handState.chatbotText = "";
                return
        }

        // Only allow the Google Meet sned message after 10 seconds
        if(interval > 10.0){
            interval = 0;
            handGestureChatBox(gestureName);
        }

    }

}
```

Figure 5.18: Sample Code to Handle Hand Gesture Action

Figure 5.18 above shows the sample code to handle the hand gesture action that perform by users. The system will use switch statement to indicate which action to be perform. For example, if user show “Help” hand gesture, then the system will retrieve Google Meet Raise Hand button and perform a click on it.

```
if (handState.idleTimeInterval != 0){
                        var currentIdleTime = new Date();

                        // Calculate time interval of idle
                        var idleInterval = Math.abs(currentIdleTime - handState.idleTimeInterval
) / 1000;

                        // Warning time
                        // In demo use 5 seconds and 10 seonds
                        if (idleInterval > 5.0){

                            if(handState.idleBoxOn == false){
                                var idleBox = document.createElement("div");
                                idleBox.setAttribute("id", "idleBoxPopup");
                                idleBox.innerText =
"Please Attend the class. After 10 Minutes will inform Organizer!";
                                document.body.appendChild(idleBox);

                                var closelink = document.createElement("div");
                                closelink.setAttribute("id", "idleBoxPopupCloseLink");
                                closelink.innerText = 'X';
                                document.getElementById("idleBoxPopup").appendChild(closelink);

                                document.getElementById("idleBoxPopupCloseLink").addEventListener
("click", removeIdleBoxPopup);

                                function removeIdleBoxPopup(){
                                    document.getElementById("idleBoxPopup").outerHTML='';
                                    handState.idleBoxOn = false;
                                }

                                handState.idleBoxOn = true;
                            }

                            if (idleInterval > 10.0) {
                                handGestureChatBox("idle");
                                handState.idleTimeInterval = 0; // update idle time interval
                            }

                        }

                    }
```

Figure 5.19: Sample Code to Show Popup Message

Figure 5.19 above shows the sample code to show the popup message when user idle for 5 minutes. If the user not idle, the popup message will not be display.

Sample Code Snippets taken from handsfree.js

```
/**
 * Send data to content scripts
 */
handsfree.use('contentScriptBus', {
    onFrame(data) {
        // Send data to content
        chrome.tabs.query({ active: true, currentWindow: true }, function (tabs) {
            try{
                for (var i = 0; i < tabs.length; ++i) {
                    chrome.tabs.sendMessage(tabs[i].id, { action: 'handsfree-data', data })
                }
            }catch (e){

            }

        })

        // Send data to active ports
        ports.webxrDevTools.forEach(port => {
            port.postMessage({
                action: 'handsfree-data',
                data
            })
        })
    }
})
```

Figure 5.20: Sample Code for Handsfree Communication

Figure 5.20 above shows the sample code that was used to pass the data from handsfree API to the content.js. This sample code will create a bus to send data to each other in the Chrome tabs. Once content.js receives the data from the handsfree API, the module will draw the hand landmarks based on the coordinates obtained from the handsfree API and display the results to the user.

5.1.3 Result

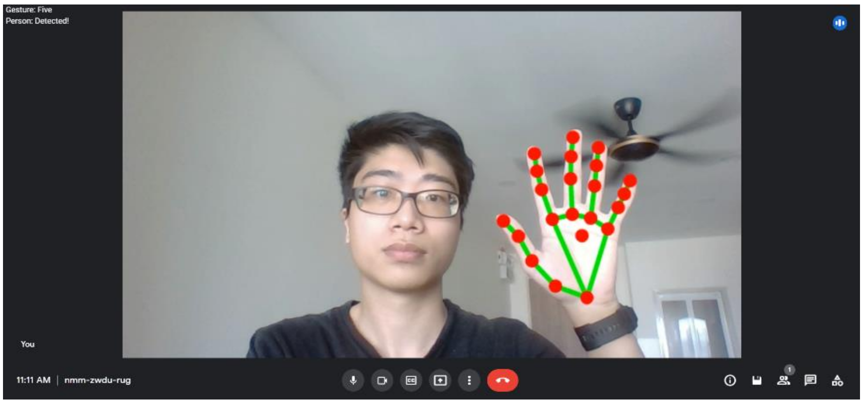


Figure 5.21: Result of Hand Gesture Recognition and Tracking Module

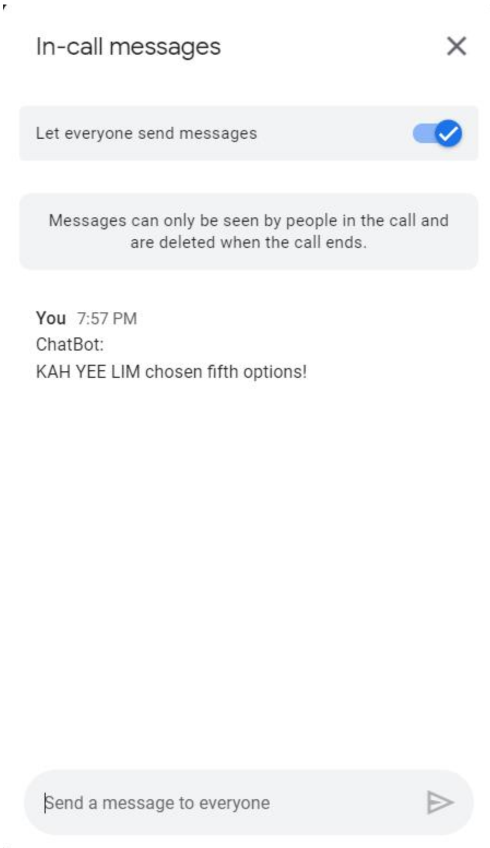
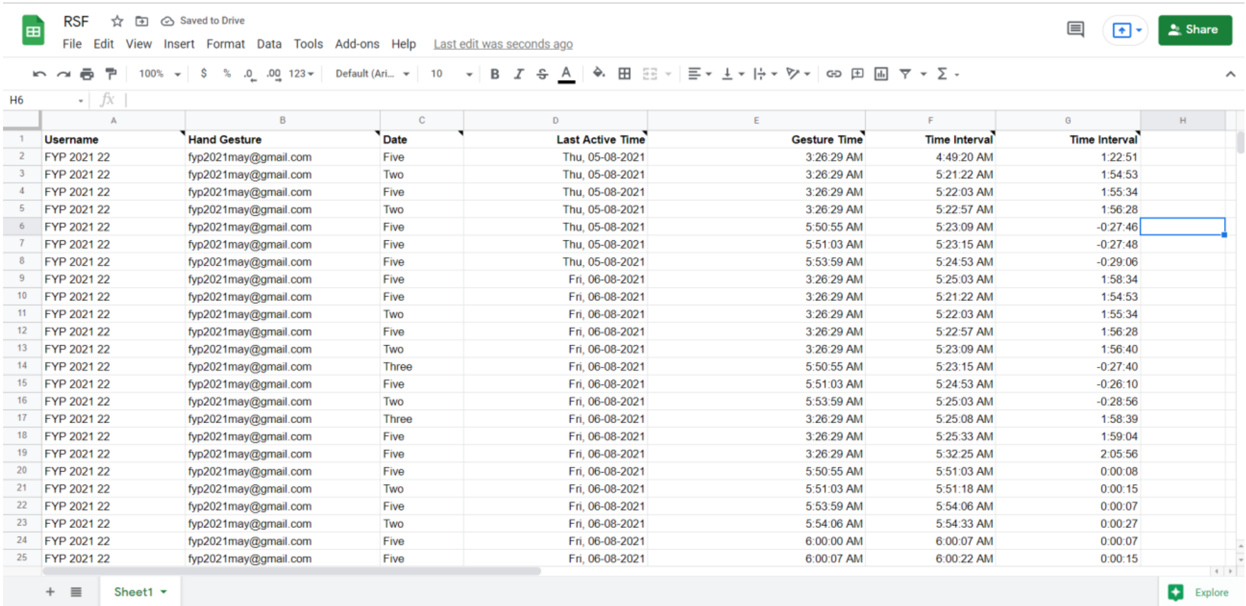


Figure 5.22: Result of Hand Gesture in Google Meet Chatbox



Username	Hand Gesture	Date	Last Active Time	Gesture Time	Time Interval	Time Interval
FYP 2021 22	fyp2021may@gmail.com	Five	Thu, 05-08-2021	3:26:29 AM	4:49:20 AM	1:22:51
FYP 2021 22	fyp2021may@gmail.com	Two	Thu, 05-08-2021	3:26:29 AM	5:21:22 AM	1:54:53
FYP 2021 22	fyp2021may@gmail.com	Five	Thu, 05-08-2021	3:26:29 AM	5:22:03 AM	1:55:34
FYP 2021 22	fyp2021may@gmail.com	Two	Thu, 05-08-2021	3:26:29 AM	5:22:57 AM	1:56:28
FYP 2021 22	fyp2021may@gmail.com	Five	Thu, 05-08-2021	5:50:55 AM	5:23:09 AM	-0:27:46
FYP 2021 22	fyp2021may@gmail.com	Five	Thu, 05-08-2021	5:51:03 AM	5:23:15 AM	-0:27:48
FYP 2021 22	fyp2021may@gmail.com	Five	Thu, 05-08-2021	5:53:59 AM	5:24:53 AM	-0:29:06
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	3:26:29 AM	5:25:03 AM	1:58:34
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	3:26:29 AM	5:21:22 AM	1:54:53
FYP 2021 22	fyp2021may@gmail.com	Two	Fri, 06-08-2021	3:26:29 AM	5:22:03 AM	1:55:34
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	3:26:29 AM	5:22:57 AM	1:56:28
FYP 2021 22	fyp2021may@gmail.com	Two	Fri, 06-08-2021	3:26:29 AM	5:23:09 AM	1:56:40
FYP 2021 22	fyp2021may@gmail.com	Three	Fri, 06-08-2021	5:50:55 AM	5:23:15 AM	-0:27:40
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	5:51:03 AM	5:24:53 AM	-0:26:10
FYP 2021 22	fyp2021may@gmail.com	Two	Fri, 06-08-2021	5:53:59 AM	5:25:03 AM	-0:28:56
FYP 2021 22	fyp2021may@gmail.com	Three	Fri, 06-08-2021	3:26:29 AM	5:25:08 AM	1:58:39
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	3:26:29 AM	5:25:33 AM	1:59:04
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	3:26:29 AM	5:32:25 AM	2:05:56
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	5:50:55 AM	5:51:03 AM	0:00:08
FYP 2021 22	fyp2021may@gmail.com	Two	Fri, 06-08-2021	5:51:03 AM	5:51:18 AM	0:00:15
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	5:53:59 AM	5:54:06 AM	0:00:07
FYP 2021 22	fyp2021may@gmail.com	Two	Fri, 06-08-2021	5:54:06 AM	5:54:33 AM	0:00:27
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	6:00:00 AM	6:00:07 AM	0:00:07
FYP 2021 22	fyp2021may@gmail.com	Five	Fri, 06-08-2021	6:00:07 AM	6:00:22 AM	0:00:15

Figure 5.23: Result of Hand Gesture Detail stored in Google Sheet

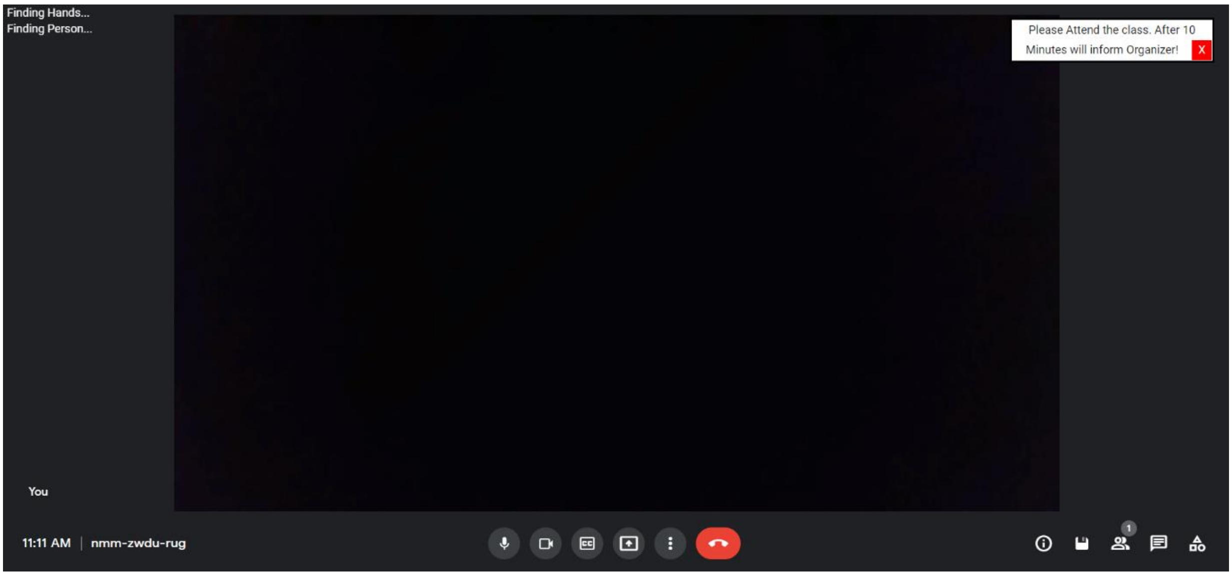


Figure 5.24: Result of Popup Message in Google Meet when User Idle

5.2 Testing

Testing is one of the important stages in the Software Development Life Cycle (SDLC). In this stage, the Hand Gesture Recognition and Tracking Module will be tested to make sure that all the module requirements are met. Several test cases will be created to test the module as shown below.

Test Case #: TC1	**Test Case Name: Start System's Hand Module**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the system's module start-up	

Pre-conditions: The system (Chrome extension) has been added to the Google Chrome browser and browsed to the Google Meet website.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Load system's user interface by clicking on the Smart Classroom Chrome Extension.	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Pass	
2	Enter valid Sheet Code, click on the *Start Model button* as follows: Sheet Code: 456535	Display text "Connecting...". After the model is started, the user interface of the system will be closed, and the Chrome extension logo of the smart classroom system will display the text "ON".	Display text "Connecting...". After the model is started, the user interface of the system will be closed, and the Chrome extension logo of the smart classroom system will display the text "ON".	Pass	
3	Show Hand in Google Meet.	Display user's Hand Landmark on Google Meet's video.	Display user's Hand Landmark on Google Meet's video.	Pass	

Post-conditions: User's hand landmark has been drew and displayed.

Test Case #: TC2	**Test Case Name: Run Hand Module Hand Gesture Number Features**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the Number Hand Gesture in the system's module.	

Pre-conditions: All the tests in TC1 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system's user interface by clicking on the Smart Classroom Chrome Extension.	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Pass	
2	Select the "Number" option in the Hand Gesture drop down list.	Drop down list of Hand Gestures shows "Number" option.	Drop down list of Hand Gestures show "Number" option.	Pass	

3	Show numbering Hand in Google Meet. For example, one to five.	Display user's Hand Landmark on Google Meet's video. After that, system will open Google Meet Chatbox and send a message as shown below. In addition, the user's hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME chosen NUMBER_GESTURE options!	Display user's Hand Landmark on Google Meet's video. After that, system will open Google Meet Chatbox and send a message as shown below. In addition, the user's hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME chosen NUMBER_GESTURE options!	Pass	

Post-conditions: Google Meet Chatbox and Google Sheet show user's hand gesture details.

Test Case #: TC3	**Test Case Name: Run Hand Module Hand Gesture Sign Features**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the Sign Hand Gesture in the system's module.	

Pre-conditions: All the tests in TC1 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system's user interface by clicking on the Smart Classroom Chrome Extension.	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Pass	
2	Select the "Sign" option in the Hand Gesture drop down list.	Drop down list of Hand Gestures shows the "Sign" option.	Drop down list of Hand Gestures shows the "Sign " option.	Pass	
3	Show Yeah Hand Gesture in Google Meet.	Display user's Hand Landmark on Google Meet's video. After that, system will open Google Meet	Display user's Hand Landmark on Google Meet's video. After that, system will open Google Meet	Pass	

		Chatbox and send a message as shown below. In addition, the user's hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME currently nice and good!	Chatbox and send a message as shown below. In addition, the user's hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME currently nice and good!		

Post-conditions: Google Meet Chatbox and Google Sheet show user's hand gesture details.

Test Case #: TC4	**Test Case Name: Run Hand Module's Number in Emergency Situation**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the backup plan in the system's module during emergency situation.	

Pre-conditions: All the tests in TC1 and TC2 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system's user	Display the user interface of Smart	Display the user interface of Smart	Pass	

	interface by clicking on the Smart Classroom Chrome Extension.	Classroom System. (i.e. Facial Module & Hand Module).	Classroom System. (i.e. Facial Module & Hand Module).		
2	Select the "Number" option in the Hand Gesture drop down list.	Drop down list of Hand Gestures shows "Number" option.	Drop down list of Hand Gestures show "Number" option.	Pass	
3	Show wrong numbering Hand in Google Meet. For example, one to five.	Display user's Hand Landmark on Google Meet's video. After that, system will open Google Meet Chatbox and send a message as shown below. In addition, the user's hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME chosen NUMBER_GESTURE options!	Display user's Hand Landmark on Google Meet's video. After that, system will open Google Meet Chatbox and send a message as shown below. In addition, the user's hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME chosen NUMBER_GESTURE options!	Pass	

4	Show correct numbering Hand in Google Meet and press the “Spacebar” key.	System will send correct message in the Google Meet Chatbox. In addition, the user’s hand gesture detail has been stored in Google Sheet.	System will send correct message in the Google Meet Chatbox. In addition, the user’s hand gesture detail has been stored in Google Sheet.	Pass	

Post-conditions: Google Meet Chatbox and Google Sheet show user’s hand gesture details.

Test Case #: TC5	**Test Case Name: Run Hand Module’s Sign in Emergency Situation**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the backup plan in the system’s module during emergency situation.	

Pre-conditions: All the tests in TC1 and TC3 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system’s user interface by clicking on the Smart Classroom Chrome Extension.	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Pass	

2	Select the “Sign” option in the Hand Gesture drop down list.	Drop down list of Hand Gestures shows the “Sign” option.	Drop down list of Hand Gestures shows the “Sign ” option.	Pass	
3	Show wrong Sign Hand Gesture in Google Meet.	Display user’s Hand Landmark on Google Meet’s video. After that, system will open Google Meet Chatbox and send a message as shown below. In addition, the user’s hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME currently nice and good!	Display user’s Hand Landmark on Google Meet’s video. After that, system will open Google Meet Chatbox and send a message as shown below. In addition, the user’s hand gesture detail has been stored in Google Sheet. ChatBot: YOUR NAME currently nice and good!	Pass	
4	Show correct sign Hand in Google Meet and press the “Spacebar” key.	System will send correct message in the Google Meet Chatbox. In addition, the user’s hand gesture detail has been stored in Google Sheet.	System will send correct message in the Google Meet Chatbox. In addition, the user’s hand gesture detail has been stored in Google Sheet.	Pass	

Post-conditions: Google Meet Chatbox and Google Sheet show user's hand gesture details.

Test Case #: TC6	**Test Case Name: Run Hand Module Mouse Features**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the Mouse feature in the system's module.	

Pre-conditions: All the tests in TC1 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system's user interface by clicking on the Smart Classroom Chrome Extension.	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Pass	
2	Select the "Mouse" option in the Hand Gesture drop down list.	Drop down list of Hand Gestures shows the "Mouse" option.	Drop down list of Hand Gestures shows the "Mouse" option.	Pass	
3	Refresh Google Meet.	N/A	N/A	Pass	

4	Show Hand in Google Meet.	Display a red dot in the Google Chrome Browser.	Display a red dot in the Google Chrome Browser.	Pass	
5	Move hand to button element and pinch hand.	Click on the button element.	Click on the button element.	Pass	

Post-conditions: System clicked the button element pointed by the red dot.

Test Case #: TC7	**Test Case Name: User is Idle in System**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**
Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the idle function when user idle.	

Pre-conditions: All the tests in TC1 and TC2 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system's user interface by clicking on the	Display the user interface of Smart Classroom System. (i.e. Facial Module &	Display the user interface of Smart Classroom System. (i.e. Facial Module &	Pass	

	Smart Classroom Chrome Extension.	Hand Module).	Hand Module).		
2	Leave the webcam.	After 5 minutes, system show popup message “Please Attend the Class. After 10 Minutes will inform Organizer!”.	After 5 minutes, system show popup message “Please Attend the Class. After 10 Minutes will inform Organizer!”.	Pass	
3	Leave the webcam.	After 10 minutes, system will open Google Meet Chatbox and send a message as shown below. In addition, the user’s idle detail has been stored in Google Sheet. ChatBot: YOUR NAME has idle for 10 Minutes!	After 10 minutes, system will open Google Meet Chatbox and send a message as shown below. In addition, the user’s idle detail has been stored in Google Sheet. ChatBot: YOUR NAME has idle for 10 Minutes!	Pass	

Post-conditions: Google Meet Chatbox and Google Sheet show user’s idle details.

Test Case #: TC8	**Test Case Name: User Not Idle in System**
System: Smart Classroom System	**Module: Hand Gesture Recognition and Tracking Module**
Design By: Lim Kah Yee	**Design Date: 14 October 2021**

Executed By: Lim Kah Yee	**Execution Date: 14 October 2021**
Short Description: Test the function when user not idle.	

Pre-conditions: All the tests in TC1 and TC2 must be passed.

Step	Action	Expected System Response	Actual Results	Pass/Fail	Comments
1	Open system's user interface by clicking on the Smart Classroom Chrome Extension.	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Display the user interface of Smart Classroom System. (i.e. Facial Module & Hand Module).	Pass	
2	Leave the webcam.	After 5 minutes, system show popup message "Please Attend the Class. After 10 Minutes will inform Organizer!".	After 5 minutes, system show popup message "Please Attend the Class. After 10 Minutes will inform Organizer!".	Pass	
3	Show yourself in front of webcam.	System popup message will be undisplay.	System popup message will be undisplay.	Pass	

Post-conditions: System undisplay the previously displayed popup message.

5.3 Chapter Summary and Evaluation

In this chapter, the code implementation of the module is discussed. In the code, it shows how to create the manifest.json for Smart Classroom System Chrome Extension, how to connect to Google API, how the Hand Gesture Recognition and Tracking Module ran, etc. In addition, the results of the Hand Gesture Recognition and Tracking Module are also shown in this chapter. Lastly, every test case has been carried out and all the requirements are met for this module.

Chapter 6

Discussions and Conclusion

6 Discussions and Conclusion

In Chapter 6, it will discuss the summary, achievements, contributions, limitations and future improvements of the implemented system as well as the issues and solutions during the project development.

6.1 Summary

The monitoring of students' activities using manpower is getting outdated. Instead, many new technologies such as Artificial Intelligent have been tried to implement in the classroom especially e-learning classroom (i.e. Google Meet). These technologies are gradually replacing the manpower as they are able to save more resources while giving high performance. As the number of students studying in academy institutions and engagement of e-learning classroom (i.e. Google Meet, Zoom, etc) are getting more and more, it will increase the burden for the tutors and lecturers to ensure all students are able to pay attention and are able to obtain the knowledge in the class. Therefore, this proposed module was designed and implemented to increase the efficiency of tutors and lecturers in monitoring students' activities in the e-learning classroom (i.e. Google Meet).

The methodology used by this project is an incremental model which is able to divide the whole software development into multiple standalone modules of the software development cycle and integrate them from time to time. The tools that are used to implement the Hand Gesture Recognition and Tracking Module into Smart Classroom System includes Mediapipe Hands, Google Sheet database, etc. Mediapipe Hands has been used because it able provide stable FPS (~50 FPS) and acceptable average accuracy compared to other hand detection libraries such as Tensorflow Handpose. In addition, Google Sheet database being used is because it is easy to implement and manage as Google Company have provided a very detailed API documentation for reference. Last but not least, in order to execute the system in Google Meet website, a chrome extension has been used as the platform to run the system in the Google Meet website in Google Chrome Browser.

6.2 Achievements

Initially the proposed project has two objectives and all objectives have been met by the implemented system. The objectives are to improve the efficiency of the lecturer and tutor in monitoring student behaviour during the class and to allow e-learning classrooms to interact with student's hand gestures and movement. As the Hand Gesture Recognition and Tracking Module is successfully implemented, the lecturer and tutor are able monitor student behaviour during the class. Lecturer and tutor will be able to monitor students' activities that have been done during the class through the Google Meet chatbox and also Google Sheet database. Besides, proposed module also allow students to use their hand gestures to interact with the e-learning classrooms (i.e. Google Meet). For example, when students show numbering gestures, the e-learning classroom can interact with it with the help of the proposed module.

For the completion of the projects, above 95 percent of the proposed functionality and requirements has been done. For another 5 percent, it could include the bugs and flaws that may not be able to sense during the system testing. In addition, through extensive research on the hand detection API to be used, the most effective hand detection API has been implemented into the proposed module. This allows the proposed module to run in low-spec and high-spec computer hardware with better performance. Last but not least, the current system can only run in Google Meet through the Google Chrome browser chrome extension. To make it run in the Google Meet mobile app, a lot of work needs to be done.

6.3 Contributions

Since the e-learning classroom may have many students, the lecturer or tutor is not able to handle everyone in the online class in the aspect of the teaching method. This may cause the student not able to catch up with the lessons. Hence, the proposed system will be able to help the lecturer and tutor as students' hand gesture will be recorded during the lesson and Google meet able to respond to it. For example, when student have questions and show "Help" hand gesture in the Google Meet, then the proposed system will activate the Raise Hand button of Google Meet. After that, the lecturer or tutor can immediately respond to the student's questions, and this will increase the effectiveness of the online class. Besides, lecturers or tutors can also focus on their teaching, because when students have any questions, the proposed system will notify the lecturer or tutor. In addition, the proposed system also able replace the dull traditional teaching methods as the hand gesture can act as a new communication tools in the e-learning classroom. For example, students can use hand gesture to answer quiz, state their current status, etc. This able motivate students to pay attention to the online class. As the quality of teaching and learning improves in the e-learning classroom, the students will graduate with full knowledge. This will help them to get better opportunities when finding jobs as well as increase the institution's reputation and images. Moreover, the proposed system can apply to companies too. The system will benefit the company that have online meetings because they can monitor the activities of participants, such as monitoring which staff have problems.

Last but not least, the proposed system also explores the possibility of implementing more and more Artificial Intelligent (AI) and Machine Learning (ML) to perfect the e-learning classroom in the future.

6.4 Limitations and Future Improvements

The most concerning limitation for the proposed project is the accuracy of hand gesture. This is because the hand gesture recognition are using Fingerpose that recognize hand gesture through finger direction (aka Fingerpose Curl). This makes it difficult to detect complex gestures because the direction of each finger is also complicated. For example, sometimes when performing the "Stick Caption" and "Webcam_Microphone" sign hand gestures, since most of the fingers are in the same direction, the system will recognize wrongly and then perform two actions. Using Fingerpose, the accuracy of recognizing complex gestures will be very low. For future improvements, the system can implement Machine Learning into the system in recognizing complex hand gesture that not able be done by Fingerpose. Therefore, if machine learning is implemented in the system and can cooperate with Fingerpose, the system will be able to detect complex gestures with high accuracy.

Another limitation of the proposed project is the lack of available complex hand gestures. This is because the currently proposed system only has some simple gestures, such as numbering gestures, signature gestures, and so on due to Fingerpose limitations. This may cause the proposed system to become unusable in complex situations. For example, user cannot use hand gestures to control the presentation slide while presenting in Google Meet. For future improvements, the system can add more complex gestures to allow users to use it in complex situations, such as using gestures to change presentation slides, use hand gesture to activate virtual background, etc.

6.5 Issues and Solutions

In the entire system development, some issues were encountered by the us. At first, every tool and technique used in the development are new for the us. For example, develop the system as a chrome extension to execute in Google Meet, using Google API to connect to Google Services, etc. However, we refer to a lot of documents, community groups such as GitHub, and other websites, such as stack overflow, to solve this issue.

Besides, due to lack of research of performance and accuracy on the Face and Hand API, we also face issue in deciding which API to be used in the Google Meet. In order to solve this issue, we have researched the available APIs. For example, for Hand Gesture Recognition and Tracking Module, I researched the performance and accuracy of the hand detection API such as Tensorflow Handpose and Mediapipe Hands. After the research is completed, compared with Tensorflow Handpose, I found that Mediapipe Hands has stable FPS and acceptable accuracy (~95%) under both low hardware specifications and high hardware specifications. Therefore, Mediapipe Hands has been implemented into the system for gesture detection.

In addition, the video stream of Google Meet cannot be input into the system for face and hand gesture detection, causing the system to fail to run. So in order to solve this problem, we have also done a lot of research, replacing the video stream of Google Meet with our own video stream, and inputting our own video stream into the system for face and gesture detection. By using our own video stream, the system is able to perform detection, and we can also draw face and hand landmarks in the video stream.

References

Andreas Schallwig, 2020. andypotato/Fingerpose, viewed 3 July 2021, <https://github.com/andypotato/fingerpose>.

BMW, 2021. What is BMW Gesture Control, viewed on 15 April 2021, <https://www.bmwoflittlerock.com/gesture-control/>.

Consumer.huawei.com, 2021. Basic Gestures | HUAWEI Support Global, viewed on 15 April 2021, <https://consumer.huawei.com/en/support/content/en-us00753853/>.

Google Open Source, 2021. MediaPipe, viewed on 16 May 2021, <https://opensource.google/projects/mediapipe>.

MediaPipe, 2019. MediaPipe – Home, viewed on 16 May 2021, <https://google.github.io/mediapipe/>.

MediaPipe, 2020. Hands, viewed on 16 May 2021, <https://google.github.io/mediapipe/solutions/hands>.

Olusola Olajide, A., 2016. Software Engineering Lecture Notes on SDLC Models, viewed on 16 April 2021, <https://www.researchgate.net/publication/311559089_Software_Engineering_Lecture_Notes_on_SDLC_Models>.

Oz Ramos, 2021. MIDIBlocks/handsfree, GitHub, viewed 3 July 2021, <https://github.com/MIDIBlocks/handsfree>.

Potnis, Swapnil & Jahagirdar, Aditi, 2014. Real Time Hand Gesture Recognition for Smart Classroom Environment. International Journal of Computer Trends and Technology. 17. 78-83. 10.14445/22312803/IJCTT-V17P115, viewed 6 April 2021.

Samsung ph. 2021, How do I use hand gestures to control my Samsung Smart TV, viewed 15 April 2021, <https://www.samsung.com/ph/support/tv-audio-video/how-do-i-use-hand-gestures-to-control-my-samsung-smart-tv/>.

TARUC, 2021, About TAR UC, viewed on 15 May 2021, <https://www.tarc.edu.my/content.jsp?cat_id=FDAA0D41-8967-4EAD-BE89-9BE26F147C47&fmenuid=2EC0DFFE-9F54-47D0-A957-48A101FF459D&fsubid=A1D6536E-597F-4A4C-8656-910E53EC6135>.

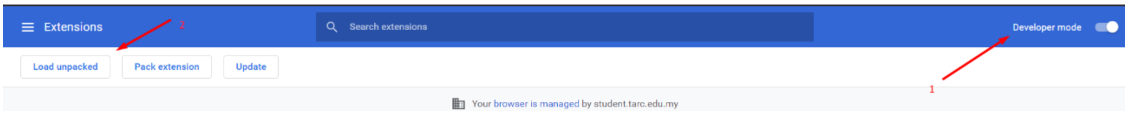
WER On Wave, 2020. Kinect Drawing, viewed on 15 April 2021, <https://www.youtube.com/watch?v=WnASSG7BwMQ&ab_channel=WEROnWave>.

Appendices

APPENDIX 1 User Guide

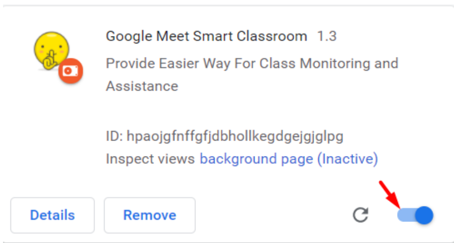
Requirements: Google Chrome Browser, Internet Connection

1. Visit https://github.com/Joan0018/SmartClassroom to download the system
2. Login account with test account for Hand Module (Google Chrome Browser)
    a. Username: smartclassroomfyp2021@gmail.com
    b. Password: smartclassroom2021
3. Visit **chrome://extensions**.
4. Enable **Developer Mode** on the top right.
5. Click **Load unpacked** and select this project's root folder.



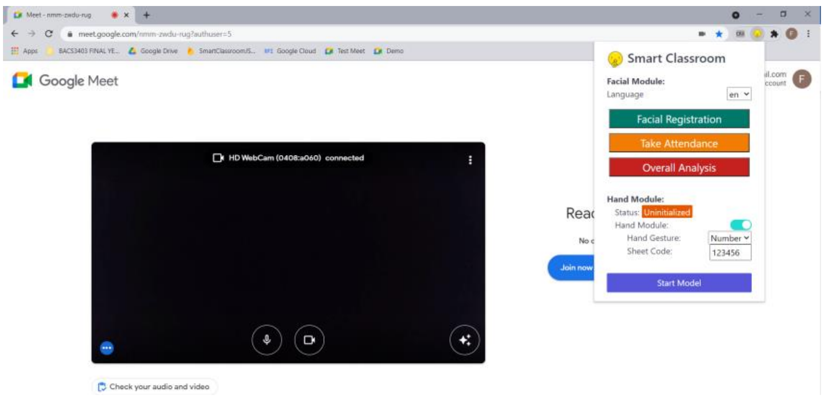
Google Chrome Browser Chrome Extension Page

6. Enable the Chrome Extension



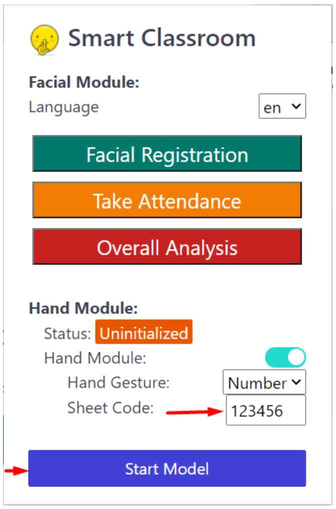
Chrome Extension Detail

7. Visit Google Meet Website and open the system user interface



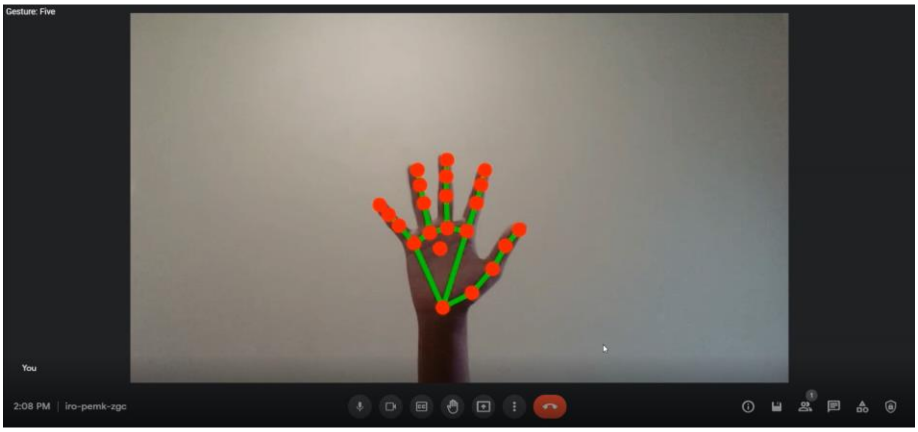
System User interface in Google Meet Website

8. Type **123456** in to the Sheet Code of Hand Module and start the model. Once Hand module started, the user interface of the system will automatically close.



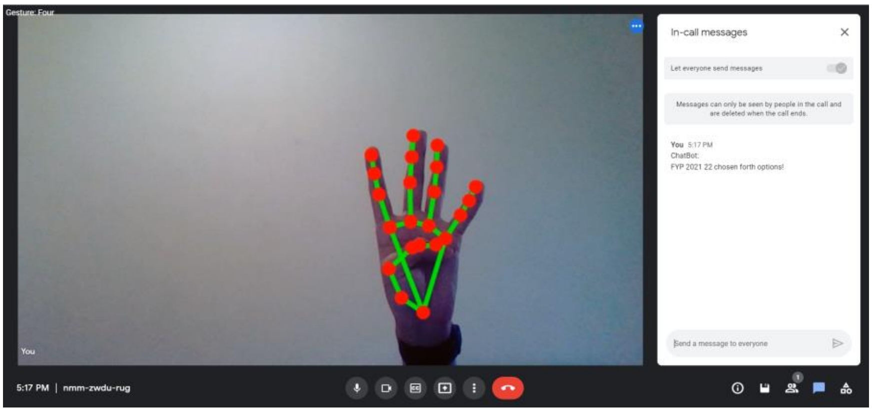
Start the Hand Module

9. Show hand and the system will draw hand landmark



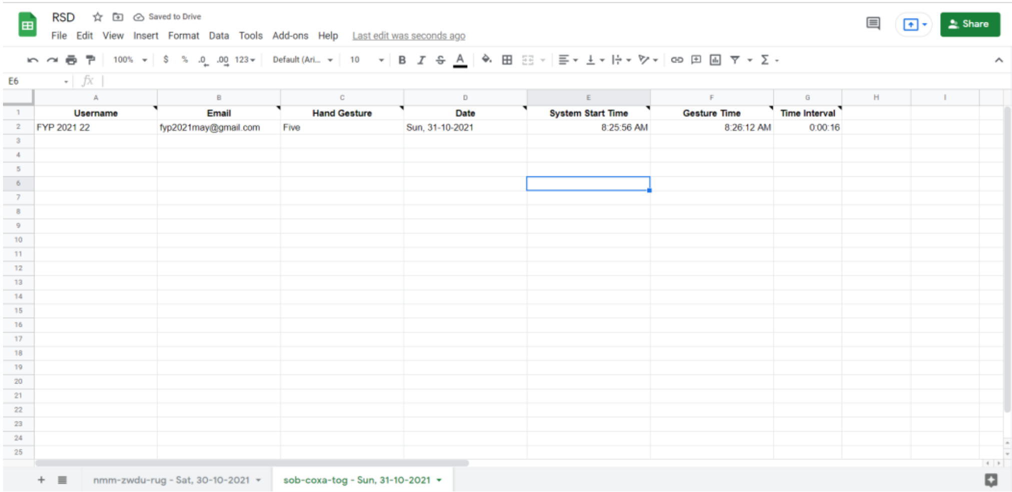
Google Meet with User's Hand Landmark

10. System open Google Meet Chatbox once detected user hand gesture



System open Google Meet Chatbox and Send Message based on Hand Gesture

11. System will store the hand gesture into the Google Sheet (https://docs.google.com/spreadsheets/d/1T_8qyXWb4ItHi4edX2uW58ZKgwrveQaaX6tGKsBHJp4/edit?usp=sharing).



Username	Email	Hand Gesture	Date	System Start Time	Gesture Time	Time Interval
FYP 2021 22	fyp2021may@gmail.com	Five	Sun, 31-10-2021	8:25:56 AM	8:26:12 AM	0:00:16

Hand Gesture Detail in Google Sheet

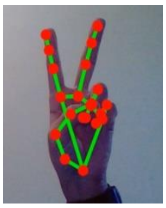
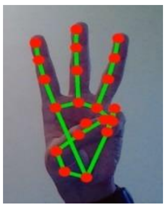
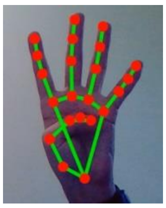
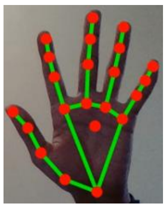
APPENDIX 2 Developer Guide

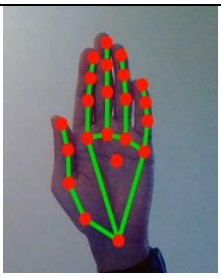
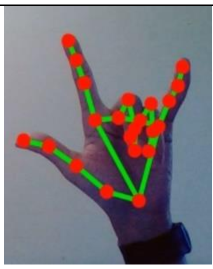
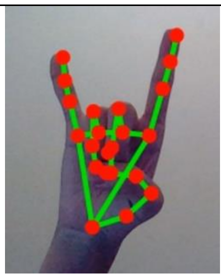
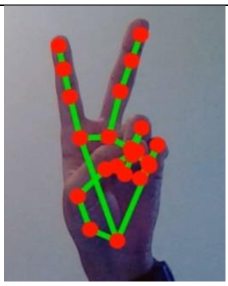
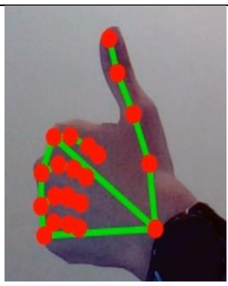
Requirements:

<u>**Software**</u>

1. Google Chrome Browser
2. Mediapipe Hands (API)
3. Visual Studio Code (Coding Tool)

APPENDIX 3 Hand Gesture Guide

<table>
<tr><th colspan="5">Number Hand Gesture</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>One</td><td>Two</td><td>Three</td><td>Four</td><td>Five</td></tr>
</table>

Sign Hand Gesture		
		
Help	Stick Caption	Webcam_Microphone
		
Nice, I’m Good	Thank You	No Question