

**FACIAL RECOGNITION-BASED ATTENDANCE MONITORING  
SYSTEM FOR EDUCATIONAL INSTITUTION**

By

Tan Shu Jing

(15ACB03979)

Supervised by

MR. TEOH SHEN KHANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology  
(Perak Campus)

JANUARY 2018

## REPORT STATUS DECLARATION FORM

**Title:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Academic Session:** \_\_\_\_\_

I \_\_\_\_\_  
(CAPITAL LETTER)

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

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

Verified by,

\_\_\_\_\_  
(Author's signature)

\_\_\_\_\_  
(Supervisor's signature)

**Address:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
Supervisor's name

**Date:** \_\_\_\_\_

**Date:** \_\_\_\_\_

**FACIAL RECOGNITION-BASED ATTENDANCE MONITORING  
SYSTEM FOR EDUCATIONAL INSTITUTION**

By

Tan Shu Jing

(15ACB03979)

Supervised by

MR. TEOH SHEN KHANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology  
(Perak Campus)

JANUARY 2018

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**FACIAL RECOGNITION-BASED ATTENDANCE MONITORING SYSTEM FOR EDUCATIONAL INSTITUTION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

## **ACKNOWLEDGEMENTS**

Upon completing this project, I would like to express my sincere expression towards my supervisor, Mr. Teoh Shen Khang whom had given me unconditional support throughout this project. Mr. Teoh had not only solved many of my encountered problems but also being very concern about my project by giving me a lot of alternative options during dealing with the project. The project couldn't complete smoothly if it wasn't his teaching dedication. Thank you very much for assisting me throughout this project.

Apart from that, I would also like to send my thanks to my parents for being very supportive in my hard times during conducting this project. Thank you for giving financial aids for the equipment in this project. Also, special thanks to my siblings and my friend William for the volunteer to provide their portrait to my project for the testing of my face database system. A million thank you.

## **ABSTRACT**

The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. Thus, by means of technology, this project will resolve the flaws existed in the current system while bringing attendance taking to a whole new level by automating most of the tasks.

The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a database system. At the end of the day, the attendance information regarding an individual can be accessed from a web server hosted by the raspberry pi.

In short, this upgraded version of attendance monitoring system not only saved many resources, but also provide huge convenience to the authority as many process are automated.

# TABLE OF CONTENTS

<b>FRONT COVER</b>	
<b>REPORT STATUS DECLARATION FORM</b>	<b>i</b>
<b>TITLE PAGE</b>	<b>ii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement and Motivation	1
1.2 Research Objectives	2
1.3 Project Scope and Direction	3
1.4 Impact, significance and contributions	3
1.5 Background Information	4
1.5.1 Project Field	4
1.5.2 Historical development prior to the project	5
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
<b>CHAPTER 3 SYSTEM DESIGN</b>	<b>9</b>
3.1 Hardware Development	9
3.1.1 Raspberry Pi 3 set up procedure	9
3.1.2 Setting up the Raspberry Pi for use	11
3.1.2.1 Connecting to Raspberry Pi remotely	11
3.1.2.2 Installing OpenCV into the Raspberry Pi	11
3.1.2.3 Setting up the Pi Camera Module	12
3.2 Software Development	13
3.2.1 The creation of the face database	13

3.2.2	The process of attendance taking	14
<b>CHAPTER 4</b>	<b>SOFTWARE DEVELOPMENT</b>	<b>16</b>
4.1	The creation of the face database	16
4.1.1	Specific requirements	16
4.1.2	Methodology	16
4.1.3	Flow Chart of the image acquisition process	18
4.1.4	Files included	20
4.1.5	Verification Test Plan	22
4.1.6	Implementation and Testing	23
4.2	The process of attendance taking	27
4.2.1	Specific requirements	27
4.2.2	Configuring system requirements	27
4.2.3	Configured Information	33
4.2.4	Methodology	34
4.2.5	Flow Chart of the attendance recording procedure	36
4.2.6	Files Included	37
4.2.7	Verification Test Plan	37
4.2.8	Implementation and Testing	38
<b>CHAPTER 5</b>	<b>DATABASE AND WEBPAGE DEVELOPMENT</b>	<b>40</b>
5.1	Database Development	40
5.1.1	Database Design	40
5.1.2	ER Diagram of the database	44
5.1.3	Relationship between the tables (in phpMyAdmin)	45
5.1.4	Database Analysis	45
5.2	Webpage Development	46
5.2.1	Service provided by the webpage	47
5.2.2	Webpage Analysis	53
<b>CHAPTER 6</b>	<b>CONCLUSION</b>	<b>55</b>
6.1	Project Review	55



6.1.1	Personal Insight	55
6.1.2	Implementation Issue and Challenges	56
6.1.3	Contributions of this project	57
6.1.4	Further Developments	57
6.2	Conclusion	58
<b>BIBLIOGRAPHY</b>		<b>59</b>
<b>PLAGIARISM CHECK RESULT</b>		<b>63</b>

## LIST OF FIGURES

<i>Figure Number</i>	<i>Title</i>	<i>Page</i>
<i>Figure 3.1.1</i>	Raspberry Pi 3 GPIO Header	
<i>Figure 3.1.2</i>	Connections of pi-fan and pi-camera on the raspberry Pi3 board	
<i>Figure 3.1.3</i>	Final assembled product	
<i>Figure 3.1.4</i>	Operating system installer	
<i>Figure 3.1.5</i>	Placement of micro SD card into the raspberry pi	
<i>Figure 3.1.6</i>	Interface of the VNC Viewer	
<i>Figure 3.2.1</i>	The system flow of the creation of face database	
<i>Figure 3.2.2</i>	The system flow of the attendance taking procedure	
<i>Figure 3.2.3</i>	Chart showing the service provided by the attendance management system webpage	
<i>Figure 4.1.1</i>	Image Acquisition and Pre-processing procedures	
<i>Figure 4.1.2</i>	Hierarchy manner of the face database	
<i>Figure 4.1.3</i>	Structure of the content in the csv file	
<i>Figure 4.1.4</i>	Flow Chart of the image retrieval process	
<i>Figure 4.1.5</i>	The relationships between the files during the face database creation	
<i>Figure 4.1.6</i>	Image provided by the AT&T Facedatabase	
<i>Figure 4.1.7</i>	Sample images taken through the system's pi camera	
<i>Figure 4.1.8</i>	Interface displayed when <i>prepare_gallery.sh</i> is executed	
<i>Figure 4.1.9</i>	Test result for invalid ID (1 <sup>st</sup> attempt)	
<i>Figure 4.1.10</i>	Test result for invalid ID (2 <sup>nd</sup> attempt)	
<i>Figure 4.1.11</i>	Test result for invalid ID (3 <sup>rd</sup> attempt)	
<i>Figure 4.1.12</i>	Test result for valid ID	
<i>Figure 4.1.13</i>	Test result for invalid face capture	
<i>Figure 4.1.14</i>	Test result for valid face capture	
<i>Figure 4.2.1</i>	How Nginx works with PHP	
<i>Figure 4.2.2</i>	Configuration steps for Nginx and PHP	
<i>Figure 4.2.3</i>	Test page for the installation of Nginx	
<i>Figure 4.2.4</i>	Test page for the configurations of Nginx with PHP	
<i>Figure 4.2.5</i>	Configuration steps for MySQL and phpMyAdmin	

<i>Figure 4.2.6</i>	Login interface for phpMyAdmin
<i>Figure 4.2.7</i>	Configuration steps for Wordpress
<i>Figure 4.2.8</i>	Completed design of the webpage
<i>Figure 4.2.9</i>	Customization login page
<i>Figure 4.2.10</i>	Process flow of attendance taking through the webpage
<i>Figure 4.2.11</i>	Method of acquiring the portrait
<i>Figure 4.2.12</i>	Process flow of the facial recognition procedure
<i>Figure 4.2.13</i>	Flow chart of attendance recording procedure
<i>Figure 4.2.14</i>	Relationships between files for the attendance taking process
<i>Figure 4.2.15</i>	Interface of the Record Attendance Page
<i>Figure 5.1.1</i>	Course Table Design View
<i>Figure 5.1.2</i>	Course Table Datasheet View
<i>Figure 5.1.3</i>	Student Table Design View
<i>Figure 5.1.4</i>	Student Table Datasheet View
<i>Figure 5.1.5</i>	Staff Table Design View
<i>Figure 5.1.6</i>	Staff Table Datasheet View
<i>Figure 5.1.7</i>	Course Timetable Table Design View
<i>Figure 5.1.8</i>	Course Timetable Table Datasheet View
<i>Figure 5.1.9</i>	Student Attendance Table Design View
<i>Figure 5.1.10</i>	Student Attendance Table Datasheet View
<i>Figure 5.1.11</i>	ER Diagram of the database
<i>Figure 5.1.12</i>	Relationship between the tables
<i>Figure 5.2.1</i>	Summary of the services provided by the Attendance Management System webpage
<i>Figure 5.2.2</i>	Registration Forms tab in the UTAR Attendance System webpage
<i>Figure 5.2.3</i>	4 Different registration forms in the UTAR Attendance Management System webpage
<i>Figure 5.2.4</i>	Record Attendance interface on the UTAR Attendance Management System webpage
<i>Figure 5.2.5</i>	Viewing Attendance Menu
<i>Figure 5.2.6</i>	Interface of View Attendance by Student
<i>Figure 5.2.7</i>	Interface of View Attendance by Subject
<i>Figure 5.2.8</i>	Login prompt from record attendance page
<i>Figure 5.2.9</i>	Login prompt from registration forms page

<i>Figure 5.2.10</i>	Login page into the UTAR Attendance System
<i>Figure 5.2.11</i>	Users created in Wordpress for this project
<i>Figure 5.2.12</i>	Staff Directory page
<i>Figure 5.2.13</i>	Student Directory page
<i>Figure 5.2.14</i>	Course Directory page

## LIST OF TABLES

<i>Table Number</i>	<i>Title</i>	<i>Page</i>
<i>Table 4.1.1</i>	Verification test plan for the creation of face database	
<i>Table 4.2.1</i>	Verification test plan of the attendance taking process	
<i>Table 4.2.2</i>	Input validation test towards Timetable ID (1 <sup>st</sup> attempt)	
<i>Table 4.2.3</i>	Input validation test towards Timetable ID (2 <sup>nd</sup> attempt)	
<i>Table 4.2.4</i>	Input validation test towards current date	
<i>Table 5.1.1</i>	Field's description in Course table	
<i>Table 5.1.2</i>	Course table properties	
<i>Table 5.1.3</i>	Field's description in Student table	
<i>Table 5.1.4</i>	Student table properties	
<i>Table 5.1.5</i>	Field's description in Staff table	
<i>Table 5.1.6</i>	Staff table properties	
<i>Table 5.1.7</i>	Field's description in Course Timetable table	
<i>Table 5.1.8</i>	Course Timetable table properties	
<i>Table 5.1.9</i>	Field's description in Student Attendance table	
<i>Table 5.1.10</i>	Student Attendance table properties	

## LIST OF ABBREVIATIONS

<i>Abbreviations</i>	<i>Description</i>
<i>CSV</i>	Comma Separated Values
<i>GPIO</i>	General Purpose Input/ Output
<i>GTK</i>	Graphical User Interface Toolkit
<i>GUI</i>	Graphical User Interface
<i>HDMI</i>	High Definition Multimedia Interface
<i>I/O</i>	Input/ Output
<i>ICT</i>	Information & Communication
<i>ID</i>	Identification
<i>IoT</i>	Internet of Things
<i>IP</i>	Internet Protocol
<i>LBPH</i>	Local Binary Patterns Histograms
<i>LED</i>	Light Emitting Diode
<i>NFC</i>	Near Field Communication
<i>NOOBS</i>	New Out of Box Software
<i>OS</i>	Operating System
<i>RFID</i>	Radio Frequency Identification
<i>SD</i>	Secure Digital
<i>VGA</i>	Video Graphics Array
<i>VNC</i>	Virtual Network Computing
<i>Wi-Fi</i>	Wireless Fidelity
<i>YML</i>	YAML Ain't Markup Language
<i>PHP</i>	Hypertext Processor
<i>SQL</i>	Structured Query Language
<i>FYP</i>	Final Year Project
<i>HTTP</i>	Hypertext Transfer Protocol
<i>FastCGI</i>	Fast Common Gateway Interface
<i>ER Diagram</i>	Entity Relationship Diagram

# **Chapter 1: Introduction**

This is a project about *Facial Recognition-Based Attendance Monitoring System for Educational Institution*. In this chapter, the problem and motivation, research objectives, project scope, project contributions and the background information of the project will be discussed in detail.

## **1.1 Problem Statement and Motivation**

According to the previous attendance management system, **the accuracy of the data** collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without the realization of the institution which violates the accuracy of the data. For example, student A is lazy to attend a particular class, so student B helped him/her to sign for the attendance which in fact student A didn't attend the class, but the system overlooked this matter due to no enforcement practiced. Supposing the institution establish an enforcement, it might need to waste a lot of human resource and time which in turn will not be practical at all. Thus, all the recorded attendance in the previous system is not reliable for analysis usage. The second problem of the previous system is where it is **too time consuming**. Assuming the time taken for a student to sign his/her attendance on a 3-4 paged name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their attendance which is obviously inefficient and time consuming. The third issue is with the **accessibility of those information by the legitimate concerned party**. For an example, most of the parents are very concerned to track their child's actual whereabouts to ensure their kid really attend the classes in college/school. However in the previous system, there are no ways for the parents to access such information. Therefore, evolution is needed to be done to the previous system to improve efficiency, data accuracy and provides accessibility to the information for those legitimate party.

## 1.2 Research Objectives

In order to solve the drawbacks of the previous system stated in 1.1, the existing system will need to evolve. The proposed system will reduce the paper work where attendance will no longer involve any manual recording. The new system will also reduce the total time needed to do attendance recording. The new system will acquire individual attendance by means of facial-recognition to secure data accuracy of the attendance.

**The followings are the objectives of this project:**

- To develop a portable Smart Attendance System which is handy and self-powered.
- To ensure the speed of the attendance recording process is faster than the previous system which can go as fast as approximately 3 second for each student.
- Have sufficient memory space to store the database.
- Able to recognize the face of an individual accurately based on the face database.
- Allow parents to track their child's attendance.
- Develop a database for the attendance management system.
- Provide a user friendly web interface for admins to access the attendance database and for non-admins (parents) to check their child's attendance.
- Allow new students or staff to store their faces in the database by using a GUI.
- Able to show an indication to the user whether the face- recognition process is successful or not.



### 1.3 Project Scope and Direction

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative smart system that can provide convenience to the institution. In this project, a smart device will be developed which is capable of recognising the identity of each individuals and eventually record down the data into a database system. Apart from that, a website will be developed to provide visual access to the information. The followings are the project scopes:

- The targeted groups of the attendance monitoring system are the students and staff of an educational institution.
- The database of the attendance management system can hold up to 2000 individual's information.
- The facial recognition process can only be done for 1 person at a time.
- There will be two types of webpage interface after the login procedure for the admins and the non-admins respectively.
- The project has to work under a Wi-Fi covered area, as the system need to update the database of the attendance system constantly.
- The smart device is powered up by power bank to improve the portability of the device.

### 1.4 Impact, significance and contributions

Many attendance management systems that exist nowadays are lack of efficiency and information sharing. Therefore, in this project, those limitations will be overcome and also further improved.

The impact and the contribution of this project is as follow:

- Students will be more punctual on attending classes. This is due to the attendance of a particular student can only be taken personally where any absentees will be noticed by the system. This can not only train the student to be punctual as well as avoids any immoral ethics such as signing the attendance for their friends.

- The institution can save a lot of resources as enforcement are now done by means of technology rather than human supervision which will waste a lot of human resources for an insignificant process.
- The smart device can operate at any location as long as there is Wi-Fi coverage which makes the attendance system to be portable to be placed at any intended location. For an example, the device can be placed at the entrance of the classroom to take the attendance.
- It saves a lot of cost in the sense that it had eliminated the paper work completely.
- The system is also time effective because all calculations are all automated.

In short, the project is developed to solve the existing issues in the old attendance system.

## **1.5 Background Information**

### **1.5.1 Project Field**

The project is developed based on the IoT (Internet of Things) concept where smart device is used to manage systems. IoT is generally about numerous device being interconnected uniquely in the existing internet infrastructure where information are shared among them. It can be viewed as a nervous system that links anything or everything together. It is usually achieved using sophisticated sensors and chips which are embedded in the physical things for real-time information retrieval. Data collected will then be analysed where intelligent decision will be taken by machines without human intervention to either solve the existing problem or to improve the current situation. In short, the IoT technology enhances many existing system to be more efficient and smarter.

The application area of this project is involved in the smart cities sub-field. Smart cities is a development vision using Information & Communication technology (ICT) in urban advancement where city's assets will be managed by smart devices to improve efficiency and also to reduce human resource consumption. By integrating these concepts, a Smart attendance monitoring system will be developed.

### **1.5.2 Historical development prior to the project**

Back in the years, attendance management system in school/colleges was done by manual reporting where the student's attendance were recorded by placing a mark or signature beside their name in a name list to indicate their presence in a particular class. While the staff in the institution will report their attendance through the punch card machine which also have to be done manually. Later on, some of those attendance systems had evolved into using smart cards to replace signature markings where each students/staff will be required to report their attendance using a smart card embedded with a unique identification chip.

## **Chapter 2: Literature Review**

Traditionally attendance was taken manually which is very time consuming and often leads to human error. Additionally, there are many uncertainties towards the sources of the attendance records which in fact, most of the attendance records are not retrieved from the actual situation. The old method that uses paper sheets for taking student's attendance can no longer be used. Based on the research, there are many solutions that are available to solve this issue.

According to research journal "*Attendance System Using NFC Technology with Embedded Camera on Mobile Device*" (Bhise, Khichi, Korde, Lokare, 2015). The attendance system is improved by using Near Field Communication (NFC) technology and mobile application. According to the research paper, each student is given a NFC tag that has a unique ID during their enrolment into the college. Attendance of each class will then be taken by touching or moving these tags on the lecturer mobile phone. The embedded camera on the phone will then capture the student's face to send all the data to the college server to do validation and verification. The advantages of this method is where the NFC is simple to use, and the speed of connection establishment is very high. It indeed speeds up the attendance taking process a lot. However, this system couldn't automatically spot the violation when the NFC tag is not personally tagged by the original owner. Apart from that, the convenience of the system which uses the mobile phone as the NFC reader was actually an inconvenience to the lecturer. Imagine if the lecturer had forgotten to bring their mobile phones to work, what would be the backup procedure for the attendance to be recorded? Moreover, most of the lecturer will not likely to prefer their personal smart phones to be used in this way due to privacy matter. Hence, unique information about the student like biometrics or face-recognition, which is guanine for a student should be used in replacement of the NFC tag. This will ensure attendance to be taken originally by the actual student.

The second research journals "*Face Recognition Based Attendance Marking System*" (Senthamil Selvi, Chitrakala, Antony Jenitha, 2014) is based on the identification of face-recognition to solve the previous attendance system's issues. This system uses camera to capture the images of the employee to do face detection and recognition. The captured image is compared one by one with the face database to search for the worker's face where attendance will be marked when a result is found in the face database. The main advantage of

this system is where attendance is marked on the server which is highly secure where no one can mark the attendance of other. Moreover, in this proposed system, the face detection algorithm is improved by using the skin classification technique to increase the accuracy of the detection process. Although more efforts are invested in the accuracy of the face detection algorithm, the system is yet not portable. This system requires a standalone computer which will need a constant power supply that makes it not portable. This type of system is only suitable for marking staff's attendance as they only need to report their presence once a day, unlike students which require to report their attendance at every class on a particular day, it will be inconvenient if the attendance marking system is not portable. Thus, to solve this issue, the whole attendance management system can be developed on an embedded design so that it can be work similarly with just batteries that makes it portable.

The third research journal "*Fingerprint Based Attendance System Using Microcontroller and LabView*" (Kumar Yadav, Singh, Pujari, Mishra, 2015) proposed a solution of using fingerprint to mark the attendance. This system is using 2 microcontrollers to deal with the fingerprint recognition process. Firstly, the fingerprint pattern will be obtained through a fingerprint sensor, then the information will be transmitted to microcontroller 1. Next microcontroller 1 will pass the information to microcontroller 2 to do the checking with the database that resides in it. After finding a student's match, the details are sent to the PC through serial communication to be displayed. This design is good as it accelerates development while maintaining design flexibility and simplifies testing. But again, this system is attached to a PC which make it not portable. Other than that, the database information cannot be accessible easily. Meaning that, for the parents whom are interested in knowing their child's attendance cannot easily or conveniently access the information. Therefore, to provide accessibility of the student's information to the legitimate concerned party, the information can be uploaded to a web server for easy access. While the authentication for the appropriate access can be enforced through a login screen.

According to the forth research journal "*RFID based Student Attendance System*" (Hussain, Dugar, Deka, Hannan, 2014), the proposed solution is almost similar to the first research journal where RFID technology is used to improve the older attendance system. In this system, a tag and a reader is again used as a method of tracking the attendance of the students. The difference between the first journals with this is where attendance's information can be accessed through a web portal. It provides more convenient for information retrieval.

Again, this system is imperfect in the sense that, firstly, it is not portable, as the RFID reader can only work when it is connected to a PC. Secondly, the RFID tag is not a genuine information that can uniquely identify a student, thus, resulting in the inaccuracy of the collected attendance information.

In conclusion, a better attendance monitoring system should be developed based on its portability, accessibility and the accuracy of the collected attendance information.

## Chapter 3: System Design

The design part of the attendance monitoring system is divided into two sections which consist of the hardware and the software part. Before the software part can be developed, the hardware part is first completed to provide a platform for the software to work. In the hardware part, the process of the raspberry pi's setup configuration will be briefly explained in this chapter. While in the software development part, there will be two major process flow which will be further discussed in Chapter 4.


### 3.1 Hardware Development

The hardware used in this project so far consists of only 4 components which are:

- Raspberry Pi 3
- Raspberry Pi 8mp Camera Module
- Power Supply Cable
- 16Gb Micro SD Card Class 10

#### 3.1.1 Raspberry Pi 3 set up procedure

When the raspberry pi 3 first arrived, its casing, pi-fan, and the pi camera are not assembled. Thus, the first thing to do is to screw the pi-fan in place on the casing provided

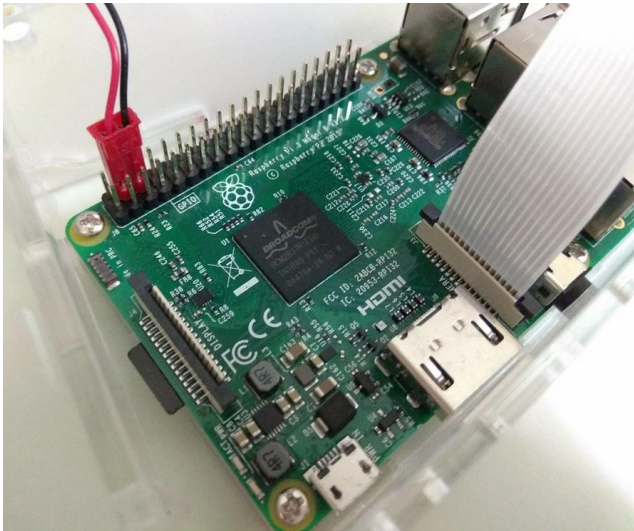
Raspberry Pi 3 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1, I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

**Figure 3.1.1 Raspberry Pi 3 GPIO Header**  
(element14 community, 2017)

and next secure the pi board onto the casing by again tightening it with screws. For the fan to work, it's end has to be connected to a 5v pin and a ground pin which are pin 4 and pin 6 respectively with the red-wired connector to pin 4 and black-wired connector to pin 6. Then, assemble the pi camera onto the raspberry pi by first lifting up the camera port's tab which is located between the Ethernet and HDMI ports on the raspberry pi to loosen it up, next insert the pi camera connector into the port with the silver



surface on the camera's connector facing the HDMI port. After that, push the port's tab downwards to secure the connection. Finally, assemble the casing and the final product is shown in Figure 3.1.3.

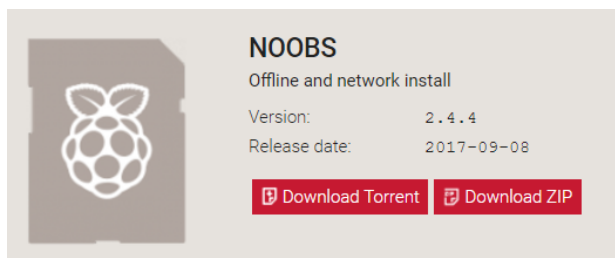


**Figure 3.1.2** Connections of pi-fan and pi-camera onto the raspberry pi 3 board



**Figure 3.1.3** Final assembled product

After the hardware is assembled, it is ready to install the appropriate operating system into the raspberry pi. The installation process starts by downloading NOOBS (New Out of Box Software) from the raspberry pi official website into the micro SD card. Then the micro SD card is inserted into the micro SD slot located below the raspberry pi board.



**Figure 3.1.4** Operating system installer (*Raspberry Pi Foundation, 2017*)



**Figure 3.1.5** Placement of micro SD card into the raspberry pi

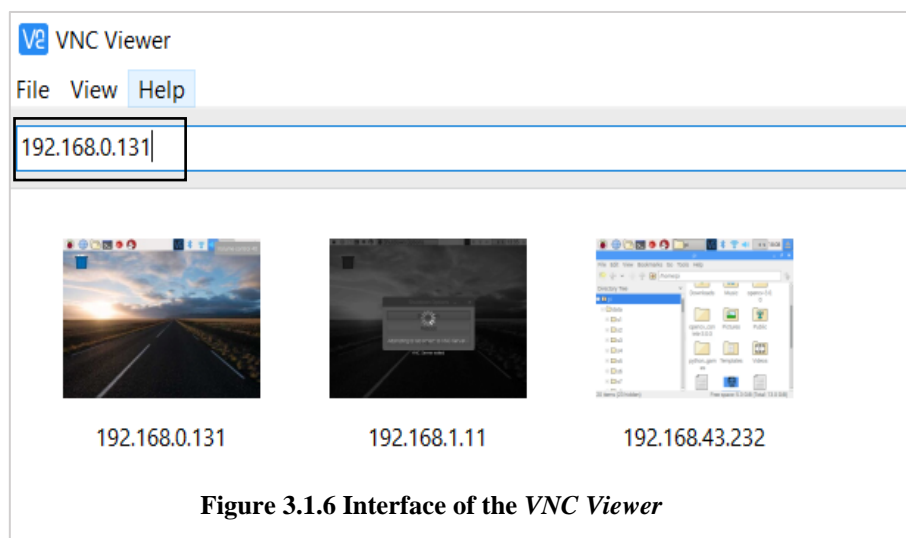
The next step would be the installation of *Raspbian* operating system (OS) into the raspberry pi. The raspberry pi is connected to a monitor via the HDMI slot on the raspberry pi through the HDMI converter to the VGA port on the monitor for display. Then, a mouse and a keyboard are connected to the raspberry pi via the USB ports to provide communication platform between the raspberry pi. Finally, the raspberry pi is powered up and the initialization is completed by following the on-screen instruction.



## 3.1.2 Setting up the Raspberry Pi for use

### 3.1.2.1 Connecting to Raspberry Pi remotely

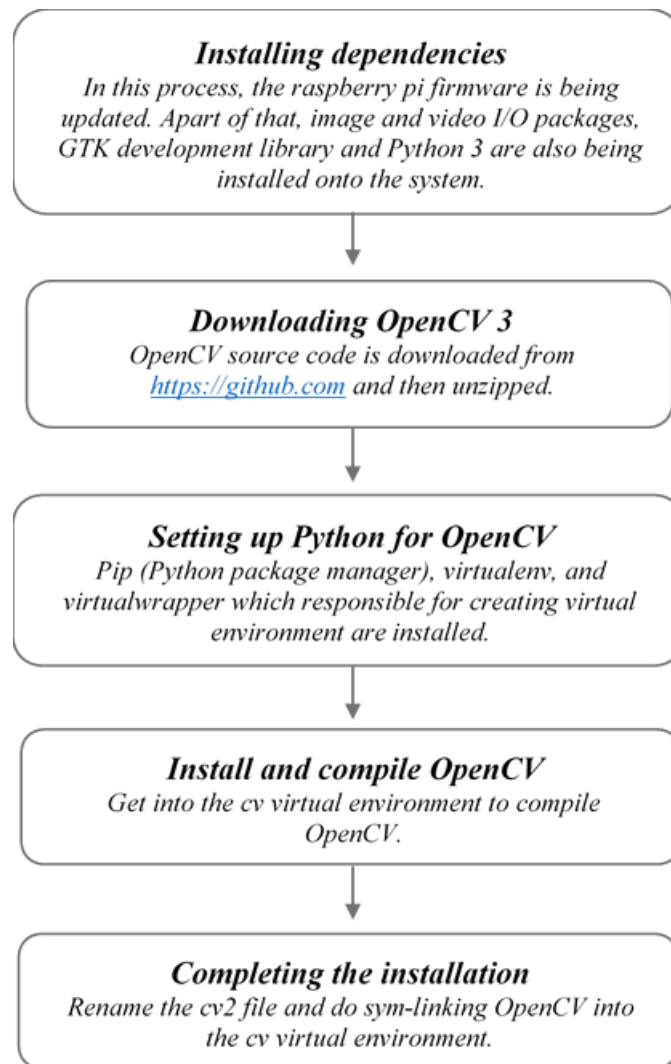
Before the raspberry pi can be accessed from a remote device through the connection of wifi, initial setup has to be done for future convenience. Therefore, when the desktop first appeared after the initialization process mentioned in the previous section, the Wi-Fi is turned on to obtain wifi connection from the hotspot provided by a mobile phone. This is to allow the raspberry pi to remember the entered password for that particular Wi-Fi, so that in the future, the raspberry pi will automatically connect to the wifi on its start-up. Then, a remote access software such as *VNC Viewer* is installed onto the laptop. Upon connecting the raspberry pi to the remote viewer software (*VNC Viewer*) on the laptop, the raspberry pi has to be in the same network as the laptop, then the connection progress can be done by just entering the IP address of the raspberry pi into the *VNC Viewer*.



**Figure 3.1.6 Interface of the VNC Viewer**

### 3.1.2.2 Installing OpenCV into the Raspberry Pi

In this project, OpenCV is used to do facial recognition where the whole program will be coded in Python Language. The installation of OpenCV is merely not enough, therefore, OpenCV is installed with Python bindings to bridge Python and C++ in OpenCV. In short, the binding is crucial to enable the calling of a C++ function from Python. The following is a brief walkthrough on the installation process that had done:



### 3.1.2.3 Setting up the Pi Camera Module

After installing the Pi Camera Module onto the raspberry pi board, the next step is to enable the camera module. Before being able to do that, the raspberry pi firmware is being updated first. Then, the camera module is being enabled by going into the configuration menu at the terminal. Next, the raspberry pi is rebooted. In order for python to interact with the pi camera, a *picamera* module with NumPy array support is installed. This is due to OpenCV takes images as NumPy arrays.

## 3.2 Software Development

There are two major system flows in the software development section as shown below:

- The creation of the face database
- The process of attendance taking

Both process mentioned above are essential because they made up the backbone of the attendance management system. In this section, the process of both flows will be briefly described. Meanwhile, their full functionality, specific requirements and also the methods/approach to accomplish such objectives will be discussed in the upcoming chapter.

### 3.2.1 The creation of the face database

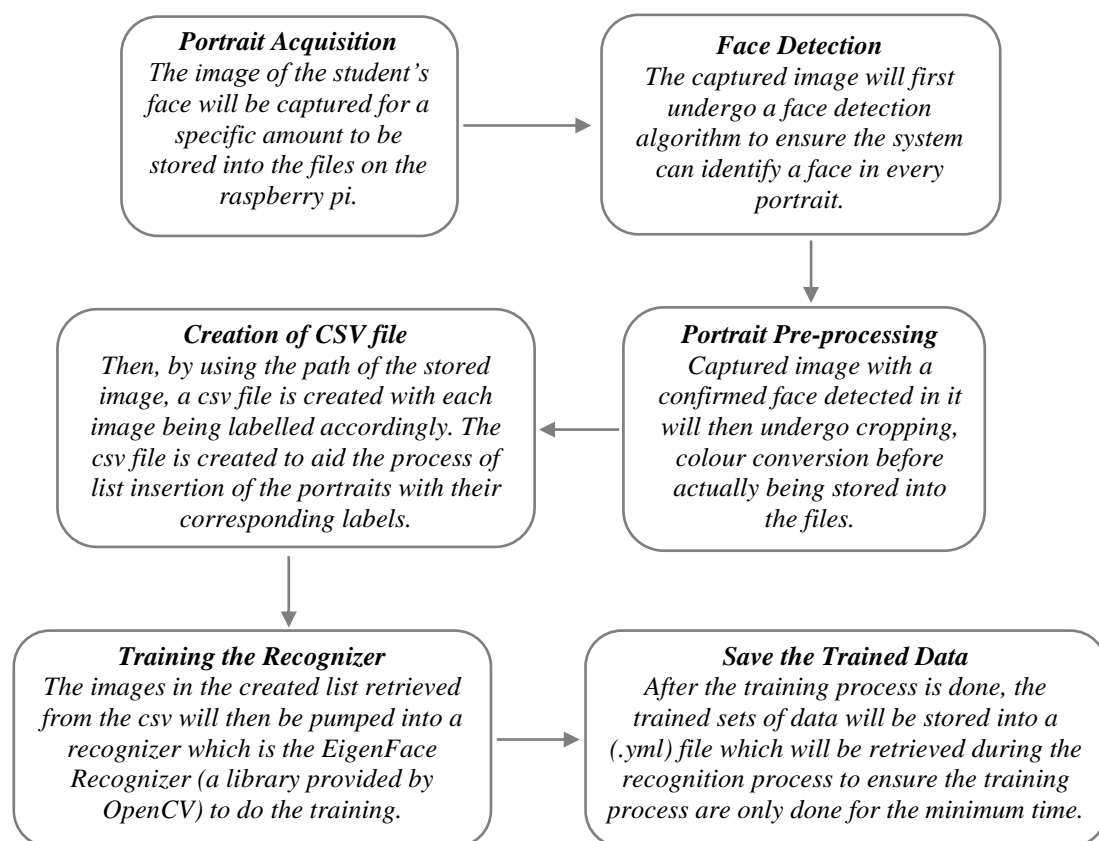
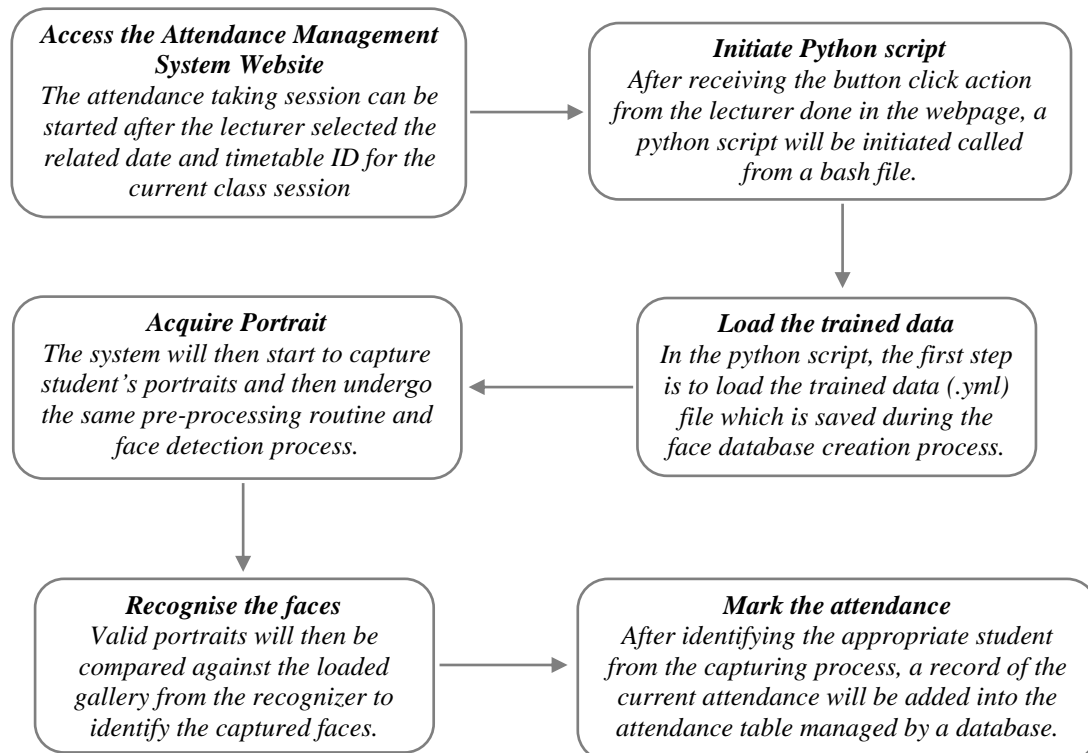


Figure 3.2.1: The system flow of the creation of face database

The face database is an important step to be done before any further process can be initiated. This is because the face database acts as a comparison factor during the recognition process which will be discussed in later section. In the process above, a csv file is created to aid the process of image labelling because there will be more than one portrait stored for each student, thus, in order to group their portraits under the name of the same person, labels are

used to distinguish them. After that, those images will be inserted into a recognizer to do its training. Since the training process is very time consuming as the face database grew larger, the training is only done right after there is a batch of new addition of student's portraits to ensure the training is done as minimum as possible.

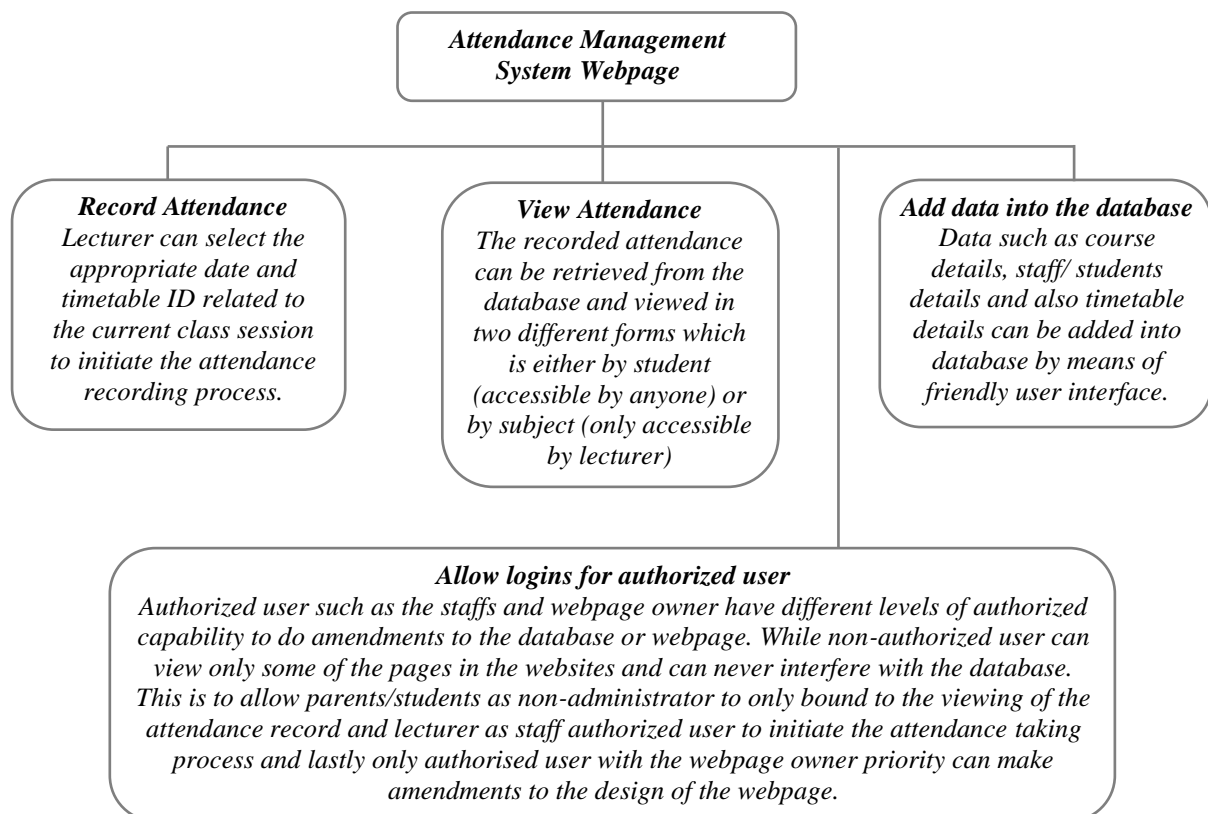
### 3.2.2 The process of attendance taking



**Figure 3.2.2: The system flow of the attendance taking procedure**

Other than the creation of face database, the rest of the remaining process can all be done through a webserver. Thus, the attendance taking procedure will also be done through a web server. This is to provide a friendly user-interface to the user (lecturer) while being able to conduct an execution on the raspberry pi to do attendance taking without the need to control the raspberry pi from a terminal which will be ambiguous for most user. Therefore, just with a click of button on the webpage, a python script will be executed which it will launch a series of initialization such as loading the trained data to the recognizer and etc. The attendance taking process will then proceed in a loop to acquire, identify and mark the attendance for each of the students that is obtained from the pi camera. In chapter 4, every step in both of the process flow above will be explained in detailed on its accomplishment. However, for the whole system to store the data effectively, a database using *MySQL* will be created to store every single piece of related information in a proper manner so that during

the data retrieval process, information can be easily tracked or filtered to be displayed. The detailed description of the database created will be further elaborated in chapter 5. Apart from the database development, another important component which is the website development will also be further discussed in the same chapter. In this project, the website acts as a platform for the user to communicate with the database which is stored in the raspberry pi. The following figure describes the service provided by the designed webpage:



**Figure 3.2.3: Chart showing the service provided by the attendance management system webpage**

## **Chapter 4: Software Development**

### **4.1 The creation of the face database**

#### **4.1.1 Specific requirements**

There will be several requirements to achieve the creation of the face database. The below are the required software or packages needed to accomplish this objective.

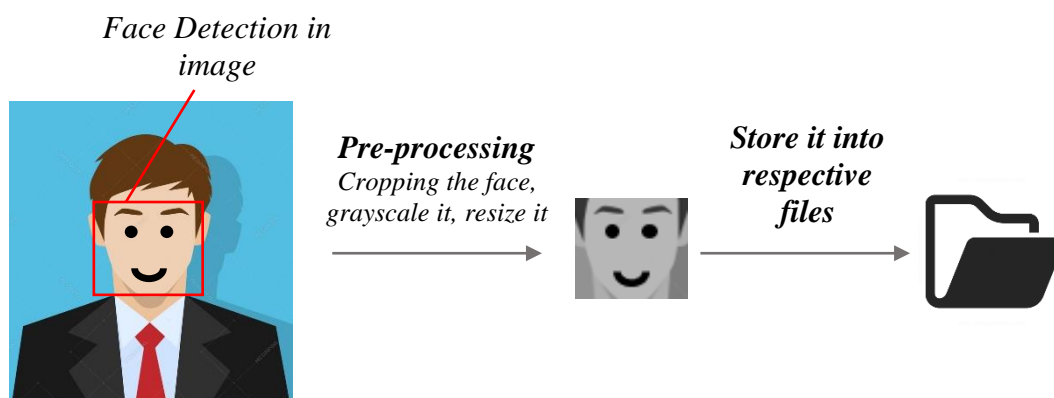
Required software: OpenCV 3.4, Python 3

Required packages: tkinter package - To provide user interface

picamera module – To interact with the raspberry pi's camera

#### **4.1.2 Methodology**

Before the attendance management system can work, there are a set of data needed to be inputted into the system which essentially consist of the individual's basic information which is their ID and their faces. The first procedure of *portrait acquisition* can be done by using the Raspberry Pi Camera to capture the faces of the individual. In this process the system will first detect the presence of a face in the captured image, if there are no face detected, the system will prompt the user to capture their face again until it meets certain number of portraits which will be 10 required portraits in this project for each student. The decision of storing only 10 portrait per student is due to the consideration of the limited storage space in the raspberry pi because the total amount of students in the university is considered heavy. Then, the images will undergo several *pre-processing* procedures to obtain a grayscale image and cropped faces of equal sized images because those are the pre-requisites of using the *EigenFaces* Recognizer. Both of the processes mentioned above can be represented in the diagram below.



**Figure 4.1.1: Image Acquisition and Pre-processing procedures**

After the images are being processed, they are stored into a file in a hierarchy manner. In this project, all the faces will be stored in a hierarchy manner under the ‘*database*’ folder. When expanding through the database folder, there will consist of many sub-folders which each of them will represent an individual where a series of face portrait belonging to the same individual will be stored in that particular sub-folder. The sub-folders that represent each individual will be named upon the ID no. of that individual which is unique for every single individual in the institution. The whole process of image retrieval, pre-processing, storing mechanism is done by the script named *create\_database.py*.

```

database
|----- 1503979
|           |-----0.pgm
|           |-----1.pgm
|           |-----2.pgm
|           |           ....
|           |-----9.pgm
|----- 1543268
|           |-----0.pgm
|           |-----1.pgm
|           |           ....
|           |-----9.pgm
|           ....
|           ....
|----- 1613765
|----- 0.pgm
|           ....
|----- 9.pgm

```

**Figure 4.1.2: Hierarchy manner of the face database**

After a successful retrieval of facial images into the respective folder, **a CSV file is created** to aid the next process of pumping the faces into the recognizer for the training process. The creation of the CSV file will be done based on a script named *create\_csv.py*. In this project, the content of CSV file will look like the following format:

<b>database</b>	<b>/1503979</b>	<b>/0.pgm</b>	<b>;0</b>
<i>Base Folder</i>	<i>Sub-folder</i> <small>Represents each individual in one sub-folder. Named by their ID.</small>	<i>Image File</i>	<i>Label</i>

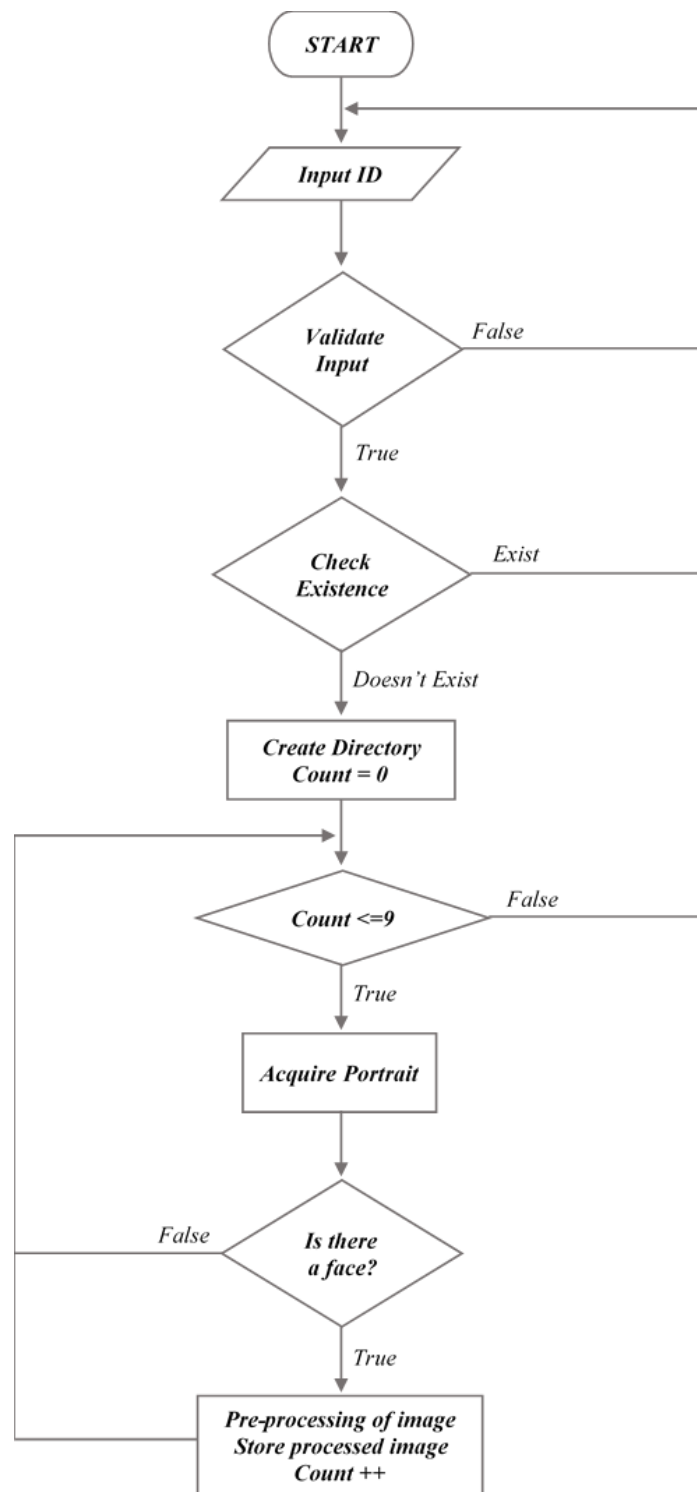
**Figure 4.1.3: Structure of the content in the csv file**

After having sufficient images in the database, those images will then be inserted into a **training mechanism**. There are generally 3 different types of training mechanism provided in OpenCV 3.4 which are *EigenFaces*, *FisherFaces*, and *Local Binary Patterns Histograms (LBPH)*. The recognizer that will be focused in this project will be the *EigenFaces* recognizer. The concept behind EigenFaces is simple – it recognizes a particular face by catching the maximum deviation in a face and then turning those identified variations into information to be compared when a new face arrives. In the training process, the csv file will be read to provide the path to all of the images where those images and labels will be loaded into a list variable. Then, the list will be passed into the training function where the training process will take a measurable time to run. The larger the face database, the longer the time will be needed to train those images. In this project there are 40 subjects, which will provide 400 images to be trained that takes approximately 50 seconds for the training session. Imagine if the system holds 5000 students there will be 50,000 images in total to be trained which might takes up roughly 1.30 hours to complete the training process. Therefore, to maintain the efficiency of the system, a **.yml file will be saved** after the training process so that during the recognition process, only the .yml file will be loaded instead of repeating the whole training process.

#### **4.1.3 Flow Chart of the image acquisition process**

The development of the face database is an important phase before any facial recognizing process can be carried out. It acts as a library to compare against with whenever the system wanted to identify a person. In the image retrieval process, the system will first prompt for an input from the user to enter their ID number. The system will then validate the entered input and then check for duplication in the system. In order to proceed, the entered input must contain only 7 digits of number. Apart from that, the ID inputted have to be a non-registered ID to ensure no duplication. After that, a directory is created for each individual where their portraits will be stored inside of it. It is a compulsory to store 10 portraits per person in the file. After the acquisition of image is done, the images undergo a pre-processing before storing it into the respective folder.





**Figure 4.1.4: Flow Chart of the image retrieval process**

The above flowchart is only the program flow for the image acquisition process which describes the program flow for the script *create\_database.py*. There are two more python scripts that responsible for the remaining execution which will be explained in the next sub-section.

#### 4.1.4 Files included

There are in total 3 python scripts, 1 bash file, 1 csv file, 1 yml file and 1 folder needed in the face database creation part. 3 of the python scripts will be included in the bash file for 2 reasons. Firstly, it is to provide convenience to the user whenever they wanted to register images for new students. By running those script in bash, the user can avoid some ambiguous steps such as tuning to the cv environment before the script is being able to run from terminal because the bash file will handle the environment tuning. Secondly, the csv file creation and also the training process can be automated after the images are added. This function is crucial as it forces the yml file to be up to date before any recognition process is done just in case the user mistakenly missed this step.

Python scripts: *create\_database.py*, *create\_csv.py*, *train\_faces.py*

Bash file: *prepare\_gallery.sh* (stored in /usr/local/bin/)

CSV file: *faceDirectory.csv* (stored in the database folder)

Yml file: *trained\_data.yml*

Folder name: *database*

*\*All the files mentioned can be found in the cd*

The following diagram will explain the relationships between these files.

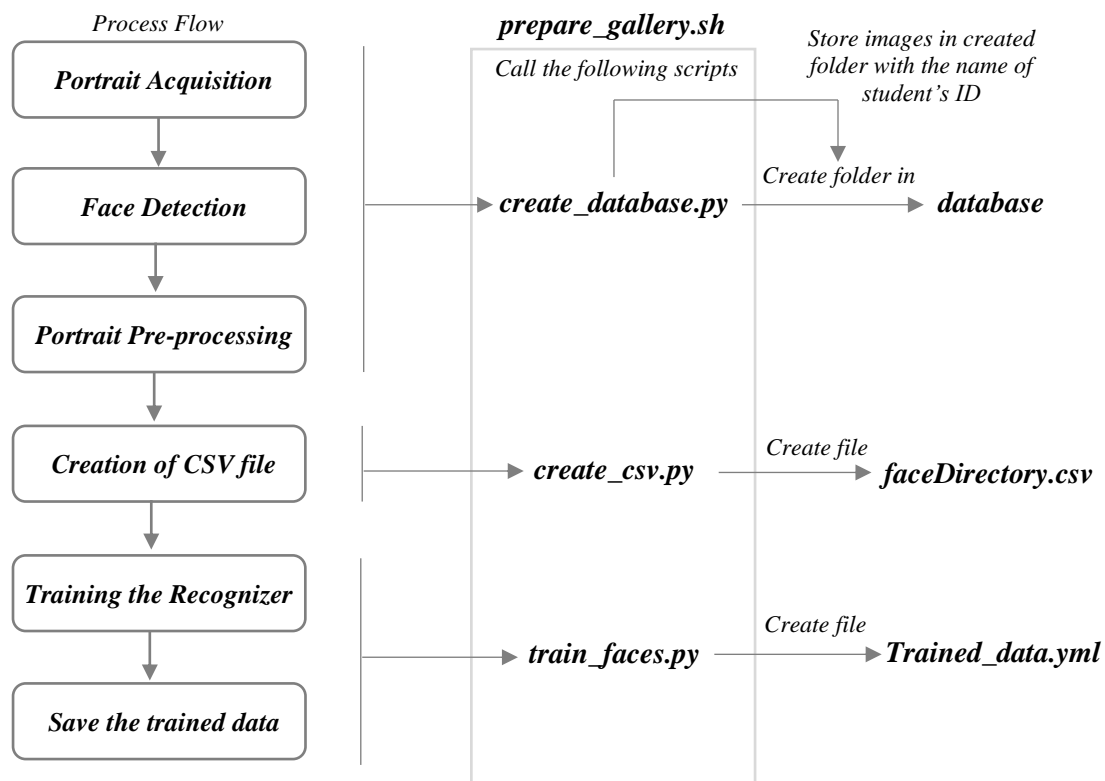
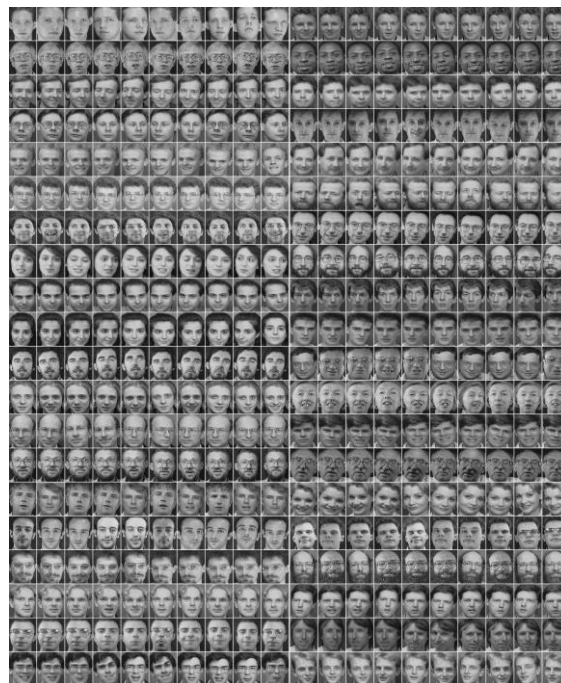


Figure 4.1.5: The relationships between the files during the face database creation

Here, the sample images of the faces used in this project are the faces downloaded from the internet. This is due the insufficient reluctance of real person to join the testing community. Thus, this problem is solved by using the downloaded faces to form the face database while adding some of the real images captured from the system's pi camera to prepare the database for verification test. The specific face database being used is named AT&T Facedatabase which can be obtained from <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>. In the provided database, there are 40 different subjects each with 10 sets of their very own portrait captured at a different timeline. This database consists of subjects with different expressions such as a smiling or frowning and etc. Apart from that, some of the subjects have different variation on their face such as wearing glasses in some of their portraits. Therefore, this database is relatively applicable to this system since the required amount of sample portrait for each person by this system is also similar as the provided amount of portrait by the downloaded database.



**Figure 4.1.6: Images provided by the AT&T Facedatabase (AT&T Laboratories Cambridge, 2002)**



**Figure 4.1.7: Sample images taken through the system's pi camera**

#### 4.1.5 Verification Test Plan

The system uses dialog box interface (supported by tkinter package) to communicate with the user which is designed to have as minimum input as possible from the user to reduce the need of heavy validation. The designed interface is also user friendly because buttons are provided to aid the process flow of the system. In this section, there are only 2 times of input prompt from the system where the first is for the input ID and the portrait of the new user.

No.	Testcase Name	Description	Test Type	Test Values	Test Output	Status (pass/fail)
1	Test Input ID	This test is performed to check whether the ID typed by the user is valid or not.	Input value more than 7 digits.	'12345678'	An error box appeared stating that 'Length of ID number can only be 7 digits!'	pass
2			Input value include alphabetic character.	'123abc7'	An error box appeared stating that 'Invalid ID number entered'	pass
3			Input ID that is already existed.	'1600001'	An error box appeared stating that 'ID number existed'	pass
4			Input value that have 7 numerical digits which is not existed in the database.	'1503979'	A notice box appeared stating that 'Directory is created successfully'	pass
5	Test Image Capture	This test performed is to ensure the captured image contains a face which is detected by the system before storing it into the face database	Captured image has no face	Capture a blank image	An error box appeared stating that 'No face detected!'	pass
6			Captured image is motion blur.	Move while the image is taken.	An error box appeared stating that 'No face detected!'	pass
7			The person is standing too close during the capturing process.	Capture the portraits 5cm away from the pi camera.	An error box appeared stating that 'No face detected!'	pass
8			Captured image has a face.	Capture an image with a person facing the pi camera	A notice box appeared stating that '1 portrait saved!'	pass

**Table 4.1.1: Verification test plan for the creation of face database**

#### 4.1.6 Implementation and Testing

The followings are the implementation test and its results produced during the creation of face database in the raspberry pi.

Interface of dialog box when the *prepare\_gallery.sh* bash file is executed:

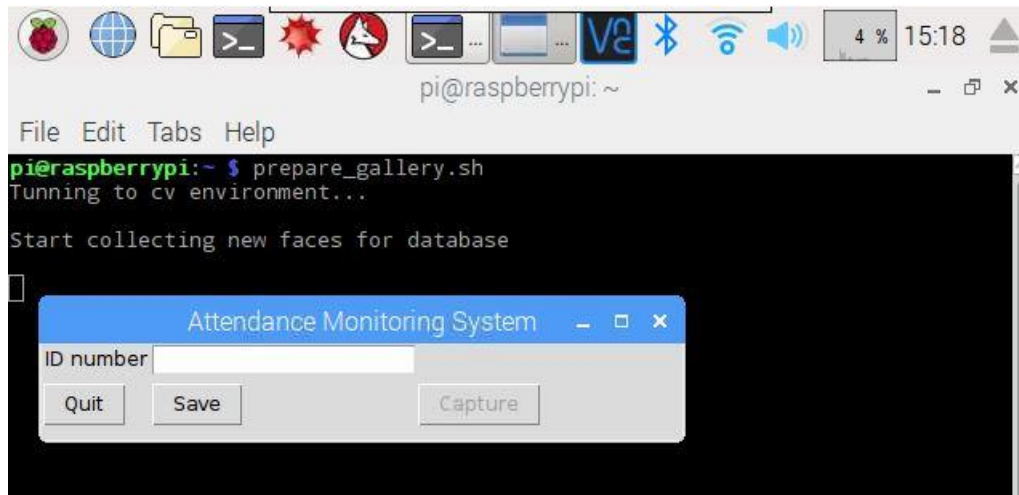


Figure 4.1.8: Interface displayed when *prepare\_gallery.sh* is executed

Invalid input ID (Existed ID) test:

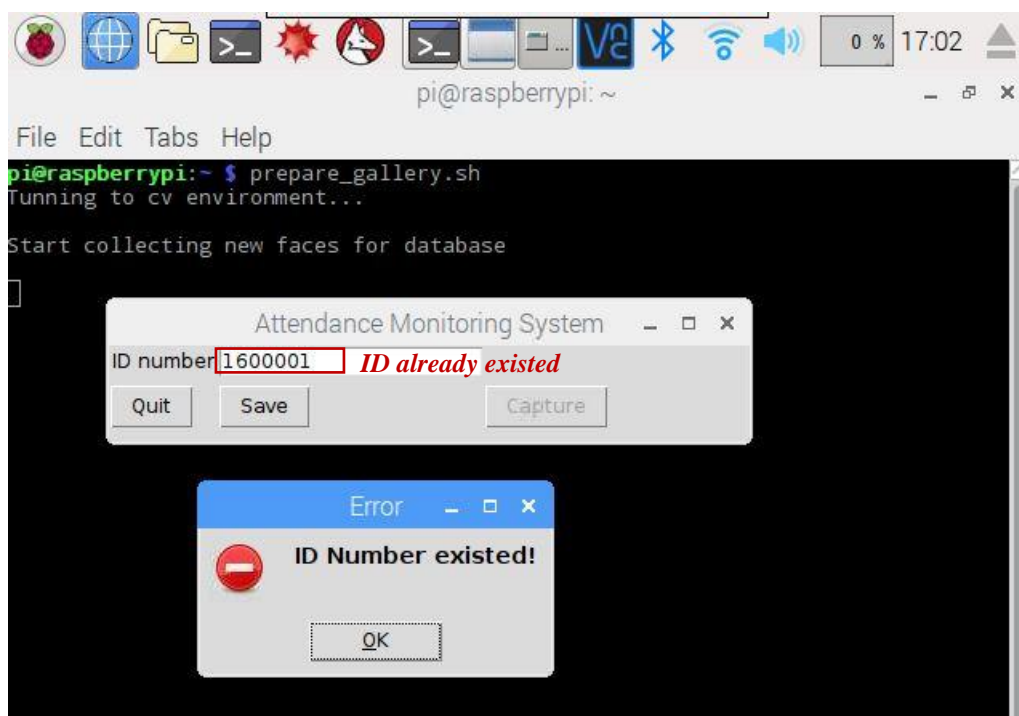


Figure 4.1.9: Test result for invalid ID (1<sup>st</sup> attempt)

### Invalid input ID (ID consist alphabet) test:

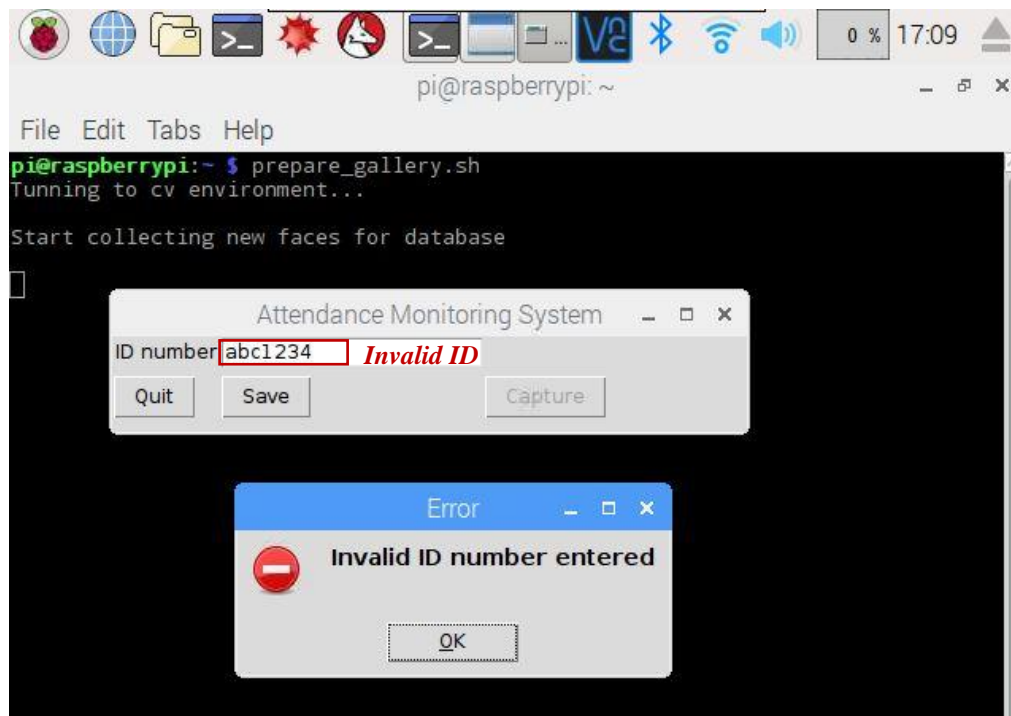


Figure 4.1.10: Test result for invalid ID (2<sup>nd</sup> attempt)

### Invalid input ID (ID longer than 7) test:

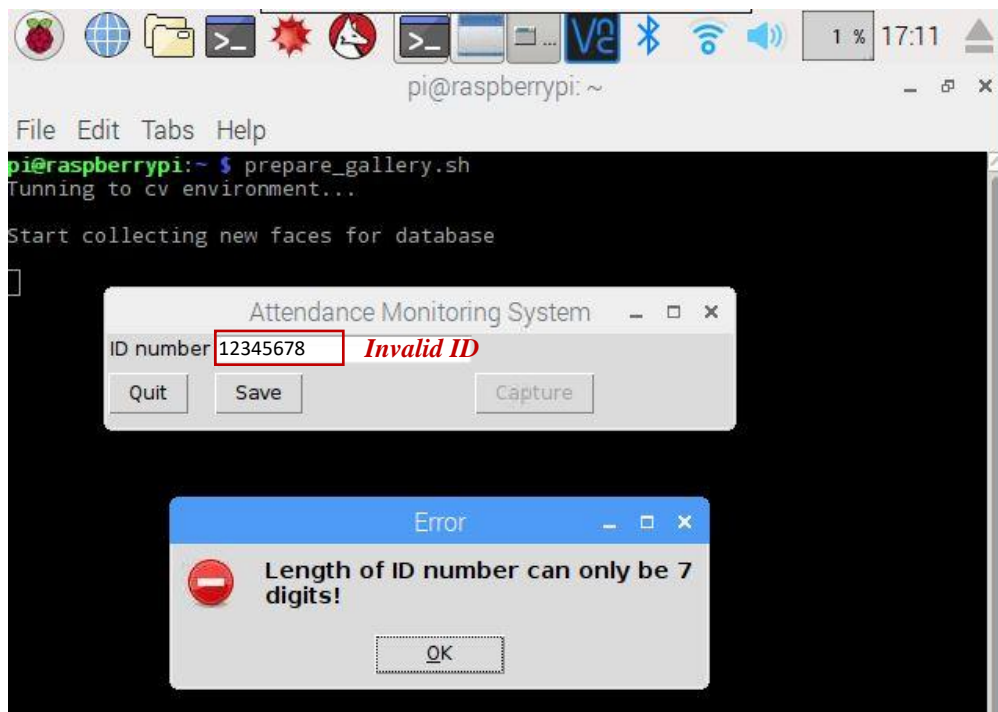


Figure 4.1.11: Test result for invalid ID (3<sup>rd</sup> attempt)



### Valid input ID (ID valid and not exist) test:

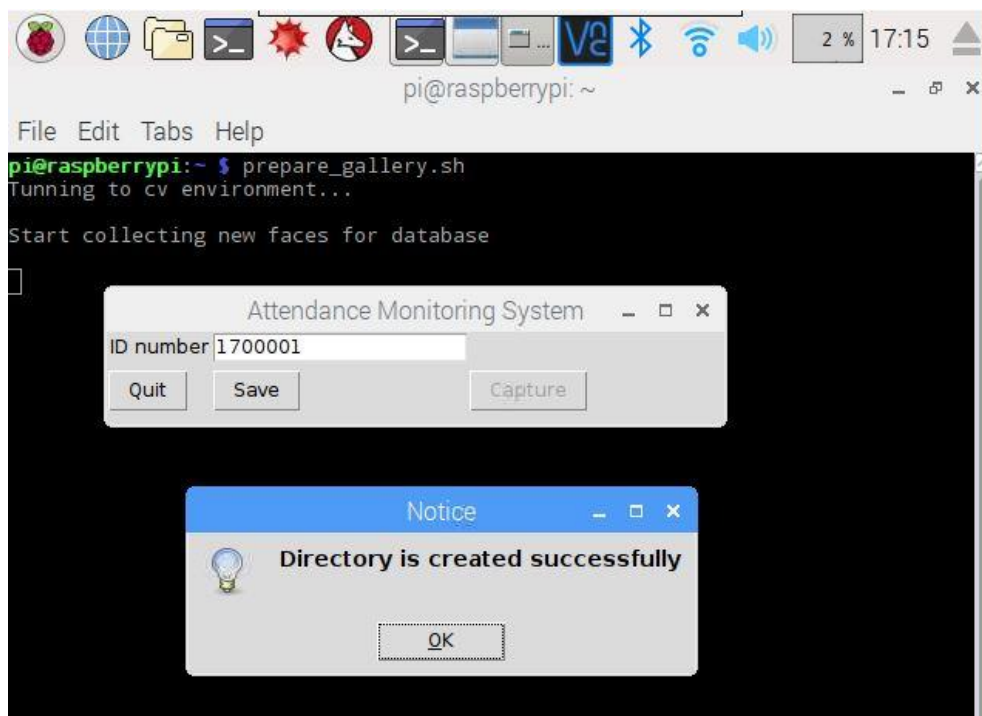


Figure 4.1.12: Test result for valid ID

### Portrait Capture (No Face) test:

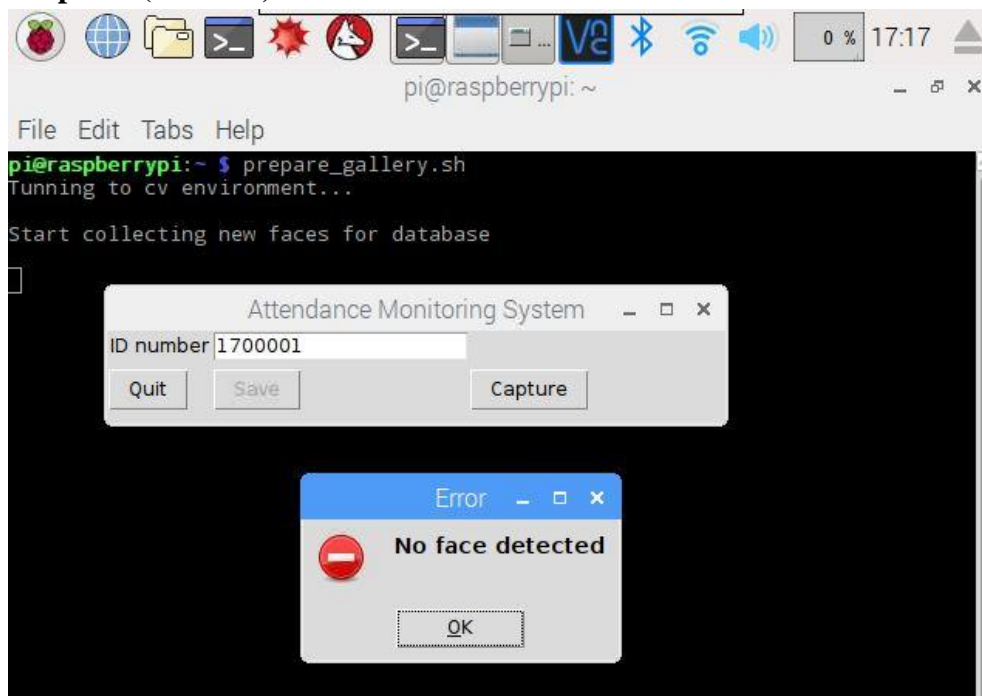
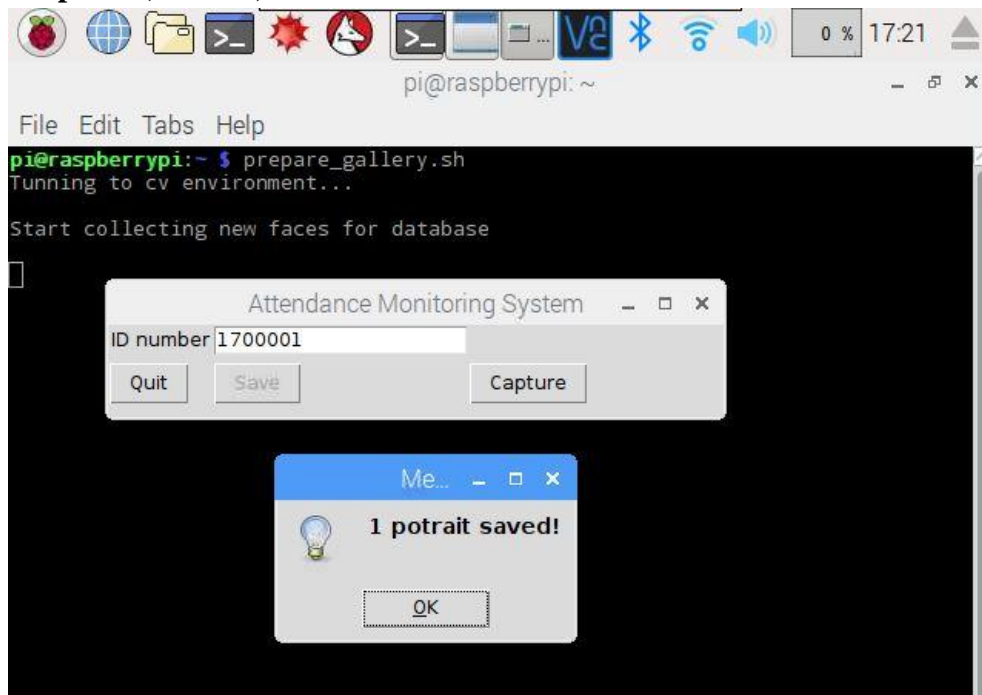


Figure 4.1.13: Test result for invalid face capture

### Portrait Capture (Normal) test:



**Figure 4.1.14: Test result for valid face capture**



## 4.2 The process of attendance taking

### 4.2.1 Specific requirements

There will be several requirements before the process of attendance taking can be initiated. The below are the required software or packages needed to accomplish this objective.

Required software: OpenCV 3.4

Python 3

PHP5-fpm – To display webpage of the hosted website by the pi

PHPMyAdmin – To aid database configuration

MySQL – To create an attendance management system database

Nginx – To allow the raspberry pi to host website

Wordpress – To aid the webpage design

Required packages: picamera module – To interact with the raspberry pi's camera

pymysql module – To be able to interact with MySQL with python

### 4.2.2 Configuring system requirements

Before proceeding to the methodology discussion, there are quite a number of installations and configurations needed to be done in order for the system to work. In this section, the procedure of the entire linkage configurations between the required software packages such as PHP 5, PHPMyAdmin, MySQL, Nginx and Wordpress will be focused. A brief explanation and reasons of selection of those software packages will also be discussed later.

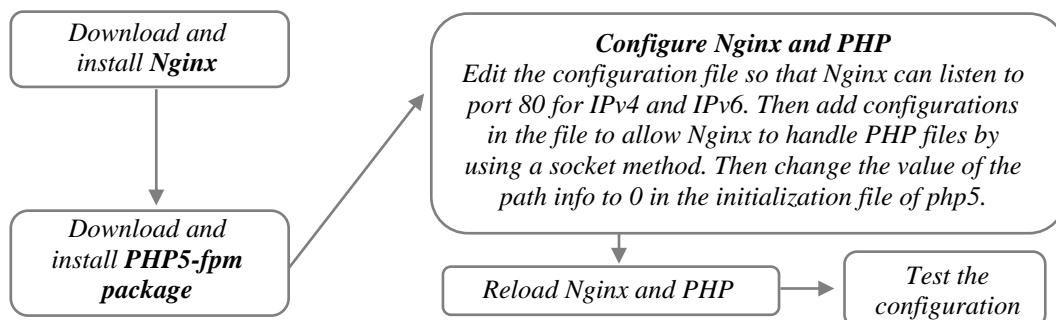
In this project, the process of attendance taking will be initiated through a webpage hosted by the raspberry pi. In order to achieve that, the raspberry pi will firstly need a HTTP Web Server to allow itself to host websites. The planned selection of the web server during FYP1 was the Apache web server. However, in this project, the web server being used is Nginx. This is due to the easier configuration of Nginx and also its ability to manage heavy loads more effectively than Apache. For raspberry pi which is a relatively resource limited device, it is more suitable to use Nginx as it is better at handling the resource distribution making it more effective. However, the only drawback while considering Nginx was its incapability to handle dynamic content. Although in this project, the ability to process dynamic content is very much needed at one of the design phase, but, these requirements can be easily overcome by sending the request to another software.



**Figure 4.2.1: How Nginx works with PHP**

Next, the raspberry pi also need a platform for web development where PHP 5 will be needed to act as server scripting language to communicate with webpages hosted by the raspberry pi. By implementing it, Nginx can be configured to serve content through PHP where the webpage content will be stored in the raspberry pi. Since Nginx is used in this project, PHP5-fpm is needed to run as FastCGI interface with Nginx. As for the webpage development, Wordpress is one of the most suited software to deal with webpage designs because firstly, it is supported by raspberry pi. Secondly, the web editing process can be done on a virtual host which is remotely accessed through the platform decorated by Nginx and PHP, this made the development process much more convenient. Thirdly, Wordpress provides fabulous webpage interface with numerous plugins which can be installed whenever needed to aid the building of the webpage interface. Also, Wordpress allows drag and drops feature during designing the webpage. At last but not least, the database used by Wordpress can be easily linked to a PHPMYAdmin account to view and perform various amendments to the data. While in PHPMYAdmin, databases can be created and managed easily. Thus, the attendance management system database can be created and stored using PHPMYAdmin and then by using Wordpress, its data can be easily retrieved and displayed in the intended manner in the website. Finally, MySQL is needed to manage the database stored in the raspberry pi. The reason why MYSQL is chosen to be used is because of its high software maturity, in simple words, it is widely used and accepted. Other than that, there are plenty of forums available on the internet making it a very practical choice whenever a database management system is ever needed.

#### **Configuration steps (Step 1: Installing Nginx and PHP):**

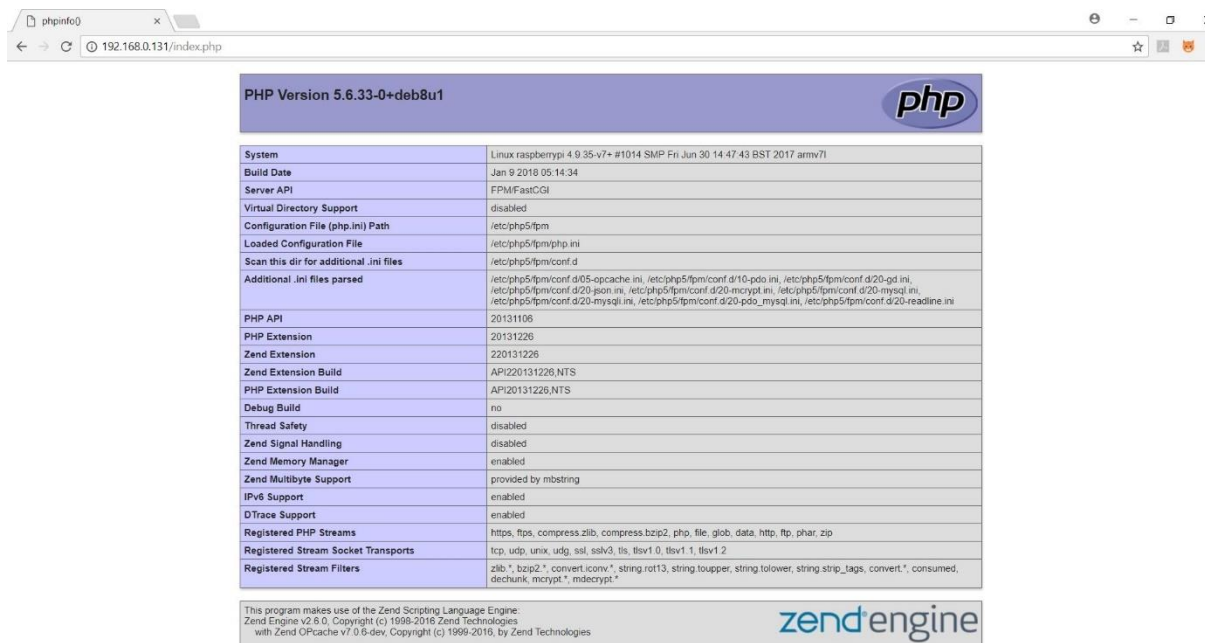


**Figure 4.2.2: Configuration steps for Nginx and PHP**

The configurations can be tested by entering the IP address of the raspberry pi to the web browser on any device that is connected to the same network as the raspberry pi. Throughout this project, the localhost address being used is **192.168.43.232**. Apart from that, the root file where all the PHP files and index pages are stored at this path **/var/www/html/**. To test the above configurations the *index.php* file should be placed under the root file which is in the html folder.

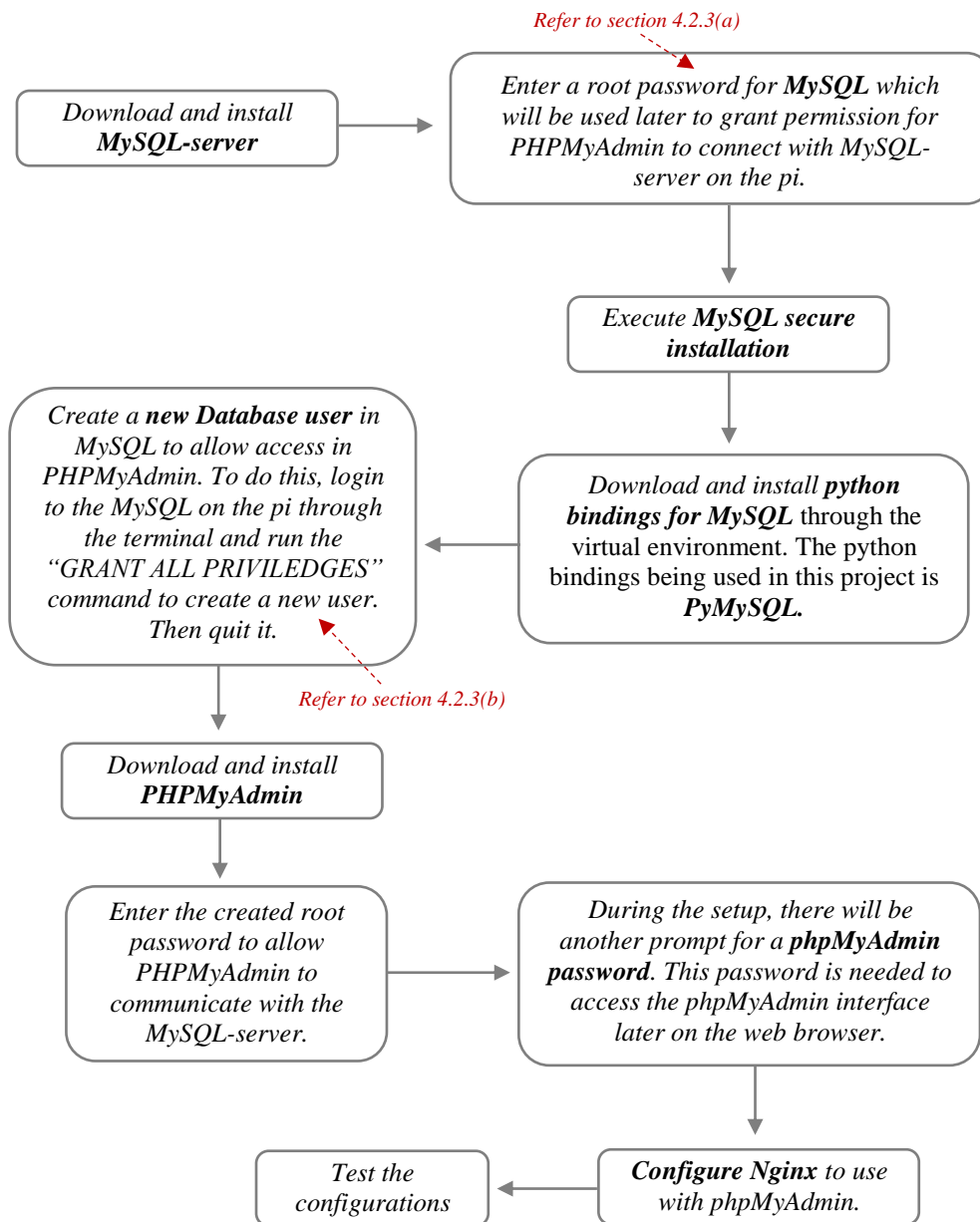


**Figure 4.2.3: Test page for the installation of Nginx**



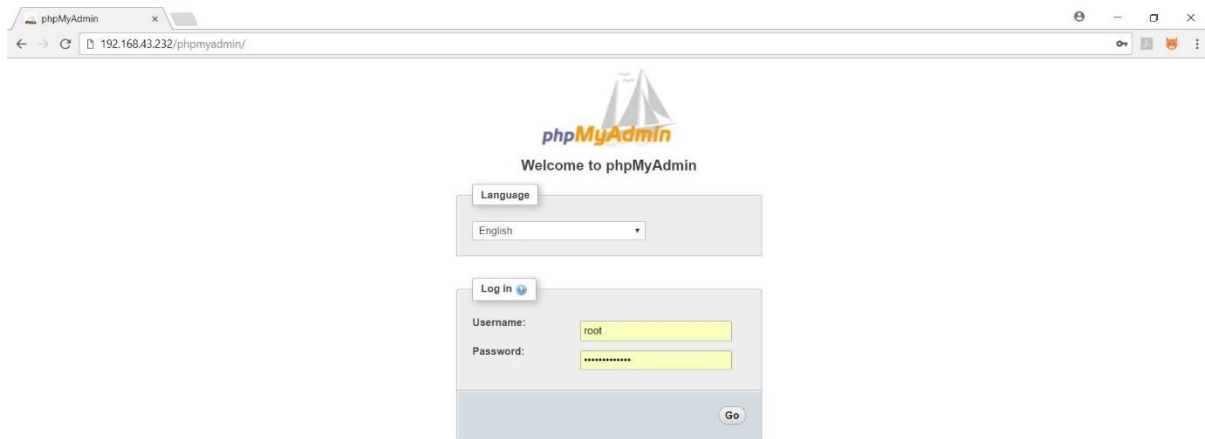
**Figure 4.2.4: Test page for the configurations of Nginx with PHP**

## Configuration steps (Step 2: Installing MySQL and PHPMyAdmin):



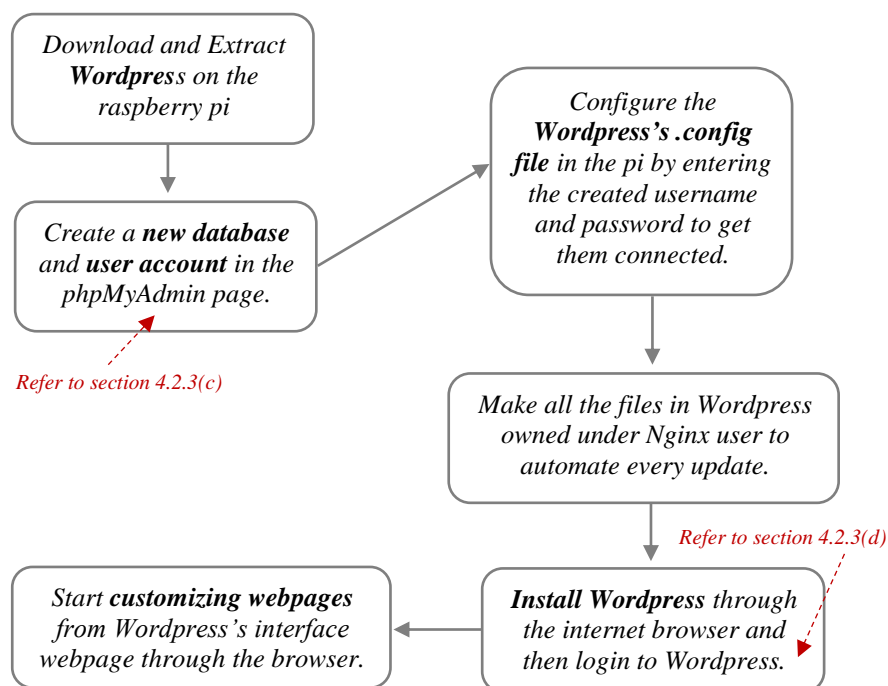
**Figure 4.2.5: Configuration steps for MySQL and PHPMyAdmin**

After the above steps are done, the interface of phpMyAdmin is now accessible from the web browser again, from any devices which is connected in the same network as the raspberry pi. PhpMyAdmin is installed in the root file (`/var/www/html/`), thus, during accessing its interface on the browser, the path **`http://192.168.43.232/phpmyadmin`** is entered to reach the login interface of phpMyAdmin. During the login, the username is set to 'root' and the password prompted is second one that is created during the setup. All the configured usernames and passwords for this project will be summarized in the upcoming sub-section.



**Figure 4.2.6: Login interface for phpMyAdmin**

### Configuration steps (Step 3: Installing and setting up Wordpress with phpMyAdmin)



**Figure 4.2.7: Configuration steps for Wordpress**

By connecting Wordpress and phpMyAdmin, the webpage information is held in phpMyAdmin. It can be easily accessed by logging into the phpMyAdmin page. In this project, Wordpress is installed to the root path `/var/www/html/wordpress` where an additional folder named **Wordpress** is created in the root path to hold all of its information. Thus, Wordpress can be accessed from the browser by typing **`http://192.168.43.232/wordpress`**. There are two phases in the development of the website. One is where the website already has

pages which are publicized, while the other is still under construction with none of the pages being publicized. In the first condition, the webpage can be accessed by typing the above-mentioned web path. This is to allow unauthorised user to view the webpage, while authorized user can access the customization page by logging in from the webpage. However, in the second condition where there are no displayable webpages, in order to access the Wordpress's dashboard to do customization, the following web path ***http://192.168.43.232/wordpress/wp-admin*** is needed instead. This path will direct the user to the login page of Wordpress where authorized admins can enter the password to customize the webpage.

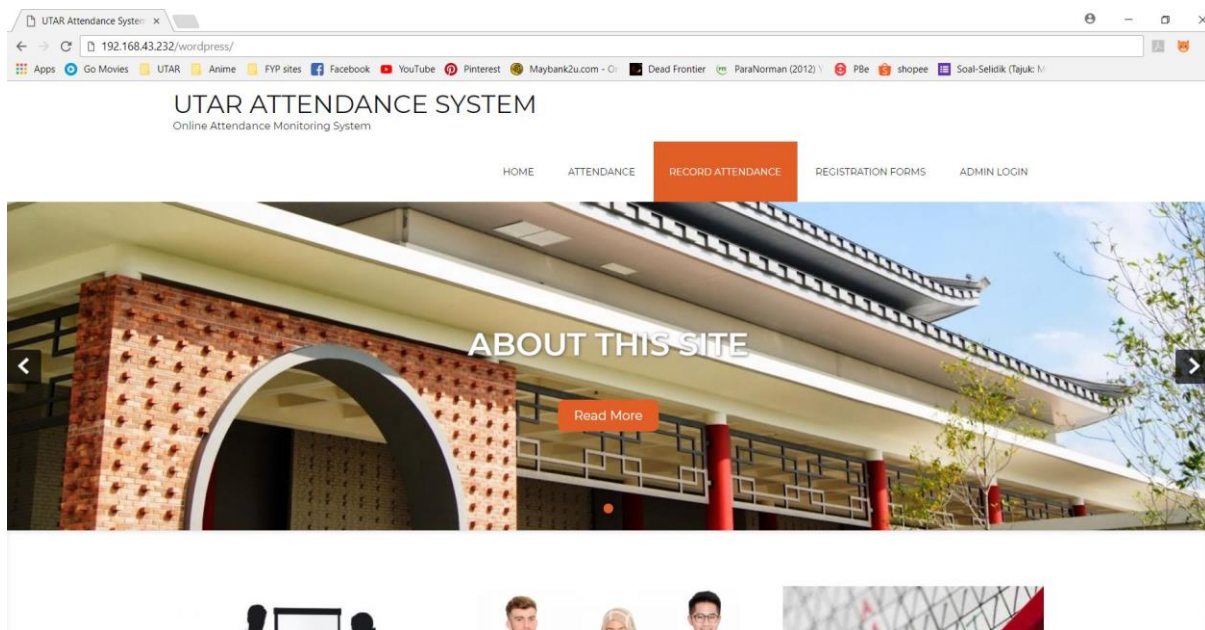


Figure 4.2.8: Completed design of the webpage

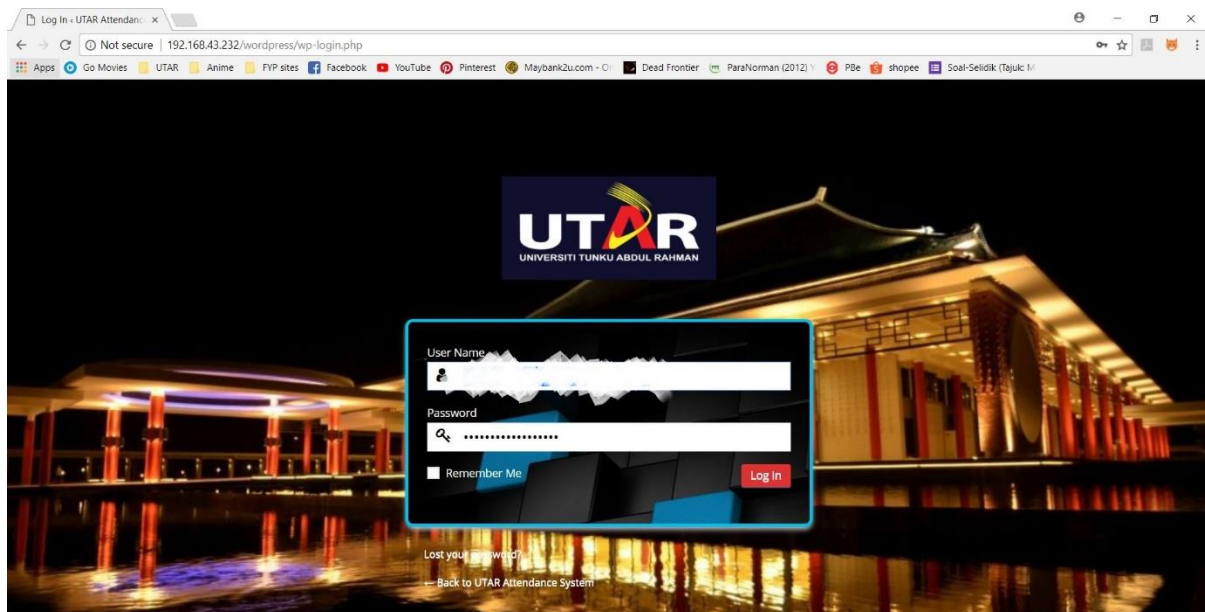


Figure 4.2.9: Customization login page



### 4.2.3 Configured Information

In this project, many usernames and password are created during the configuration process, the below are the configured information.

**(a) MySQL root user**

Password: mysql123abc

**(b) New user created to login to PHPMyAdmin**

Username: root

Password: phpmyadmin123abc

**(c) New user account to store Wordpress's database**

Username: wordpress

Host: localhost

Password: 5julKmzJZe9pxtqx

**(d) Wordpress root user**

Email: tanshujing2004@yahoo.com

Password: 5ul0CVfYJZ!RwUcnlx

**(e) Wordpress staff user**

Email: testing@yahoo.com

Password: q0grkfYjIPf@cf0Tot9uHJ&r

The followings are the summarized file path and web path for this project.

**Root file in raspberry pi:** /var/www/html/

**PhpMyAdmin file:** /var/www/html/phpmyadmin

**Wordpress file:** /var/www/html/wordpress

**Access path to phpmyadmin web interface:** <http://192.168.43.232/phpmyadmin>

**Access path to wordpress web interface:** <http://192.168.43.232/wordpress>

**Access path to wordpress login page:** <http://192.168.43.232/wordpress/wp-admin>

\*Note: These web paths are only accessible by devices that are connected to the same network as the raspberry pi. Meaning the website can only be accessed locally. In order for them to be accessed outside the network, port forwarding mechanism can be done to allow this. However, in this project, there are no administrative power to use the university's network.

#### 4.2.4 Methodology

After the all the configurations are done, the website now is available for use and thus users such as the lecturer can start taking down attendance simply by just accessing the “Record Attendance” tab in the **Attendance Management System website**. However, in order for user to access this tab, he/she has to be authorized to do so by logging in into the system as a “contributor” role (will be further explained in Chapter 5). After logging into the respective tab, the user will be prompted to select the current timetable ID from a table and also select the current date to start the attendance taking process. When the user selected the “submit” button a *.php* file called *student\_attendance.php* will be called which a python script named *record\_attendance.py* will be executed from the *.php* file. In this *student\_attendance.php* file, two variables which are the timetable ID and the current date (passed by POST method from the html) will be passed into the python script upon calling it.

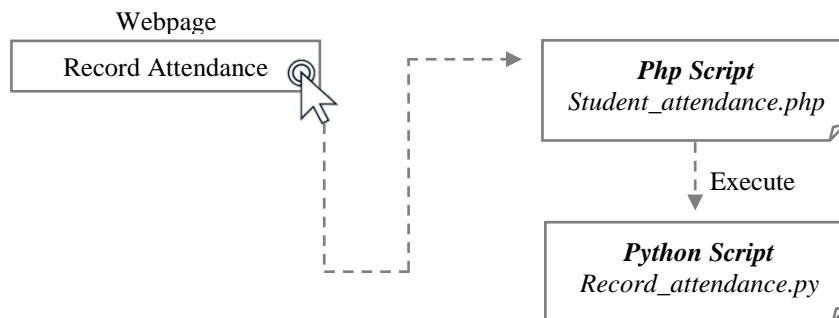


Figure 4.2.10: Process flow of attendance taking through the webpage

The facial-recognition process is started when the python script is executed. When the script is first executed, variables are initialized and most importantly, the *.yml* file created previously during the recognizer training process is now loaded into the system. After the system is loaded with the intended data, the process of capturing student faces is initiated. Firstly, all the students are advised to stand approximately 20cm away from the attendance taking device (packaged pi camera + raspberry pi) with their face remained parallel to the pi camera. Usually this can be done by ensuring the eye position is same level as the pi camera.

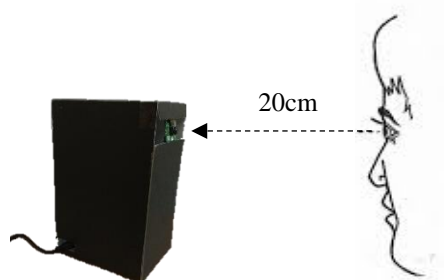
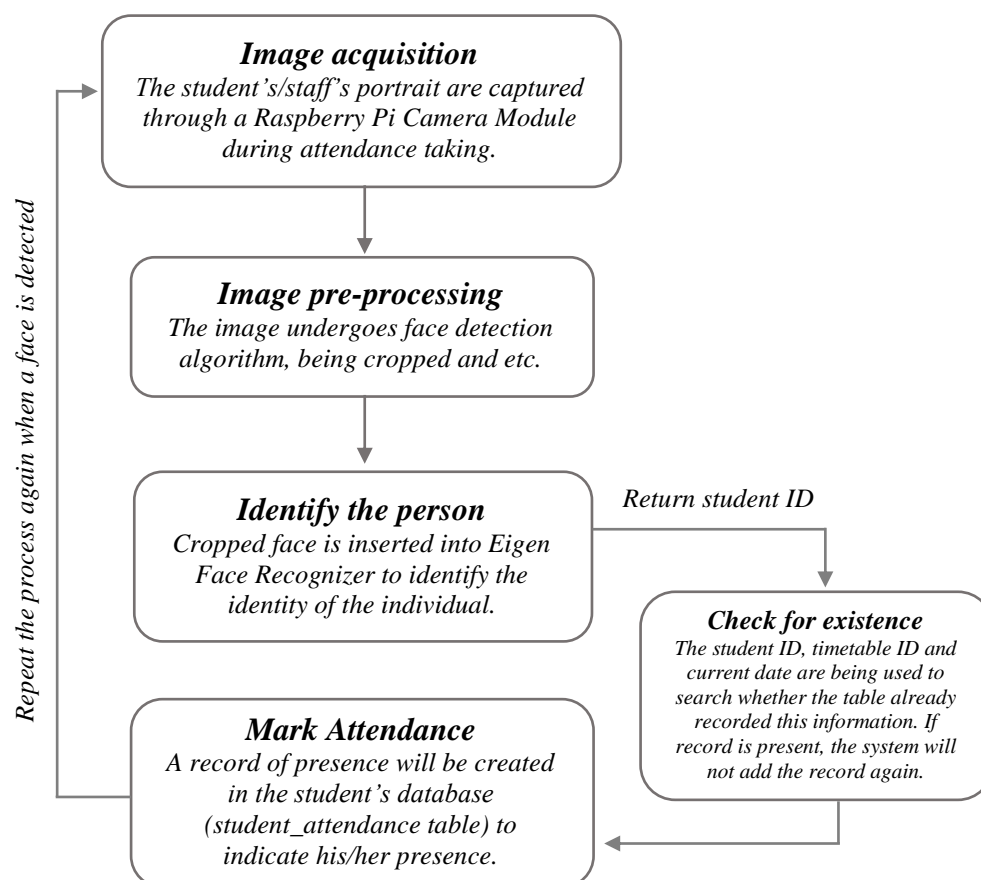


Figure 4.2.11: Method of acquiring the portrait



During the capturing process, students are advised to remain still for approximately 3s for a higher rate of successful recognition. After the portrait is successfully acquired from the pi camera, the system will start to do face detection to ensure a face is present before proceeding to the pre-processing process. During the **pre-processing** process, images are cropped and converted to grayscale. Then, it is pumped into the trained recognizer to identify the identity of the captured face. If the face captured is valid meaning that the detected face is present in the student database, the system will return a label corresponding to the identified person in the face database. Then, by using the label, the student's ID number can be retrieved since in the face database, student's portraits are distinguished and grouped using their ID number. After that, the **individual's attendance will be recorded** into the attendance database which is specifically the *student\_attendance* table. However, before adding the record into the table, the student ID and the related information will be checked against the table to ensure there are no redundant attendance added into the table. Apart from that, the Green LED will be triggered when a capture is valid, else the Red LED will be triggered. Finally, the attendance information in the database can be viewed through the Attendance Management System website at the "Attendance" tab.



**Figure 4.2.12: Process flow of the facial recognition procedure**

#### 4.2.5 Flow Chart of the attendance recording procedure

Since the face database are downloaded from the internet, thus to test whether the face recognizer is working, facial images were manually pumped into the code to check for identification. However, the following flow chart will explain the intended process flow of the face recognition process.

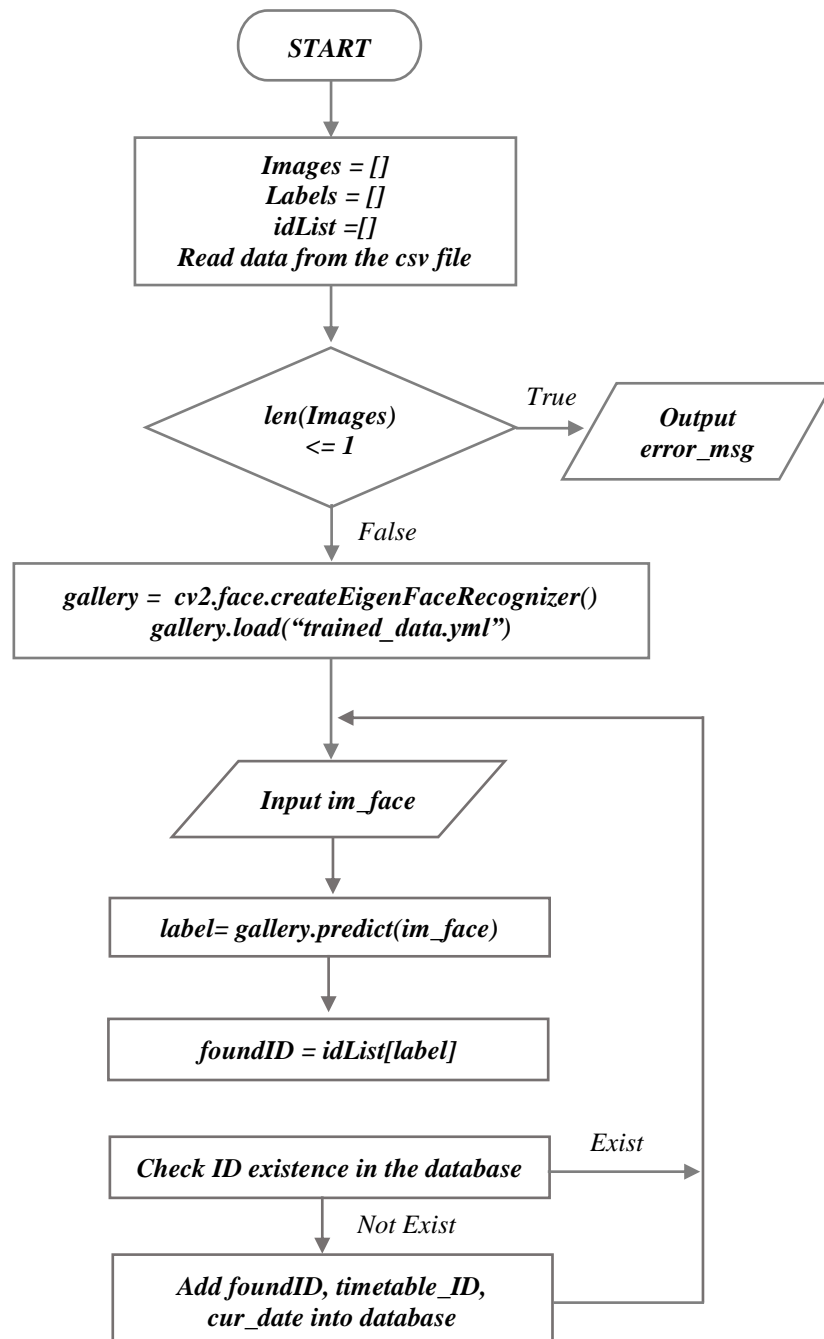


Figure 4.2.13: Flow Chart of attendance recording procedure

#### 4.2.6 Files Included

In this section, there are only 1 php file, 1 bash file, 1 python script and 1 yml file used to execute the attendance taking process. The followings are the related files and their storage path.

PHP file: ***student\_attendance.php*** (stored in /var/www/html/wordpress)

Bash file: ***runcv\_env.sh*** (stored in /usr/local/bin)

Python script: ***record\_attendance.py***

Yml file: ***trained\_data.yml***

*\*All the files mentioned can be found in the cd*

The php file is the first file that will be called after the user clicked the submit button to start the attendance taking process. Then, in the php file, a bash file will be executed where the values entered by the user will also be passed in into the bash file. The bash file will then again tune the environment to cv environment before executing the python script. During the execution of the python script, the values passed into the bash file will also be passed into it. The following diagram explains the relationship between these files.

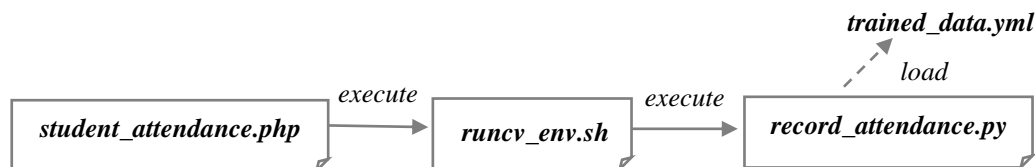


Figure 4.2.14: Relationships between files for the attendance taking process

#### 4.2.7 Verification Test Plan

During the attendance taking process which is conducted on the webpage, the user (lecturer) is required to enter the current timetable ID and date before the process can be initiated. Thus, the validation is done through the *php* file where it will produce an error page if the inputs entered are invalid.

No.	Testcase Name	Description	Test Type	Test Values	Test Output	Status (pass/fail)
1	Test input Timetable ID	This test is performed to check whether the Timetable entered is valid or not	Input value have more than 7 digits.	'12345678'	Invalid entered input! Timetable ID must be 7 digits long.	pass
2			Input value have less than 7 digits.	'123456'	Invalid entered input! Timetable ID must be 7 digits long.	pass

3			Input value doesn't exist in the database.	'9999999'	Timetable ID 9999999 doesn't exist in the database.	pass
4			Input value that have 7 numerical digits which existed in the database.	'1000001'	No error page.	pass
5	Test input date	This test is performed to check whether the user select a date before starting the attendance taking process.	Doesn't select any date.	-	Error! Please select a date before submitting.	pass
6			Selected a date	'9/4/2018'	No error page.	pass

Table 4.2.1: Verification test plan of the attendance taking process

#### 4.2.8 Implementation and Testing

The followings are the implementation test and its results produced during the attendance taking process.

##### Interface of the “Record Attendance” page

RECORD ATTENDANCE

Timetable ID	Subject Name	Day	Start Time	End Time	Session
1000001	BASIC MICROPROCESSOR AND INTERFACING	Tuesday	1100	1300	L1
1233211	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES	Saturday	1500	1700	P2
2221111	ALGORITHMS ANALYSIS	Friday	0900	1200	T14
6666666	COMMUNITY PROJECT	Wednesday	0800	1000	L3
9812523	CIRCUIT THEORY	Wednesday	1600	1730	P6

Pages: 1 2 Next Last 1 of 2

Select the timetable ID according to the current session.

Timetable ID:  Date:

Figure 4.2.15: Interface of the Record Attendance Page

<b>Invalid Input of Timetable ID</b>	<p>Select the timetable ID according to the current session.</p> <p>Timetable ID: <input type="text" value="123456"/> Date: <input type="text" value="08/04/2018"/></p> <p><input type="button" value="Submit"/> <i>Less than 7 digits</i></p>
<b>Results</b>	<p>← → ↻ 192.168.43.232/wordpress/student_attendance.php</p> <p>Invalid entered input! Timetable ID must be 7 digit long.</p> <p><a href="#">Back to previous page</a></p>

**Table 4.2.2: Input validation test towards Timetable ID (1<sup>st</sup> attempt)**

<b>Non-Exist Input of Timetable ID</b>	<p>Select the timetable ID according to the current session.</p> <p>Timetable ID: <input type="text" value="9999999"/> Date: <input type="text" value="08/04/2018"/></p> <p><input type="button" value="Submit"/> <i>ID doesn't exist</i></p>
<b>Results</b>	<p>← → ↻ 192.168.43.232/wordpress/student_attendance.php</p> <p>Timetable ID 9999999 doesn't exist in the database!</p> <p><a href="#">Back to previous page</a></p>

**Table 4.2.3: Input validation test towards Timetable ID (2<sup>nd</sup> attempt)**

<b>Invalid Input of Current Date</b>	<p>Select the timetable ID according to the current session.</p> <p>Timetable ID: <input type="text" value="1000001"/> Date: <input type="text" value="dd/mm/yyyy"/></p> <p><input type="button" value="Submit"/> <i>Unselected Date</i></p>
<b>Results</b>	<p>← → ↻ 192.168.43.232/wordpress/student_attendance.php</p> <p>Error! Please select a date before submitting.</p> <p><a href="#">Back to previous page</a></p>

**Table 4.2.4: Input validation test towards current date**

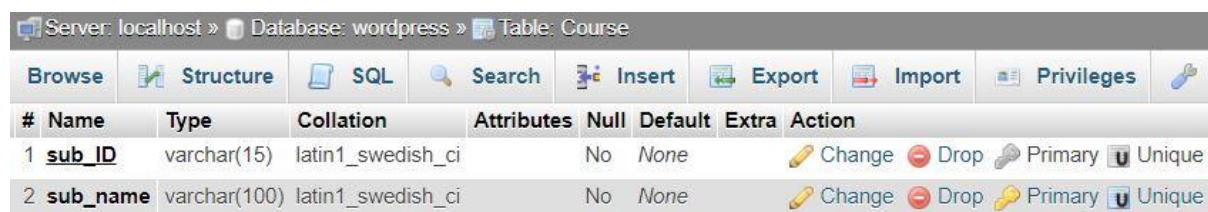
## Chapter 5: Database and Webpage Development

### 5.1 Database Development

#### 5.1.1 Database Design

##### Course Table (Table name: Course)

##### Course Table Design View (in phpMyAdmin)

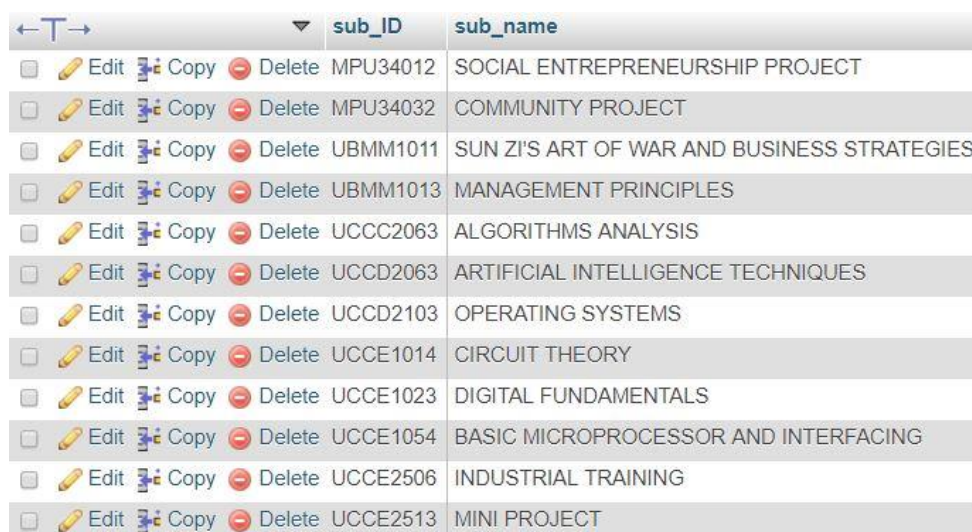


The screenshot shows the phpMyAdmin interface for the 'Course' table in the 'wordpress' database on 'localhost'. The 'Structure' tab is selected. The table has two columns: 'sub\_ID' (varchar(15), latin1\_swedish\_ci, No Null, None Default, Primary, Unique) and 'sub\_name' (varchar(100), latin1\_swedish\_ci, No Null, None Default, Primary, Unique). Action buttons for 'Change', 'Drop', 'Primary', and 'Unique' are visible for each column.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	sub_ID	varchar(15)	latin1_swedish_ci		No	None		Change Drop Primary Unique
2	sub_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique

Figure 5.1.1: Course Table Design View

##### Course Table Datasheet View (in phpMyAdmin)



The screenshot shows the phpMyAdmin interface for the 'Course' table in the 'wordpress' database on 'localhost'. The 'Datasheet' tab is selected. The table displays 12 rows of data, each with a checkbox, 'Edit', 'Copy', and 'Delete' buttons, followed by the 'sub\_ID' and 'sub\_name' columns.

	sub_ID	sub_name
<input type="checkbox"/> Edit Copy Delete	MPU34012	SOCIAL ENTREPRENEURSHIP PROJECT
<input type="checkbox"/> Edit Copy Delete	MPU34032	COMMUNITY PROJECT
<input type="checkbox"/> Edit Copy Delete	UBMM1011	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES
<input type="checkbox"/> Edit Copy Delete	UBMM1013	MANAGEMENT PRINCIPLES
<input type="checkbox"/> Edit Copy Delete	UCCC2063	ALGORITHMS ANALYSIS
<input type="checkbox"/> Edit Copy Delete	UCCD2063	ARTIFICIAL INTELLIGENCE TECHNIQUES
<input type="checkbox"/> Edit Copy Delete	UCCD2103	OPERATING SYSTEMS
<input type="checkbox"/> Edit Copy Delete	UCCE1014	CIRCUIT THEORY
<input type="checkbox"/> Edit Copy Delete	UCCE1023	DIGITAL FUNDAMENTALS
<input type="checkbox"/> Edit Copy Delete	UCCE1054	BASIC MICROPROCESSOR AND INTERFACING
<input type="checkbox"/> Edit Copy Delete	UCCE2506	INDUSTRIAL TRAINING
<input type="checkbox"/> Edit Copy Delete	UCCE2513	MINI PROJECT

Figure 5.1.2: Course Table Datasheet View

Field Name	Description
sub_ID	Identification number for a subject
sub_name	Name of a subject

Table 5.1.1: Field's description in Course table

Field Name	Data Type	Field Length	Nulls	PK/FK	Reference to
sub_ID	Varchar	15	No	PK	-
sub_name	Varchar	100	No	-	-

Table 5.1.2: Course table properties



## Student Table (Table name: Student)

### Student Table Design View (in phpMyAdmin)

Server: localhost » Database: wordpress » Table: Student

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>stu_ID</u>	int(7)			No	None		Change  Drop  Primary  Unique
2	stu_name	varchar(50)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique
3	stu_fac	varchar(10)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique

Figure 5.1.3: Student Table Design View

### Student Table Datasheet View (in phpMyAdmin)

	stu_ID	stu_name	stu_fac
Edit  Copy  Delete	1503979	Tan Shu Jing	FICT
Edit  Copy  Delete	1513344	LOKE BEI AN	FBF
Edit  Copy  Delete	1545767	Wong Mae Kae	FBF
Edit  Copy  Delete	1607123	MA ZI QI	FMHS
Edit  Copy  Delete	1703657	Tan Shu Ting	CFS
Edit  Copy  Delete	9999999	Baby Fat Fat	FAS

Figure 5.1.4: Student Table Datasheet View

Field Name	Description
stu_ID	Identification number for a student
stu_name	Name of a student
stu_fac	Faculty of the student

Table 5.1.3: Field's description in Student table

Field Name	Data Type	Field Length	Nulls	PK/FK	Reference to
stu_ID	integer	7	No	PK	-
stu_name	Varchar	50	No	-	-
stu_fac	Varchar	10	No	-	-

Table 5.1.4: Student table properties

## Staff Table (Table name: Staff)

### Staff Table Design View (in phpMyAdmin)

Server: localhost » Database: wordpress » Table: Staff

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>stf_ID</u>	int(7)			No	None		Change  Drop  Primary  Unique
2	stf_name	varchar(50)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique
3	stf_fac	varchar(10)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique
4	stf_dpt	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique

Figure 5.1.5: Staff Table Design View

### Staff Table Datasheet View (in phpMyAdmin)

	stf_ID	stf_name	stf_fac	stf_dpt
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1098765	ZYAN MALIK	ICS	Department of Student Affairs
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1231233	LEE AH MING	FAS	Human Resource
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2345672	LUNA GOH	FEGT	Department of Consultancy & Commercialisation
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5555555	BABY SHARK	FICT	IT Infrastructure and Support Centre
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6664444	CHEN LI XIN	CEE	Division of Human Resource

Figure 5.1.6: Staff Table Datasheet View

Field Name	Description
<i>stf_ID</i>	Identification number for a staff
<i>stf_name</i>	Name of a staff
<i>stf_fac</i>	Faculty of the staff
<i>stf_dpt</i>	Department of the staff

Table 5.1.5: Field's description in Staff table

Field Name	Data Type	Field Length	Nulls	PK/FK	Reference to
<i>stf_ID</i>	integer	7	No	PK	-
<i>stf_name</i>	Varchar	50	No	-	-
<i>stf_fac</i>	Varchar	10	No	-	-
<i>stf_dpt</i>	Varchar	200	No	-	-

Table 5.1.6: Staff table properties

### Course Timetable Table (Table name: course\_timetable)

### Course Timetable Table Design View (in phpMyAdmin)

Server: localhost » Database: wordpress » Table: course_timetable									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	<b>timetable_ID</b>	int(7)			No	None	AUTO_INCREMENT	<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique
2	<b>sub_ID</b>	varchar(15)	latin1_swedish_ci		No	None		<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique
3	<b>stf_ID</b>	int(7)			No	None		<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique
4	<b>day</b>	varchar(10)	latin1_swedish_ci		No	None		<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique
5	<b>timestart</b>	varchar(5)	latin1_swedish_ci		No	None		<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique
6	<b>timeend</b>	varchar(5)	latin1_swedish_ci		No	None		<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique
7	<b>session</b>	varchar(3)	latin1_swedish_ci		No	None		<input type="checkbox"/> Change <input type="checkbox"/> Drop	<input checked="" type="checkbox"/> Primary <input checked="" type="checkbox"/> Unique

Figure 5.1.7: Course Timetable Table Design View



### Course Timetable Table Datasheet View (in phpMyAdmin)

		timetable_ID	sub_ID	stf_ID	day	timestart	timeend	session
<input type="checkbox"/>	Edit	1000001	UCCE1054	5555555	Tuesday	1100	1300	L1
<input type="checkbox"/>	Edit	1000002	UBMM1011	2345672	Saturday	1500	1700	P2
<input type="checkbox"/>	Edit	1000003	UCCC2063	6664444	Friday	0900	1200	T14
<input type="checkbox"/>	Edit	6666666	MPU34032	5555555	Wednesday	0800	1000	L3
<input type="checkbox"/>	Edit	9812523	UCCE1014	1098765	Wednesday	1600	1730	P6
<input type="checkbox"/>	Edit	9812524	UCCE1023	5555555	Monday	0800	0900	T5

Figure 5.1.8: Course Timetable Table Datasheet View

Field Name	Description
<b>timetable_ID</b>	Identification number for a specific timetable (Auto-increment)
<b>sub_ID</b>	Identification number of the related subject
<b>stf_ID</b>	Identification number of the staff incharge
<b>day</b>	Day of the current session
<b>timestart</b>	Time Start of the current session
<b>timeend</b>	Time End of the current session
<b>session</b>	Session of the subject

Table 5.1.7: Field's description in Course Timetable table

Field Name	Data Type	Field Length	Nulls	PK/FK	Reference to
<b>timetable_ID</b>	integer	7	No	PK	-
<b>sub_ID</b>	Varchar	15	No	FK	Subject(sub_ID)
<b>stf_ID</b>	integer	7	No	FK	Staff(stf_ID)
<b>day</b>	Varchar	10	No	-	-
<b>timestart</b>	Varchar	5	No	-	-
<b>timeend</b>	Varchar	5	No	-	-
<b>session</b>	Varchar	3	No	-	-

Table 5.1.8: CourseTimetable table properties

### Student Attendance Table (Table name: student\_attendance)

#### Student Attendance Table Design View (in phpMyAdmin)

Server: localhost » Database: wordpress » Table: student_attendance									
Browse Structure SQL Search Insert Export Import Privileges Operations									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	num	int(100)			No	None	AUTO_INCREMENT	Change	Drop  Primary  Unique
2	timetable_ID	int(7)			No	None		Change	Drop  Primary  Unique
3	stu_ID	int(7)			No	None		Change	Drop  Primary  Unique
4	attnnd_date	varchar(10)	latin1_swedish_ci		No	None		Change	Drop  Primary  Unique

Figure 5.1.9: Student Attendance Table Design View

### Student Attendance Table Datasheet View (in phpMyAdmin)

	num	timetable_ID	stu_ID	attnd_date
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	6666666	1503979	21/3/2018
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	9812523	1703657	22/3/2018
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	1000001	1503979	5/4/2018

Figure 5.1.10: Student Attendance Table Datasheet View

Field Name	Description
<b>num</b>	Index number of this table (Auto-increment)
<b>timetable_ID</b>	Identification number of the related timetable
<b>stu_ID</b>	Identification number of the student
<b>attnd_date</b>	The date where this attendance is taken

Table 5.1.9: Field's description in Student Attendance table

Field Name	Data Type	Field Length	Nulls	PK/FK	Reference to
<b>num</b>	integer	100	No	PK	-
<b>timetable_ID</b>	integer	7	No	FK	Course_timetable (timetable_ID)
<b>stu_ID</b>	integer	7	No	FK	Student(stu_ID)
<b>attnd_date</b>	Varchar	10	No	-	-

Table 5.1.10: Student Attendance table properties

### 5.1.2 ER Diagram of the database

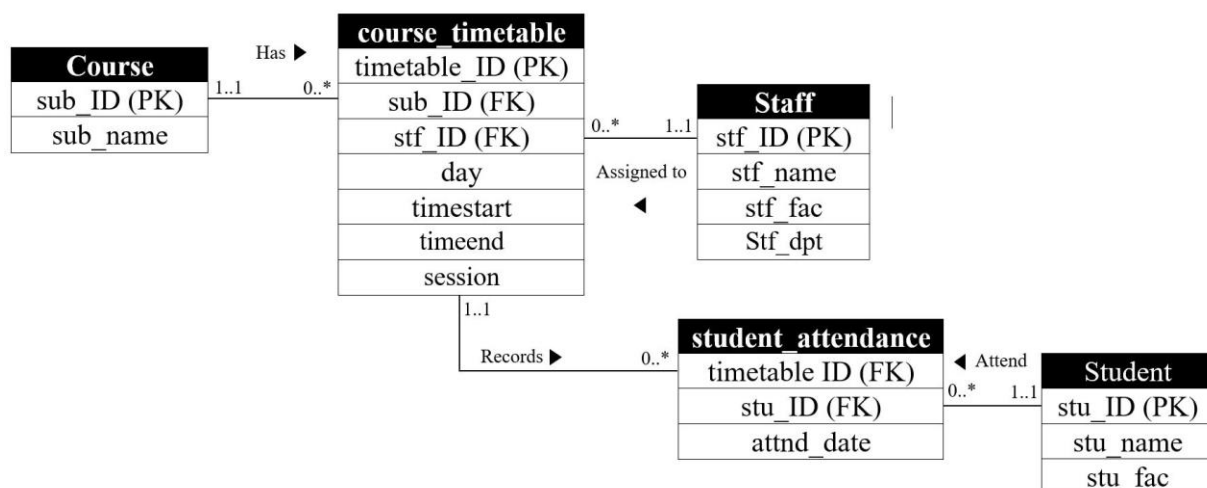


Figure 5.1.11: ER Diagram of the database

### 5.1.3 Relationship between the tables (in phpMyAdmin)

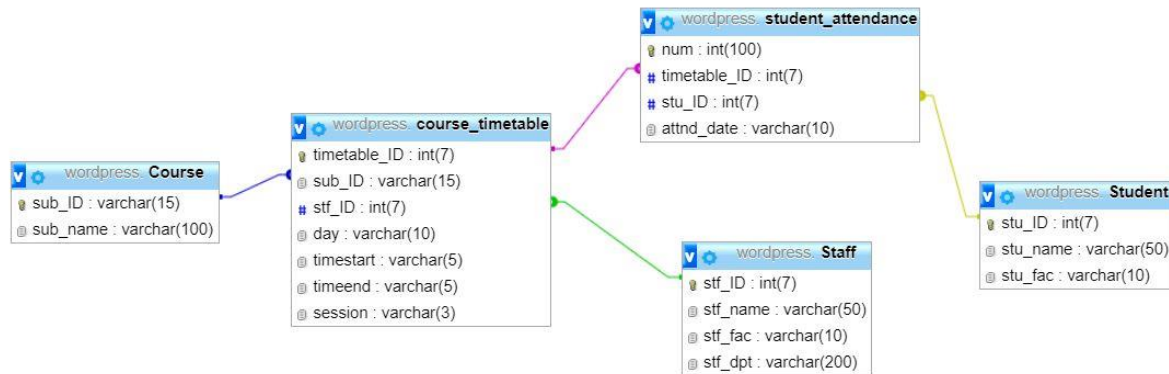


Figure 5.1.12: Relationship between the tables

### 5.1.4 Database Analysis

From the designed database above, there are reasons for the creation of every table. In an institution attendance management system, the main focus is on the staff and students. Thus, the *Student* and *Staff* tables are created to store their basic information. In most institution, every student is attending classes based on their registered course which are conducted according to the arranged sessions. While different students can be taking the same subject in the same semester but are conducted at a different time session. Therefore, there is a need to store the information of every class sessions according to the subjects so that the system can track the exact class session that the students had attended. Thus, the *course\_timetable* table is created to manage that. After having almost every relevant tables created, another table is needed to store the attendance information of the students which will be the *student\_attendance* table.

The three main parent tables in this database are the *Course*, *Student* and *Staff* responsible to hold basic information for the subject details, student's information and staff's information respectively. While there are two child tables in this database which are made up of the *course\_timetable* table and the *student\_attendance* table. In the *course\_timetable* table, information of different class session of a particular subjects are stored. In this table, 2 referenced keys are assigned to connect this table with another two tables which are the *sub\_ID* (connecting it to the *Course* table) and *stf\_ID* (connecting it to the *Staff* table). While in the *student\_attendance* table, there are 3 important elements being stored for each attendance record which are the *stu\_ID*, *timetable\_ID* and the *attnd\_date*. The *timetable\_ID* and the *stu\_ID* are connected to the *course\_timetable* table and *Student* table respectively.

## 5.2 Webpage Development

### 5.2.1 Services provided by the Webpage

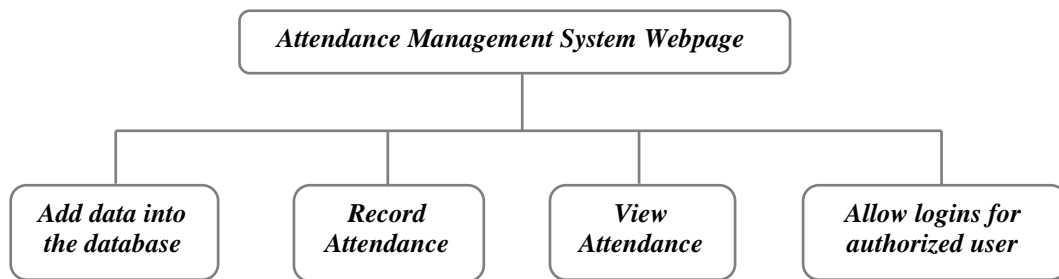


Figure 5.2.1: Summary of the services provided by the Attendance Management System webpage

Based on the chart above, there are in total four types of different utility provided by the Attendance Management System webpage. In Chapter 4, the setup procedure of the website is reviewed and the access path to this website is also discussed in detail. In this Chapter, the created webpage which are hosted by the raspberry pi by **Wordpress** will be illustrated and explained here. Since the website created is connected to the database which can be accessed from the phpMyAdmin page, the ultimate goal provided by this website is to allow the user (usually the staff or lecturer) to communicate with the database without the need to access the phpMyAdmin page nor the raspberry pi. And by logging in into the system, various actions such as **adding records to the table**, **viewing the records** and **filtering the intended records** can all be performed through the webpage at ease.

#### Adding data into the database:

In this section, there are 4 different pages created in Wordpress to deal with this service. By considering what data are needed to be stored into the database, the created database at the previous section will provide a rough idea of what is needed. By looking back to the previous section, there are three major parent tables which consist of **Course**, **Student** and **Staff**. Thus, the details for these three tables are guaranteed to be obtained from the input interface of the website. Then, the **course\_timetable** table also needed its data to be inputted by a user (staff) to arrange class sessions for each subject. The followings are the requirements.

- Able to input course details into the **Course** table
- Able to input student details into the **Student** table
- Able to input staff details into the **Staff** table
- Able to input timetable details for each class session into the **course\_timetable** table

The followings are the created interface pages for this purpose.

## REGISTRATION FORMS

CLICK ON THE LINK BELOW TO FILL IN THE INTENDED FORM.

[Course Registration Form](#)

[Course Timetable Form](#)

[Staff Registration Form](#)

[Student Registration Form](#)

**Figure 5.2.2: Registration Forms tab in the UTAR Attendance System webpage**

## COURSE REGISTRATION FORM

Subject Code:

Subject Name:

## STAFF REGISTRATION FORM

Staff ID:

Staff Name:

Faculty:

Department:

## STUDENT REGISTRATION FORM

Student ID:

Student Name:

Faculty:

## COURSE TIMETABLE FORM

Subject ID	Subject Name
MPU34012	SOCIAL ENTREPRENEURSHIP PROJECT
MPU34032	COMMUNITY PROJECT
UBMM1011	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES
UBMM1013	MANAGEMENT PRINCIPLES
UCCC2063	ALGORITHMS ANALYSIS

Pages: 1 2 3 Next Last 1 of 3

Staff ID	Staff Name
1098765	ZYAN MALIK
1231233	LEE AH MING
2345672	LUNA GOH
5555555	BABY SHARK
6664444	CHEN LI XIN

Subject ID:

Staff ID:

Day:

Time Start:

Time End:

Session:

**Figure 5.2.3: 4 Different registration forms in the UTAR Attendance Management System webpage**

### Attendance Recording:

There is a page in this website specifically designed to initiate a python script containing the recognition algorithm during the attendance taking process is started. But before that could happen, flashing back to the previous section, the *student\_attendance* table require 3 elements to make a record which are the student ID, timetable ID and also the current date. Thus, in this page, the user (lecturer) will be required to enter the related timetable ID and



current date, so that, those values can be passed into the python script and then executed to be stored into attendance table. The following is the created interface.

## RECORD ATTENDANCE

Timetable ID	Subject Name	Day	Start Time	End Time	Session
1000001	BASIC MICROPROCESSOR AND INTERFACING	Tuesday	1100	1300	L1
1233211	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES	Saturday	1500	1700	P2
2221111	ALGORITHMS ANALYSIS	Friday	0900	1200	T14
6666666	COMMUNITY PROJECT	Wednesday	0800	1000	L3
9812523	CIRCUIT THEORY	Wednesday	1600	1730	P6

Pages: [1](#) [2](#) [Next](#) [Last](#) 1 of 2

Select the timetable ID according to the current session.

Timetable ID:  Date:

**Figure 5.2.4: Record Attendance interface on the UTAR Attendance Management System webpage**

### Attendance Viewing:

After the attendance is being recorded into the database, the next requirements is to be able to retrieve it. By considering what different user are interested in the form of data to be displayed, it appears that the lecturer are more interested to view the attendance by the subjects they teach whilst, students are more interested to view their own attendance for each of the subjects they registered. Therefore, the attendance viewing page is designed to have two different access methods.

- View attendance by Student ID
- View attendance by Subject ID (only accessible by lecturer)

The followings are the created interface for the purpose.

## ATTENDANCE

[View attendance by Student](#)

[View attendance by Subject \(Only accessible by authorized user\)](#)

**Figure 5.2.5: Viewing Attendance Menu**

### VIEW ATTENDANCE BY STUDENT

Date	Student ID	Student Name	Subject Name	Session
21/3/2018	1503979	Tan Shu Jing	COMMUNITY PROJECT	L3
22/3/2018	1703657	Tan Shu Ting	CIRCUIT THEORY	P6
5/4/2018	1503979	Tan Shu Jing	BASIC MICROPROCESSOR AND INTERFACING	L1

Enter student ID to search for its attendance information.

Student ID:

[Edit](#)

**Figure 5.2.6: Interface of View Attendance by Student**

### VIEW ATTENDANCE BY SUBJECT

Subject ID	Subject Name
MPU34012	SOCIAL ENTREPRENEURSHIP PROJECT
MPU34032	COMMUNITY PROJECT
UBMM1011	SUN ZI'S ART OF WAR AND BUSINESS STRATEGIES
UBMM1013	MANAGEMENT PRINCIPLES
UCCC2063	ALGORITHMS ANALYSIS

Pages: [1](#) [2](#) [3](#) [Next](#) [Last](#) 1 of 3

Enter the subject ID to check its attendance.

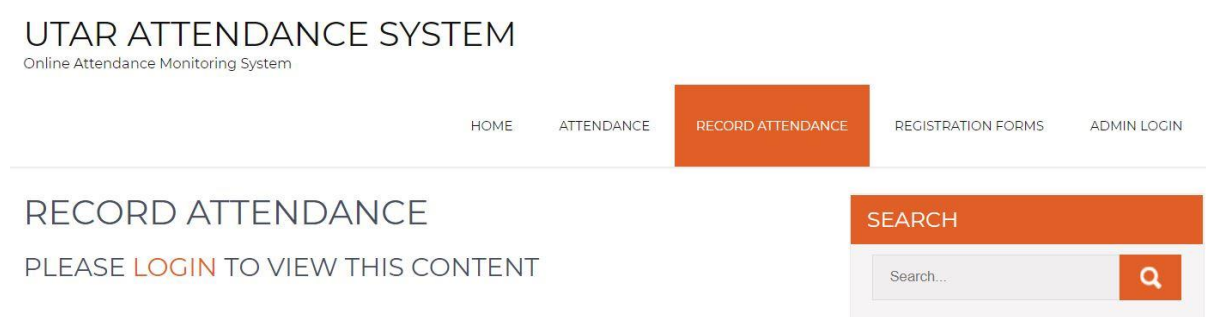
Subject ID:

**Figure 5.2.7: Interface of View Attendance by Subject**

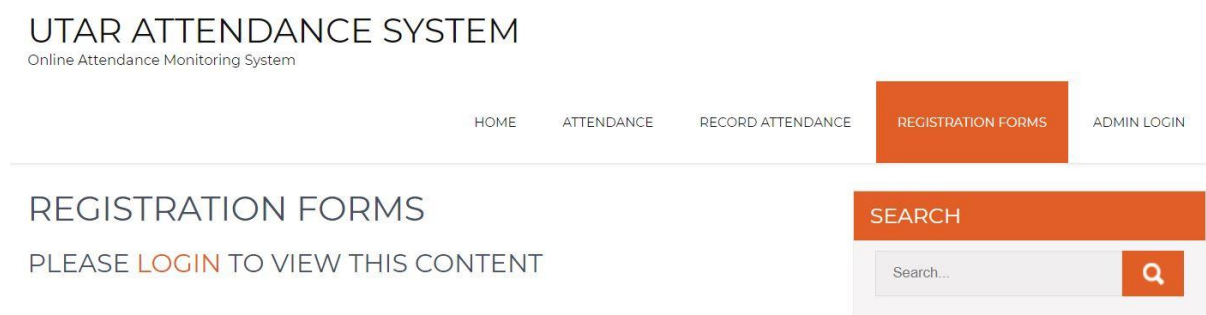


### Authorized User Login:

In this project, the attendance management system webpage designed is protected by authorized logins. Some of the pages are not accessible by a non-authorized user such as the “Record Attendance” page and “Registration Forms” page. When either one of this page is clicked by the user, the following interface will appear to prompt for a login.

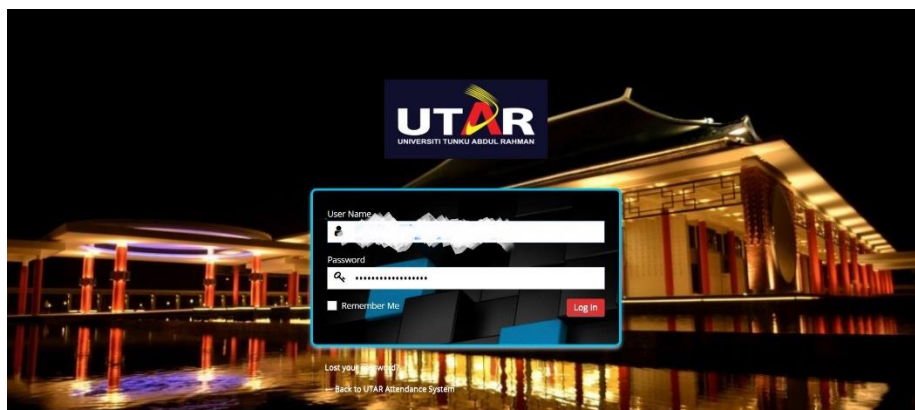


**Figure 5.2.8: Login prompt from record attendance page**





**Figure 5.2.9: Login prompt from registration forms page**

As for authorized user, they can easily login into the system by either clicking on the orange “LOGIN” word or by choosing the “ADMIN LOGIN” tab which a login page will then appear as below.



**Figure 5.2.10: Login page into the UTAR Attendance System**

Apart from that, in the authorized user category, there are two different authorized users with different access priority. To be able to gain full control on the customization of the webpage, the user have to login as the “**root**” user, whereas the “**contributer**” type user can only access both the record attendance and registration forms page to record attendance or communicate with the database. This user is not allowed to make any changes to the design of the webpage. Usually, the “**root**” user will be held by the website owner while the “**contributor**” user will be held by the staff or lecturer in the institution. As the “**root**” user, he/she has the power to create more “**contributor**” users. The Email and Password for both of the mentioned users can be obtained in *Chapter 4 (4.2.3 (d)) & (4.2.3 (e))*.

Username	Name	Email	Role
 root	—	tanshujing2004@yahoo.com	Administrator
 staff_admin	staff admin	testing@yahoo.com	Contributor

**Figure 5.2.11: Users created in Wordpress for this project**

### Other Pages:

Apart from the pages created above, there are 3 additional pages created just for viewing purpose where the records in the database can be viewed here.

- Staff Directory page
- Student Directory page
- Course Directory page

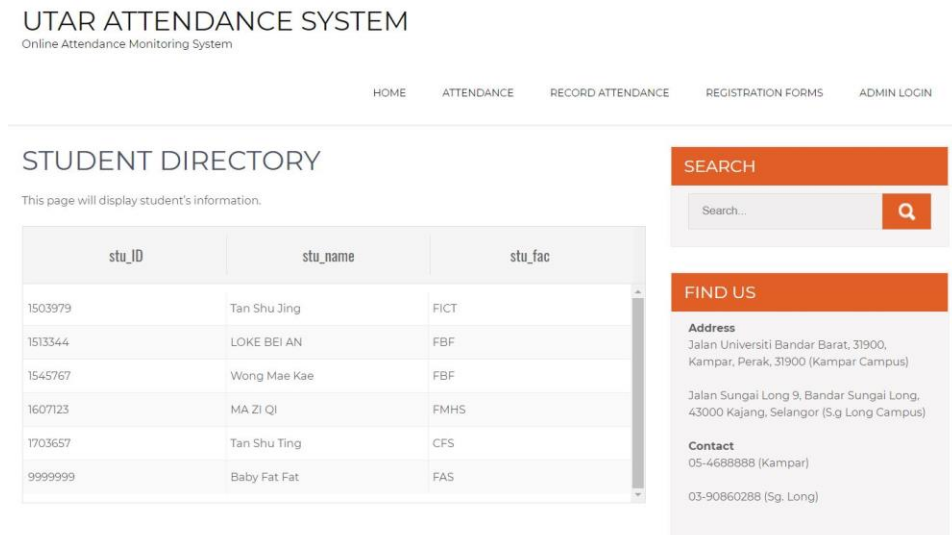
The Staff and Student Directory pages only display a direct Staff and Student tables from the database, whilst in the Course Directory page, it allows user to enter a specific Subject ID to enable the viewing of all its timetable details.

UTAR ATTENDANCE SYSTEM			
Online Attendance Monitoring System			
<a href="#">HOME</a> <a href="#">ATTENDANCE</a> <a href="#">RECORD ATTENDANCE</a> <a href="#">REGISTRATION FORMS</a> <a href="#">ADMIN LOGIN</a>			
STAFF DIRECTORY			
This page will display staff's information.			
Staff ID	Staff Name	Faculty	Department
1098765	ZYAN MALIK	ICS	Department of Student Affairs
1231233	LEE AH MING	FAS	Human Resource
2345672	LUINA GOH	FECT	Department of Consultancy & Commercialisation
5555555	BABY SHARK	FICT	IT Infrastructure and Support Centre
6664444	CHEN LI XIN	CEE	Division of Human Resource

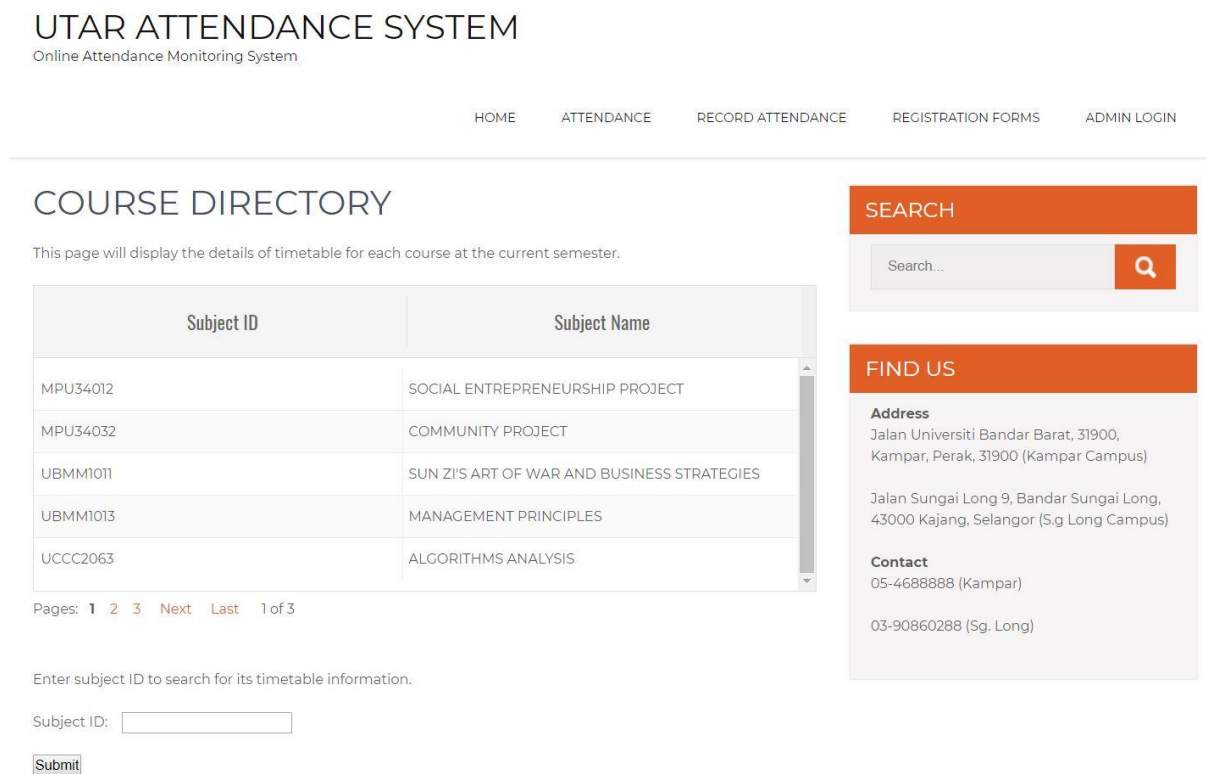
**SEARCH**

**FIND US**  
**Address**  
Jalan Universiti Bandar Barat, 31900, Kampar, Perak, 31900 (Kampar Campus)  
  
Jalan Sungai Long 9, Bandar Sungai Long, 43000 Kajang, Selangor (Sg Long Campus)  
  
**Contact**  
05-4688888 (Kampar)  
03-90860288 (Sg. Long)

**Figure 5.2.12: Staff Directory page**



**Figure 5.2.13: Student Directory page**



**Figure 5.2.14: Course Directory page**

## 5.2.2 Webpage Analysis

Most of the webpage content are displayed by Wordpress, however some of the webpage have filtering functions that are actually executed from a *php* file where query is used to filter the required data from the database. In this project, all the created *php* files are stored under

the path */var/www/html/wordpress* in the raspberry pi. The followings are the corresponding *php* file working behind the service provided by the webpage.

**Registration Forms:**

- Course Registration Form → *student\_details.php*
- Course Timetable Form → *timetable\_details.php*
- Staff Registration Form → *staff\_details.php*
- Student Registration Form → *student\_details.php*

**Attendance:**

- View Attendance by Student → *show\_attendance.php*
- View Attendance by Subject → *show\_course\_attendance.php*

Record Attendance → *student\_attendance.php*

Course Directory → *show\_timetable.php*

\*Note: All the *php* files above can be found in the attached cd.

## **Chapter 6: Conclusion**

### **6.1 Project Review**

After conducting this project, attendance can now be taken with a portable mini box (raspberry pi + pi camera) in a Wifi coverage area. This technology can reduce the effort of enforcing students to attend classes as everything is automated. Since Wifi coverage is not a problem for most of the institution, by using a mobile phone, the lecturer can enter the current class session's information into the Attendance Management System Webpage hosted by the raspberry pi to start the attendance taking process. This had provided convenience not only to the lecturer but also to the students because the attendance taking process for a class of approximately 100 students can be done in 5 minutes which is way more faster than the old method of passing attendance sheet around in the classroom which created a lot of issues to the institutions and inconvenience to the students. Other than that, this system provides excellent graphical interface to the user. Data accessing can be easier nowadays simply by logging in into the webpage where searching of a record can be done easily. This also reduces the need of the lecturer to keep on entering the attendance record manually into the system.

#### **6.1.1 Personal Insight**

The raspberry pi is indeed a very powerful portable device that can perform many tasks to solve our every day's life problem. It is not only a mini computer, but it also allows embedded systems to work where huge computers are no longer needed to navigate simple yet helpful task. Throughout this project, I've learnt that building and hosting an own website is totally possible. I am a big fan of website design and development thus, it is a little unfortunate that I didn't manage to create a domain name for the webpage, however, it is possible to get it done without charges, thus, I would definitely explore more about this in the future to make use of this advantage. Although raspberry pi has limited resources, however its portability is the best thing to ever exist because I can actually manage or work on the project anywhere I want. Apart from that, I've learnt that phpMyAdmin is a very useful tool for managing a database, it provides a lot of convenient to me throughout the development of this project because it reduced the need to communicate with the database using sql language which can be disastrous if I am not fluent enough to handle it. Overall, this project had given me the chance to discover the world that can barely exist in the class. The thirst for solving each and every problem encountered in this project had made me acquire determinations

towards responsibilities. If I am ever given the chance to work on such projects with a larger scales in the future, I will definitely take a leap into it.

### 6.1.2 Implementation Issue and Challenges

During the development of the project, there are several issues that cause minor hindrance to the development. Initially, a GUI is created to aid the user for storing their portrait for the formation of face database. To achieve that, an external library called *guizero* is downloaded to aid the creation of the GUI. However, there are many limitations to this library as it does not support the view of other image file type except for *.gif* image file type. Therefore, images cannot be displayed through the window. Besides that, there are many restrictions on the layout of the GUI window which makes the created interface undesirable. Thus, the usage of *guizero* is abandoned in later times which is then replaced by *Tkinter*.

Apart from that, before being able to test out the recognizer, there are insufficient faces in the created database as there are only a small amount of volunteer willing to help out to form the face database. However, this problem is overcome by doing some research on the internet which came out with the solution of using a pre-prepared face database that are downloadable from the internet. The downloaded face database is normalized and greyscaled, thus making the testing process very convenient.

However, in this project, the website developed can only be accessed locally by devices that are using the same network as the raspberry pi. This is due to the lack of administrative power to alter the institution's networking system. But, it is feasible/possible for the raspberry pi to gain access from an outer network if there is allowance from the authority to allow implementation of the port forwarding configuration.

It is also very challenging while dealing with the pre-processing of the captured image. Fortunately, those problems can be resolved by surfing through the internet for recommended solutions. In short, developing a face recognition system can be very easy when there is sufficient background knowledge of how those process worked because most of the complicated algorithm are provided in the library itself which only requires understanding in order to be able to integrate it into the developing system.

### **6.1.3 Contributions of this project**

The attendance taking process had never been an easy task to every institution. The old method of using paper to collect the attendance had created numerous troubles to the institution. However, with the invention of this project using the raspberry pi to solve issues like this can somehow be very effective for the institution. This project utilized the presence of genuine characteristic in every student to conduct the attendance taking procedure which never been practised before in history. In directly, the achieved objectives of this project had induced the following effects.

- Students cannot sign the attendance for their friend.
- The process of attendance taking is now paperless where resources can be saved.
- Apart from resources, this system also saves a lot of time.
- Enforcement is done indirectly without human workforce.
- Students will be more punctual to class.
- It is almost impossible to sabotage the system.

By implementing this system, lecturer can just sit back and collect the attendance without worries.

### **6.1.4 Further Developments**

Since the development time for this project is very limited, the designed system only consists of the minimum function required for it to work. However, it can be further improved to maximise the usage of the raspberry pi to produce a better system. The followings are the further developments for the project to be improved.

- Provide a better domain name for the webpage.
- Improve the face recognition algorithm.
- Provide better search functions in the webpage.
- Expand the storage of the raspberry pi.
- Develop a fingerprint recognition mechanism to enhance the recognition system.
- Improves the system so that it can eliminates the need of lecturer input before the recognition procedure can start.
- Improves the database so that it can also stores the information of the subjects taken by each student to facilitate the attendance marking procedure.

## 6.2 Conclusion

Before the development of this project. There are many loopholes in the process of taking attendance using the old method which caused many troubles to most of the institutions. Therefore, the facial recognition feature embedded in the attendance monitoring system can not only ensure attendance to be taken accurately and also eliminated the flaws in the previous system. By using technology to conquer the defects can not merely save resources but also reduces human intervention in the whole process by handling all the complicated task to the machine. The only cost to this solution is to have sufficient space in to store all the faces into the database storage. Fortunately, there is such existence of micro SD that can compensate with the volume of the data. In this project, the face database is successfully built. Apart from that, the face recognizing system is also working well. A webpage is also successfully built with fully functioning feature which is user-friendly. The database built is hidden from the user, however they can still access and make changes to it through the developed webpage with excellent interface.

At the end, the system not only resolve troubles that exist in the old model but also provide convenience to the user to access the information collected which perfected the existence of technology to assist human's needs.



## Bibliography

- Diepen, G. (2017). *Detecting and tracking a face with Python and OpenCV*. [online] [www.GuidoDiepen.nl](http://www.GuidoDiepen.nl). Available at: <https://www.guidodiepen.nl/2017/02/detecting-and-tracking-a-face-with-python-and-opencv/> [Accessed 19 Oct. 2017].
- Docs.opencv.org. (2017). *Face Recognition with OpenCV — OpenCV 2.4.13.4 documentation*. [online] Available at: [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html#face-database](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-database) [Accessed 19 Oct. 2017].
- Element14.com. (2017). *Community: Raspberry Pi 3*. [online] Available at: [https://www.element14.com/community/community/raspberry-pi/raspberrypi3/content?filterID=contentstatus\[published\]~objecttype~objecttype\[document\]&filterID=contentstatus\[published\]~language~language%5Bcpl%5D](https://www.element14.com/community/community/raspberry-pi/raspberrypi3/content?filterID=contentstatus[published]~objecttype~objecttype[document]&filterID=contentstatus[published]~language~language%5Bcpl%5D) [Accessed 21 Oct. 2017].
- GitHub. (2017). *iashris/Face-Detection-OpenCV*. [online] Available at: <https://github.com/iashris/Face-Detection-OpenCV/blob/master/FaceDet.py> [Accessed 19 Oct. 2017].
- Opencv-python-tutroals.readthedocs.io. (2017). *Getting Started with Images — OpenCV-Python Tutorials 1 documentation*. [online] Available at: [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_gui/py\\_image\\_display/py\\_image\\_display.html](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html) [Accessed 19 Oct. 2017].
- Projects.raspberrypi.org. (2017). *Cite a Website - Cite This For Me*. [online] Available at: <https://projects.raspberrypi.org/en/projects/getting-started-with-guis> [Accessed 18 Oct. 2017].
- Raja, R. (2017). *Face Detection using OpenCV and Python..* [online] SuperDataScience - Big Data | Analytics Careers | Mentors | Success. Available at: <https://www.superdatascience.com/opencv-face-recognition/> [Accessed 19 Oct. 2017].
- Raspberry Pi. (2017). *Download NOOBS for Raspberry Pi*. [online] Available at: <https://www.raspberrypi.org/downloads/noobs/> [Accessed 19 Oct. 2017].
- Rosebrock, A. (2017). *Accessing the Raspberry Pi Camera with OpenCV and Python - PyImageSearch*. [online] PyImageSearch. Available at: <https://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/> [Accessed 19 Oct. 2017].
- Rosebrock, A. (2017). *How to install OpenCV 3 on Raspbian Jessie - PyImageSearch*. [online] PyImageSearch. Available at: <https://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/> [Accessed 18 Oct. 2017].
- Rosebrock, A. (2017). *Practical Python and OpenCV: Learn Computer Vision in a Single Weekend*. [online] Pyimagesearch.com. Available at: <https://www.pyimagesearch.com/practical-python-opencv/?src=sidebar-face-detection> [Accessed 18 Oct. 2017].

- The Pi Hut. (2017). *How to install / use the Raspberry Pi Camera*. [online] Available at: <https://thepihut.com/blogs/raspberry-pi-tutorials/16021420-how-to-install-use-the-raspberry-pi-camera> [Accessed 20 Oct. 2017].
- Authority.com. (n.d.). *Software Development Life Cycle / Software Testing is an Art!*. [online] Available at: <http://www.authority.com/sdlc/> [Accessed 6 Apr. 2017].
- Cha, B. (2014). *A Beginner's Guide to Understanding the Internet of Things*. [online] Recode. Available at: <https://www.recode.net/2015/1/15/11557782/a-beginners-guide-to-understanding-the-internet-of-things> [Accessed 6 Apr. 2017].
- Gordiyenko, S. (2014). *Waterfall Software Development Life Cycle (SDLC) Model: Steps, Stages, Case Studies*. [online] XB Software. Available at: <https://xbsoftware.com/blog/software-development-life-cycle-waterfall-model/> [Accessed 6 Apr. 2017].
- Half, R. (2015). *6 Basic SDLC Methodologies: Which One is Best?*. [online] Robert Half. Available at: <https://www.roberthalf.com/technology/blog/6-basic-sdlc-methodologies-the-pros-and-cons> [Accessed 6 Apr. 2017].
- Ieeexplore.ieee.org. (2014). *Internet of Things for Smart Cities - IEEE Xplore Document*. [online] Available at: <http://ieeexplore.ieee.org/document/6740844/> [Accessed 6 Apr. 2017].
- Libelium.com. (2016). *Top 50 Internet of Things Applications - Ranking / Libelium*. [online] Available at: [http://www.libelium.com/resources/top\\_50\\_iot\\_sensor\\_applications\\_ranking/](http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/) [Accessed 6 Apr. 2017].
- Qastation.wordpress.com. (2008). *SDLC / Qastation / Page 2*. [online] Available at: <https://qastation.wordpress.com/tag/sdlc/page/2/> [Accessed 6 Apr. 2017].
- Santiago, D. (n.d.). *edu-scrum*. [online] Slidedeck.io. Available at: <http://slidedeck.io/Dawlys/edu-scrum> [Accessed 6 Apr. 2017].
- SearchSoftwareQuality. (2015). *What is systems development life cycle (SDLC)? - Definition from WhatIs.com*. [online] Available at: <http://searchsoftwarequality.techtarget.com/definition/systems-development-life-cycle> [Accessed 6 Apr. 2017].
- Smartcitysystems.com. (2015). *Smart City Systems : Time and Attendance Machines, HR and Payroll Software UAE, HR and Payroll Software dubai, HR and Payroll UAE, HR and Payroll dubai, HRMS UAE, HR and Payroll Software Abu Dhabi, HR and Payroll Software Sharjah, Payroll Software UAE, Payroll Software Saudi Arabia, Payroll Software Oman, Payroll Software Qatar, Payroll Software Kuwait, Payroll Software Dubai, payroll Software Abu Dhabi, payroll Software Sharjah, HRMS dubai*. [online] Available at: [http://www.smartcitysystems.com/content/default.aspx?categorycode=Time\\_and\\_Attendance\\_Machines](http://www.smartcitysystems.com/content/default.aspx?categorycode=Time_and_Attendance_Machines) [Accessed 6 Apr. 2017].
- Ti.com. (2012). *Application Areas for the Internet of Things / TI.com*. [online] Available at: [http://www.ti.com/ww/en/internet\\_of\\_things/iot-applications.html](http://www.ti.com/ww/en/internet_of_things/iot-applications.html) [Accessed 6 Apr. 2016].

- A2hosting.com. (2018). *Connecting to MySQL using Python - A2 Hosting*. [online] Available at: <https://www.a2hosting.com/kb/developer-corner/mysql/connecting-to-mysql-using-python> [Accessed 10 Feb. 2018].
- Agarwal, N. (2018). *How to Create Customized Form in WordPress | InkThemes*. [online] InkThemes. Available at: <https://www.inkthemes.com/how-you-can-easily-create-customized-form-in-wordpress/> [Accessed 11 Feb. 2018].
- Balkhi, S. (2018). [online] Wpbeginner.com. Available at: <http://www.wpbeginner.com/beginners-guide/beginners-guide-to-wordpress-database-management-with-phpmyadmin/> [Accessed 6 Feb. 2018].
- Bazerra, G. (2018). *How to use python 3.5.1 with a MySQL database*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/35684400/how-to-use-python-3-5-1-with-a-mysql-database> [Accessed 10 Feb. 2018].
- Breuning, S. (2018). *Raspberry Pi web server - Setting up Nginx and uWSGI for CGI scripting*. [online] Raspberrywebserver.com. Available at: <http://raspberrypiwebserver.com/cgiscripting/setting-up-nginx-and-uwsgi-for-cgi-scripting.html> [Accessed 11 Feb. 2018].
- command, I. (2018). *Image Processing (OpenCV with PHP) - Issue with the exec command*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/35381705/image-processing-opencv-with-php-issue-with-the-exec-command/35382122> [Accessed 15 Feb. 2018].
- Docs.opencv.org. (2018). *Saving and Loading a FaceRecognizer — OpenCV 2.4.13.6 documentation*. [online] Available at: [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec\\_save\\_load.html](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec_save_load.html) [Accessed 10 Mar. 2018].
- Hobbs, S. (2018). *How to Install WordPress on a Raspberry Pi | samhobbs.co.uk*. [online] Samhobbs.co.uk. Available at: <https://samhobbs.co.uk/2014/02/how-to-install-wordpress-on-a-raspberry-pi> [Accessed 6 Feb. 2018].
- Linux.com | The source for Linux information. (2018). *Understanding Linux File Permissions*. [online] Available at: <https://www.linux.com/learn/understanding-linux-file-permissions> [Accessed 3 Mar. 2018].
- Lovely, S. (2018). *How to set up a web server on the Raspberry Pi - The Pi*. [online] The Pi. Available at: <https://thepi.io/how-to-set-up-a-web-server-on-the-raspberry-pi/> [Accessed 3 Feb. 2018].
- page, C. (2018). *Calling OpenCV python script with php page*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/44745170/calling-opencv-python-script-with-php-page> [Accessed 15 Feb. 2018].

- PHP, R. (2018). *Running a Python script from PHP*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/19735250/running-a-python-script-from-php> [Accessed 15 Feb. 2018].
- Pi My Life Up. (2018). *Raspberry Pi MYSQL & PHPMyAdmin Tutorial*. [online] Available at: <https://pimylifeup.com/raspberry-pi-mysql-phpmyadmin/> [Accessed 6 Feb. 2018].
- place?, W. (2018). *Why do I have to keep using `source ~/.profile` to get settings in place?*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/10320885/why-do-i-have-to-keep-using-source-profile-to-get-settings-in-place> [Accessed 20 Feb. 2018].
- Projects.raspberrypi.org. (2018). [online] Available at: <https://projects.raspberrypi.org/en/projects/lamp-web-server-with-wordpress/9> [Accessed 6 Feb. 2018].
- Raspberrypi.org. (2018). *Setting up an NGINX web server on a Raspberry Pi - Raspberry Pi Documentation*. [online] Available at: <https://www.raspberrypi.org/documentation/remote-access/web-server/nginx.md> [Accessed 3 Feb. 2018].
- script, R. (2018). *Raspberry pi call workon cv in bash or python script*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/48034370/raspberry-pi-call-workon-cv-in-bash-or-python-script> [Accessed 16 Feb. 2018].
- script, S. (2018). *Shell Script: Execute a python program from within a shell script*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/4377109/shell-script-execute-a-python-program-from-within-a-shell-script> [Accessed 20 Feb. 2018].
- Ste Wright the Web Developer. (2018). *Install MySQL on Raspberry Pi tutorial. Install MySQL on Raspbian*. [online] Available at: <https://www.stewright.me/2014/06/tutorial-install-mysql-server-on-raspberry-pi/> [Accessed 3 Feb. 2018].
- Wright, S. (2018). *Tutorial - Install Nginx and PHP on Raspbian - Ste Wright the Web Developer*. [online] Ste Wright the Web Developer. Available at: <https://www.stewright.me/2014/06/tutorial-install-nginx-and-php-on-raspbian/> [Accessed 6 Feb. 2018].

# Turnitin: Plagiarism check summary

feedback studio

Tan Shu Jing | Facial Recognition-Based Attendance Monitoring System for Educational Institution

?

1

2

3

4

5

6

7

8

9

Match Overview

2%

1

Guy Hart-Davis, "Deplo...

Publication

<1%

>

2

Submitted to TAR Univ...

Student Paper

<1%

>

3

Submitted to Institute ...

Student Paper

<1%

>

4

www.ijert.org

Internet Source

<1%

>

5

documents.mx

Internet Source

<1%

>

6

Submitted to University...

Student Paper

<1%

>

7

Priyanka Wagh, Roshan...

Publication

<1%

>

8

Harris, Preethi and Raju...

Publication

<1%

>

9

Submitted to 60892

Student Paper

<1%

>

Page: 1 of 59

Word Count: 12951

Text-only Report

Q

Q

## ABSTRACT

The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. Thus, by means of technology, this project will resolve the flaws existed in the current system while bringing attendance taking to a whole new level by automating most of the tasks.

The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the