

DEVELOPMENT OF A
PORTABLE FINGERPRINT ATTENDANCE SYSTEM
FOR UNIVERSITY STUDENT

KHAIR SYUHADAK BIN BASIRAN

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
SEPTEMBER 2012

Development of a Portable Fingerprint Attendance System for University Student

by

Khair Syuhadak Bin Basiran

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

SEPTEMBER 2012

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Development of a Portable Fingerprint Attendance System for University Student

by

Khair Syuhadak Bin Basiran

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved:

(Ms. Zazilah Binti May)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

September 2012

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

KHAIR SYUHADAK BIN BASIRAN

ABSTRACT

As a partial fulfillment of Final Year Project, the main aim of this project is to develop an accurate, fast and very efficient portable attendance system using fingerprint verification technique. This portable fingerprint attendance management system is specially designed and implemented for university student. This system is based on biometrics, software and database technique to solve the problem of spurious attendance and any else made by the students and others. The current product of fingerprint attendance systems offers minimal, nonflexible and costly system. With an increasing awareness of efficiency of fingerprint attendance system, the implementation of portable fingerprint attendance system has gain interest. The implementation is achieved using a combination of hardware and software development. A Motorola handphone model Atrix 4G is being used as a hardware device while software development is based on the Android operating system. Software called Eclipse is used for the development of android application which based on the Java script language. In this paper, the result of the system were shown sequentially, demonstrating that the system able to run safely and efficient in any real-time conditions.

ACKNOWLEDGEMENTS

I hereby would like to grab this opportunity to express my deepest gratitude and gratefulness firstly to God, The Almighty for giving me a chance to acquire new knowledge and with His will, this project was successfully completed. I would like to express my appreciation to Universiti Teknologi PETRONAS for providing the facilities and technical expertise to implement the project. My special thanks and appreciation to my supervisor, Ms. Zazilah Binti May for all the guidance, advice and commitment in assisting me through out this project. Her suggestions for this project are very valuable. A warmness thanks to my fellow friends for all the help, support and opinion given in completing this project. Last but not least I would like to thank my beloved parents and sisters for giving me moral supports and all parties that involved in completing this project. This project would not have been possible to finish without any help of many people who had involved.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL	2
CERTIFICATION OF ORIGINALITY.....	3
ABSTRACT.....	4
ACKNOWLEDGEMENT	5
CHAPTER 1: INTRODUCTION.....	10
1.1 Background of Study	10
1.2 Problem Statement	11
1.3 Objectives and Scope of Study	12
1.4 Project Relevancy	13
1.5 Feasibility of the Project within the Scope and Time Frame	13
CHAPTER 2: LITERATURE REVIEW.....	14
2.1 Fingerprint.....	14
2.2 Fingerprint Sensor.....	15
2.3 Literature Analysis	17
2.3.1 Hardwired Fingerprint System.....	18
2.3.2 Wireless Fingerprint System.....	18
2.3.3 RFID Attendance System	19
2.4 System Analysis	20
CHAPTER 3: RESEARCH METHODOLOGY	21
3.1 Procedure	21
3.2 Procedure Identification.....	22
3.2.1 FYP 1	22
3.2.2 FYP 2	23
3.3 Project Gantt Chart and Key Milestone	24
3.4 Project Tools Required.....	25
3.4.1 Fingerprint Reader Module.....	25
3.4.2 AC to DC adaptor charger	25
3.4.3 PC to hardware USB connector	25
3.4.4 Programming software.....	25

CHAPTER 4: RESULT AND DISCUSSION.....	26
4.1 Project Design Flow	26
4.2 Result	29
4.2.1 Data Gathering and Analysis	29
4.3 Discussion	34
4.3.1 Eclipse (software)	34
4.3.2 Android SDK (Software Development Kit).....	36
4.3.3 Motorola Atrix 4G	38
4.4 Experimentation/Modeling the Project Application	40
CHAPTER 5: CONCLUSION AND RECOMMENDATIONS	43
5.1 Conclusion	43
5.2 Recommendations	44
REFERENCES.....	45
APPENDICES	47

LIST OF FIGURES

Figure 1: The image and details of fingerprint pattern	15
Figure 2: The minutiae points	15
Figure 3: Laptop fingerprint sensor	17
Figure 4: Handphone fingerprint sensor	17
Figure 5: Diagram of hardwired fingerprint system	18
Figure 6: Diagram of wireless fingerprint system	19
Figure 7: Diagram of RFID system.....	19
Figure 8: System Design Flow.....	26
Figure 9: Flow chart of the application.....	28
Figure 10: Application of main page	30
Figure 11: Class setup section page	30
Figure 12: Student profile Section	31
Figure 13: Calendar section page.....	31
Figure 14: Taking attendance system	32
Figure 15: Student attendance report	32
Figure 16: Result of application running on desired hardware.....	33
Figure 17: Attendance report exporting option.....	33
Figure 18: Screenshot of Eclipse software.....	35
Figure 19: Develop new android application in Eclipse.....	35
Figure 20: Android Software Development Kit manager.....	36
Figure 21: Android emulator main interface software.....	37
Figure 22: Fingerprint scanner operation on the desired hardware	39
Figure 23 and 24: The technique to swipe the finger on the scanner.....	39
Figure 25: Main page layout	40
Figure 26: Adding class layout	40
Figure 27: Adding student layout	41
Figure 28: Class attendance and report layout	41
Figure 29: Email attendance report.....	42

LIST OF TABLES

Table 1: Current attendance system technology	17
Table 2: Analysis of the system technology	20

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

Nowadays, tracking of class attendance by using fingerprint sensor is one of the newest technologies that have been applied in some universities in this world [7]. Compared to the previous system, this system can avoid the students from escaping the class. Furthermore, this system also can avoid student from cheating during the attendance taking session as the procedure was done manually by the lectures. By using this system, it also can help to lighter the burden of the lectures to key in and record the attendances of the students. As the fingerprint attendance system support the electronic storage of data, the captured digital image of the student fingerprint can be recorded and stored in the database of the computer which also can help to decrease to probability of losing the data. As the fingerprint attendance system will be in portable device, it is a bonus to a user as it will give a lot of advantages to them. The user of this portable fingerprint attendance system can bring the device everywhere and it will be easily to use to record and store the needed data to database system. It is not like the earliest method where the lecturers need to take attendance manually by passing the attendance sheet around the class and the data need to be maintained in the paper and if missing of just a page used to cause severe loss to the lecturers. Currently, most of the fingerprint attendance product offer limited or minimal flexibility of the system as usually it will be integrated with the main system. The design of portable fingerprint attendance system attempt to provide enhancement to the attendance system with the current android-based software

application system. User friendly graphical user interface that is low cost, accurate and time efficient will be designed and implemented for all parties including administrator and user. The application platform of real time system will be implemented and capable of sending report through an email for further action or reference.

Introducing flexibility in the student attendance system software would prove to be beneficial in term of accurateness and energy saving. Especially in the real time situation of recording the student attendance, the parties involved in the recording the attendance report causes too much energy and time wasting. The accurateness also decreasing as there might be some uncertain recorded data during data compiling process. Importantly, the application system must adhere to the real environment where the error could happen. This includes setting the fingerprint detection method upon event of recording the attendance, and the system platform for transporting the attendance report through an email of the administrator efficiently. The paper is organized as follows. Next section presents the theoretical development and the concept behind the proposed system. The component of the proposed system and the design flow is then shown. The following section demonstrates the results of the system as well as discussion on several important findings. Final section concludes the paper, referring perspectives of future work.

1.2 PROBLEM STATEMENT

As the number of truancy around the world is rising incredibly from time to time, the fingerprint students attendance system has been widely produces worldwide in order to overcome the problem situation. But, even though the automated system has been produced, the efficiency of the current system was still not meet the specific criteria and need to be improved as there are still minor problem occur in the system. For instant, in university, the procedure for student attendance system is taken manually as the lecturer need to pass the

attendance sheet around the class and the student are required to sign according to their name. Then, the efficiency for this process has greatly affected as the student are able to cheat and help to sign their friend name. If the fingerprint student attendance system is used to overcome this situation, the efficiency for attendance taking procedure will get better, but there are still some minor problems as the lectures, students and management will face the problem of managing the fingerprint attendance system. So, the best option is to produce the portable fingerprint attendance system with the system and the database can be individually manageable by the owner of that portable devices. In other way, this will help all the people around the campus to get well-managed by this useful stuff.

1.3 OBJECTIVE AND SCOPE OF STUDY

Nowadays, fingerprint identification for attendance system has been widely used in many places including industries and university area. It offers significant benefits for individuals, businesses and governments but there are still small problem that need to be solved in order to propose the better system. Therefore, the new version of portable fingerprint identification system for university student has been proposed to achieve those objectives:

- a) To develop portable attendance system using fingerprint verification technique based on Android operating system.
- b) To develop user friendly graphical user interface application system.
- c) To increase the efficient of the system compared to the previous design.

The scopes of study are:

- a) To study the current fingerprint identification system technology.
- b) To study the latest mobile device technology
- c) To study about Mobile Application Development (Android)
- d) To study Java language coding system
- e) To study SQL database language

1.4 PROJECT RELEVANCY

Based on the research that had been done, this project acquire the application and knowledge of electrical and electronics field of study especially in the area of programming as the design of the project is mostly dealing with the software and database system. Besides that, some knowledge from other part like electrical power also necessary as this prototype will dealing with the power management efficiency. Then, in the term of necessity, this product will be very helpful and expected to be highly demanded in the future. This project will involve a lot of engineering knowledge and discipline that certainly relevant to the purpose of this final year project.

1.5 FEASIBILITY OF THE PROJECT WITHIN THE SCOPE AND TIME FRAME

The project will start by collecting materials such as books, journals and technical papers specifically on fingerprint sensor, programming language, and database management system. Research will be done from time to time as part of getting a better understanding on this issue. This project will then focus on developing and conducting simulating the programming code and data to test the efficiency of the idea and proceed with the hardware design and assembly. Some improvement will be done if needed in order to produce the best and successful designed product.

CHAPTER 2

LITERATURE REVIEW

2.1 FINGERPRINT

For every person in this world, fingerprint or thumbprint is a one unique identification that everyone will have its own pattern between one another (see figure 1) [6]. It is against law if there were any person who try to copy or steal another person fingerprint. A fingerprint is made up of furrows and ridges, which shows good similarities like average width and parallelism [1]. As time goes by, there were many research have been done through this matter and the result shows that we can distinguish fingerprint with the help of minutiae, which are the some abnormal points that can be found on the ridges. Based on the theoretical knowledge, the minutiae come up with three type of termination which is called bifurcation, ridge ending or termination and short ridges. The bifurcation is the point on the ridge from which other branch drives or a point from where ridge splits into two or more branches. Then, the ridge ending or termination is the immediate ending of ridges or a point where ridge end abruptly[10]. Finally, the short ridges are the ridges that shorter than the average ridge length on the fingerprint. This can be referring in figure 2.

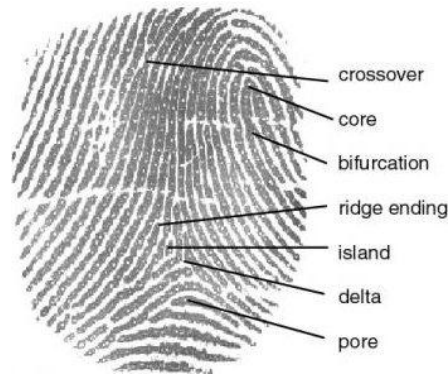


Figure 1: The image and details of fingerprint pattern

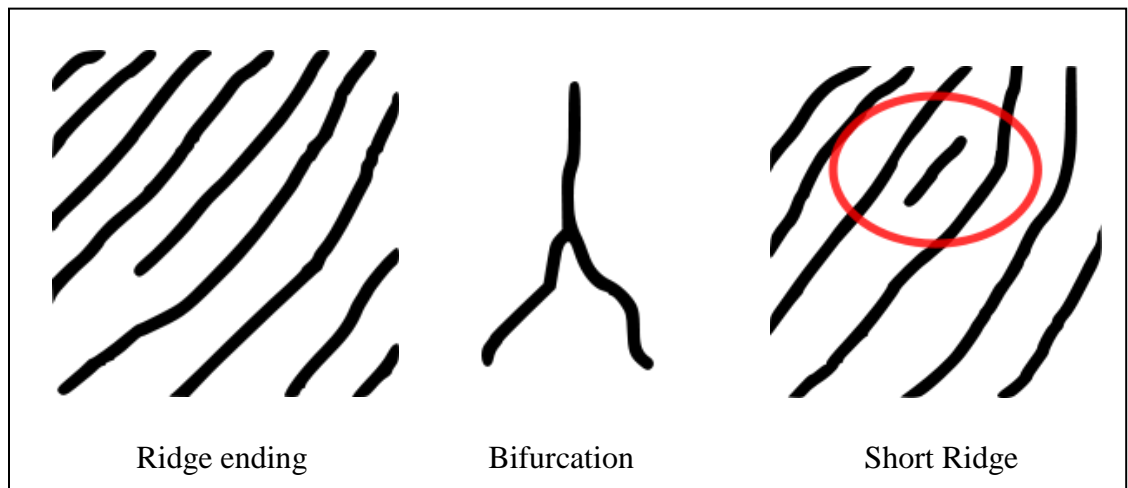


Figure 2: The minutiae points

2.2 FINGERPRINT SENSOR

About a decade ago, we can still remember of cleaning our finger after being dotted with black ink so that we can make a stamp to verify something by using our fingerprint. We can see this at many office or public service around the country during that area of time. Nowadays, technology has rapidly increased from day to day until the new modern design of fingerprint sensor has been created. A fingerprint sensor or reader is designed and implemented for transforming the topological form of human finger into an electric output signal. In the general term, the electric output signal can be called as the captured digital

image of the fingerprint pattern [2]. After there were so many new technology have been invented, fingerprint sensor technology also have exposed too many types of sensor used. The types of fingerprint sensor technologies used are optical fingerprint imaging, ultrasonic sensors, capacitance sensors, active capacitance sensors, and passive capacitance sensors [4]. The optical fingerprint imaging involves capturing a fingerprint pattern by using the visible light. This type of sensor is using a specialized digital camera to take the pattern of the finger. The finger is places at the top layer of the sensor where it also can be known as the touch surface. Under this layer of sensor is a light-emitting phosphor layer which it will illuminates the surface of the finger. The visual image of the fingerprint will be captured as the light reflected from the finger passes through the phosphor layer to an array of solid state pixels [11]. For the ultrasonic type of fingerprint sensor, it order to create the visual image of the fingerprint, it make use the principles of medical ultrasonography [12]. Different with the optical imaging, in order to form an image of the fingerprint, ultrasonic sensors will use a very high sound waves frequency in order to penetrate the epidermal layer of skin. Then, the reflected wave measurements will be used to form an image of the fingerprint since the dermal skin layer displays the same characteristic pattern of the fingerprint. The sound waves used in this type of sensor are generated using piezoelectric transducers. Piezoelectric materials also have been used in this sensor in order to measure the reflected energy [5]. For capacitance sensors, it utilizes the principle that associated with capacitance to form the fingerprint digital images. Those active and passive capacitance sensors are using different approach to generate the digital images of the fingerprint. Like the ultrasonic method, the active capacitance sensors will measure the ridge pattern of the dermal layer. This happens as the distance values are calculated mathematically, using some equations and uses it to form the image of the fingerprint [4]. While the passive capacitance will use the measured value of the capacitance in order to differentiate between the fingerprint ridges and valleys [3]. There are several example of fingerprint sensor in the market nowadays, the sample are shown in the figure below.



Figure 3: Laptop fingerprint sensor



Figure 4: Handphone fingerprint sensor

2.3 LITERATURE ANALYSIS

Portable fingerprint attendance system is designed as a proposed solution from current system that not meets enough criteria to handle some attendance in some situation. Table 1 shows the research and analysis that have been done on the current attendance system technology.

Project Block	Reference	Description / Finding
Hardwired fingerprint system	Ronald B. Roy, John C. Sadlon (2008)	Comprises an integrated network in electronic communication with a computer database
Wireless fingerprint system	Joseph D. Cayen (2010)	Comprises fingerprint scanner and a computer server
RFID attendance system	M.Helmi, A.Mutalib, Herdawatie, M.Farhan (2010)	Comprises RFID smart card and the RFID reader

Table 1: Current attendance system technology

2.3.1 Hardwired Fingerprint System

Based on this system, all the connection between the fingerprint scanner and server is being hardwired in an integrated network [13]. Figure 5 shows the example of current hardwired fingerprint system in the market.

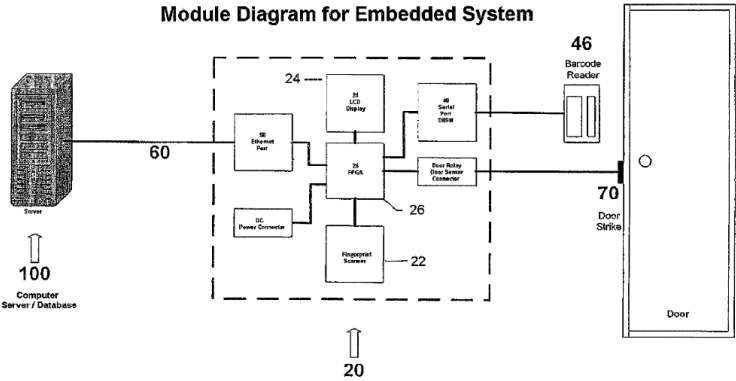


Figure 5: Diagram of hardwired fingerprint system

2.3.2 Wireless Fingerprint System

Compared to the hardwired fingerprint system, the different for this system is that the server and the fingerprint scanner are connected through wireless connection technology [14]. Thus, the concept of storing data and management is slightly the same. Figure 6 show the sample of current wireless fingerprint system in the market.

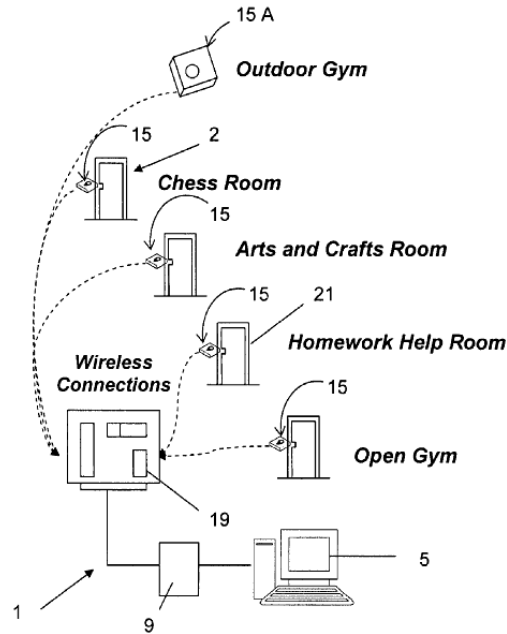


Figure 6: Diagram of wireless fingerprint system

2.3.3 RFID Attendance System

For this system, the concept is slightly different compared to the system before as RFID technology does not need any fingerprint pattern. All the attendance record is depend on the smartcard that need to be detected by RFID reader and send through the database system. The database system can be any device or system such as computer. Figure 7 shows the model of current RFID system that being implemented in the market.

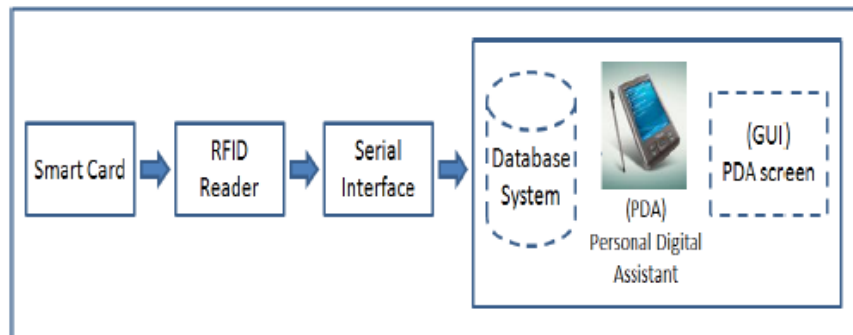


Figure 7: Diagram of RFID system

2.4 SYSTEM ANALYSIS

Based on the comparison that had been done on the current technology, table 2 below shows the result on the advantages and disadvantages on the technology with an additional of the proposed project solution, portable fingerprint attendance system which based on the android system.

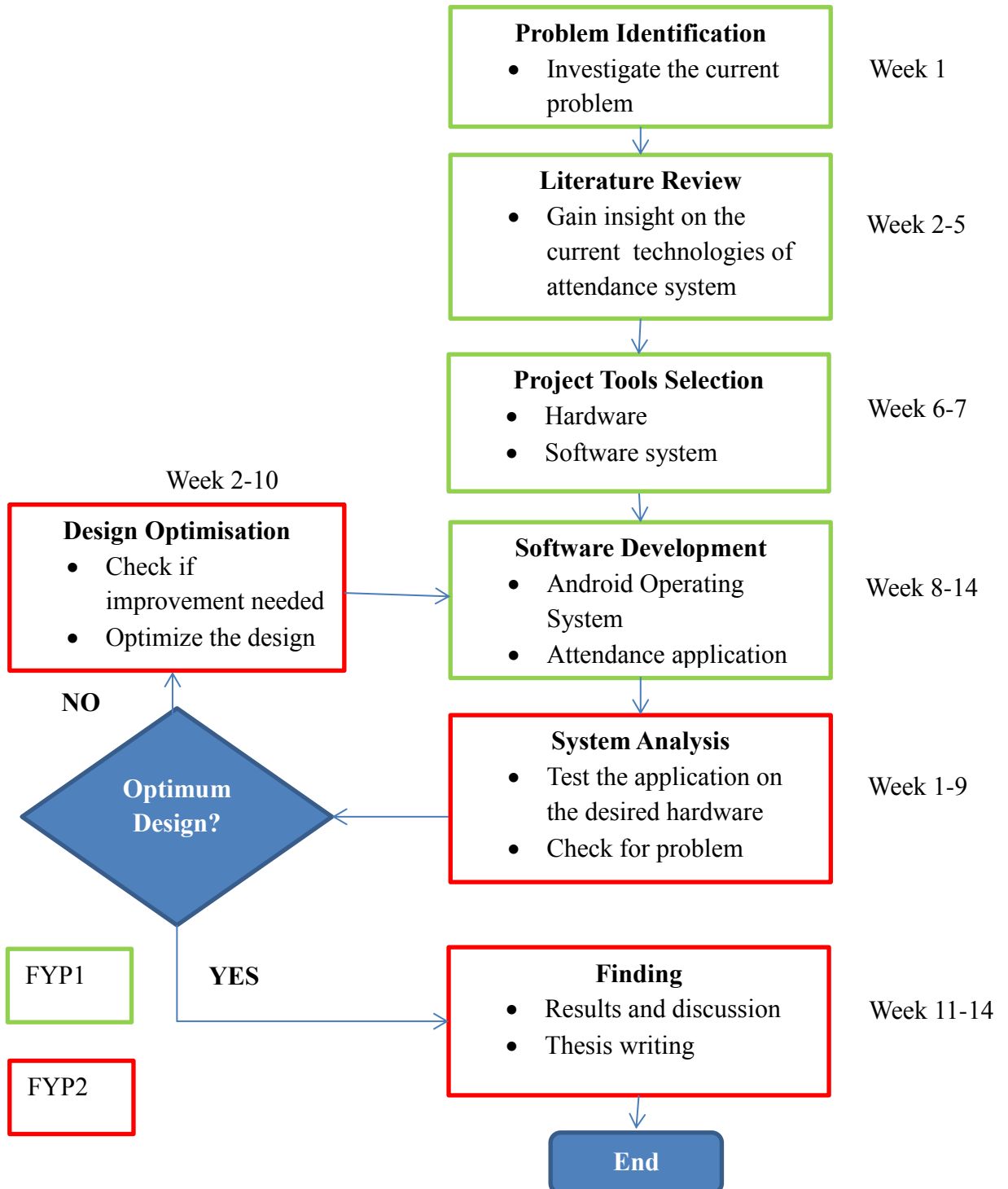
Systems	Advantages	Disadvantages
Hardwired fingerprint system	<ul style="list-style-type: none"> • Stable system • Low maintenance cost 	<ul style="list-style-type: none"> • High development cost • Setup at limited area
Wireless fingerprint system	<ul style="list-style-type: none"> • Cover more area 	<ul style="list-style-type: none"> • High development cost • High maintenance
RFID system	<ul style="list-style-type: none"> • Static or portable system 	<ul style="list-style-type: none"> • Low security system
New Product		
Portable fingerprint attendance system (android)	<ul style="list-style-type: none"> • Portable • High security • Low cost (development and maintenance) 	

Table 2: Analysis of the system technology

CHAPTER 3

RESEARCH METHODOLOGY

3.1 PROCEDURE



3.2 PROCEDURE IDENTIFICATION

3.2.1 FYP 1

This stage is the beginning process of the project development. Starting with the topic selection process, the problem identification process regarding the current problem has been investigated in order to choose the best option to solve the problem. The flow continued on understanding and studying the topic selected. During this understanding process, literature review and further research must be done in order to get the better overview of the chosen topic which is regarding on the current technologies of attendance system. The current technology and past paper would be helping to get the better overview of the new invented project. After completing all of those procedures, the process continued on selecting the best hardware and software system for the new invented portable students attendance system. This project development process can be rapidly done by the help of the current product of fingerprint scanner. Then, as the software system is selected based on the Android operating system, the process is continued on developing the student attendance system based on the Java programming language. In this area, the fingerprint sensor hardware must be properly synchronized with the programming system so that the efficiency of the system can be maximized. Finally, the final procedure for this FYP 1 part will be on combining both student database and the fingerprint programming system. All connected system part must be properly managed as there will no problem during system testing procedure.

3.2.2 FYP 2

In this second stage of the project development process, the first step that needs to be done is to load the system into the desired hardware. After completing the programming part, system analysis procedure must be done as the application will be tested on the desired hardware to check if there any problem exists. If there is any problems occur, improvement must be done on the project and recheck again all the software and hardware as this will help the produce the optimum design of the project. Then, the pre-complete project must go through the testing procedure again that will include all the criteria needed to achieve the project objective. If problem has been solved, thesis writing and report can be done as all the result and discussion regarding the project will be recorded and included in this project documentation.

3.3 PROJECT GANTT CHART AND KEY MILESTONE

FYP 1														
Activities	Week Number													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Topic selection	Red	Red												
Study on the topic selected		Red	Red											
Literature review			Red	Red	Red									
Submit the extended proposal (27/6)						Red								
Search and buy the component						Red	Red							
Proposal defense presentation (18/7)							Red	Red	Red					
Developing all the hardware and software system								Red	Red	Red	Red	Red	Red	Red
Interim report submission (15/8)													Red	Red
FYP 2														
Testing the system	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow					
Solve the project problem		Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow				
Submit the progress report (5/11)								Yellow						
ELECTREX (28/11)											Yellow			
Submit the draft report (10/12)													Yellow	
Submit the final report and technical paper (17/12)														Yellow
Viva session (20/12 – 21/12)														Yellow
Do some improvement and finalize												Yellow	Yellow	Yellow
Submit the final report (7/1)														

3.4 PROJECT TOOLS REQUIRED

No.	Component	Unit
1	Fingerprint Reader Module (Motorola Atrix 4G)	1
2	AC to DC adaptor charger	1
3	PC to hardware USB connector	1
4	Programming software	1

3.4.1 Fingerprint Reader Module

This fingerprint scanner device is portable so that it is convenient to carry anywhere.

3.4.2 AC to DC adaptor charger

Use to charge the prototype when the battery is running out.

3.4.3 PC to hardware USB connector

This connector will be used to synchronize all the data between the hardware and computer.

3.4.4 Programming software

This software will be used during the software developing process.

CHAPTER 4

RESULT AND DISCUSSION

4.1 PROJECT DESIGN FLOW

Figure 8 illustrates the idea of how the application system being implemented and used by the owner of the application. Below of the figure shows an explanation for the illustrated picture.



Figure 8: System Design Flow

This project consists of two main parts that is very important to be completed which are coming from the software development part and the selected hardware. For hardware part, after further research that had been done, it can be concluded that the best option for hardware is the handphone come from model Motorola Atrix 4G. As the handphone has been integrated with the fingerprint scanner, it will help to easy the process of taking the student attendance as hand phone is the most common device that people always bring everywhere. For software development part, as the software is come from the android operating system application, it need to always being synchronized with the hardware (Motorola Atrix 4G) so that the problem occur can be rapidly improve.

After the process of making the application on the desired hardware is completely done, the device will be given to the lecture as the lecturer is the main administrator for the application. Then, the lecturer can bring the handphone to their class or any session to take the attendance for their students. Before taking the attendance, the lecturer must register their students name in the application so that the database of the student profiles and fingerprint pattern can be stored in the application safely. Then, the lecturer can pass around the phone to the students so that the student can tick their name on the attendance software and this will be verified by their own fingerprint pattern.

Finally, after the attendance taking procedure has been completed, the lecturer can export the data of the signed attendance list to the document view such as from the format of Microsoft Office Word or Excel. Furthermore, the lecturer also can export their student attendance data to their email so that they can store their data securely on the internet. All this process can be done easily as nowadays all the handphone technologies can always being connected to the internet anytime at anywhere.

For software development process, there will be a flowchart on how the idea of application will run. Figure 9 illustrate the main idea on how the application will run between the owner and the user of the application.

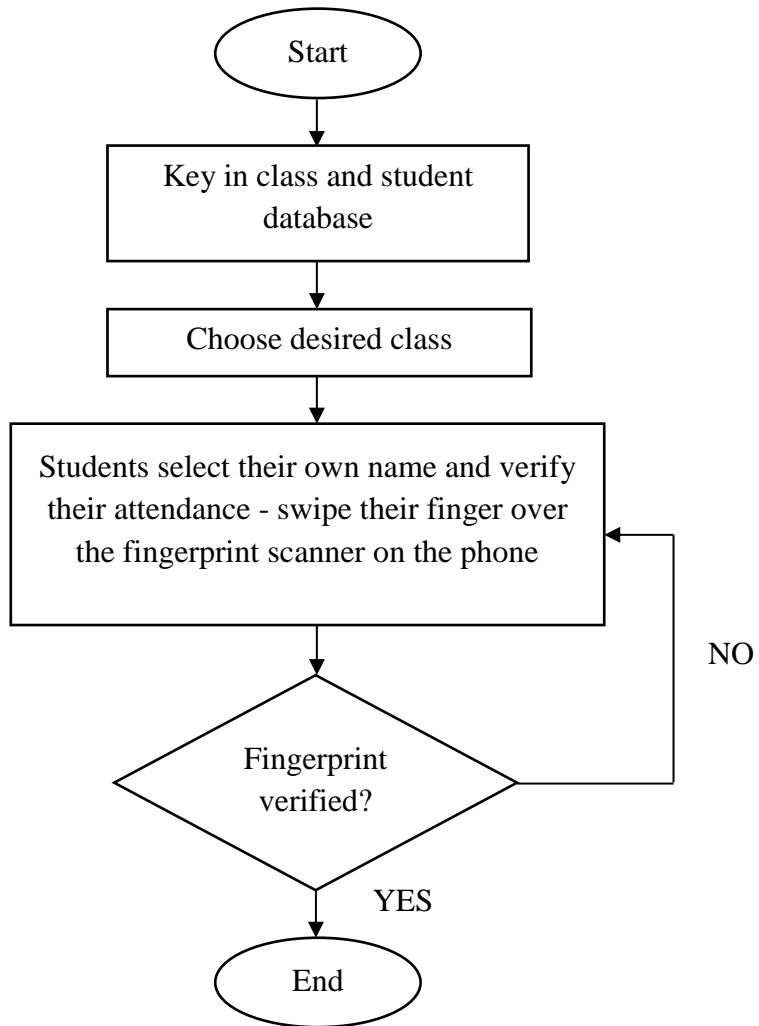


Figure 9: Flow chart of the application

4.2 RESULT

In this section, the project result will be shown and discussed more thoroughly. It will be mostly discussed on the software development as there is not much thing that needs to be discussed about the hardware part. As the software development for this project is from the Java programming language, specific software is needed in order to undergo the application development process.

4.2.1 Data Gathering and Analysis

Software Development

For the android application development, the software that commonly used by all programmer around the world is called as eclipse. Further clarification regarding eclipse will be explained in the next discussion part. As some research and studied that had been done, there were many samples of android attendance application around the world. Unfortunately, there were still no applications for student attendance system that can be verified by their own fingerprint. As a pioneer for this application project, I managed to refer some current software regarding this attendance system. After further research, the application system that will be invented in this project task will be look alike below. All explanation regarding my project application will be described by the figure shown as follow.

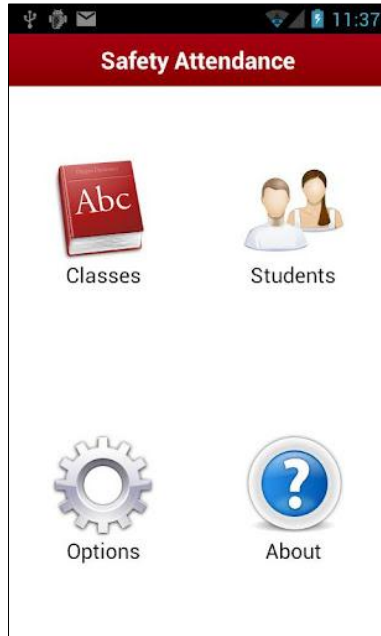


Figure 10: Application of main page

Based on the figure 10 besides, it shows the main page of my expected Portable Student Attendance System. The will be some option buttons that can be selected and modified which are “Classes”, “Students”, “Options”, and “About” button. For “Classes” button, the administrator of the software can manage the class they have by adding or remove the class session they have. For “Students” button, the administrator can register their student profiles

that later will be added to the class. In this section, the fingerprint pattern database of the student will be recorded and stored in the system so that it can be used later when they want to verify their own name in the attendance list. Then, for “Option” button, it will help the administrator to modify the application setting while the “About” button will help to give info to the administrator regarding the background of the application.

Then, if the “Classes” button is pressed, the application will go through the next page as shown in the figure 11. Based on the page, it clearly shown that New Class is being added to the application system. Then, the administrator can edit their class name, email and some notes according to their desired notes or other else. As shown in figure, the list of student also can be directly added to the class just by pressing the

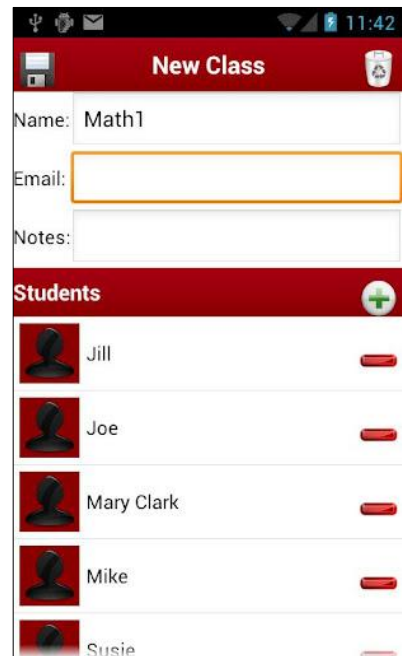


Figure 11: Class setup section page

(+) icon. But, this student only will exist if the name of student has been register in the “Students” section page before.

Other than “Classes” part, the registered students part also can be modified by the administrator. For this part, the administrator can just click on the “Students” icon at the main page, and after completed the registration process, there will be one more page section inside the student name. After the student name icon is pressed, there will one more new page section as shown on figure 12. In this section page, the administrator can edit the student name, notes and QR code data. There also will be some option that will help the administrator to have more information about the student as this software can record their fingerprint pattern and the picture of the students.

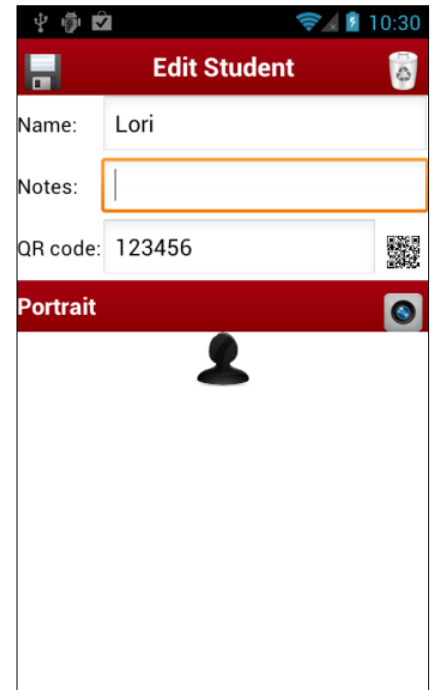


Figure 12: Student profile section



Figure 13: Calendar section page

As the user usually will have their time to take the attendance, this application has been designed so that the administrator can take their student attendance according to their planner. Based on the figure 13, it shows that the date of the attendance can be chosen manually according to their plan. Because of that, this new designed system will absolutely help the administrator to take the attendance of their students.

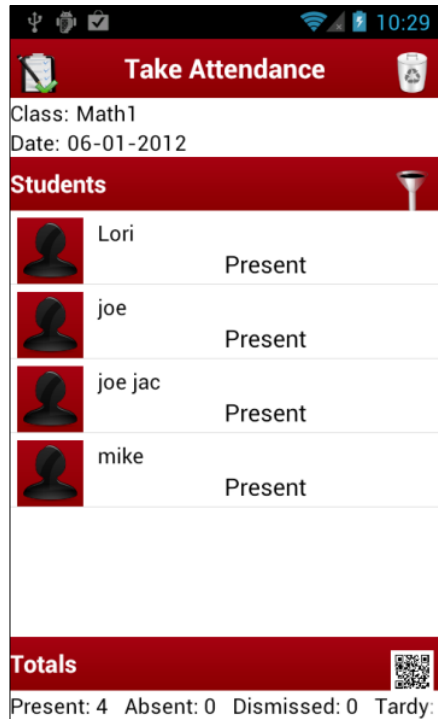


Figure 14: Taking attendance section

Finally, after all attendance has been properly recorded in the application system, the application is able to give the report for the attendance record for each of the registered students. Based on the figure 15, it shows that the application is able to give the percentage of “Present” for each of the students of the class. Furthermore, the attendance report also can be shown according to the start date and end date depending on the administrator desire. Based on the figure, there is also an icon of message which it means that the report for the attendance record can be send through an internet or email. Another option that can be selected is that the application can manage to

During the attendance taking process, the portable device will be passing around so that the student can change the state of their attendance (present/absence) and then being verified by their own fingerprint pattern. For example, if they want to change their state to “present”, they need to verify the application system by swiping their finger on the portable hardware fingerprint scanner. The state will only change if the fingerprint scanner recognised the owner fingerprint that also has been recorded in the system before.



Figure 15: Student Attendance Report

export the attendance record to a documentation format such as Microsoft Office Word and Excel. All of these purposes hopefully will help the administrator to manage their student attendance system more easily.

Figure below shows the actual result of the portable student attendance system android application running on the desired hardware (Motorola Atrix 4G).



Figure 16: Result of application running on desired hardware

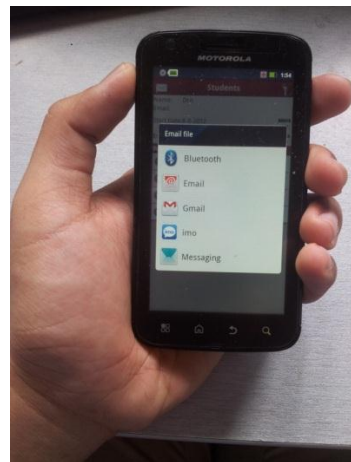


Figure 17: Attendance report exporting option

4.3 DISCUSSION

Based on the result shown before, all the background process is generated by the programming code that had been hardly designed and constructed by using Java programming language. As been mention before, the application is generated by using software called eclipse that can be downloaded from the internet.

4.3.1 Eclipse (software)

Eclipse is a multi-language software development environment that comprises an Integrated Development Environment (IDE) and an extensible plug-in system. For this software, it is mostly popular to be written in Java programming language. Nevertheless, it also can be used to develop applications in other programming languages including C, C++, PHP, Python, Android, and etc. Released under the terms of the Eclipse Public License, Eclipse Software Development Kit (SDK) is free and open source software[9]. It can be downloaded through several website on the internet or from the actual eclipse website www.eclipse.org. Please see figure 18 and 19 for the sample of the screenshot from the Eclipse v4.2.

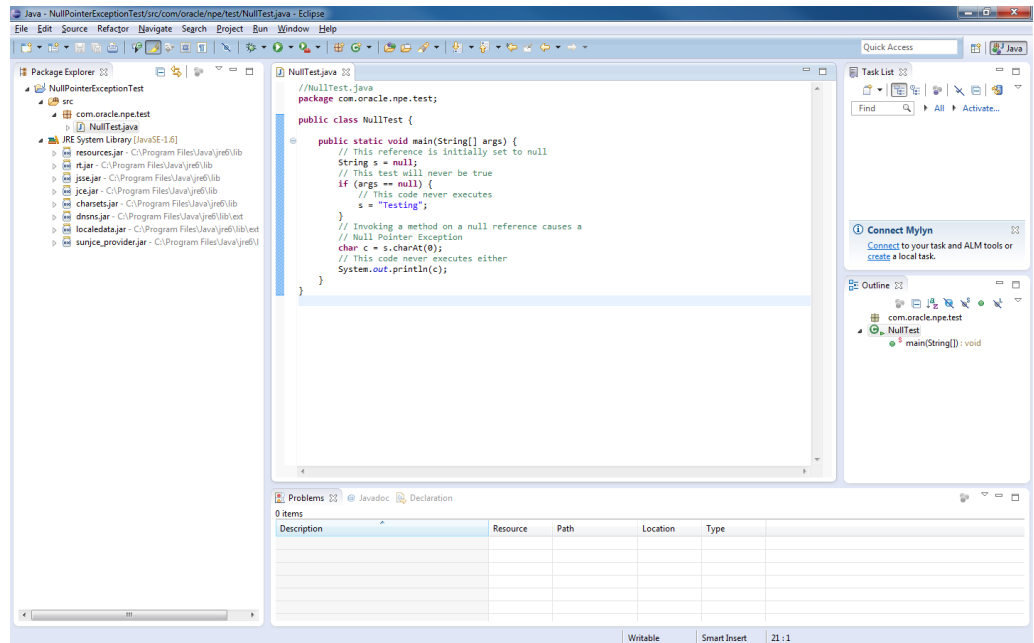


Figure 18: Screenshot of Eclipse software

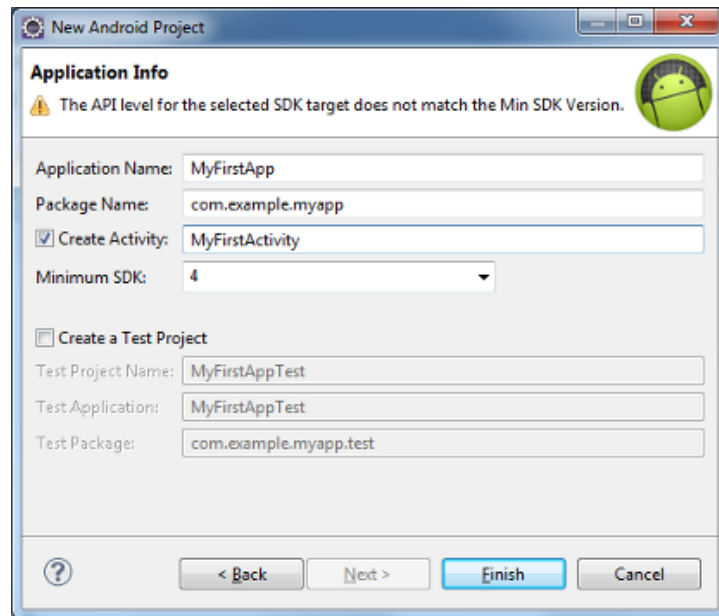


Figure 19: Develop new android application in Eclipse

4.3.2 Android SDK (Software Development Kit)

In the development of android application software, the android software development kit (SDK) includes a comprehensive set of development tools. This android SDK will include libraries, debugger, a handphone emulator, documentation, tutorials, and sample code from several simple applications. Currently supported development platforms includes the computer that running on Mac OS X, Linux, Windows XP, Vista, 7 or later [8]. There is android SDK manger software that will help the user to download all tools and android platform that are available in the internet. Figure 20 below show the screenshot of the android SDK manager which is now available for free open-source on the internet.

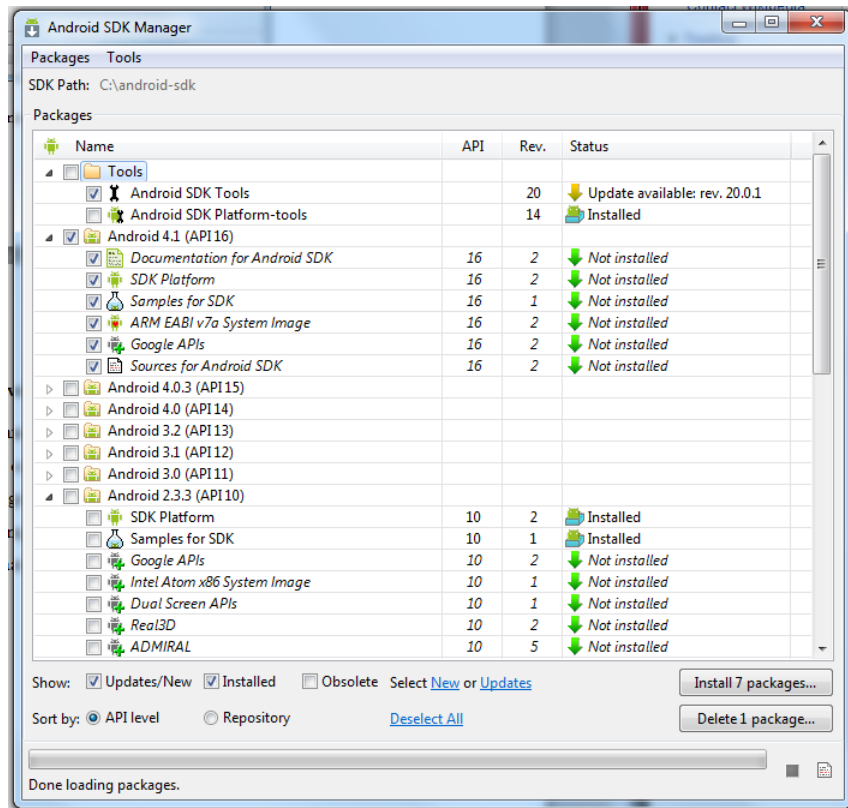


Figure 20: Android Software Development Kit manager

During the process of developing the android application, there will be easier if the programmer have the emulator of the android device on their computer so that they no need to always connected and synchronize between their actual handphone and computer. Because of that, Google has made their android SDK more useful as the user can download the android emulator machine directly over the internet for free. For that reason, it has helped the people around the world to develop application easier. Figure 21 shown below is the main interface for the Android emulator software over the computer.

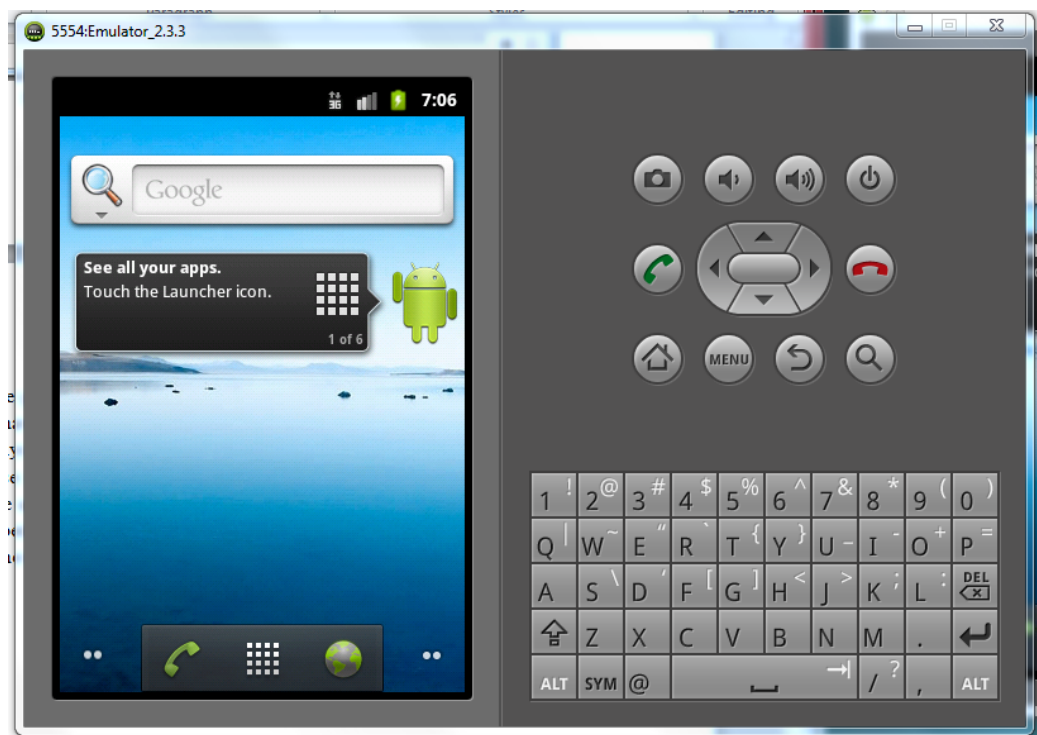


Figure 21: Android emulator main interface software

4.3.3 Motorola Atrix 4G

As this project is dealing with the fingerprint scanner device, it had been decided to choose the best device which comes from the most common things in human life which is a handphone. Currently in the market there is not so much handphone technology which integrated with the fingerprint technology. There only a few handphone brand which produce the model that is integrated with the fingerprint scanner such as Motorola and Toshiba. Thus, after a several research that had been done, Motorola Atrix 4G has been chosen as the hardware for this project as it comprises many specifications which are up to date to current technology. Please see figure below for some of the Motorola Atrix 4G specifications and features.

Key Features:

- Quad-band GSM and dual-band 3G with HSDPA and HSUPA
- 4" 16M-color capacitive touchscreen of qHD (960 x 540 pixels) resolution, scratch-resistant Gorilla glass
- Dual-core 1GHz ARM Cortex-A9 processor, ULP GeForce GPU, Tegra 2 chipset; 1GB of RAM
- Android OS v2.2; MOTOBLUR UI (update to Gingerbread planned)
- Web browser with Adobe Flash 10.1 support
- 5 MP autofocus camera with dual-LED flash; face detection, geotagging
- 720p video recording @ 30fps (to be upgraded to 1080p Full-HD)
- Wi-Fi ab/g/n; Wi-Fi hotspot functionality; DLNA
- GPS with A-GPS; Digital compass
- Fingerprint scanner that doubles as a power key
- 16GB storage; expandable via a microSD slot
- Accelerometer and proximity sensor
- Standard 3.5 mm audio jack
- microUSB port (charging) and stereo Bluetooth v2.1 with A2DP
- standard microHDMI port
- Smart and voice dialing
- Office document editor
- Active noise cancellation with a dedicated secondary mic
- DivX/XviD video support
- Lapdock and HD Dock versatility
- Web browser with Adobe Flash 10.2 support



Figure 22: Fingerprint scanner operation on the desired hardware

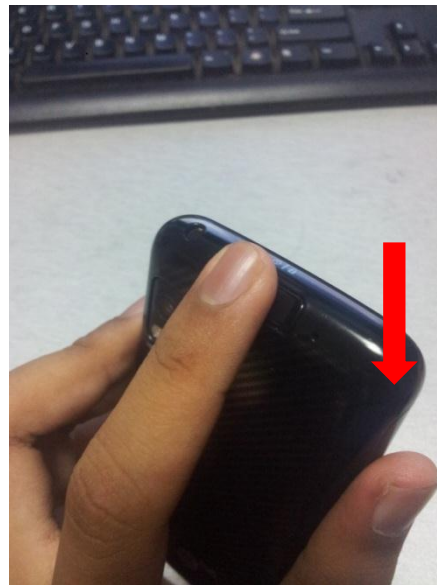


Figure 23 and 24: The technique to swipe the finger to the scanner

4.4 EXPERIMENTATION / MODELING THE PROJECT APPLICATION

After completing data gathering and analysis, there were so many ideas that can be applied to the modelling of project application. The process of modelling or developing the project application will be go through the software called eclipse and for the purpose of testing, the software application will be debug directly to the hardware (handphone). After some experimentation and modelling process, this project has been partially completed as all the explanation and some layout will be described by the figure as shown below.



Figure 25: Main page layout

Based on figure 25, the application will bring the user to another layout if the “Class” icon was selected. In order to add for the class database, there will be another layout as shown in figure 26. The user can store all their class profiles by fill in the space given. Then, there will be other useful option such as save, delete and adding the student that participate in the class.

Based on the figure 25, it shows the main page of my Portable Student Attendance Application. There will be three options which are “Class”, “Students”, and “Option”. All of this icon will bring user to the other layout that will show another option to setup the student database for the attendance taking process.

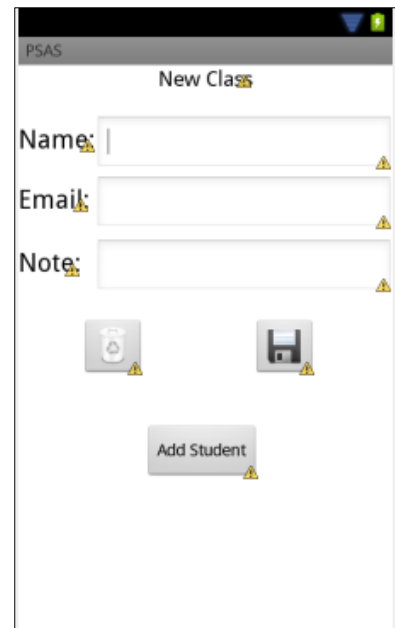


Figure 26: Adding class layout

Then, if the “Student” icon is being selected, the application will bring the user to another layout that will have option to add the database of the student as shown in figure 27. On this layout, the user can key in all the student profile by fill in the space given. Then, the user will have option either save or delete the data of the student.

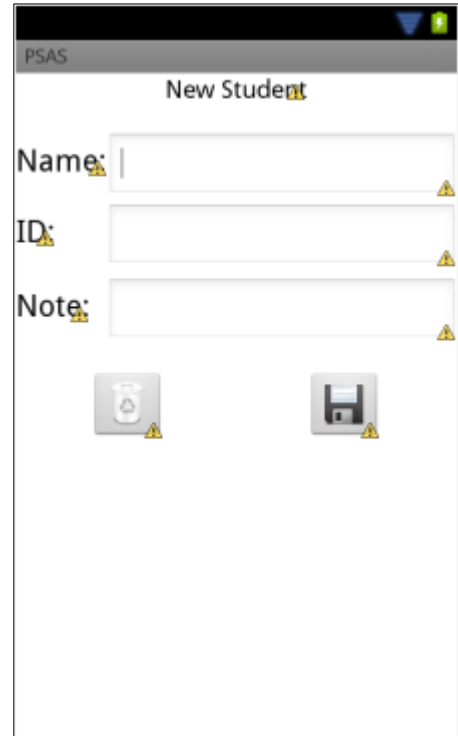


Figure 27: Adding student layout

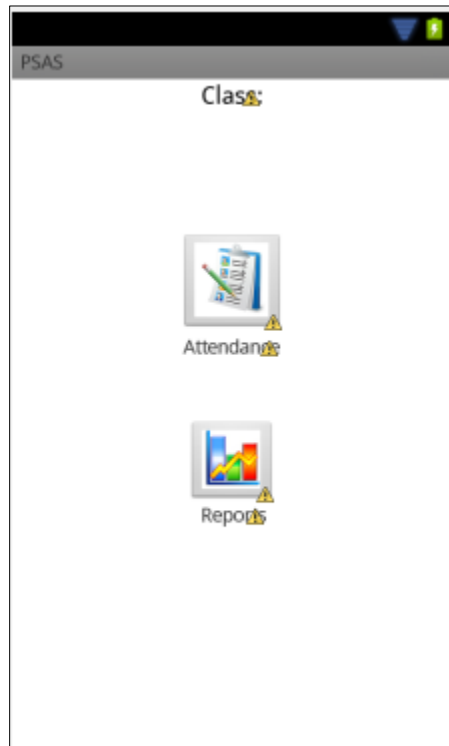


Figure 28: Class attendance and report layout

After completed the class database, there will be another layout that will give the option to the user either take the student attendance or showing the attending report of the student as shown in figure 28. Attendance of the student can be taken by selecting the “Attendance” icon and attendance report can be shown by selecting the “Report” icon.

E-mail Attendance Report

Upon completing the recording of student attendance, the attendance report from the application can be send and see through an email. The administrator can always monitor and update their user attendance record anytime and anywhere as there can modify the data either through the application or internet mail. Figure 29 shows the contents of the mail present during application system test process.

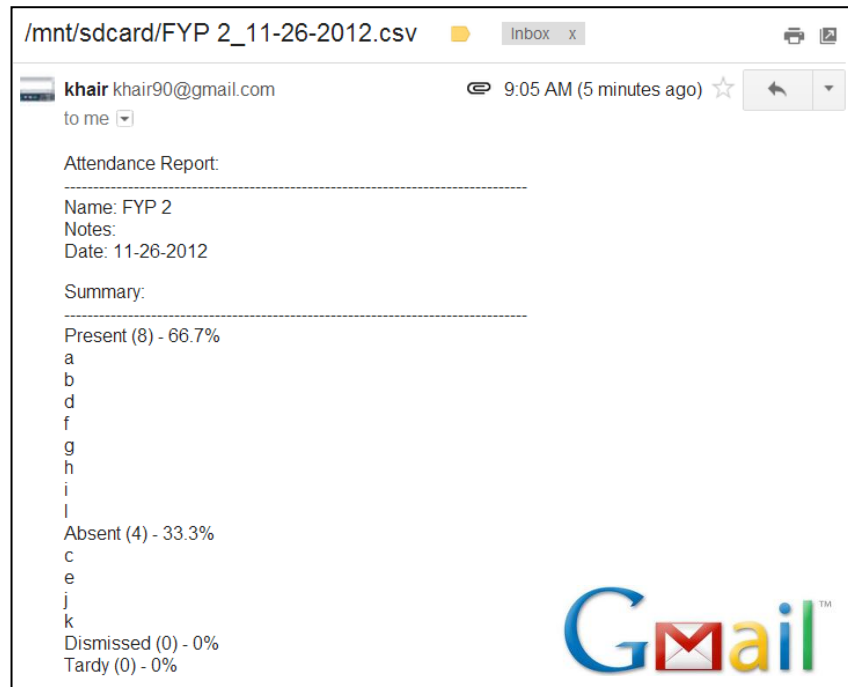


Figure 29: Email attendance report

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

Portable fingerprint attendance system can help increase the efficiency of taking the university students attendance as the current system are not enough to meet the requirement needed. The architecture of the system is to provide a simple and user friendly graphical user interface. It is able to assist the administrator of the application to always monitor and update their user attendance report through an email. This can help all the staff and the management in control the university attendance situation and in other way will help the people to become more discipline in their life. By having this study of fingerprint sensor technology and programming stuff, I hoped to expand my knowledge towards this area so that I will have the advantages of wider perspective and understandings of electrical and electronics course.

5.2 RECOMMENDATIONS

Once this project is completed, there also a few recommendations that can be recommended in other to improve the performance of the hardware and software system. In the software part, the android application must always being updated or in other word the programming code must always being checked for errors and always revises to improve the system stability and efficiency. The system could also be configured to perform automatic alert system via software notification and controlling and update the database via internet. For the hardware part, the hardware used for the designed application can be updated by checking if there any latest hardware available on the market that using latest technology and also can be suitable with the designed application.

REFERENCES

- [1] J Edgar Hoover, *Guide to Fingerprint Recognition*, DigitalPersona, Inc. 720 Bay Road Redwood City, CA 94063 USA, Retrieved from <http://www.digitalpersona.com>
- [2] Setlak, D. R, *Advances in Biometric Fingerprint Technology are Driving Rapid Adoption in Consumer Marketplace*. December 13, 2005 of Malaya 1998/1999.
- [3] Mazumdar, Subhra, Dhulipala, Venkata (2008)., *Biometric Security Using Finger Print Recognition*. University of California, San Diego. p. 3. Retrieved 30 August 2010.
- [4] Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar, *Handbook of Fingerprint Recognition*; June 12, 2009.
- [5] Diaz, Raul (2007)., *Biometrics: Security Vs Convenience*. SecurityWorld Magazine. Retrieved 30 August 2010.
- [6] Langenburg, Glenn, *Are one's fingerprints similar to those of his or her parents in any discernable way?*. Scientific American. Retrieved 28 August 2010.
- [7] L. O’Gorman, *Overview of fingerprint verification technologies*, Elsevier Information Security Technical Report, Vol. 3, No. 1, 1998.
- [8] Morrill, Dan (25 September 2009). *A Note on Google Apps for Android*. *Android Developers Blog*. Archived from the original on 25 October 2009. Retrieved 2009-09-26.
- [9] Eclipse Web Tools Platform Project".<http://eclipse.org/>: Eclipse. Retrieved 2 May 2011.
- [10] Dankmeijer, J., Waltman, J.M, and Wild, AG.D, *Biological foundations for forensic identification based on fingerprints*. Acta Morphologica Neerlandica scandivancia 18, 1 (1980),67-83.
- [11] Lin Hong., *Automatic Personal Identification Using Fingerprints*, Ph.D. Thesis, ISBN: 0-599-07596-1, 1998.
- [12] H. C. Lee, R. E. Gaensslen (eds.), *Advances in Fingerprint Technology*, 2nd Edition, CRC Press, New York, 2001.
- [13] Ronald B. Roy, John C. Sadlon, *Biometric Access Control and Time and Attendance Network Including Configurable System-on-Chip (CSOC) Processors with Embedded Programmable Logic*. Retrieved 9 September 2008.

- [14] Joseph D. Caven, *Wireless Fingerprint Attendance System*. Retrieved 2 November 2010.
- [15] M. Helmi, A. Mutalib, K. Herdawatie, M. Farhan, *Design and Development of Portable RFID for Attendance System*. Retrieved June 2010.
- [16] Burd. 2011, *Android Application Development All in one for Dummies*. United States, For Dummies.
- [17] Reto Meier. 2010, *Professional Android 2 Application Development*. New York, John Wiley & Sons.
- [18] M. Helmi, A. Mutalib, K. Herdawatie, M. Farhan, *Design and Development of Portable RFID for Attendance System*. Retrieved June 2010.

APENDICES

As this project is coming from the software development process, there will be a coding part rather than the layout interface that had been shown before. Based on Java programming language, below is one part of the full programming code that had been specially designed for the attendance system application.

```
package com.her.pfas;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.support.v4.app.NavUtils;

    public class MainActivity extends Activity {

        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            getMenuInflater().inflate(R.menu.activity_main, menu);
            return true;
        }

        public void onClickData(View v){
            setContentView(R.layout .for_class );
        }

        public void onClickData2(View v){
            setContentView(R.layout .for_student );
        }
    }
}
```

```
package com.her.pfas;

import android.app.Activity;
import android.app.AlertDialog;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
```



```

public class AddEditStudent extends Activity {

    private long rowID;
    private EditText nameEt;
    private EditText matrixEt;
    private EditText noteEt;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.new_student);

        nameEt = (EditText) findViewById(R.id.NewClass_name);
        matrixEt = (EditText) findViewById(R.id.NewStudent_id);
        noteEt = (EditText) findViewById(R.id.NewStudent_note);

        Bundle extras = getIntent().getExtras();

        if (extras != null)
        {
            rowID = extras.getLong("row_id");
            nameEt.setText(extras.getString("name"));
            matrixEt.setText(extras.getString("matrix"));
            noteEt.setText(extras.getString("note"));
        }

        Button saveButton =(Button) findViewById(R.id.btn_save);
        saveButton.setOnClickListener(new OnClickListener() {

            public void onClick(View v)
            {
                if (nameEt.getText().length() != 0)
                {
                    AsyncTask<Object, Object, Object> saveContactTask =
                        new AsyncTask<Object, Object, Object>()
                        {
                            @Override
                            protected Object doInBackground(Object...
params)
                            {
                                saveContact();
                                return null;
                            }

                            @Override
                            protected void onPostExecute(Object result)
                            {
                                finish();
                            }
                        };

                    saveContactTask.execute((Object[]) null);
                }
            }
        });
    }
}

```

```

        else
        {
            AlertDialog.Builder alert = new
AlertDialog.Builder(AddEditStudent.this);
            alert.setTitle(R.string.errorTitle);
            alert.setMessage(R.string.errorMessage);
            alert.setPositiveButton(R.string.errorButton, null);
            alert.show();
        }
    }
});
}

private void saveContact()
{
    DatabaseConnector dbConnector = new DatabaseConnector(this);

    if (getIntent().getExtras() == null)
    {
        dbConnector.insertContact(nameEt.getText().toString(),
            matrixEt.getText().toString(),
            noteEt.getText().toString());
    }
    else
    {
        dbConnector.updateContact(rowID,
            nameEt.getText().toString(),
            matrixEt.getText().toString(),
            noteEt.getText().toString());
    }
}
}
}

```

```

package com.her.pfas;

```

```

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

```

```

public class DatabaseConnector {

    private static final String DB_NAME = "Student";
    private SQLiteDatabase database;
    private DatabaseOpenHelper dbOpenHelper;

    public DatabaseConnector(Context context) {
        dbOpenHelper = new DatabaseOpenHelper(context, DB_NAME, null,
1);
    }

    public void open() throws SQLException
    {
        //open database in reading/writing mode

```

```

        database = dbOpenHelper.getWritableDatabase();
    }

    public void close()
    {
        if (database != null)
            database.close();
    }

    public void insertContact(String name, String matrix, String note)
    {
        ContentValues newCon = new ContentValues();
        newCon.put("name", name);
        newCon.put("matrix", matrix);
        newCon.put("note", note);

        open();
        database.insert("student", null, newCon);
        close();
    }

    public void updateContact(long id, String name, String
matrix,String note)
    {
        ContentValues editCon = new ContentValues();
        editCon.put("name", name);
        editCon.put("matrix", matrix);
        editCon.put("note", note);

        open();
        database.update("student", editCon, "_id=" + id,
null);
        close();
    }

    public Cursor getAllContacts()
    {
        return database.query("student", new String[]
{"_id", "name"},
        null, null, null, null, "name");
    }

    public Cursor getOneContact(long id)
    {
        return database.query("student", null, "_id=" + id,
null, null, null, null);
    }

    public void deleteContact(long id)
    {
        open();
        database.delete("student", "_id=" + id, null);
        close();
    }

```

```

    }
}

package com.her.pfas;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseOpenHelper extends SQLiteOpenHelper {

    public DatabaseOpenHelper(Context context, String name,
        CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createQuery = "CREATE TABLE country (_id integer primary key
autoincrement,name, cap, code);";
        db.execSQL(createQuery);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {

    }

}

package com.her.pfas;

import android.os.AsyncTask;
import android.os.Bundle;
import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.CursorAdapter;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.AdapterView.OnItemClickListener;

public class StudentList extends ListActivity {

    public static final String ROW_ID = "row_id";
    private ListView conListView;
    private CursorAdapter conAdapter;

```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.for_student);
    conListView=getListView();
    conListView.setOnItemClickListener(viewConListener);

    // map each name to a TextView
    String[] from = new String[] { "name" };
    int[] to = new int[] { R.id.StudentTextView };
    conAdapter = new SimpleCursorAdapter(StudentList.this,
R.layout.for_student, null, from, to);
    setListAdapter(conAdapter); // set adapter
}

@Override
protected void onResume()
{
    super.onResume();
    new GetContacts().execute((Object[]) null);
}

@Override
protected void onStop()
{
    Cursor cursor = conAdapter.getCursor();

    if (cursor != null)
        cursor.deactivate();

    conAdapter.changeCursor(null);
    super.onStop();
}

private class GetContacts extends AsyncTask<Object, Object, Cursor>
{
    DatabaseConnector dbConnector = new
DatabaseConnector(StudentList.this);

    @Override
    protected Cursor doInBackground(Object... params)
    {
        dbConnector.open();
        return dbConnector.getAllContacts();
    }

    @Override
    protected void onPostExecute(Cursor result)
    {
        conAdapter.changeCursor(result); // set the adapter's Cursor
        dbConnector.close();
    }
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.student_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    Intent addContact = new Intent(StudentList.this,
AddEditStudent.class);
    startActivity(addContact);
    return super.onOptionsItemSelected(item);
}

OnItemClickListener viewConListener = new.OnItemClickListener()
{
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long
arg3)
    {
        Intent viewCon = new Intent(StudentList.this, ViewStudent.class);
        viewCon.putExtra(ROW_ID, arg3);
        startActivity(viewCon);
    }
};
}

package com.her.pfas;

import com.her.pfas.R;
import com.her.pfas.R.id;
import com.her.pfas.R.layout;
import com.her.pfas.R.menu;
import com.her.pfas.R.string;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.TextView;

public class ViewStudent extends Activity {

    private long rowID;
    private TextView nameTv;

```

```

private TextView matrixTv;
private TextView noteTv;

@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.view_student);

    setUpViews();
    Bundle extras = getIntent().getExtras();
    rowID = extras.getLong(StudentList.ROW_ID);
}

private void setUpViews() {
    nameTv = (TextView) findViewById(R.id.textNewStudent_name);
    matrixTv = (TextView) findViewById(R.id.textNewStudent_id);
    noteTv = (TextView) findViewById(R.id.textNewStudent_note);
}

@Override
protected void onResume()
{
    super.onResume();
    new LoadContacts().execute(rowID);
}

private class LoadContacts extends AsyncTask<Long, Object, Cursor>
{
    DatabaseConnector dbConnector = new
DatabaseConnector(ViewStudent.this);

    @Override
    protected Cursor doInBackground(Long... params)
    {
        dbConnector.open();
        return dbConnector.getOneContact(params[0]);
    }

    @Override
    protected void onPostExecute(Cursor result)
    {
        super.onPostExecute(result);

        result.moveToFirst();
        // get the column index for each data item
        int nameIndex = result.getColumnIndex("name");
        int capIndex = result.getColumnIndex("cap");
        int codeIndex = result.getColumnIndex("code");

        nameTv.setText(result.getString(nameIndex));
        matrixTv.setText(result.getString(capIndex));
        noteTv.setText(result.getString(codeIndex));

        result.close();
    }
}

```

```

        dbConnector.close();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.view_student_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.editItem:
            Intent addEditContact =
                new Intent(this, AddEditStudent.class);

            addEditContact.putExtra(StudentList.ROW_ID, rowID);
            addEditContact.putExtra("name", nameTv.getText());
            addEditContact.putExtra("matrix", matrixTv.getText());
            addEditContact.putExtra("note", noteTv.getText());
            startActivity(addEditContact);
            return true;

        case R.id.deleteItem:
            deleteContact();
            return true;

        default:
            return super.onOptionsItemSelected(item);
    }
}

private void deleteContact()
{
    AlertDialog.Builder alert = new
AlertDialog.Builder(ViewStudent.this);

    alert.setTitle(R.string.confirmTitle);
    alert.setMessage(R.string.confirmMessage);

    alert.setPositiveButton(R.string.delete_btn,
        new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int button)
            {
                final DatabaseConnector dbConnector =
                    new DatabaseConnector(ViewStudent.this);

```



```

        AsyncTask<Long, Object, Object> deleteTask =
            new AsyncTask<Long, Object, Object>()
            {
                @Override
                protected Object doInBackground(Long... params)
                {
                    dbConnector.deleteContact(params[0]);
                    return null;
                }

                @Override
                protected void onPostExecute(Object result)
                {
                    finish();
                }
            };

        deleteTask.execute(new Long[] { rowID });
    }
}
);
alert.setNegativeButton(R.string.cancel_btn, null).show();
}
}

```