Fingerprint Recognition Student Attendance Management System

By

Liew Ken Nam

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology
(Perak Campus)

MAY 2015

UNIVERSITY TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Fingerprint Recognition Student Attendance Management System

Academic Session: MAY 2015

I _____LIEW KEN NAM_____
(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**: _____

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Fingerprint Recognition Student Attendance Management System**" 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 : _____

ACKNOWLEDGEMENT

I would like to present very special thanks to my supervisors, Mr. Sohail Safdar who had guide me since the beginning of Proposal Writing subject. Thanks to him for provide guidance, advice, and useful feedback regarding the final year project either in documentation parts or technical parts. In addition, guideline and briefing on how to produce a good quality report for this project also provided to me.

Next, I would like to thank to my academic supervisor Mr. Albert Yong who willing to take some time every tri-semester to advice me on how to manage time in order to finish the final year project on time. In addition, he also provides me with useful information in the way on how to study well.

Besides that, my thanks also go to my parents who keep on encouraging me while I facing problem and when I going to give up. Without their support, I may not be able finish my final year project on time. Their supports are truly appreciated.

Last but not least, my sincere thanks and appreciation go to my moderator, Mr. Lee Chen Kang who gives me a chance to express and present my final year project idea. And very thanks to him for take some time in evaluating the quality of my final year project either in the documentation part or technical part.

ABSTRACTS

This project document aims at introducing the presentation phase of a system. There are four chapters introduced in this project documents which is introduction part, literature review part, proposed method/approach part, and conclusion part. This project is about to study on biometric technologies and develop a hybrid student attendance system that based on fingerprint recognition of student in order to verify their attendance. In this system, desktop-based attendance system will be developed for student to scan their fingerprint with provided hardware for a purpose to verify their attendance in all classes. At the same time, web-based attendance system will be developed for admin/lecturer to view and analyze student attendance by generate the attendance report. The main purpose to develop this project is to replace the current traditional attendance system by provide faster, accurate, and efficient system. With this new fingerprint recognition attendance system, it can eliminate some problems such as buddy signing, loss of attendance sheet, and control student skip class rate. In developing this project, evolutionary prototyping had been applied as methodology that guides the direction of whole project development. Besides that, few fact-finding methods are used to collect the data for analysis such as survey questionnaire methods, review journals method, and observation method. This project is planned to develop using Microsoft Visual Studio 2013, Structured Query Language (SQL) Server, GrFinger Software Development Kit (SDK), and Microsoft Fingerprint Reader. Other than that, system analysis and design technique is used to illustrate necessary diagrams for purpose to illustrate the whole system in more clear way. Lastly, the implementation of this system will definitely provide more efficient, reliable, and accurate way to manage the student attendance data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Project Background

Student academic attendance is very important since it will affect the students from gaining knowledge and skills as well as their grades. This project has related about the student attendance system through the matching of their fingerprint to confirm their attendance. The main purpose of carrying out this project is to develop a hybrid student attendance system for which desktop-based application is developed to obtain the attendance of student by fingerprint and post/review the attendance results using web-based student attendance system. As we know, there is one and only one fingerprint occurs in the world for each person which will never has duplication. So, fingerprint attendance system can be known as the best authentication to detect the individual student attendance record. In addition, according to the technology nowadays, it is not unusual anymore to take the attendance of students through their fingerprint.

Nowadays, most universities and colleges are still using the traditional attendance system which requires student to sign on a piece of paper every time they attend a class throughout the whole semester. Using the traditional attendance system, we can obviously see that there are few problems such as it will be no backup for the attendance records once the lecturer accidentally lost the attendance sheet, course mate help those who did not attend the class sign the attendance which also known as buddy-signing as well, hard in analyzing and tracking student performances based on attendance factor, student lack of knowledge and skills due to the poor attendance in attending classes, and etc. It is important to overcome these problems since it will help in improving the academic performance of students as well as the teaching environment of the lecturers. Hence, the purpose of carrying out this project is to prevent unwanted situation occur and to find out the problems that causes these problems as well as find the solutions to overcome these problems.

Thus, through the problems analyzed, the objective of this project is to develop a desktop-based and web-based fingerprint student attendance system in recording their attendance effectively in every class in order to prevent student skip classes. Next, the developed system will provide the report generation regarding to the student attendance in order to assist the lecturer/staff in analyze and tracking the

student attendance. By implementing the developed system, lecturers will no more facing the empty classroom every time while they are lecturing in front the stage. Other than that, student will not be able to ask their buddy to sign for them anymore since the system requires their fingerprint to prove their attendance in the class. In addition, it will be easier to evaluate and analyze the student performance based on their attendance since the system will record the attendance more accurately and efficiently with minimum possible error. Furthermore, student academic performance will increase as well since they cannot fake their attendance through the developed system which means they have to attend all the classes in order to prevent them from get bar.

Last but not least, the system have includes several modules which are attendance module, email module, report module, fingerprint module, schedule module, lecturer module, student module, and etc. in order to ensure the system can help in improving the student attendance as well as provide the accurate and efficient information regarding the student attendance. As a conclusion, using an electronic-based system is better than using a paper-based system in order to collect, process, store, and produce the attendance results and perform long-term analysis.

1.2 Motivation and Problem Statement

1.2.1 Motivation

The motivation to develop this project is to solve some problems that are currently occurring in every colleges/universities. This project purpose is to improve the current paper-based traditional attendance management system that is still in use by many colleges and universities. From the observation, most of the problems found are normally caused by the use of traditional attendance system in these colleges and universities. Therefore, a fingerprint-based student attendance management system will be developed in order to solve these problems. The system is believed will be needed in order to improve the ways the colleges/universities in managing their student's attendance.

Since most of the colleges/universities still using the traditional attendance system, so, a bold assumption is made which most of the problems faced by these

colleges/universities are almost same. In addition, this project will be able to reduce the workload of every lecturer in key-in the student's attendance records to the system at every end of trimester since this system will record all student's attendance accurately and automatically in every classes attended by students throughout the whole trimester. Besides that, students have paid the colleges/universities in order to gain knowledge that helps them in building up their future career. Therefore, every colleges/universities must provide the responsibilities in ensuring their students will really attend all the classes for the subjects they had registered.

1.2.2 Problem Statement

a. No backup for the attendance records once the lecturer accidentally lost the attendance sheet.

Throughout the whole semester, lecturer will only record and evaluate the student attendance through a piece of attendance sheet. Almost last few weeks before the semester end, the student attendance will be key-in to the current system by lecturer in order to generate the bar-list report. But if the lecturer lost the attendance sheet, which mean the record will be lost as well and lecturer will end up unable to enter the correct attendance records to the system at the last few weeks of the semester.

b. Course mate help those who did not attend the class sign the attendance which also known as 'buddy-signing'.

Most of the time, lecturer facing a problem which the classroom is empty but the attendance list is full. It is because most of the student will only attend the class for the first few weeks but after then they will request their friends who always attend the class to help them sign the attendance. Since lecturer always busy in lecturing and have no time to check their attendance one-by-one, so students take advantages from this point to help their friends sign.

c. Hard in analysing and tracking student performances based on attendance factor.

In evaluating an individual student academic performance, it is important to review back their attendance records. It is because through the attendance record, lecturer

may easily get to know whether the student with poor performance is result from poor attendance factor or due to another factor. Without correct and accurate student attendance, it is hard to evaluate the real factor of poor performance.

d. Student lack of knowledge and skills due to the poor attendance in attending classes.

Student who absent from the class will not be able to learn what lecturer had taught in the classes which may result them to know nothing about the subject and end-up with poor academic performances. Without the student participation in a class, they may unable to absorb what lecturer teaches in the class and at the same time, it will affect the passing rate of that subject as a result.

1.3 Project Objectives

In developing this system, some project objectives had been specified. The main purpose of this project is to improve the current existing student attendance system that in use by most of the colleges/universities by develop a fingerprint-based student attendance management system. Some objectives of this project had been identified and listed below.

i. To replace the current existing student attendance system process to fully-computerized and automated student attendance system.

ii. To develop a desktop-based application that obtains the student fingerprint every time they attend the classes for attendance marking purpose.

iii. To develop a web-based student attendance system in displaying every student attendance results effectively.

iv. To generate reports regarding to the student attendance in order to assist the lecturer/staff in analyze and tracking the student attendance.

v. To eliminate the chances for student to ask their buddy sign attendance for them through the implementation of fingerprint attendance system.

vi. To provide easier method in evaluate and analyze the student performance based on their attendance since the system will record the attendance more accurately and efficiently with minimum possible error.

1.4 Project Scope

The project scope of this project is to develop a hybrid student attendance management system through the fingerprint scanning. In this project, desktop-based student attendance system will be developed for a purpose just to obtain the fingerprint of students who attend the class. In addition, web-based student attendance system will be developed for purposes to display the attendance status/condition of every student, generate reports related to the student attendance, and etc. Besides that, proper planning will be carry on in order to perform this project by using the project methodology that had been chosen.

Next, the student attendance management system will only developed for managing the student attendance status and allow lecturers/faculty staffs to easily analyze the information regarding the student attendance. In other words, it means that this attendance system will only cover the functions related to student attendance but not any others function related to another thing. So at the end of the project, a system will be developed which used for record the attendance of students more efficiently and effectively through the fingerprint scanning. The purpose to carry out this system is to overcome the current problems in their current attendance system facing by the school, college, and university. There are some modules covered in the product scope.

1.5 Module Scope

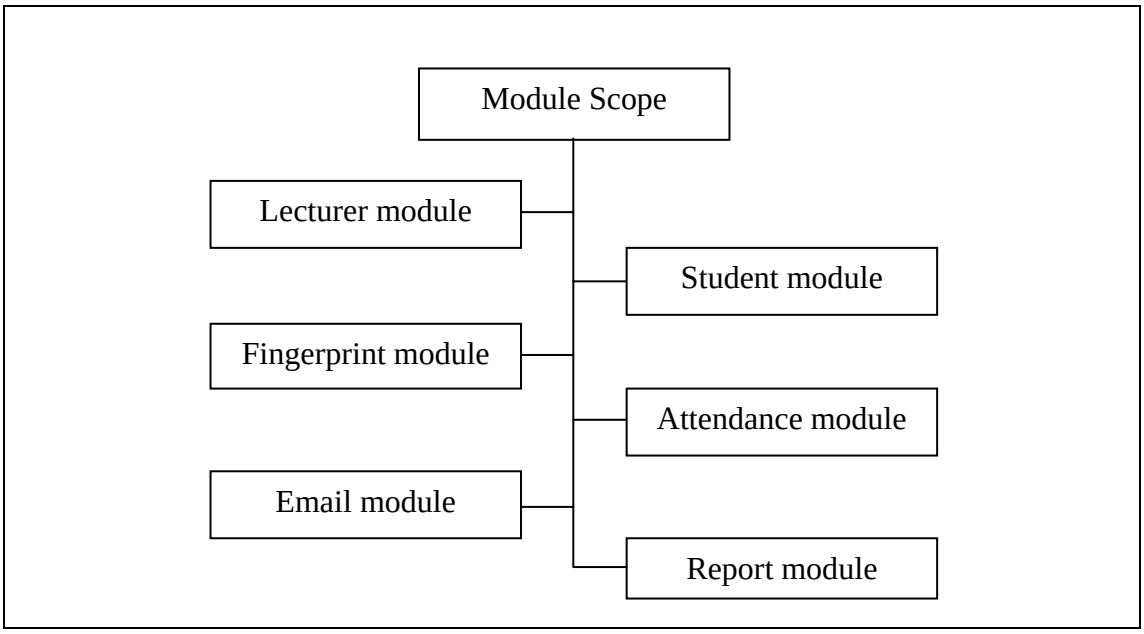

Figure 1.5-F1: Modules covered in the Project Scope.

a. Lecturer Module

In the lecturer module, it allows the user to enter the information of all lecturers in a school, college, or university. There is some functionality provided in the lecturer module which includes add new lecturer, edit lecturer information, and delete lecturer information. However, profile of the lecturer who works at the school before is unable to delete although that lecturer have leave the school or being fired. If the lecturer have leave the school or being fired, the system will allows user to set the status of that lecturer to “Fired” or “Resigned”. Besides that, information of lecturer that will be stored are includes their staff identification, first name, last name, date of birth, start work date, staff status, and etc.

b. Student Module

In the student module, it allows the user to enter the information of all students in a school, college, and university. There is some functionality provided in the student module which includes view student records, enter new student records, and update student records. In this module, users will be allowed to set the status of the student to “Withdrawn”, “Undergraduate”, and “Postgraduate”. It is for the purpose to identify whether the student is a withdrawn student, undergraduate student, or postgraduate

student. Information of students will not be remove/delete and will be kept into the system as a student history for future references.

c. Fingerprint Module

In the fingerprint module, it is used to store the fingerprint of all the student into the system for future matching while the student attend every class for the subjects they had registered before. First of all, students will require to registers their fingerprint into the system while first coming to the school, college or university after successfully applies as a student in that school. Then, every time they attend a class, they will need to scan their fingerprint through the hardware provided within a class in order to do fingerprint matching with the fingerprint record that had been stored in the system. Attendance will be automatically signed by the system once the student's fingerprint match the fingerprint record in the system. If the fingerprint of the student does not exist in the system while doing matching, system will prompt out message showing "Invalid fingerprint" whereas once the matching successful, system will also prompt out successful message showing "Attendance taken" to let the student know.

d. Attendance Module

In the attendance module, it is used to take the attendance of the student who attends the classes. First of all, for those students who late to class more than 15 minutes will be automatically known as absent by the system. It is because due to the university's policy, students are considered late and attendance will not be taken after 15 minutes. So same to this system if following the university's policy, student's attendance will not be taken for those who late for more than 15 minutes. However, if the student coming late with a reason and it is accepted by the lecturer, the lecturer will have the right to manually change the attendance status of the student through the website provided.

e. Email Module

In the email module, it is used to send the email to the student as a notification to let the student know that their attendance had been successfully taken by the system. First of all, student who scan their fingerprint while attend a class, if the system successfully matching their fingerprint in the system, email will be send to them to tell them that their attendance had been taken. It is to prevent if the system unable to

match the student's fingerprint which mean attendance not taken and student do not know about that. In addition, lecturer will be given authority to send email to those who had been barred from taking exam start from the last two weeks before the tri-semester end.

f. Report Module

In the report module, it is used to let the users generate the report regarding to the student attendance record. First of all, every time the class ended, the student attendance list record for that class will be automatically sent to the system and stored. The report will be student attendance for whole semester as well as auto calculated bar-list report at the end of the semester. It is to ease the lecturer and user in analyzing the student attendance for the whole semester. In addition, the system will automatically calculate each individual student attendance percentage for that semester and come out with bar-list report that automatically showing the name of student who had been barred. User will only need to generate the report without do much calculation and customization.

1.6 Impact, Significance, and Contribution

In the developed project, there are some contributions are identified. By implementing the fingerprint attendance system, it will definitely provide a more accurate and efficient record of student attendance record it is fully utilized by the computerized system. By using this system, users will surely realize that this system process will be faster and simpler compare to the current complex-existing system. Other than that, this system can assist lecturer in taking the student attendance more effectively without worry about losing or damaging their attendance sheets while passing it among the students around the whole class.

In addition, as mentioned earlier, most university is currently using the traditional attendance system method which requires students to mark attendance in a piece of paper and pass it among student around the whole class until every student who attends the class done signing. Although the traditional attendance system seem like very effective but it is a kind of time consuming process and will disturb the student who are concentrate in listening to the lecture as well as while in the test

situation. So, in prevent these things to happen again, in the enhanced system, the system will be developed which requires student to match their fingerprint once entered the class with the hardware provided in the class. Using fingerprint to retrieve student attendance can be considered as one of the enhanced system innovation. Other than that, this will definitely assists the school to eliminate the buddy-signing issues as well.

Besides that, enhanced system will also automatically set the status of those students who late to the class to “Absent” in order to prevent some student coming in at last 10 minutes (subject to amendment based on college/university procedures) before the class end and sign the attendance sheet and leave. Furthermore, lecturer no more need to key in the student attendance to the system by themselves at almost the end of the semester which may requires them to recheck again and again to confirm that they key it in correctly to the current system. In the enhanced system, lecturer do not need to key in the attendance by their own as the system will automatically record all the attendance status into the system by its own once the class ended.

Lastly, the contribution presented in this system will definitely help lecturer in handling the student attendance record more convenient and accurate. It will provide better teaching environment for the lecturers as lecturers will no more facing the empty classroom every time while they are lecturing in front the stage through the implementation of fingerprint attendance system. Not only that, student will also definitely increase their student academic performances. Therefore, this system has to be developed because it will help in save time from unnecessary process and promote a very accurate and efficient ways in recording the student’s attendance.

Chapter 2: Literature Review

2.1 Literature Review

A literature review can be refers to as a review of current system that the researcher had done previously and the review of the system that will be developed. Literature review also focuses on the knowledge and ideas established on a topic as well as their strengths and weaknesses. Nowadays, technology is getting better and better to replacing the traditional system to speed up the process by introducing the computerized system. There are few types of attendance system that had been introduced nowadays in school, college, and university.

2.1.1 Student Attendance Management

In the journal paper of "Student Attendance Management" (Dhanashree A. G., 2011), it had known that attendance system software is required to replace the traditional attendance system for a purpose to reduce the processing time and the paper work cost used in traditional attendance system. This journal paper main objective is to indicate the disadvantage of the traditional attendance system compared to the proposed system. In addition, the journal paper had also indicated the characteristic of the proposed system. On the other hand, from the journal, it had shown that the proposed system only provide a very simple application with some features that allow the users to manage their student attendance more easily and effectively.

As mentioned in the journal paper, it did not include any special hardware (such as fingerprint scanner, mobile phone, bar code scanner, and etc.) in order to make the software work. It only requires basic equipment such as a set of desktop computer. In short, the proposed system only requires the users to install the software to their laptop/desktop for managing their student attendance. Based on the journal paper, there are two modules introduced which is admin module for managing the classes and report module for generating the attendance report.

2.1.1.1 Strengths:

From the journal paper, there is some strength that can be found in the desktop-based attendance system. First of all, the system can be known as a computerized system that reduces the workload of the users in managing the attendance records. Compared to the traditional attendance system, it offered cost-effectiveness since it reduces the paper needed to process the data of the student attendance records. In addition, it will also help the lecturers save up a lot of time in recording the attendance of the students throughout the whole semester.

In addition, since the proposed system does not require any special hardware to make it workable, so it will be very easy to use due to its simplicity. At the same time, it will reduce the training cost required to train the user in operate that student attendance system. Besides that, it only require local database to store all the information of the students attendance which mean the lecturers do not need to worry about the network coverage issues. Other than that, as mentioned in the journal paper, the attendance system was designed in extremely simple graphical user-interface which allows user to easily deal with the system.

2.1.1.2 Weaknesses/Limitations:

However, there is some drawback that can found from the solution in using desktop-based student attendance system. First of all, the system will still require the lecturers to key-in student attendance records in every class. Although the system had developed to be work in computerization, but it still require computer operator to manage the storing and retrieving of information frequently. So, the proposed system had only converted the traditional attendance system to computerization but everything still have to be done manually such as key-in student attendance and etc.

In addition, it only reduces the cost used for paper work but the system does not save up much time from record the student attendance. Other than that, those kind lecturers may try to help those student who always absent from the class to prevent get barred from the final exam by fake the student attendance by their own since lecturers is the one who key-in the attendance records in every class. Other than that, the proposed system only provides very basic functions to be used by the lecturers.

2.1.2 RFID Based Attendance Management System

In the article paper of "RFID Based Attendance Management System" (Microtronics Technologies, 2013), it had known that the attendance is needed to be taken in several places like school, college, university, and workplaces. This article paper main objectives had concerned about to replace the old traditional attendance system technology with Radio Frequency Identification (RFID) technology. It is carry out to overcome some existing problems occur in the traditional attendance system. In the article paper, it also mentioned that the RFID system is developed and is suitable to take the attendance of the students as well as employees. There are two modules introduced in the article which includes reader module and RFID module. In details, each student/employee must have a valid RFID card of RFID tags with them in order to communicate with the RFID reader placed on their workplace/school.

As mentioned in the article, the RFID reader will automatically detect the student/employee attendance and record it while the RFID card gets closer to the RFID reader which means it is using the non-contact type of reader and passive types of card. From the article paper, we can get to know that the attendance system using RFID technology is much better than the traditional attendance system in school/workplace as almost whole the system is done in automation and with high transparency process.

2.1.2.1 Strengths:

From the article paper, there is some strength that can be found in the attendance system that using RFID technology. First of all, the system can be known as a fully-automated system which requires less human interaction in the attendance record process. It is because the student/employee will just need to bring their RFID card every time they attend classes or check-in for works. Compare to the existing attendance system, it do not requires the student to pass the attendance around the whole class which cause time consuming as well as do not requires worker to write down their check-in and check-out time as well.

In addition, as mentioned in the article paper, the user will be able to view the attendance lists on the spot through the computer interface or view remotely through

another computer. It is because the database will be automatically updating the attendance status once the reader read the card. Besides that, in using the RFID technology, the information provided by the system will be more accurate than manually record as it is recorded by the computer to the system. At the same time, it will be able to avoid the student from signing for their buddy who did not attend the class as every student will have their own unique RFID card.

2.1.2.2 **Weaknesses/Limitations:**

However, there is some drawback that can found from the solution in using RFID technology in the attendance system. First of all, the system will require the student/employee to bring the RFID cards always with them while they are in class or workplace in order to check-in or check-out for the attendance. If the student/employee lost their card, they may need to go to the office to make a new card which will cause them to pay for the lost as well as waiting for the new card to be generated and pass to them.

As we know, student will normally attend a few different classes per day which mean they will need to tell the lecturer if their card had been lost and will require them to report to the faculty by themselves with lecturer approval as evidence in order to gain back their attendance as the system do not provide a manual key in function for the lecturer which may lead to complex process. However, although the system can help prevent the student from making fake attendance but nobody can guarantee that the student will not take their friend's RFID cards along with them to the classes which mean it is still get the same result which scanning the cards on behalf of their buddies.

2.1.3 Bar Code Scanner Based Student Attendance System (SAS)

In the journal paper of "Bar Code Scanner Based Student Attendance System (SAS)" (Subramaniam H. et al., 2013), it had known that student attendance and participation among a class is very important in order to achieve good academic outcome of a student and school. This journal paper main objectives had concerned about to replace the non-automated attendance record system with the barcode scanner technology in order to record and manage the student attendance records more efficiently and effectively. As mentioned in the journal paper, RFID-based technology and biometric-based technology is sometime too costly to implement into a school since it requires purchasing of certain hardware in order to get the system work. Compare to both RFID-based technology and biometric-based technology, barcode technology obviously shown that it is cheaper than both the technology. So through the journal paper, barcode scanner attendance system had been introduced to improve the admin staff managing process such as process daily, weekly and yearly student attendance report.

In the barcode scanner technology, student will be issued a student card for each of them with the barcode displayed on the card for a scanning purpose every time they attend the classes. Student attendance status will be automatically checked and record into the system once lecturer scan their student card with barcode scanner. From the journal paper, we can get to know that the attendance system using barcode scanner technology is much better than the traditional attendance system in school as the lecturer just requires to scan the barcode of the student cards as prove that the student attend the class.

2.1.3.1 Strengths:

From the journal paper, there is some strength that can be found in the attendance system that using barcode scanner technology. First of all, the system provide a report modules which allow the lecturer to generate daily, weekly and monthly report that do not requires them to manually calculate the percentage of attendance of each individual student. This is definitely will speed up the processing speed of report generation as compared to the current existing system as well as assist

the lecturer/admin staff in obtain more accurate and efficient student attendance information.

In addition, as mentioned, the barcode scanner system also provide a functionality which is if the student does not meet the attendance requirements, the system will automatically generate warning letter to the student to be deliver to their parent. Besides that, barcode scanner requires less cost development compare to the RFID technology and biometrics technology as the hardware equipment cost of the barcode scanner system is slightly cheaper than both the hardware equipment cost of RFID technology and biometrics technology. Other than that, it can minimize the mistake that may occur by human errors since it is almost a fully-automated system which just highly depends on the barcode scanner.

2.1.3.2 **Weaknesses/Limitations:**

However, there is some drawback that can found from the solution in using barcode scanner technology in the attendance system. First of all, the system will require the student to have their student card with the unique barcode displayed on their student card every time they attend a class. So sometime if the student forgot to bring their student card along with them while go to the class which may cause the lecturer will need to go to the office to ask the admin staff to change their student attendance status. It is because lecturer is not authorized to manually key-in the student attendance as the system will automatically update the student attendance status into database after scan the student card.

In addition, although the system allow the admin staff to generate warning letter if the student does not meet the attendance requirements, however, the generated warning letter will be given to the student and then pass it to their parent through the student themselves without giving an instant message or email to their parent. So, student may end up with just throw away the letter and pretending that they already surrender the warning letter to their parents.

2.1.4 Integrated System for Monitoring and Recognizing Students during Class Session

In the journal paper of "Integrated System for Monitoring and Recognizing Students during Class Session" (Mohammad A. et al., 2013), it had known that the attendance system using face recognition is more efficient than other student attendance system methods. As mentioned in the journal paper, it is because they believe that the face recognition method is the most accurate and fastest method among the biometric attendance system. The journal paper main objectives had concerned about to replace the manual attendance record system with the face recognition technology in order to eliminate the waste of using paper and response time from students.

As mentioned in the journal paper, face detection is the best among all the biometric attendance system because face can represent the identity of an individual. In the journal paper, it has mentioned that the picture of whole class will be taken by classroom's camera and upload to the system to do face filtering and then the attendance of the student will be checked automatically by the system once the face matching of a student is successfully performed. The system also allows the lecturer to drag and drop their student's picture into the system if the system failed to recognize their student face. From the journal paper, we can get to know that the attendance system using face recognition technology is much better than the manual attendance system in school as the lecturer just requires to active the classroom's camera that allocated inside the classroom in order to capture the picture of whole class.

2.1.4.1 Strengths:

From the journal paper, there is some strength that can be found in the attendance system that using face recognition technology. First of all, the system provides a high security feature since the face recognition will only start to record attendance by perform face filtering after the lecturer has login using their ID and password and capture the picture of whole class. Every student attendance will not be recorded before the lecturer uploads the picture and it makes the system security

become more convenient. Next, sometimes the student face may not be recognize by the face detection system but the system provide a good features which let lecturer to upload the student photo that used to match the photo of the student stored in the database. By having this features, student who does not recognized by the face detection would also be able to take the attendance.

In addition, as mentioned in the article paper, the system will reduce the percentage of fake attendance or maybe almost zero chances for student to make a fake attendance since everyone have a different look as their unique identity. Because of this, lecturer does not need to double check the attendance every time in order to prevent fake attendance as the face recognition can solve this problem automatically.

2.1.4.2 **Weaknesses/Limitations:**

However, there is some drawback that can found from the solution in using face recognition technology in the attendance system. First of all, the system will require the lecturer to manually upload photo to the system in order to complete the attendance through face recognition process. Since nowadays people talking about mobility and automation, manually control is not a best solution for the system as it still can be improved. Next, although it has provided the report features, but the report only can produce immediately after the attendance has done. User is not able to print again if the report is lost. Although the record is backed up to the database server, but there is no web services provided for the user to print the report and it cause the user have to manually retrieve the data from the database server if they want to print the report again.

In additional, the hardware for the face recognition is too expensive to implement at the every class as it makes the system become invaluable anymore. The university has to prepare one CCTV hardware and extra computer to handle the face recognition as the lecturers need to use it to upload the attendance records. Besides that, face recognition attendance system sometime may unable to capture all the face in the classroom as well as if a class got twins brother, the face recognition attendance system will more likely to match the wrong student identity.

2.1.5 Wireless Attendance Management System based on Iris Recognition

In the journal paper of "Wireless Attendance Management System based on Iris Recognition" (Kadry S. et al., 2010), it had known that iris recognition is one of the biometrics method that is a very reliable identification methods for every person. From the journal paper, it had mentioned that iris recognition was once ago still a science fiction. Due to the rapid evolvement on technology, iris recognition had successfully implemented to some applications for use in verification and identification. Reason of implementation of iris recognition in attendance system is because of it can obtain easily, unalterable, unique, and etc. like any other biometrics method. The journal paper main objectives had concerned about to apply the biometrics to wireless attendance management system to make the user's attendances more easily and effectively.

From the journal paper, we can get to know that iris recognition is one of the ways to make the attendance of students/staffs to be manageable in more effective and automated ways. In the iris recognition, the system will requires to acquire the iris image of students/staffs and store inside the database after go through several process by the system. The journal paper also had mentioned that the use of traditional methods in attendance management system such as magnetic card, barcode scanner, identification card, and etc. is unable to avoid those students/staffs to fake their attendance since cards can be easily separated from the owners.

2.1.5.1 Strengths:

From the journal paper, there is some strengths can be found in the attendance system that implement the iris-recognition feature. First of all, the greatest advantage of this method is that iris recognition is irreplaceable and it is unique for everyone. Other than that, iris is placed behind the cornea and eyelid which make it always protected from the external environments not like other biometric methods. For example, people may accidentally damage their face, voice, and etc. due to any reason which may cause the system unable to recognize him/her anymore. In this situation, it will cause the user unable to make use of the data anymore due to the verification failure.

Besides that, implementation of iris-recognition in the attendance system will definitely improve the attendance of the staffs/students in class/company. In short, for the long-term effects, it will definitely help improve the staff morale, increase the work efficiency, and create a good business image of the company. On the other hand, for students, it will help in reduce the poor attendance of the student and make them attend all the classes obediently. In addition, the journal paper had mentioned the implementation of wireless technique (Bluetooth, Wi-Fi, and etc.) rather than the wired-based which can help in reduce the risk of malfunctions of the background wired-based machine as the iris-recognition will be performed wirelessly.

2.1.5.2 **Weaknesses/Limitations:**

However, there is some drawback that can be found in the solution in implement the wireless iris recognition attendance system. First of all, the startup cost in implementing this attendance system to the company/university will be a burden as the hardware devices for the iris recognition is quite expensive compare to any other biometric recognition devices. In addition, although wireless technique is used in this attendance system which means to be replace and reduce the risk of malfunction of wired-based machine. However, implementation of wireless technique will requires high transmission speed of network in order to verify the students/staffs identify. Without the high transmission speed, it will take time to verify the students/staffs identities as the data will continuously transmit between the data server and the device for attendance verification.

2.1.6 A Low-cost Remote Attendance Tracking System for Developing Regions

In the journal paper of "A Low-cost Remote Attendance Tracking System for Developing Regions" (Reda A. et al., 2011), it had known that tracking attendance of the staffs/students is a very important consideration in improving the work efficiency of the employees, academic results of the students, and etc. From the journal paper, it had mentioned that many existing systems only focused on tracking the attendance of students or patients rather than the lecturers or workers. Besides that, the attendance system techniques introduced in the journal paper are combination of voice recognition technology and location tagging in obtaining the attendance of the remote staffs. The main objectives of the journal paper had concerned about tracking the attendance of remote agents (such as lecturers, employees, and etc.) rather than the participants (such as students, patients, and etc.) by using low-cost methods.

As mentioned in the article, the system will be developed in mobile platform in order to reduce the hardware cost in obtaining the attendance of the remote agents. In addition, location tagging tools will be implemented to the system as well to track the location of the remote agents while they make verification for their attendance. Other than that, among all the biometric forms, voice recognition can be known as the less privacy-sensitive forms of verification.

2.1.6.1 Strengths:

From the journal paper, there is some strength that can be found in the attendance system that using voice recognition technology. First of all, the attendance system mentioned in the journal paper is for tracking the attendance of remote agents rather than the students. The system was developed in the mobile platform in order to reduce the deployment costs which mean the company do not need to supply additional hardware devices in order to mark the attendance of the remote agents which provide them cost-effectiveness. In short, remote agents only need to verify their attendance through their own mobile devices by send their voice and with the combination of location tagging tools which also offer them with simplicity in using the system.

Besides that, voice recognition is known as the less privacy-sensitive forms of verification compared to other biometric forms such as fingerprint, palm-print, and etc. So, remote agents do not need to worry so much about the loss of privacy issue in sending their voice for verification.

2.1.6.2 Weaknesses/Limitations:

However, there is some drawback that can be found from the solution in using voice recognition technology in the attendance system. First of all, the system will require the remote agents to have a smart phone in order to verify their attendance since the system was developed in mobile platform. Other than that, the cellular network must be available in using this system since it will take the attendance remotely and Global Positioning System (GPS) will be required in order to track the location of the users while they verify their attendance. Although nowadays mostly everyone will have at least a smartphone device, but there is still small area of people who do not have smartphone as well as the network coverage issue. Hence, this will cause difficulty for those people who do not have smartphone or cellular network to verify their attendance.

Other than that, it is still possible for voice recognition technology to occur errors in verification as voice can be easily affected by the body condition. For example, sore-throat may temporarily affect a person's voice which causes that person unable to verify his/her attendance and noise in the surrounding may also increase errors to the verification as well. Therefore, voice recognition is not the best biometric forms to be apply to the system in tracking the attendance of the students in the university as biometric methods must guarantee the permanency of the data.

2.1.7 Wireless Fingerprint Based College Attendance System Using Zigbee Technology

In the journal paper of "Wireless Fingerprint Based College Attendance System Using Zigbee Technology" (Talaviya G. et al., 2013), it had known that the attendance system using fingerprint recognition is more efficient than the manual attendance system since it provide more automation in managing the attendance of the students. As mentioned in the journal, previously, there are some works such as RFID-based and facial recognition had been done in improving the method of managing the student attendance, but, cost of hardware of facial recognition method is very high compared to the fingerprint recognition method. The journal paper main objectives had concerned about to make use of fingerprint-based attendance system in marking the attendance of the students who attending the classes for purpose to save the time taken to record down every students attendance and fasten the report generation.

In the fingerprint recognition technology, students will require to register their fingerprint into the database for future matching while they attend every class. Other than that, hardware required in implementing the fingerprint attendance system can be easily acquired compare to other biometric method as fingerprint recognition is very common in use nowadays in the community.

2.1.7.1 Strengths:

From the journal paper, there is some strength that can be found in the attendance system that using fingerprint recognition technology. As mentioned in the journal paper, fingerprint can be known as one of the most accurate and successful biometric forms for verify identity of students. First of all, the system had provide the report generation module which will automatically send to few parties such as Head of Department (HOD), lecturers, and student's parent regarding their attendance status in the specified days. Due to the provided report generation features, it can help speed up the report generation without needing the lecturers to key-in the attendance of the students before the system allows them to generate the report.

At the same time, it will also help in reducing the errors occur for the report generated as the report will be compiled automatically based on the attendance record obtained in very class. In addition, due to the uniqueness of the fingerprint, buddy signing will be able to eliminate since they will be unable to mark the attendance on behalf of their friends.

2.1.7.2 Weaknesses/Limitations:

However, there is some drawback that can found from the solution in using fingerprint technology in the attendance system. First of all, the journal paper had mentioned that they did not provide manual key-in feature in their attendance system which means those students who come late to the class with reason will still consider absent since manually key-in is not provided. Other than that, the journal paper does not mention that the admin can set the period that considers the student is late to class. The journal paper only mentioned that the system will not update the student attendance record that came late to the class after 20 minutes by default. However, not all university/college have the same policy which mean some university/college will consider their students late to class after 15 minutes. Since this feature is not included in the system, then it will still cause the inaccuracy data in the student attendance records.

2.2 Overcome the Limitations/Weaknesses

After gone through the literature review of the few attendance systems, several limitations or weaknesses of the previous researcher's solution had been identified. In the proposed system, the limitations and weaknesses found from the literature review will be improved and enhances in order to make the student attendance system operate more efficiently and effectively in assisting the lecturer to take attendance of the student.

One of the main weaknesses/limitations that can be identified from these solutions is the unique identifier of the students which mean major attendance system currently using in the school is highly based on the student card which may easily lost by the student. Hence, in my proposed system, the student attendance will be recorded by scan their fingerprint every time they attend a class. As we know, fingerprint can be considered as the most unique identifier of an individual that will never be duplicated or change owner. Through the scanning of student's fingerprint, student will no more worry of losing their student card and causes them to unable to verify their participation in a class. In addition, fingerprint provides a high recognition compare to other biometrics attendance system as the mistake occur is very minimum in detect the student's fingerprint.

Besides that, one of the weaknesses/limitations that can be identified from these solutions is fake attendance by student's friends which mean sometime those students who absent from class will request their friends who attend the class to sign for them. It is also can be known as buddy-signing. Since my proposed system is going to take the attendance of student through their fingerprint scanning, so it is hard and can be considered as no chance to let the students to fake their attendance as the fingerprint of each person is totally unique unless the student who absent from class cut off their finger and surrender it to their friends for attendance.

Next, one of the weaknesses/limitations that can be identified from these solutions is report generation function. From these literature reviews, it had been identified that their system just provide a report regarding to the student attendance status but did not mentioned about the bar list report and etc. So, in my proposed system, it will be included the report function as well. Once the class it finished, the system will automatically send the student attendance status report of that class to the

faculty. And almost at the end of the semester, user will be able to generate the bar-list report based on the student whole semester attendance status and the system will automatically calculate the attendance percentage of each individual as well as highlight the name of student that been get barred due to the poor attendance. Other than that, user will also be able to generate daily, weekly and yearly report.

Then, one of the weaknesses/limitations that can be identified from these solutions is student lateness detection. Through the literature review, it had been identified that their system only provides two attendance statuses in their system which are present and absent. So in my proposed system, it will also only have two status as their system but once the student come in to the class in 15 minutes late, the system will automatically assuming that they skipped the class. So, attendance will not be recorded for those students who late to the class for 15 minutes as according to the college/university policy. However, chances will be given to those students who late with reason by allow the lecturer manually make correction to the attendance with reason provided before the class end.

Last but not least, one of the weaknesses that can be identified from these solutions is email services which refer to the email send to the parent or student when they absent from class or once they attend the class. Once the student attendance is verified through their fingerprint scanning, one email will be send to the student to tell them that their attendance had been verified whereas if the student attendance status is still in absent status at the end of the class, one email will be send to their parent for a purpose to inform their parent that their son/daughter is absent from the class.

2.3 Comparisons between Existing Solutions and Proposed Solutions

Comparisons / Systems	Unique Identifier	Eliminate Buddy-Signing	Student Lateness Detection	Accurate and efficient attendance records	Save Time	Email Notification	Report Generation
1. Student Attendance Management	NO	NO	NO	NO	NO	NO	NO
2. RFID Based Attendance Management System	NO	NO	YES	YES	YES	NO	NO
3. Bar Code Scanner Based Student Attendance System (SAS)	NO	NO	NO	YES	YES	NO	YES
4. Integrated System for Monitoring and Recognizing Students during Class Session	YES	YES	NO	NO	NO	NO	YES
5. Wireless Attendance Management System based on Iris Recognition	YES	YES	NO	YES	YES	NO	NO
6. A Low-cost Remote Attendance Tracking System for Developing Regions	YES	NO	NO	NO	NO	NO	YES

7. Wireless Fingerprint Based College Attendance System Using Zigbee Technology	YES	YES	NO	YES	YES	NO	NO
8. Fingerprint Student Attendance Management System (Proposed System)	YES	YES	YES	YES	YES	YES	YES

Table 2.3-T1: Comparisons between Existing Solutions and Proposed Solutions

2.4 Fact Finding

There are many types of fact-finding method that can be used to gather useful information that help in analyzing the areas that related to the system the developer planned to develop. Fact-finding is a very important technique that is required to uncover important information and assist people in making quality decision. Fact-finding methods are classified into several categories which includes questionnaire, observation, interview, survey, review written sources, and etc. These methods are used to collect the data the researcher needed to perform analysis. In carrying this project, several fact-finding methods such as survey questionnaire method, review written sources method, and observation had been applied to collect the important data.

2.4.1 Observation

The first method used in carrying out this project is observation method. Observation method is a method used to collect information based on observation. It is a fundamental way that helps observer to find out the information or situation in their surroundings. Other than that, observation is always based on the natural sense and consideration after someone observes something. Through the observation, it can allow observer to find out some information that they may not be able to gather through another fact-finding method that based on probability or report generation.

There is some strength of using observation method which one of the strength is that the observation gives observer chance to consider about the problem or situation that existing currently rather than rely on some kind of report. Sometime people may refuse to provide accurate and efficient information using other fact-finding methods which lead to possible data errors. Through the observation, they can found out that the information they observed can be more trusted compare to data analysis. It is because observer will take time to think about the problem and find a solution to resolve the problems.

In this project, observation method had been applied to observe the current problems faced in using the traditional student attendance system. From the observation, the attendance paper sometime can be lost while passing around the

whole classes which at the same time causes the lecturer lost their whole tri-semester student attendance records and make them unable to key-in the student attendance records accurately to the current existing system since the attendance paper had lost. In addition, buddy-signing behavior is detected through observation method.

2.4.2 Review Written Sources

The second method use to obtain the information about the related study areas is reviewing written sources. It is a method of reviewing current existing work or literature review that had been done by other researchers regarding the project. Written sources can be as the report generated, result of previous research by others, official files, research paper, and etc.

There is some strength that can be found in using the review written sources method. One of the strength is that the researchers do not need to start everything from zero in analysis something related to their study areas. For example, if data needed by the researcher is very hard to obtain, then by review the existing written sources that other researcher had done previously, they may be able to get the information they need from the previous work easily. This method definitely will help save up a lot of time and reduce work efforts. In addition, researcher can easily obtain the written sources through the internet nowadays which definitely will cost them very low compare to other fact-finding methods.

Therefore, in this project, review written sources method had been applied by reviewing the literature review regarding the study areas. By review the written resources, it help to obtain the problem that currently faced in similar study area. In addition, it can help increasing the inspiration through other researcher's written documents. One of the problems detected in current student attendance system through this method is that many college/university still using the paper-based traditional method that is inaccurate and inefficient.

2.4.3 Survey Questionnaire

The third method use to obtain the information about this project is survey questionnaire method. Survey questionnaire method is a method that can be performing face-to-face through distribution of set of survey questions or through online. Those who take the survey are allowed to keep their identity anonymous. It usually requires having sample population of people to take the survey in order to obtain the result accurately. The larger the total number of people who take the survey, the more accurate the result will be. The result obtained then will be perform probability calculation to make it easier to analyze and turn into final useful information.

There is some strength that can be found from the survey questionnaire method which one of the strength is that the surveyors do not need to know who has respond to the survey. They do not need to record the personal data who had taken the survey one-by-one. In addition, those who take the survey will not need to worry while they answering the question since their identity can be in anonymous. People are tends to answer question more honestly when their identity is not exposed. In addition, survey information allow surveyor to turn it into sampling data that can be easily analyze through some data sampling technique.

Therefore, survey questionnaire method will be applied to this project in order to make the data analysis can be perform easily. In using survey method, it can help to fasten the process of gather information compare to observation since many people can perform the survey at the same time while observation required more time to truly observe the thing clearly with no doubt. Survey method has help in knowing the student opinion more clearly regarding the current existing system.

2.5 Data Collection

Data collection can be referred to as the process use to collect and measure relevant data needed to justify the research problems, significance of the solutions, and evaluate outcomes. Data collection is a very important stage in carrying out a research. Without performing data collection, no matter how good the system is designed, the researcher may still not be able to complete the research. There are two types of data in the data collection which named quantitative data and qualitative data.

According to the survey that had conducted, 40 sets of questionnaire (15 questions per set) were prepared for students to fill up in order to obtain the data regarding the student attendance system. The questionnaires are distributed to few different faculties among students and are conducted inside UTAR. Questions inside the questionnaires are prepared to focus/analyze on few things which include the problem analysis, current attendance system analysis, and new fingerprint-attendance system analysis. Through the data collected, analysis was carried out justify the problems and need for solutions.

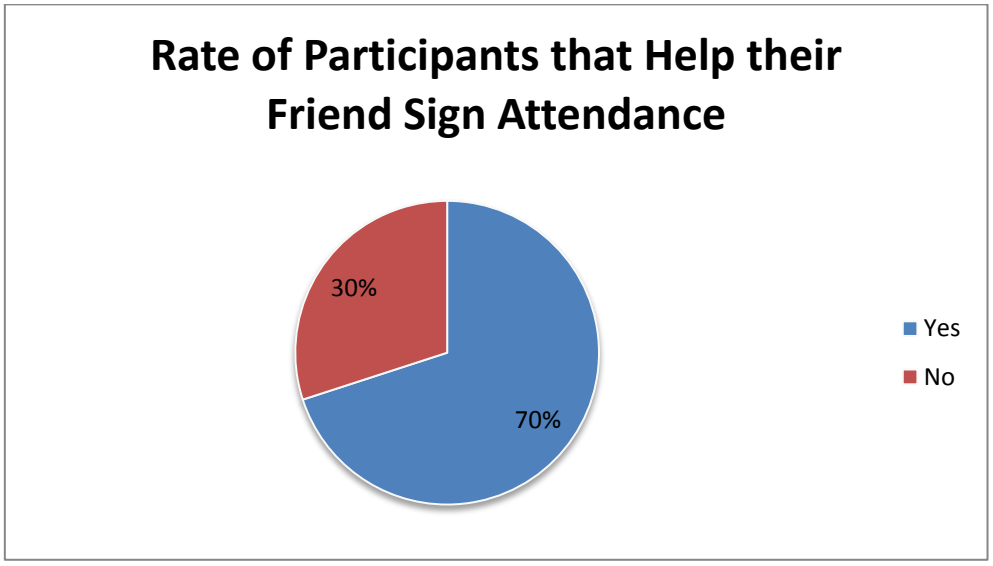

Figure 2.5-F1: Rate of Student who helps Friend Sign Attendance

According to the Figure 2.10.1, the data collected shows that the rates of participants who help their friends sign the attendance while they absent from class had reach up to 70% while only 30% of them emphasize that they never sign for their friends.

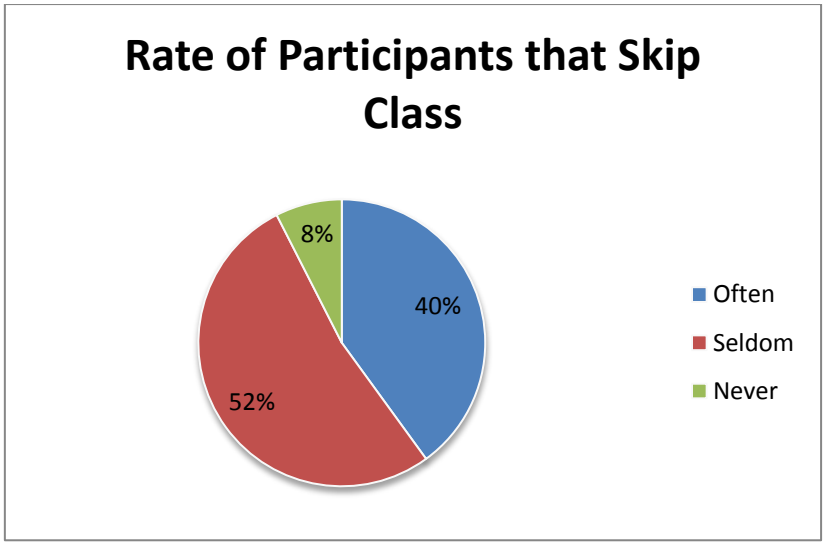

Figure 2.5-F2: Rate of Participants that Skip Class

Next, the information collected shown that 40% of them are always skip class. While 52% of them emphasize that they seldom skip class following by the 8% of students emphasize that they never skip class.

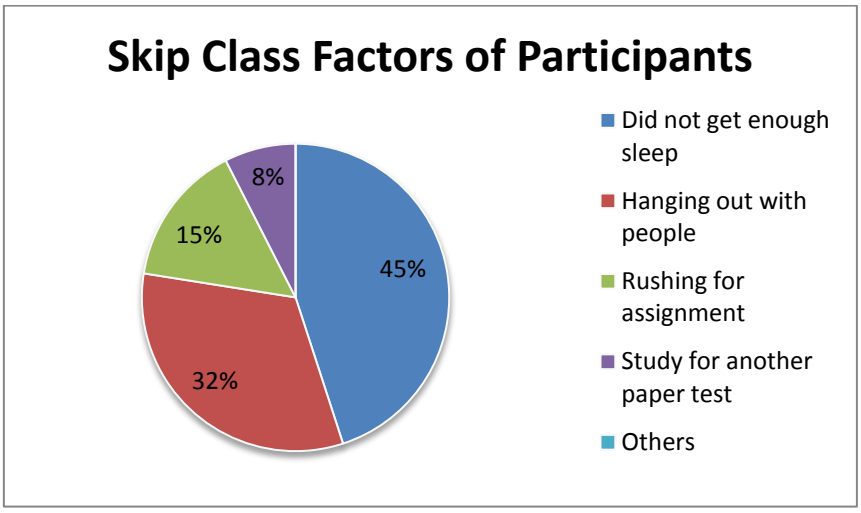

Figure 2.5-F3: Skip Class Factors of Participants

According to the data collected, it can be clearly seen that there are some major factors that causes the student to skip the class. Results shown that 45% of the participants choose to skip class because they did not get enough sleep following by the factors of hanging out with people (32%), rushing for assignment (15%), study for another paper test (8%), and other reasons (0%). Other than that, there are others sub-factors that causes the student unable to sign the attendance which include forgotten

to sign, loss of paper attendance, attendance record wasn't updated although they attends the class, and etc.

According to the data collected, it shows that 70% of the participants never forgot to sign their attendance while 30% of them had encountered this problem before. In addition, data collected also shows that there are 90% of students who never try to make lost the attendance paper to avoid being barred due to poor attendance while 10% of them had tried to do so. Besides that, 52.5% of students emphasize that they had encountered a problem which they attended the class but attendance wasn't updated while 47.5% of them emphasize that they never facing this problem.

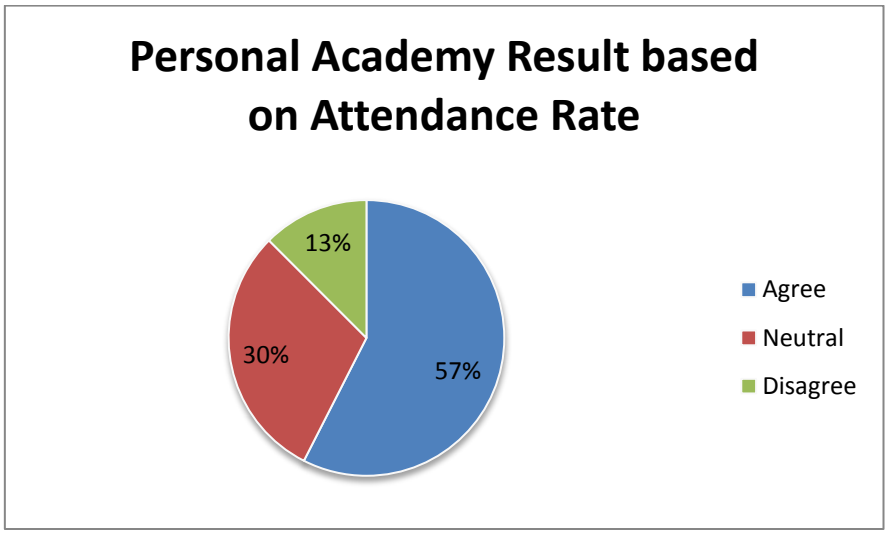

Figure 2.5-F4: Personal Academy Result based on Attendance Rate

Next, data shows that 57% of participants believe that attendance record will affect their personal academy result which means attending the class is very important in obtaining good academy result. While 30% of them emphasize there is no differences between attending the class or not in order to obtain good academy result following by 13% of them are disagree with the statement.

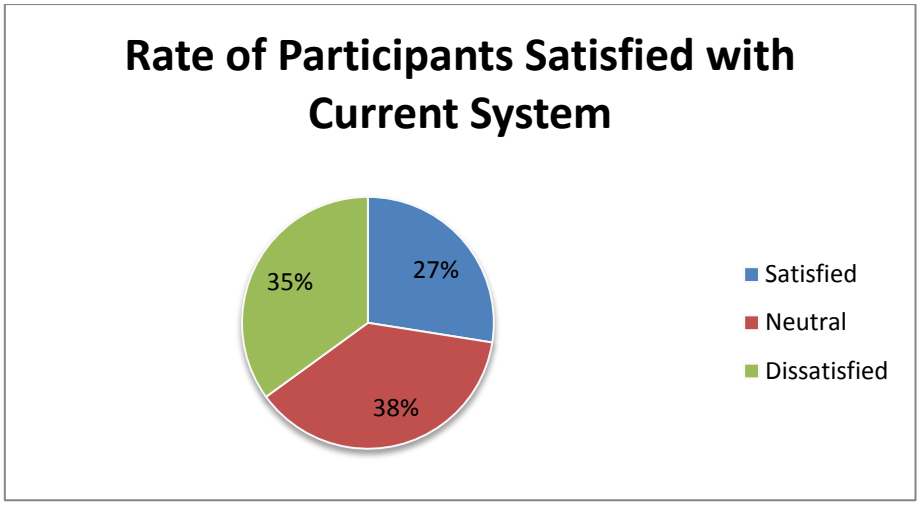

Figure 2.5-F5: Rate of Participants Satisfied with Current System

From the collected data, it shows that only 27% of participants are satisfied with current attendance system while 35% of them dissatisfied with the current system and 38% of them give neutral response. Other than that, data also shows that 65% of them agree that the current system is very troublesome when there are large amount of students in a class since they need to pass the paper-attendance around the whole class. Data shows only 10% of them disagree that the current system is troublesome while 25% of them give neutral response.

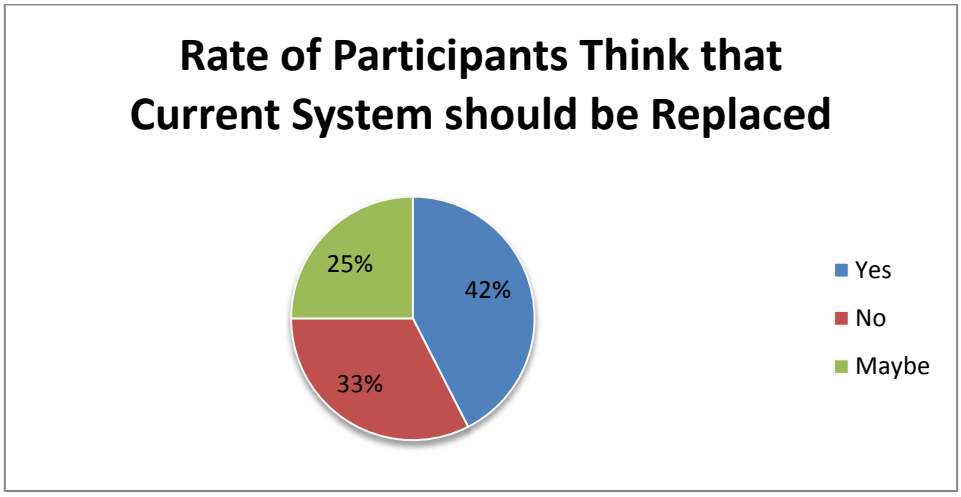

Figure 2.5-F6: Rate of Participants think Current System should be Replaced

The data shown indicates that 42% of the participants agree that the current system should be replaced while only 33% of them disagree with this statement following by 25% of them are hesitate whether the system should be replace or not.

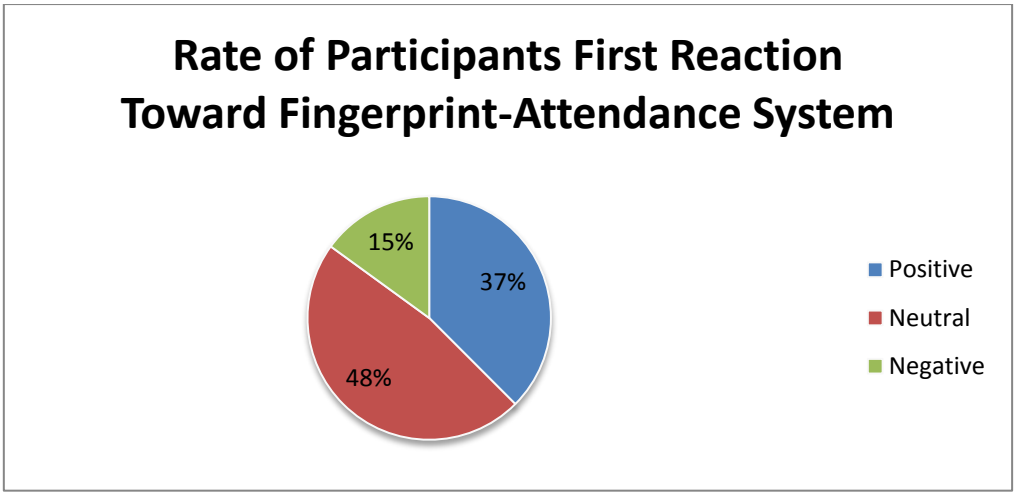

Figure 2.5-F7: First Reaction toward Fingerprint-Attendance System

Besides that, data collected also shows that 37% of the participants give out positive reaction to the new fingerprint-attendance system while 48% of them keep neutral response following by 15% of them give negative response regarding this matter. Furthermore, students had given response according to the unfairness of current attendance system. In short, 70% of them agree the possibility of student absent from class can be reduces since it will become hard to trick the lecturers by implementing the new attendance system. Another 10% of participants disagree with this statement while 20% of them give neutral response.

After go through the whole data collection process, some problems are clearly seen in current student attendance management system. Due to the analysis, it had shown that the new fingerprint-attendance system should be conduct to overcome these problems spotted from the survey. In short, most of the problems are likely can be solve by implementing the new system. Therefore, this project should be conduct.

Chapter 3: Proposed Method/Approach

3.1 Design Specifications

Firstly, the methodology chosen in carry out this project is Evolutionary Prototyping which is one of the prototyping methodologies. Evolutionary prototyping can be referred to as a form of software development method in which an initial prototype is developed and refined through a number of cycles and lastly to the final complete system (Sommerville, 2000). The reason to implement the evolutionary prototyping methodology in this project is because it allows the developer continuously getting feedbacks and suggestions for system improvements of the prototype that had been presented to the user until the final system is delivered. Evolutionary prototyping had consist of four phases which includes initial concept, design and implement initial prototype, refine prototype, and deliver complete system.

There is some strength that can be found from the evolutionary prototyping methodology. Evolutionary prototyping can help developers to speed up their system development. In addition, evolutionary prototyping help in improve the quality of final product since it needs to undergo few prototypes and lastly to the final version after the overall functionalities and requirements met. Besides that, chances to increase the satisfaction of end-users in using the system will be high since every generated prototype is based on the requirement specified by the end-users.

However, there are some weaknesses that can be found from the evolutionary prototyping methodology. First of all, it is hard to predict the completion date and the cost of the project since requirement can be change from time to time based on the end-users requirement. In addition, if there is any uncertainty, it can cause the developer to feel frustrated due to the money, time, and effort sacrificed previously. Besides that, the behind code of the software may occur high chances to be damaged or poorly structured due to the frequent changes make to meet the requirement specified by the end-users from time-to-time.

3.2 Methodologies and General Work Procedures

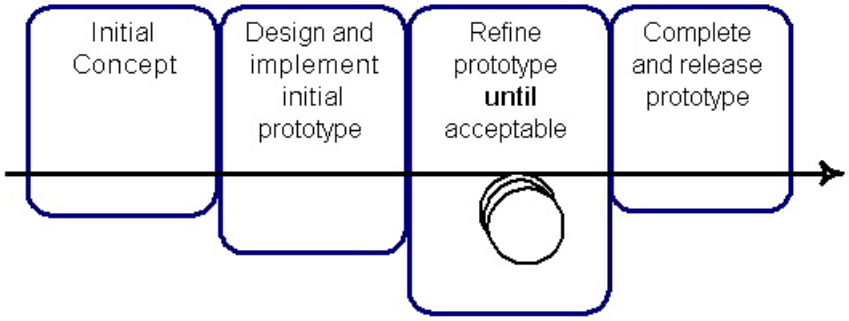

Figure 3.2-F1: *Evolutionary Prototyping Model [ONLINE]*. Available at: http://joshfolgado.com/rapiddev/html/evolutionary_prototyping.html [Accessed 02 August 2014]

a. Initial Concept

During the first phase, there is a need to identify the all basic requirements in developing the system. The developer will come out with the creation of initial concept and identify the required materials that related to the project in this phase. Thus, the developer may need to gather the requirements by conducting survey to those who seem likely will involve in the system development such as lecturer, student, and etc. From the result obtained from the survey, the developer will identify the business plan as an initial input from the user in developing this system. Through the business plan, the developer will be able to specify and create the project plan regarding to the project. In addition, the established project plan will provide information to the developer which includes the schedule of the project, clear description of the project, and etc. At the end of this phase, the developer will need to come out with business plan, project plan, and a list of initial requirements from the users as an output for this phase. Therefore, the initial concept to replace the current existing student attendance system process to fully-computerized and automated student attendance system had been stated. So, one of the objectives had been stated in this phase.

b. Design and implement initial concept

In the previous phase, the developer had identified all the basic information which includes the business plan, project plan, and a list of initial requirements from the users. So, in this phase, the developer will perform the design activities and implement the initial concept in developing the system. First of all, the development of prototyping of the basic user interface will be established for a purpose to show to the users as to make sure that the development is moving toward the right direction. The established basic user interfaces will includes the interface of fingerprint student attendance system in order to provide the clear images to the users what the system will look like. At the same time, the developer will perform the initial system design analysis based on the listed requirements in this phase. After developing the interface and identified the initial system design, the user interface is created and show to the users for a purpose to let the users evaluate the prototypes of the interface of the system. Next, the developer will then record the feedback from the users after the evaluation and modify the requirements according to the information provided by the users in this phase. After all, the basic user interfaces developed in this phase are archived and will not be use anymore. At the end of this phase, the developer will need to come out with a list of validated requirements, system design, and evaluation and feedback from the users about the missing requirements and etc.

c. Refine prototype until acceptable

In the previous phase, the developer had come out with a list of validated requirements, system design, and evaluation and feedback from the users about the missing requirements. So, in this phase, the developer will modify the system design through the feedback and evaluation from the users at the previous phase. In the development of the first prototype, the developer will ensure that the development of the system has fulfilled the validated requirements and system design obtained from the users. In addition, the developer will also consider about the quality of the system while developing it as well as perform operations such as program the system, perform debug activities, and test the system. Hence, the first prototype of the fingerprint attendance system is created. After that, the established first prototype of the system in this phase will be shown to the users for them to perform evaluation.

After the evaluation of the first prototype, the developer will record down the new or modified requirements that stated by the users for a purpose as a review for next prototype development. This phase will be repeating again and again until the final set of requirements stated by the users is established. At the end of this phase, the developer will need to come out with a validated final set of requirements.

d. Complete and release prototype

In this phase, the developer will develop a complete student attendance system based on the validated final set of requirements obtained from the users. Then, the developer will implement the system and perform testing to the final system. After all the requirements met, the system will be delivers to the final users as the approved system with the needed functionality and quality built-in. Hence, the overall objectives/sub-objectives of the system will be met at the end of this phase. The current attendance system will be replaced by fully-automated and computerized student attendance system at the end of this phase. The hybrid fingerprint student attendance system will be achieved in the end of this phase. The report generation regarding to the student attendance will be available at the end of this phase.

3.3 Technology Involved

Software

i. Microsoft Visual Studio 2013

Figure 3.3-F1: *Microsoft Visual Studio 2013 Logo [ONLINE]*. Available at: http://logo-kid.com/microsoft-visual-studio-2013-logo-png.htm [Accessed 02 Feb 2014]

Microsoft Visual Studio 2013 is a freeware to let the computer programmers to develop software and it is provided by the Microsoft. It allows the programmers to develop applications and websites in platforms such as Visual Basic, Visual C#, Visual C++, and etc. The Express Editions is more suitable for novice developers.

ii. Structured Query Language (SQL) Server

Figure 3.3-F2: *Microsoft SQL Server Logo [ONLINE]*. Available at: https://databasesmurf.wordpress.com/ [Accessed 02 Feb 2014]

SQL Server is a local database storage provided by the Microsoft along with the Microsoft Visual Studio. It is provided to let the computer programmers to manage and store information while develop the applications and websites. SQL Server appears to be a Relational Database Management System.

iii. Flexcode Software Development Kit (SDK)

Figure 3.3-F3: *Flexcode Software Development Kit (SDK) Logo [ONLINE]*. Available at: http://www.flexcodesdk.com/ [Accessed 02 Feb 2014]

Flexcode Software Development Kit (SDK) is a fingerprint recognition software development kit that allows computer programmers to integrate the fingerprint technology with the applications or websites. This software is available from the Flexcode official with purchase.

Hardware

i. DigitalPersona U.Are.U 4500 Reader

Figure 3.3-F4: *DigitalPersona U.Are.U 4500 [ONLINE]*. Available at: http://www.crossmatch.com/UareU4500Reader/ [Accessed 02 Feb 2014]

DigitalPersona U.Are.U 4500 Reader is a device that developed and sold for the use of biometrics fingerprint. This hardware is developed for used to detect the fingerprint of student or worker. However, DigitalPersona U.Are.U 4500 Reader does not provide the SDK file for the computer programmers to integrate into their applications or websites. So, in order to integrate the hardware with the developed applications or websites, the Flexcode SDK is necessary.

3.4 System Performance Definition

In developing the system, several targeted improvements had been made for a purpose to improve the accuracy, cost, time, and so on for the student attendance management system.

First of all, the new student attendance system will help in ensure the accuracy of the student attendance records since every student will be required to mark their attendance by scanning their fingerprint to the provided fingerprint reader. Therefore, none of the students will be able to fake their attendance in the new system. In addition, since the system will automatically update the student attendance record in database while students attend every class, so, the report generation feature will promote a very accurate data and result for analysis purpose. In the report generation feature, calculation of the percentage of student attendance will be included in the behind code of the system which mean lecturers/staffs will only need to generate the report in a very simple way anytime.

Besides that, the new student attendance system will help in save cost and time for paperwork. Current existing student attendance system will require students to sign on a piece of paper which require them to pass the paper around whole class in order to prove their attendance. Therefore, it is very time consuming since the paper will need to pass around the class to obtain the entire students signature. In long term investment, it will help in reduce the cost of paper used in every tri-semester just for students to mark their attendance. In using the new system, it will eliminate the usage of paper and save time from key-in the records into current existing system for a purpose to compile the end tri-semester report.

3.5 Implementation Issues and Challenges

Difficulties and challenges are always occurring in implementing something new to the existing system. There are several issues and challenges had been identified in implementing the new student attendance management system. These difficulties and challenges are includes the cost of hardware, lack of implementation time, stable network required, developer skills, and etc.

First of all, the first identified implementation issue and challenge is the cost of hardware. Although it is believed that in long term investment, university will be able to save up the cost of paper usage by using the new system. However, as the current attendance system is using paper-based to obtain the students attendance record. Therefore, initial cost to purchase the fingerprint reader hardware for every class is very expensive which require the university to support a huge amount of modal in implementing this system to the university. Other than that, the second implementation issue and challenge is lack of implementation time. As we know, the whole project from initial stage until the final stage of the development is handled by only one developer. Therefore, time given to complete the whole project might not enough and sometime may lead to negligence. In this situation, the system might not be able to meet the final requirement since developer will need to complete the project in rush mode.

Besides that, the third implementation issue and challenge is stable network required in implementing this system. Since the attendance obtained in every class will be update to database server directly through the network, so, stable network is required to faster the processing. Other than that, every student identity will need to authenticate on the network while they scan their fingerprint through the fingerprint reader. Last but not least, the developer skill also is an implementation issue and challenge since the developer never writes programs that work with hardware. So, time taken to learn how the software will work with hardware is kind of time consuming which may slow down the whole project progress.

3.6 Timeline

In this session, Gantt chart is provided to show the timeline and planning for FYPI and current semester (FYPII). Therefore, planning for previous and current semester is clearly stated in the Gantt chart provided below.

3.6.1 Gantt Chart Table

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Fingerprint Student Attendance Management System**	**161 days**	**Mon 12/1/15**	**Mon 24/8/15**	
2		**1.0 Initial Stage**	**36 days**	**Mon 12/1/15**	**Mon 2/3/15**	
3		1.1 Meeting with Supervisor	1 day	Wed 21/1/15	Wed 21/1/15	
4		1.2 Identify Project Background	1 day	Thu 22/1/15	Thu 22/1/15	3
5		**1.3 Conduct Fact-finding**	**9 days**	**Fri 23/1/15**	**Wed 4/2/15**	
6		1.3.1 Observation	3 days	Fri 23/1/15	Tue 27/1/15	4
7		1.3.2 Review Existing Manuals/Procedures	3 days	Wed 28/1/15	Fri 30/1/15	6
8		1.3.3 Survey	3 days	Mon 2/2/15	Wed 4/2/15	7
9		**1.4 Conduct Feasibility Studies**	**3 days**	**Thu 5/2/15**	**Mon 9/2/15**	
10		1.4.1 Technical Feasibility	1 day	Thu 5/2/15	Thu 5/2/15	8
11		1.4.2 Schedule Feasibility	1 day	Fri 6/2/15	Fri 6/2/15	10
12		1.4.3 Economic Feasibility	1 day	Mon 9/2/15	Mon 9/2/15	11
13		**1.5 Initial Investigation**	**7 days**	**Tue 10/2/15**	**Wed 18/2/15**	
14		1.5.1 Identify Motivation and Problem Statement	2 days	Tue 10/2/15	Wed 11/2/15	12
15		1.5.2 Identify Project Scope	2 days	Thu 12/2/15	Fri 13/2/15	14
16		1.5.3 Identify Project Objectives	2 days	Thu 12/2/15	Fri 13/2/15	14
17		**2.0 Analysis Stage**	**10 days**	**Mon 16/2/15**	**Fri 27/2/15**	
18		2.1 Meeting with Supervisor	0 days	Mon 16/2/15	Mon 16/2/15	16
19		**2.2 Literature Review**	**4 days**	**Tue 17/2/15**	**Fri 20/2/15**	
20		2.2.1 Benchmark Similar System	1 day	Tue 17/2/15	Tue 17/2/15	18
21		2.2.2 Identify Strengths and Weaknesses	1 day	Wed 18/2/15	Wed 18/2/15	20
22		2.2.3 Analysis Existing Solutions	1 day	Thu 19/2/15	Thu 19/2/15	21
23		2.2.4 Evaluate Existing Solutions	1 day	Fri 20/2/15	Fri 20/2/15	22
24		2.3 Identify Project Requirements	1 day	Mon 23/2/15	Mon 23/2/15	23
25		2.4 Define Methodology	1 day	Tue 24/2/15	Tue 24/2/15	24
26		2.5 Identify Project Contribution and Innovation	1 day	Wed 25/2/15	Wed 25/2/15	25
27		**2.6 Identify Technology Involved**	**2 days**	**Thu 26/2/15**	**Fri 27/2/15**	
28		2.6.1 Software	1 day	Thu 26/2/15	Thu 26/2/15	26
29		2.6.2 Hardware	1 day	Fri 27/2/15	Fri 27/2/15	28

Figure 3.6.1-F1: Gantt chart Table (Part1)

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
29		2.6.2 Hardware	1 day	Fri 27/2/15	Fri 27/2/15	28
30		**3.0 Prototype 1**	**41 days**	**Mon 16/2/15**	**Mon 13/4/15**	
31		**3.1 Design Stage**	**13 days**	**Mon 16/2/15**	**Wed 4/3/15**	
32		**3.1.1 Develop Diagram**	**11 days**	**Mon 16/2/15**	**Mon 2/3/15**	
33		3.1.1.1 Basic Diagrams	4 days	Mon 16/2/15	Thu 19/2/15	16
34		3.1.1.2 Advanced Diagrams	7 days	Fri 20/2/15	Mon 2/3/15	33
35		3.1.2 Design System Architecture	1 day	Tue 3/3/15	Tue 3/3/15	34
36		3.1.3 Design User Interface	1 day	Wed 4/3/15	Wed 4/3/15	35
37		**3.2 Implementation Stage (Prototype)**	**15 days**	**Thu 5/3/15**	**Wed 25/3/15**	
38		3.2.1 Module Design	5 days	Thu 5/3/15	Wed 11/3/15	36
39		3.2.2 Module Generation	5 days	Thu 12/3/15	Wed 18/3/15	38
40		3.2.3 Module Implementation	5 days	Thu 19/3/15	Wed 25/3/15	39
41		**3.3 Testing Stage**	**12 days**	**Thu 26/3/15**	**Fri 10/4/15**	
42		3.3.1 Unit Testing	4 days	Thu 26/3/15	Tue 31/3/15	40
43		3.3.2 Integration Testing	4 days	Wed 1/4/15	Mon 6/4/15	42
44		3.3.3 System Testing	4 days	Tue 7/4/15	Fri 10/4/15	43
45		3.4 Evaluation Stage (Prototype)	1 day	Mon 13/4/15	Mon 13/4/15	44
46		**4.0 Prototype 2**	**42 days**	**Tue 14/4/15**	**Wed 10/6/15**	
47		**4.1 Design Stage**	**16 days**	**Mon 20/4/15**	**Mon 11/5/15**	
48		4.1.1 Review and Evolve Diagram	6 days	Mon 20/4/15	Mon 27/4/15	
49		4.1.2 Design Graphical User Interface	10 days	Tue 28/4/15	Mon 11/5/15	48
50		**4.2 Implementation Stage (Prototype)**	**10 days**	**Tue 12/5/15**	**Mon 25/5/15**	
51		4.2.1 Refine Design	2 days	Tue 12/5/15	Wed 13/5/15	49
52		4.2.2 Refine Generation	3 days	Thu 14/5/15	Mon 18/5/15	51
53		4.2.3 Refine Implementation	5 days	Tue 19/5/15	Mon 25/5/15	52
54		**4.3 Testing Stage**	**10 days**	**Tue 26/5/15**	**Mon 8/6/15**	
55		4.3.1 Unit Testing	3 days	Tue 26/5/15	Thu 28/5/15	53
56		4.3.2 Integration Testing	3 days	Fri 29/5/15	Tue 2/6/15	55
57		4.3.3 System Testing	4 days	Wed 3/6/15	Mon 8/6/15	56
58		4.4 Evaluation Stage (Prototype)	2 days	Tue 9/6/15	Wed 10/6/15	57
59		**5.0 Deployment Stage**	**53 days**	**Thu 11/6/15**	**Mon 24/8/15**	
60		5.1 Construct Final Product	35 days	Thu 11/6/15	Wed 29/7/15	58
61		**5.2 Testing Stage**	**15 days**	**Thu 30/7/15**	**Wed 19/8/15**	
62		5.2.1 Unit Testing	5 days	Thu 30/7/15	Wed 5/8/15	60

Figure 3.6.1-F2: Gantt chart Table (Part2)

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
63		5.2.2 Integration Testing	5 days	Thu 6/8/15	Wed 12/8/15	62
64		5.2.3 System Testing	5 days	Thu 13/8/15	Wed 19/8/15	63
65		5.3 Evaluation of Final Product	2 days	Thu 20/8/15	Fri 21/8/15	64
66		5.4 Final Presentation	1 day	Mon 24/8/15	Mon 24/8/15	65
67		5.5 Project Closure	0 days	Mon 24/8/15	Mon 24/8/15	65

Figure 3.6.1-F3: Gantt chart Table (Part3)

3.6.2 Gantt Chart Diagram

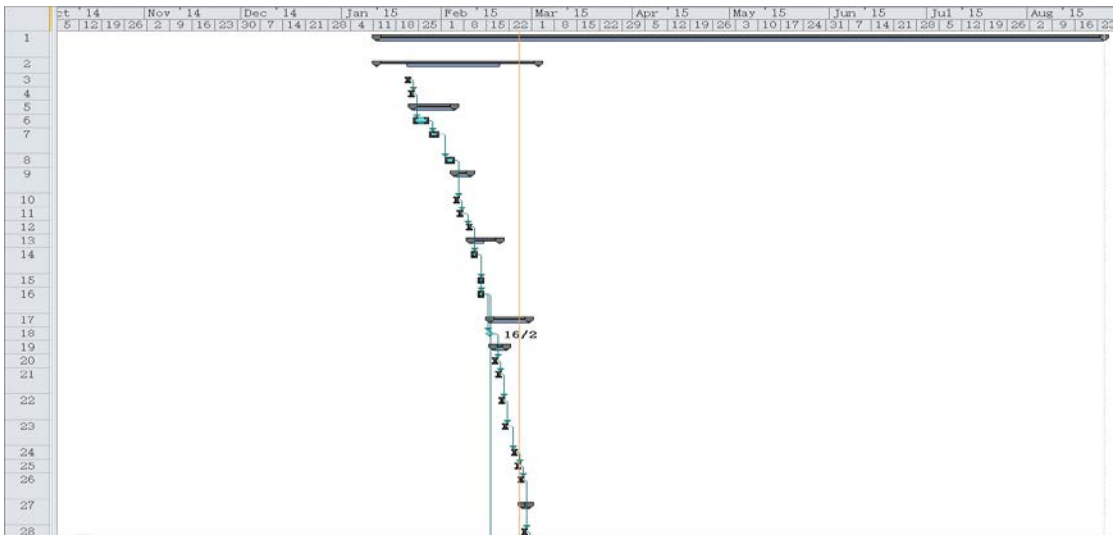

Figure 3.6.2-F1: Gantt chart Diagram (Part1)

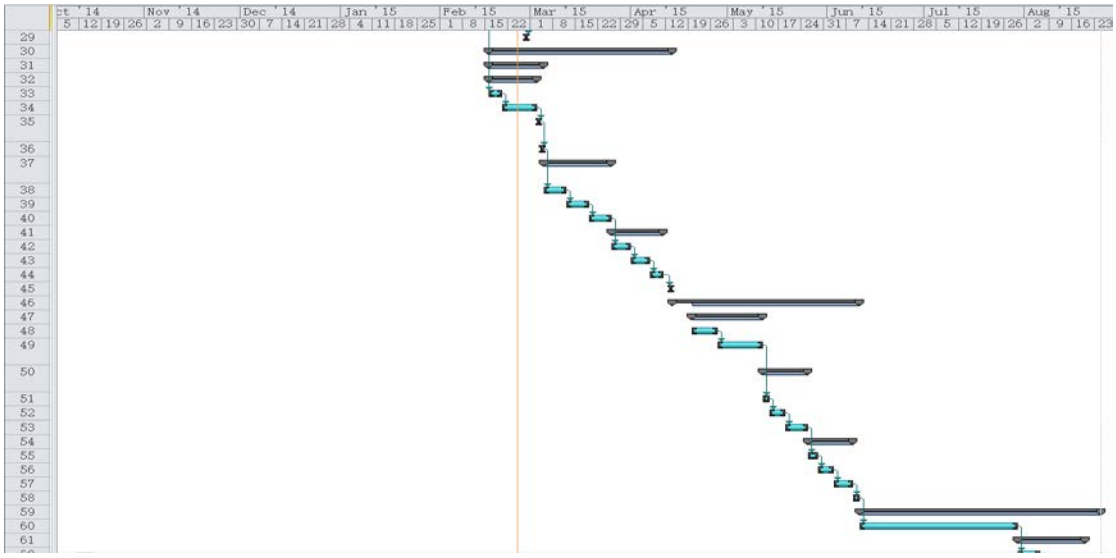

Figure 3.6.2-F2: Gantt chart Diagram (Part2)

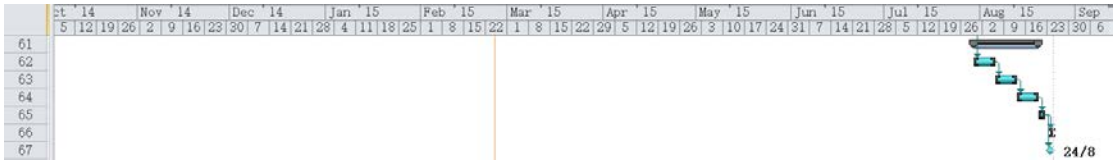

Figure 3.6.2-F3: Gantt chart Diagram (Part3)

Chapter 4: System Analysis and Design

4.1 System Design/Overview

In describing how the project is designed, diagrams will be sketch in this session to make clear on how the project will be written, how it function, how it install, and etc. In previously, I have studied the Object-Oriented System Analysis and Design (OOSAD). Therefore, this session will includes all the diagrams that I had learnt from the OOSAD subject. The diagrams included are use-case diagram, activity-flow diagram, entity-relationship diagram, sequence diagram, object diagram, and etc. With the help of the diagrams, it will be able to assists the readers to understand more easily regarding the new system.

Other than diagrams, the database design will also be perform in this session for a purpose to specify all the required entities in this system. These also include the CRC cards, use-case description, and etc. In addition, drawing prototype of whole system interface will be included in this session to give a clear image to the reader/user how the system will look like and how it will function. The purpose of the drawing prototype is to obtain initial feedback from the reader/user before the first real prototype begins development.

4.2 Use-Case Diagram

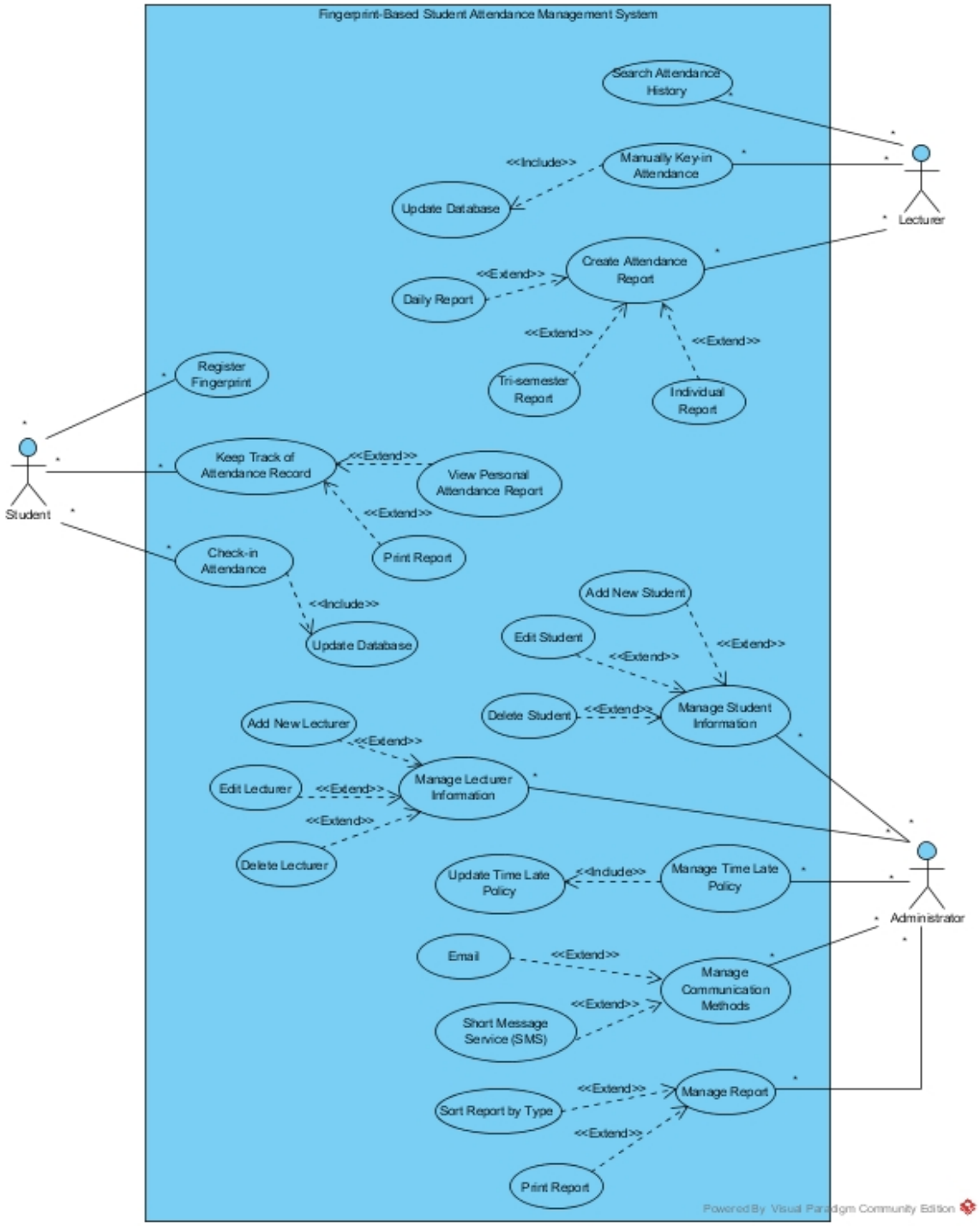

Figure 4.2-F1: Use-Case Diagram

4.3 Activity Diagram

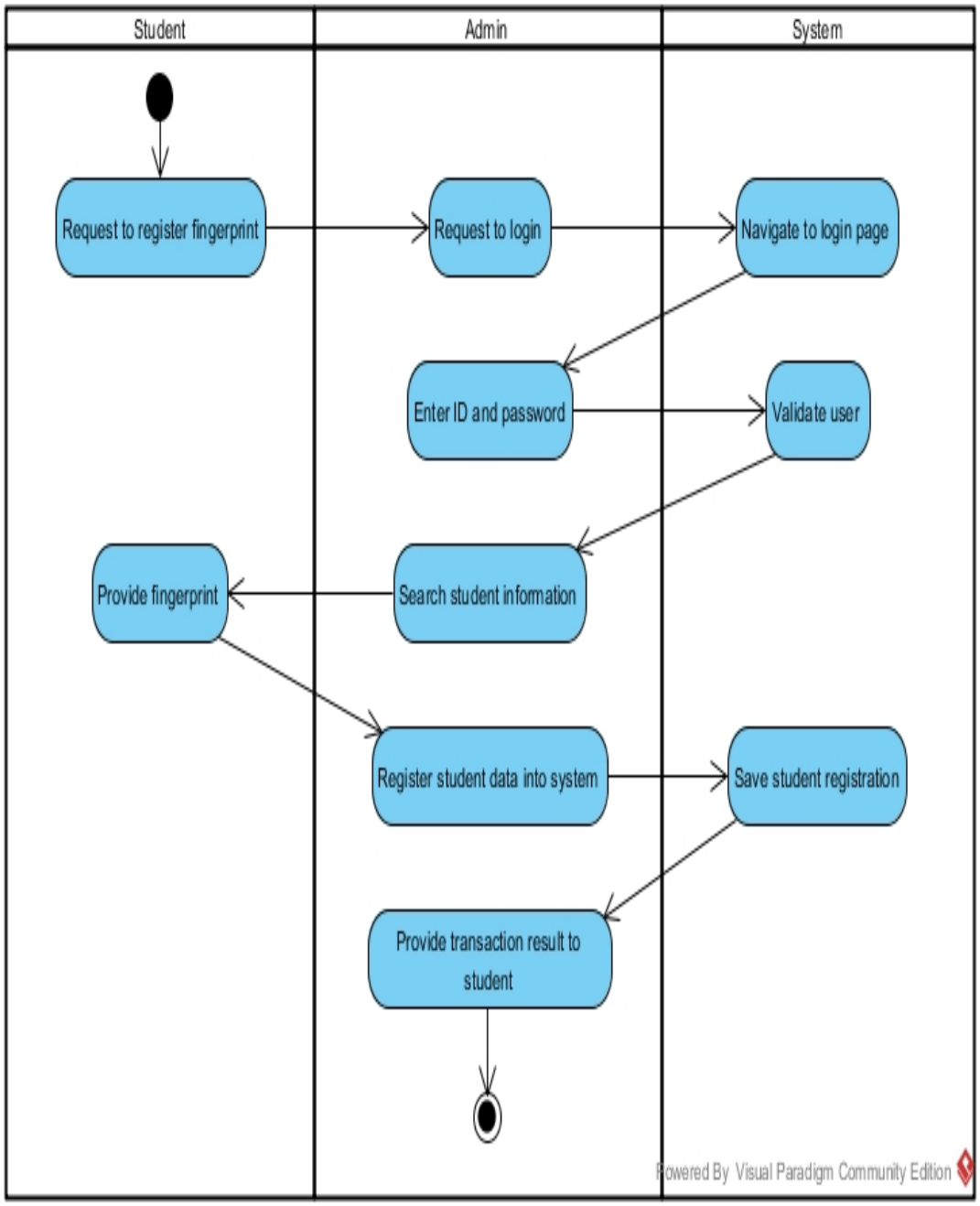

Figure 4.3-F1: Activity Diagram for Register Fingerprint (Student)

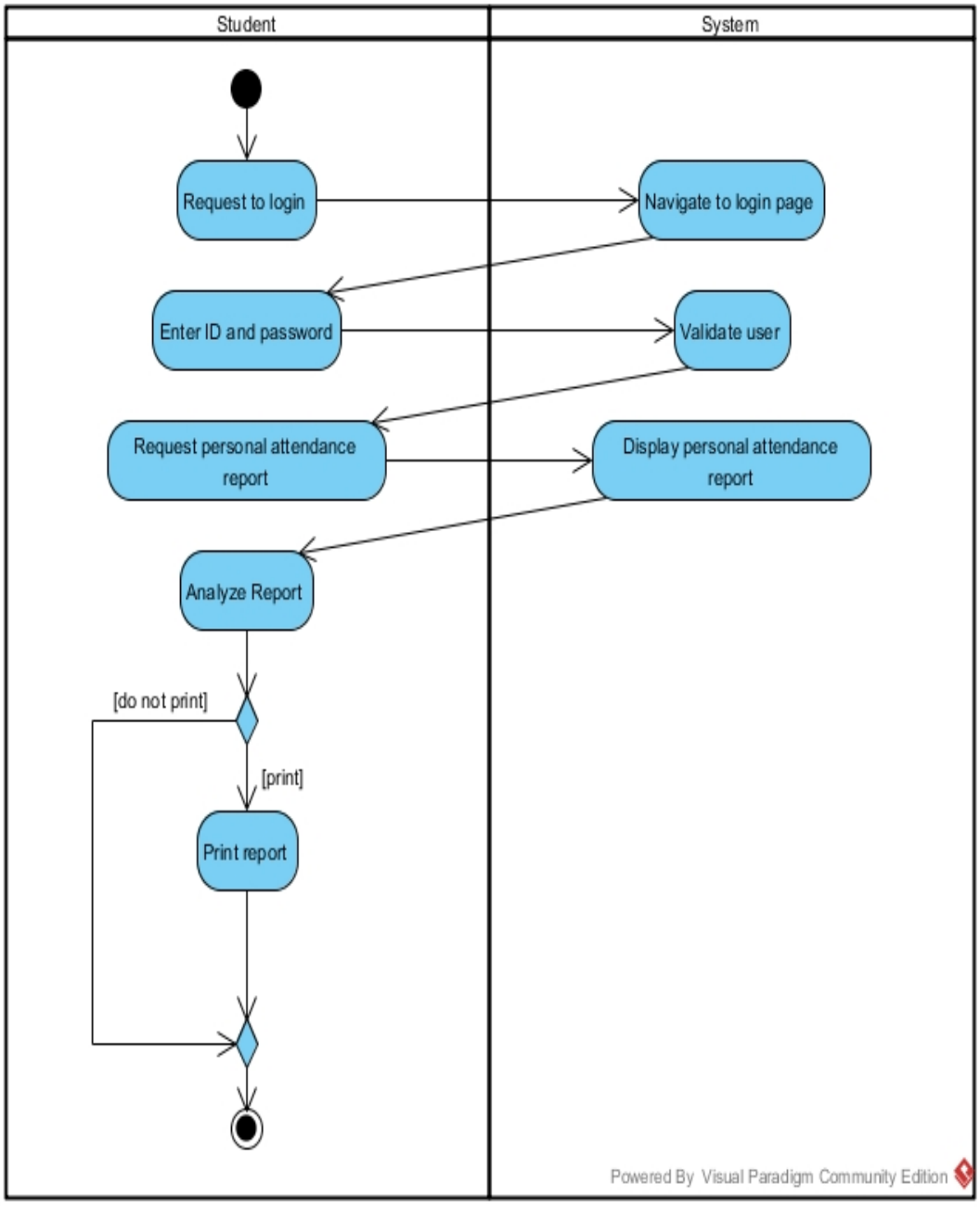

Figure 4.3-F2: Activity Diagram for Keep Track of Personal Attendance Record (Student)

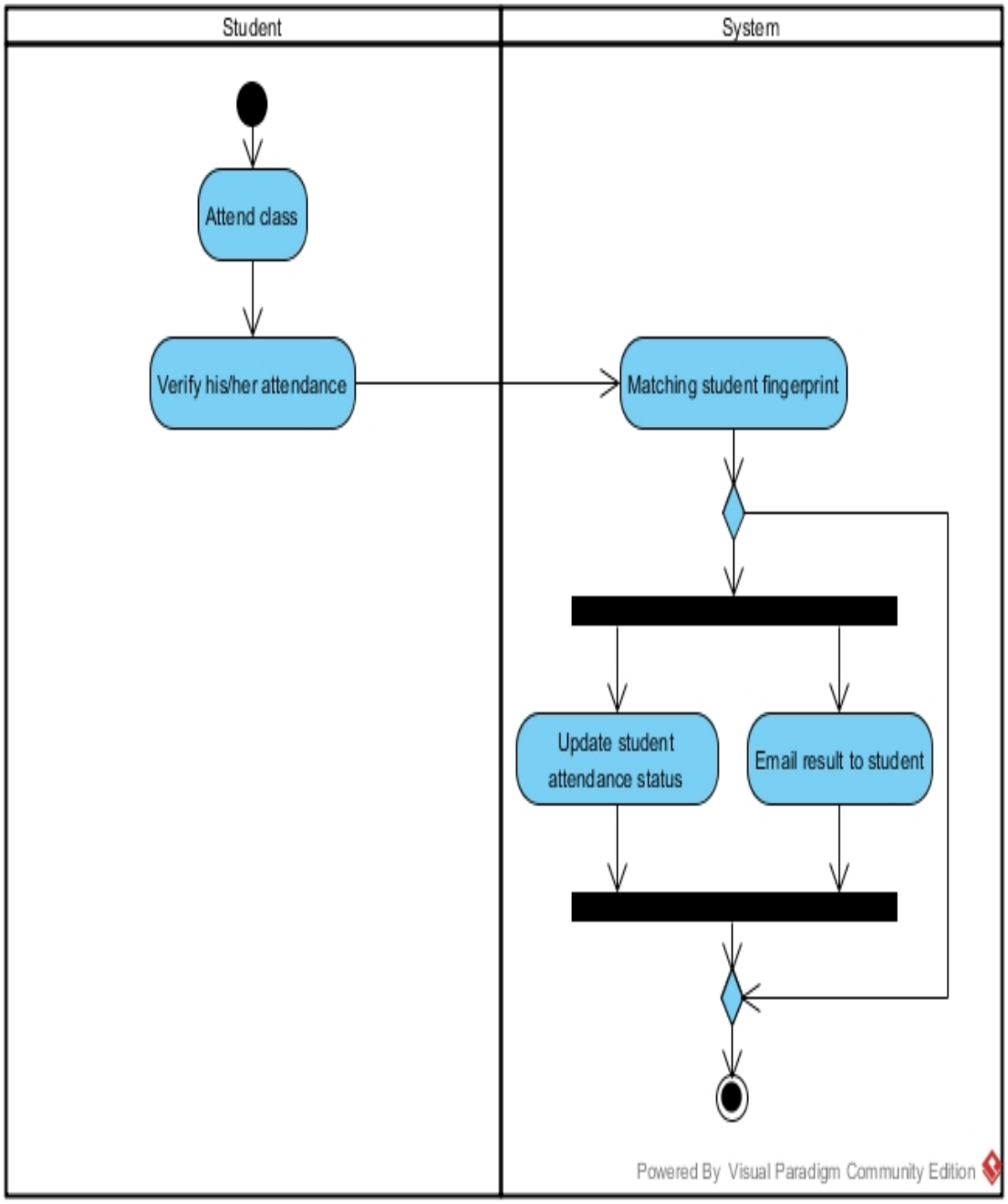

Figure 4.3-F3: Activity Diagram for Check-in Attendance (Student)

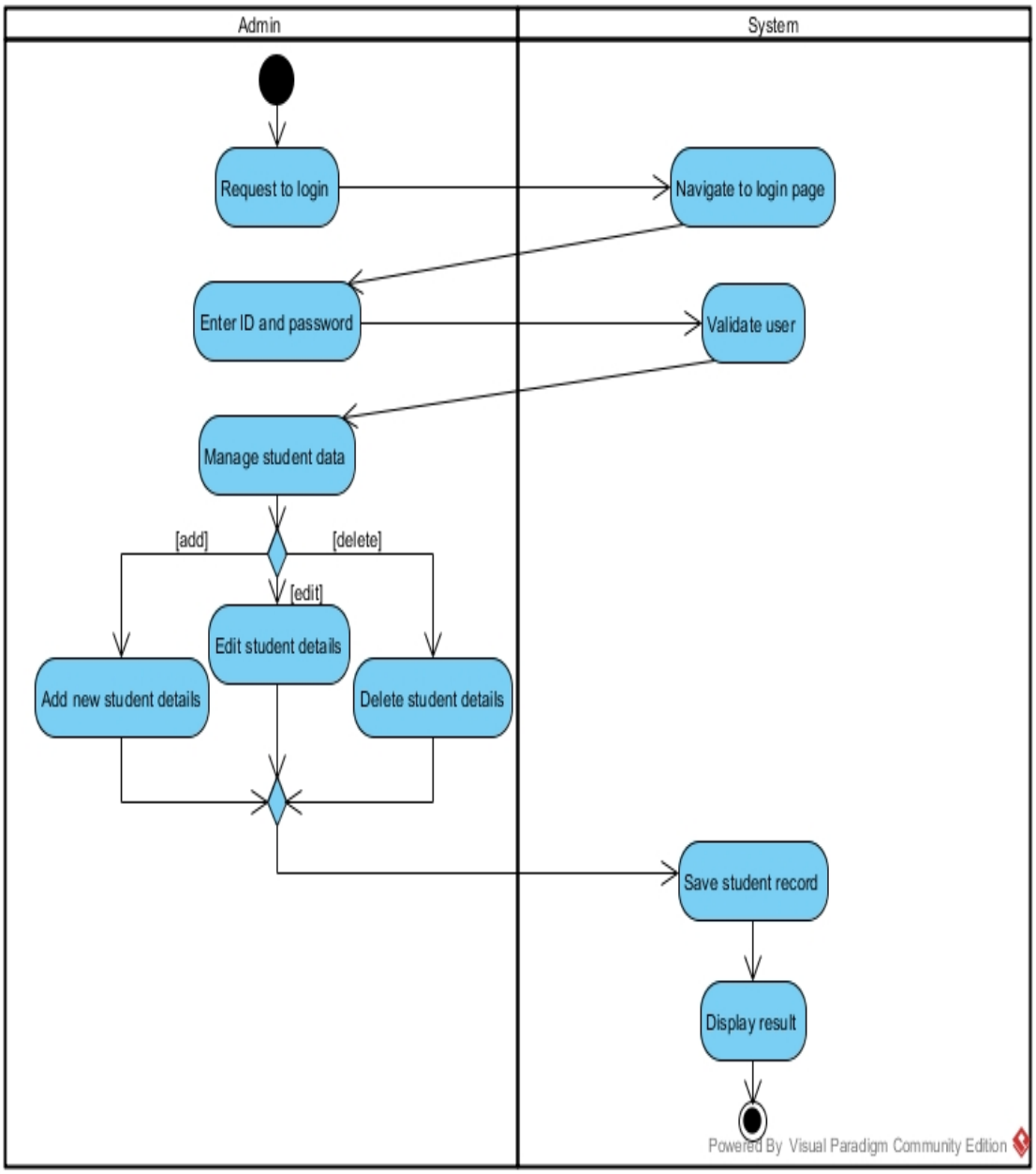

Figure 4.3-F4: Activity Diagram for Manage Student Information (Admin)

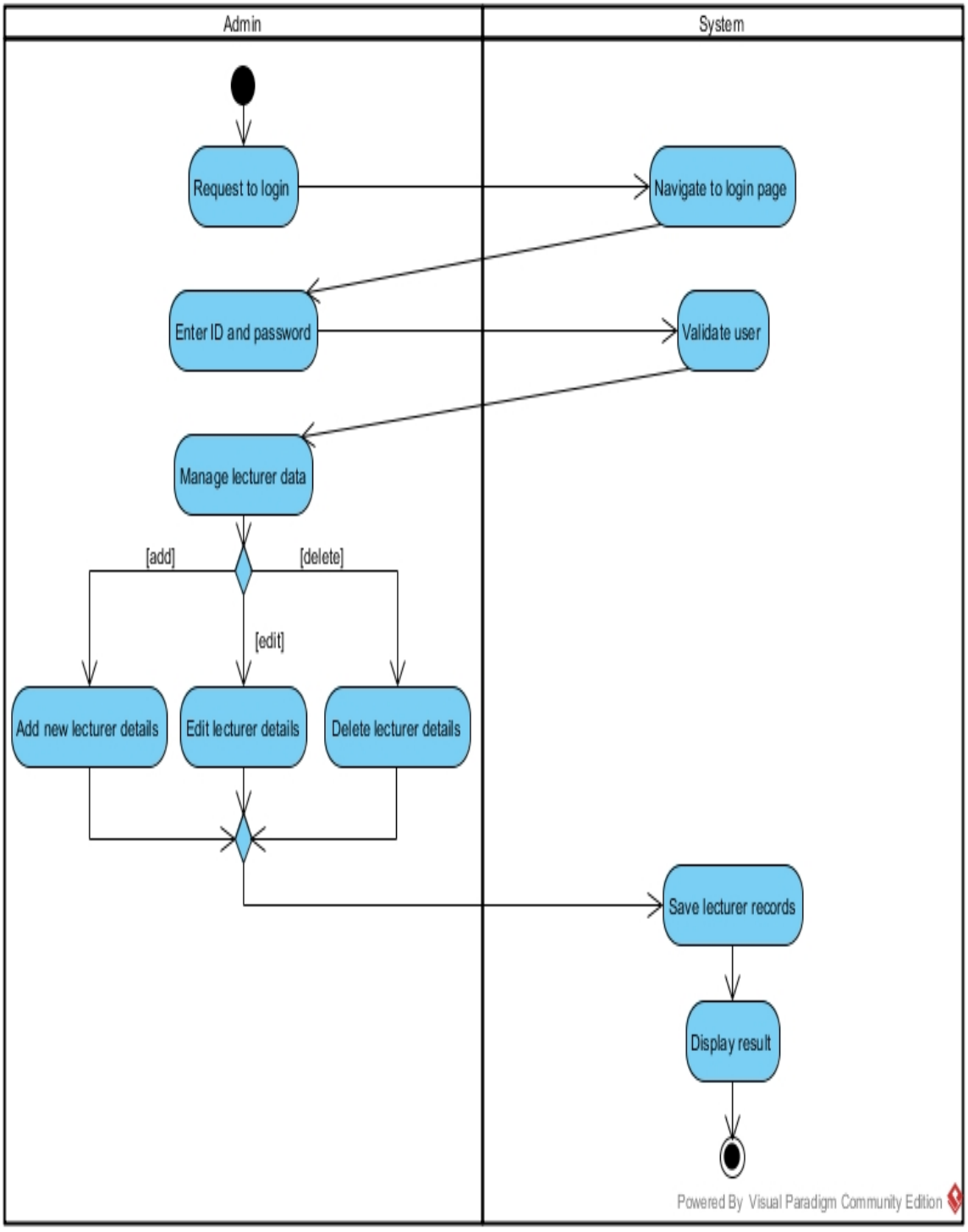

Figure 4.3-F5: Activity Diagram for Manage Lecturer Information (Admin)

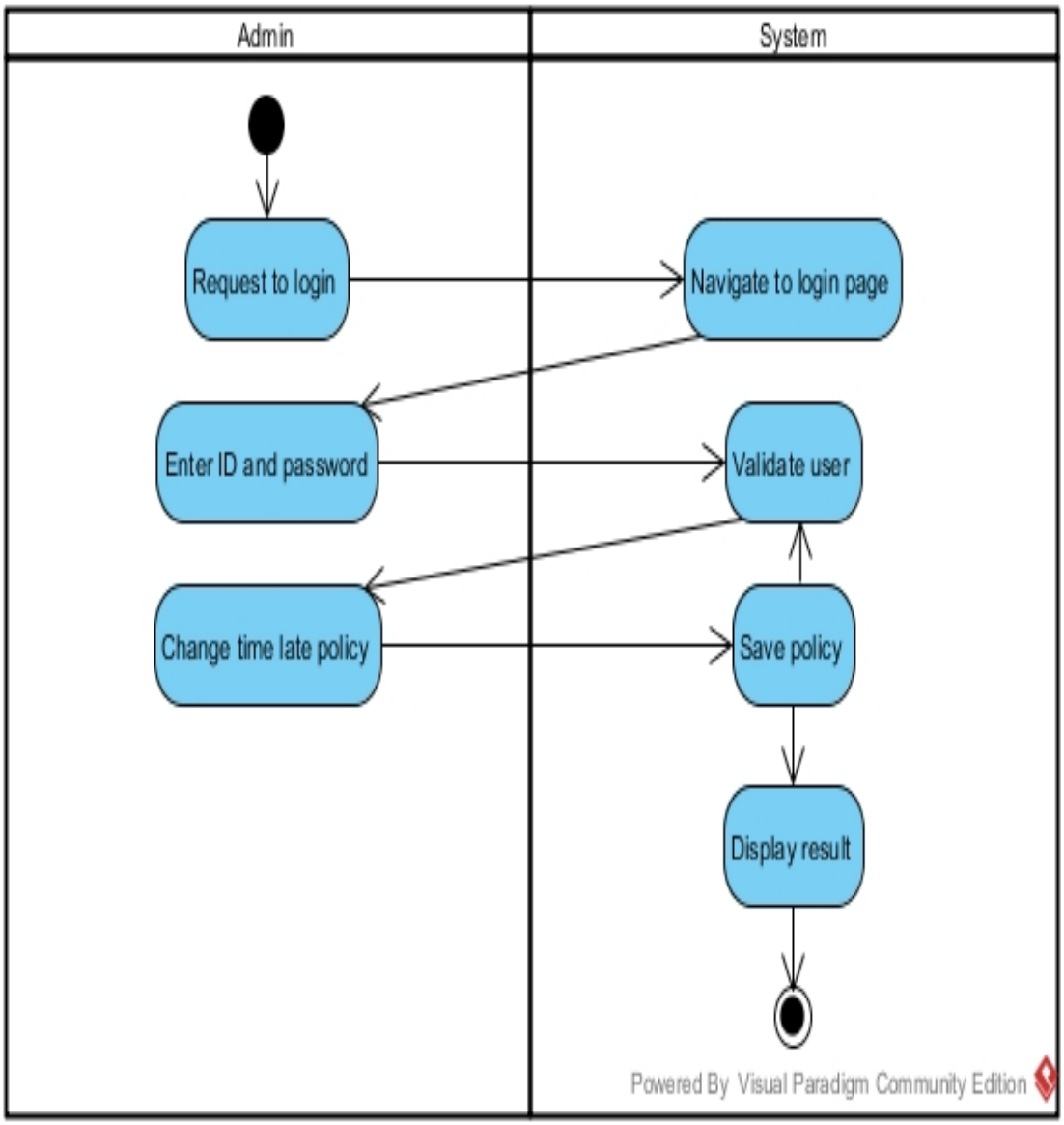

Figure 4.3-F6: Activity Diagram for Manage Time Late Policy (Admin)

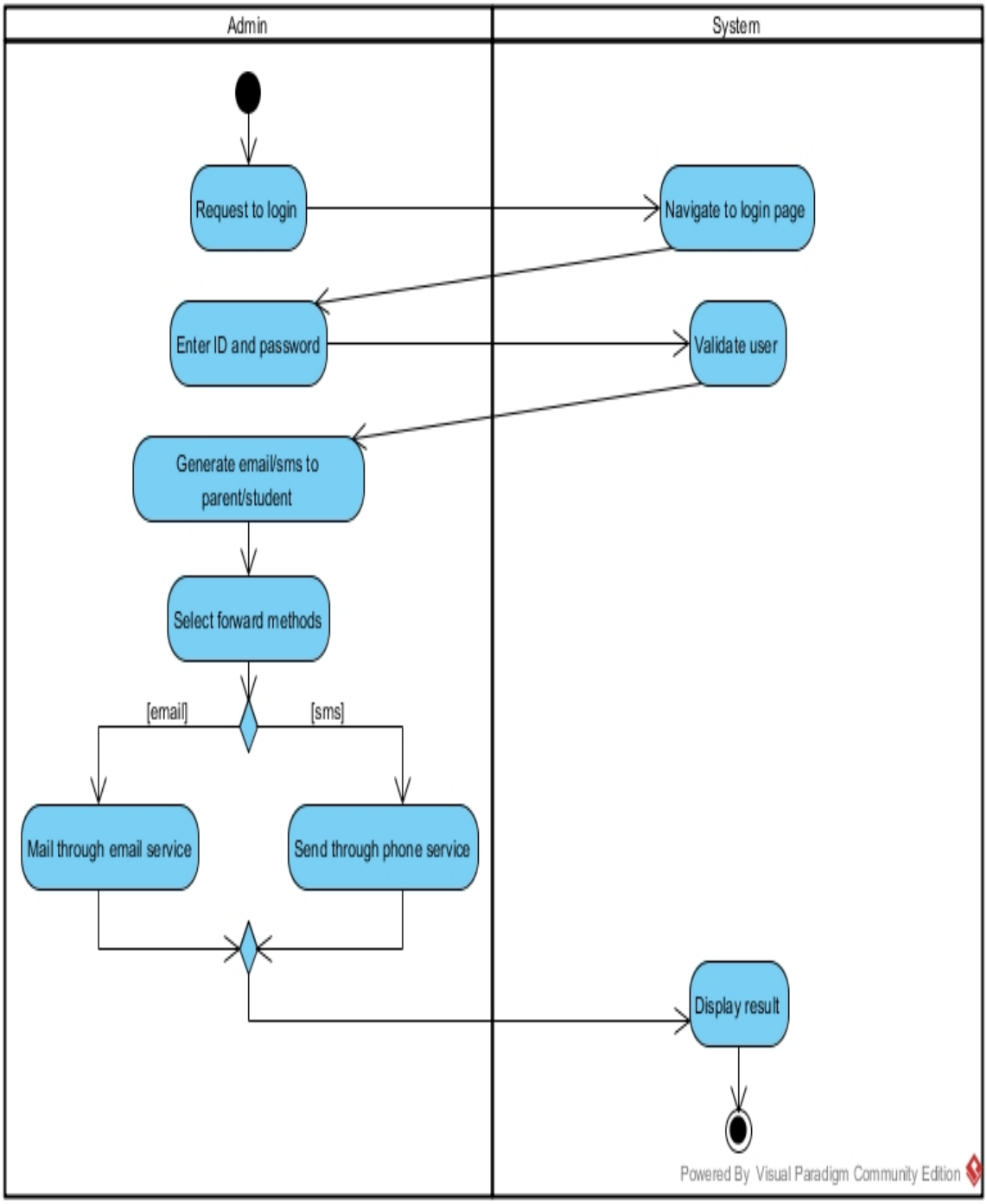

Figure 4.3-F7: Activity Diagram for Manage Communication Methods (Admin)

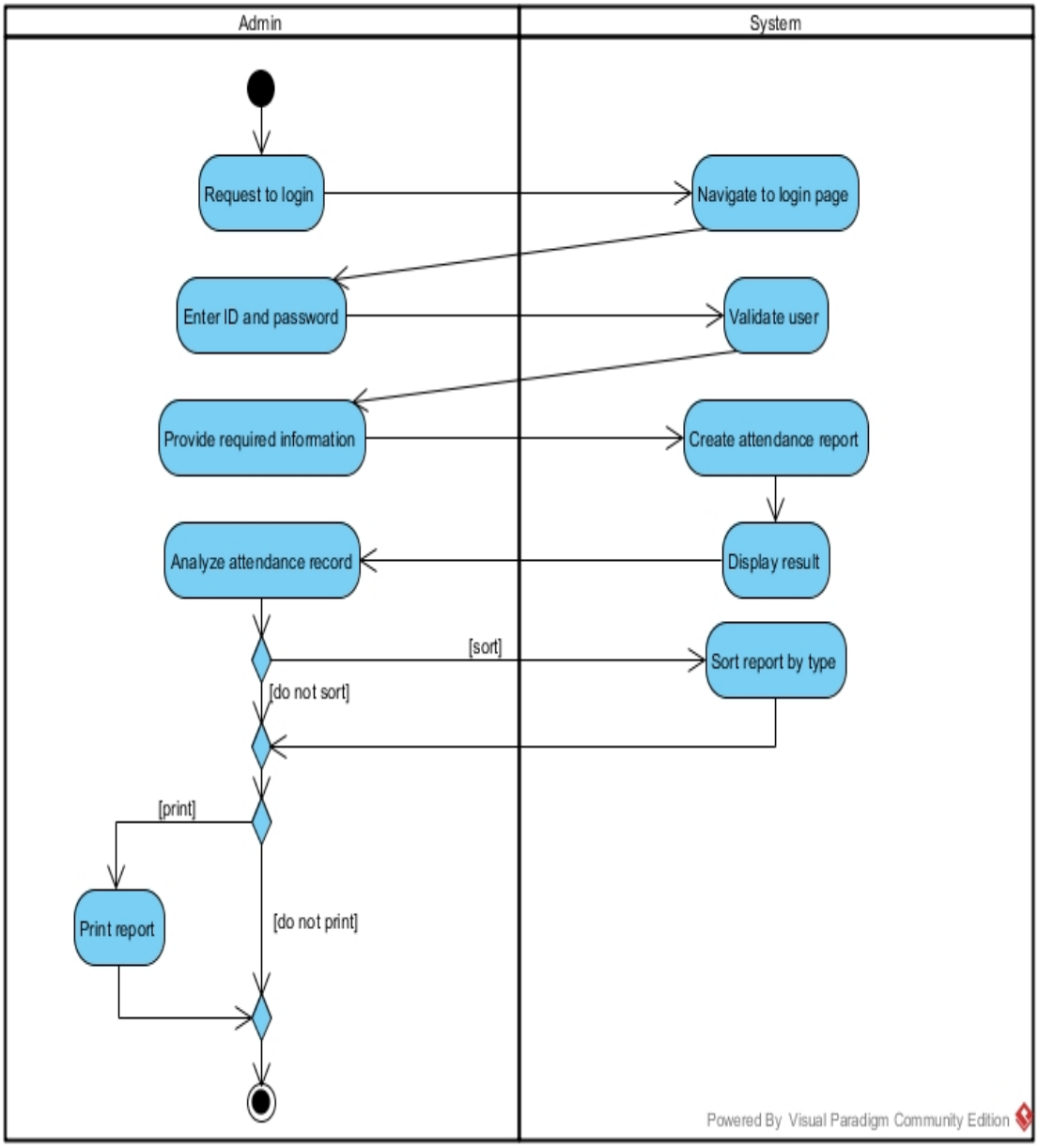

Figure 4.3-F8: Activity Diagram for Manage Report (Admin)

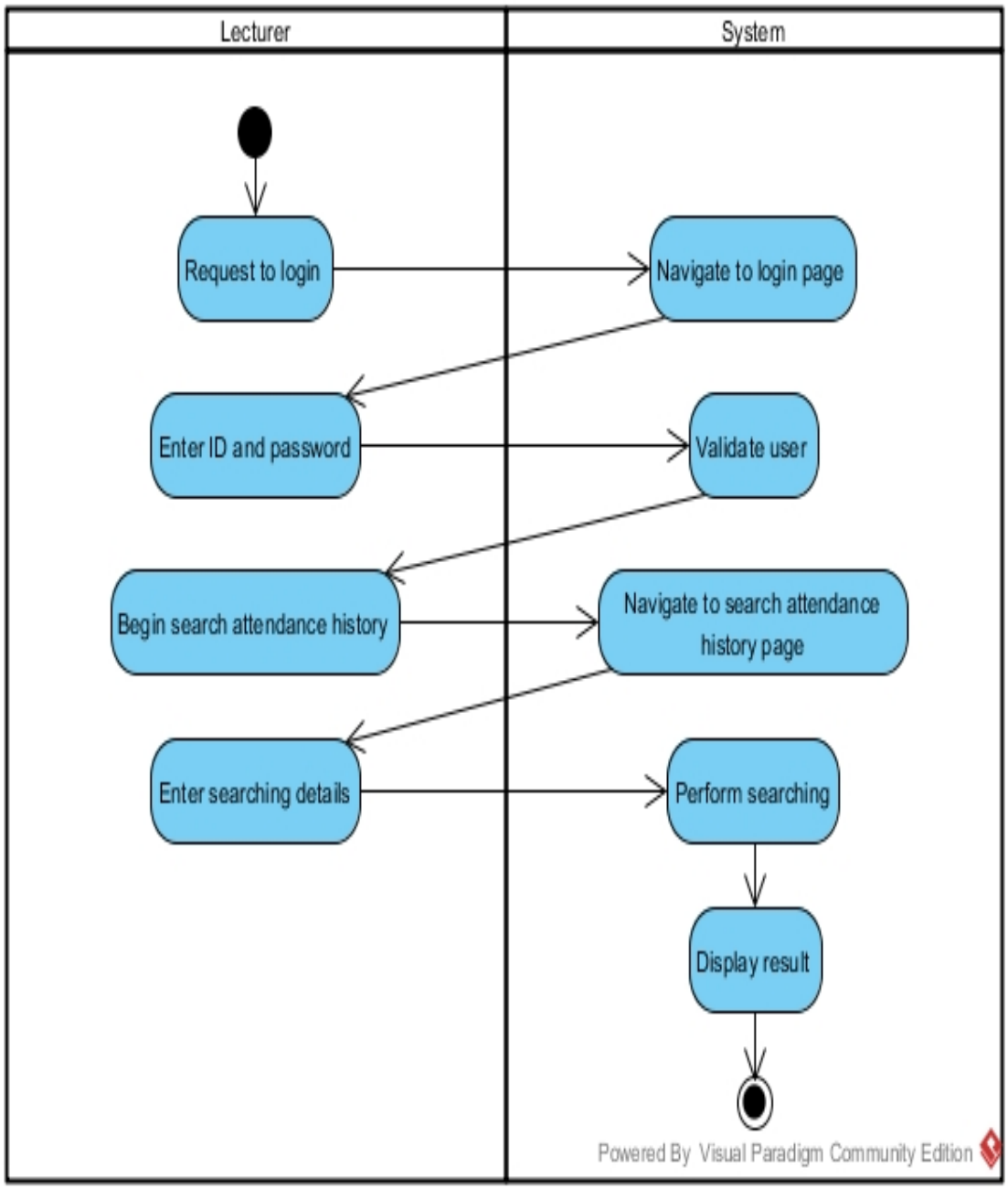

Figure 4.3-F9: Activity Diagram for Search Attendance History (Lecturer)

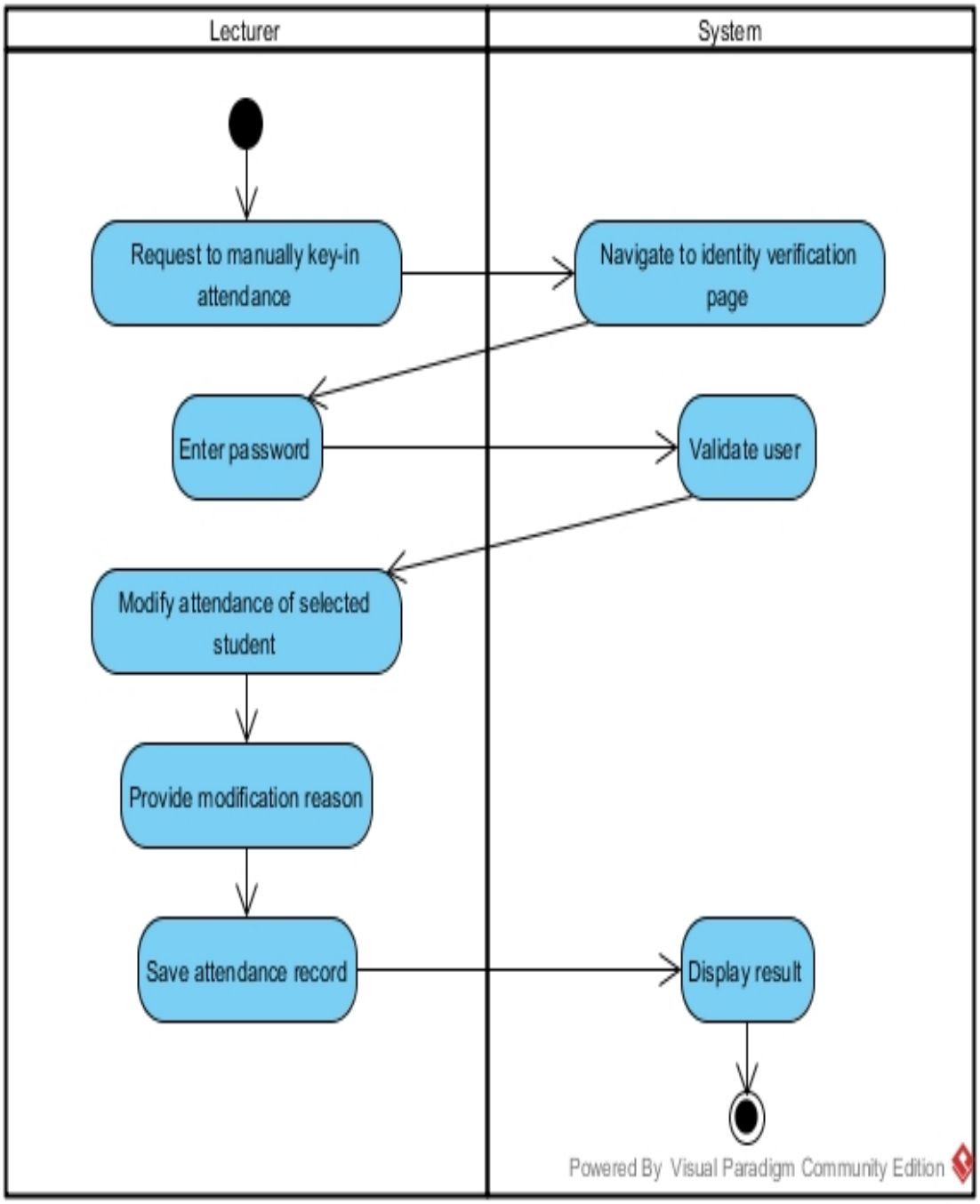

Figure 4.3-F10: Activity Diagram for Manually Key-in Attendance (Lecturer)

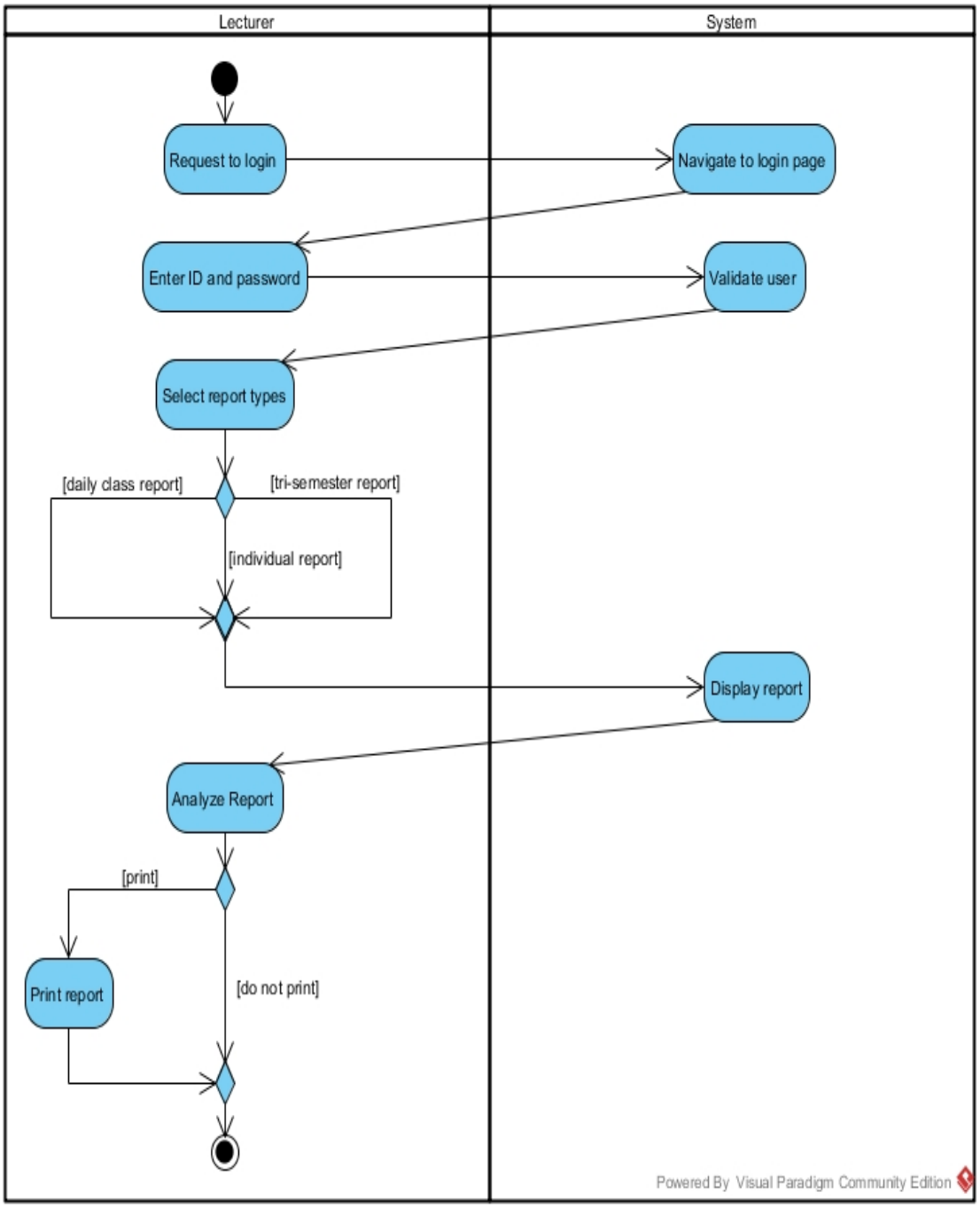

Figure 4.3-F11: Activity Diagram for Create Attendance Record (Lecturer)

4.4 Use-Case Description

Use Case Name: Register Fingerprint	ID: 1	Important Level: High
Primary Actor: Student	Use Case Type: Essential, Detail	
Stakeholders and Interests: Student – wants to register his/her fingerprint. Admin – wants to save the student fingerprint into system.		
Brief Description: This use case describes how student register his/her fingerprint into the system assisted by admin.		
Trigger: The student come and request to register his/her fingerprint into the system. Type: External		
Relationships: Association: Student. Include: Extend: Generalization:		
Normal Flow of Events: 1. The student come and request to register his/her fingerprint into the system. 2. The admin request to login. 3. The system navigates to login page. 4. The admin enter ID & password to the system. 5. The system validates the admin ID & password from database. 6. The admin search for student information. 7. The student provides his/her fingerprint. 8. The admin register student data into system. 9. The system save student registration information. 10. The admin provides the result of the transaction to student. 11. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F1: Register Fingerprint Use-Case Description

Use Case Name: Keep Track of Personal Attendance Record	ID: 2	Important Level: High
Primary Actor: Student	Use Case Type: Essential, Detail	
Stakeholders and Interests: Student – wants to keep track of his/her own attendance record.		
Brief Description: This use case describes how student keep track on his/her personal attendance record by print out report.		
Trigger: The student wants to keep track of his/her own attendance record. Type: External		
Relationships: Association: Student. Include: Extend: View Personal Attendance Report, Print Report. Generalization:		
Normal Flow of Events: 1. The student wants to keep track of his/her own attendance record. 2. The student request to login. 3. The system navigates to login page. 4. The student enters ID & password to the system. 5. The system validates the student ID & password from database. 6. The student request to view his/her own personal attendance report. 7. The system displays the personal attendance report. 8. The student analyzes the report. 9. If print out the report 10. The report printed out. 11. The system end. 12. Else 13. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F2: Keep Track of Personal Attendance Record Use-Case Description

Use Case Name: Check-in Attendance	ID: 3	Important Level: High
Primary Actor: Student	Use Case Type: Essential, Detail	
Stakeholders and Interests: Student – wants to check in his/her attendance status.		
Brief Description: This use case describes how student handle the process of check-in his/her attendance.		
Trigger: The student wants to check in his/her attendance status. Type: External		
Relationships: Association: Student. Include: Update Database. Extend: Generalization:		
Normal Flow of Events: 1. The student wants to check in his/her attendance status. 2. The student attends the class. 3. The student verifies his/her attendance using fingerprint. 4. The system matching student fingerprint. 5. If fingerprint matched 6. The student attendance status updated. 7. The system email transaction result to student. [E1] 8. The system end. 9. Else 10. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: E1: If student fail to verify his attendance, the system won't send email to the student and the process ends.		

Figure 4.4-F3: Check-in Attendance Use-Case Description

Use Case Name: Manage Student Information	ID: 4	Important Level: High
Primary Actor: Admin	Use Case Type: Essential, Detail	
Stakeholders and Interests: Admin – wants to manage the student information in system.		
Brief Description: This use case describes how admin handle the process of managing student information.		
Trigger: The admin wants to manage the student information in system. Type: External		
Relationships: Association: Admin. Include: Extend: Add New Student Information, Edit Student Information, Delete Student Information. Generalization:		
Normal Flow of Events: 1. The admin wants to manage the student information in system. 2. The admin request to login. 3. The system navigates to login page. 4. The admin enters ID & password to the system. 5. The system validates the admin ID & password from database. 6. The admin select action to be performed. 7. If the admin wants to add new student information 8. The S – 1: Add new student info performed. 9. If the admin wants to edit existing student information 10. The S – 2: Edit student info performed. 11. If the admin wants to delete existing student information 12. The S – 3: Delete student info performed. 13. The system displays the result. 14. The system end.		
Sub Flows: S – 1: Add new student info 1. The admin enter required information into the system. 2. The admin saves the record into the system.		

S – 2: Edit student info 1. The admin search and navigate to specific student info. 2. The admin edit the student info. 3. The admin saves the record into the system. S – 3: Delete student info 1. The admin search and navigate to specific student info. 2. The admin edit the student info. 3. The admin saves the record into the system.
Alternative/Exceptional Flows: Not applicable

Figure 4.4-F4: Manage Student Information Use-Case Description

Use Case Name: Manage Lecturer Information	ID: 5	Important Level: High
Primary Actor: Admin	Use Case Type: Essential, Detail	
Stakeholders and Interests: Admin – wants to manage the lecturer information in system.		
Brief Description: This use case describes how admin handle the process of managing lecturer information.		
Trigger: The admin wants to manage the lecturer information in system. Type: External		
Relationships: Association: Admin. Include: Extend: Add New Lecturer Information, Edit Lecturer Information, Delete Lecturer Information. Generalization:		
Normal Flow of Events: 1. The admin wants to manage the lecturer information in system. 2. The admin request to login. 3. The system navigates to login page. 4. The admin enters ID & password to the system. 5. The system validates the admin ID & password from database. 6. The admin select action to be performed. 7. If the admin wants to add new lecturer information 8. The S – 1: Add new lecturer info performed. 9. If the admin wants to edit existing lecturer information 10. The S – 2: Edit lecturer info performed. 11. If the admin wants to delete existing lecturer information 12. The S – 3: Delete lecturer info performed. 13. The system displays the result. 14. The system end.		
Sub Flows: S – 1: Add new lecturer info 1. The admin enter required information into the system. 2. The admin saves the record into the system.		

S – 2: Edit lecturer info 1. The admin search and navigate to specific lecturer info. 2. The admin edit the lecturer info. 3. The admin saves the record into the system. S – 3: Delete lecturer info 1. The admin search and navigate to specific lecturer info. 2. The admin edit the lecturer info. 3. The admin saves the record into the system.
Alternative/Exceptional Flows: Not applicable

Figure 4.4-F5: Manage Lecturer Information Use-Case Description

Use Case Name: Manage Time Late Policy	ID: 6	Important Level: High
Primary Actor: Admin	Use Case Type: Essential, Detail	
Stakeholders and Interests: Admin – wants to manage the time late policy of every class.		
Brief Description: This use case describes how admin handle the process of setting the time late policy of every class.		
Trigger: The admin wants to manage the time late policy of every class. Type: External		
Relationships: Association: Admin. Include: Update Time Late Policy. Extend: Generalization:		
Normal Flow of Events: 1. The admin wants to manage the time late policy of every class. 2. The admin request to login. 3. The system navigates to login page. 4. The admin enters ID & password to the system. 5. The system validates the admin ID & password from database. 6. The admin change the time late policy of every class. 7. The system saves the policy. 8. The system displays the result. 9. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F6: Manage Time Late Policy Use-Case Description

Use Case Name: Manage Communication Methods	ID: 7	Important Level: High
Primary Actor: Admin	Use Case Type: Essential, Detail	
Stakeholders and Interests: Admin – wants to manage the methods to contact/mail the student/parent.		
Brief Description: This use case describes how admin handle the process of email or SMS to the student/parent.		
Trigger: The admin wants to manage the methods to contact/mail the student/parent. Type: External		
Relationships: Association: Admin. Include: Extend: Email, Short Message Service (SMS). Generalization:		
Normal Flow of Events: 1. The admin wants to manage the methods to contact/mail the student/parent. 2. The admin request to login. 3. The system navigates to login page. 4. The admin enters ID & password to the system. 5. The system validates the admin ID & password from database. 6. The admin generate email/SMS to parent/student. 7. The admin select the forward methods. 8. If email method selected 9. The system emails the mail to student/parent through email service. 10. The system end. 11. Else 12. The system SMS the message to student/parent through phone service. 13. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F7: Manage Communication Methods Use-Case Description

Use Case Name: Manage Report	ID: 8	Important Level: High
Primary Actor: Admin	Use Case Type: Essential, Detail	
Stakeholders and Interests: Admin – wants to manage and analyze report about the student attendance record.		
Brief Description: This use case describes how admin manage the process of report analysis and generation.		
Trigger: The admin wants to manage and analyze report about the student attendance record. Type: External		
Relationships: Association: Admin. Include: Extend: Sort Report by Types, Print Report. Generalization:		
Normal Flow of Events: 1. The admin wants to manage and analyze report about the student attendance record. 2. The admin request to login. 3. The system navigates to login page. 4. The admin enters ID & password to the system. 5. The system validates the admin ID & password from database. 6. The admin provide required information to the system. 7. The admin create attendance report of all students in a class. 8. The system display report result. 9. The admin analyze attendance record result. 10. If admin choose to sort report 11. The system sort report by categories. 11. If admin choose to print out report 12. The system print out the report. 13. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F8: Manage Report Use-Case Description

Use Case Name: Search Attendance History	ID: 9	Important Level: High
Primary Actor: Lecturer	Use Case Type: Essential, Detail	
Stakeholders and Interests: Lecturer – wants to search history of the student attendance record.		
Brief Description: This use case describes how lecturer searches the history of student attendance record.		
Trigger: The lecturer wants to search history of the student attendance record. Type: External		
Relationships: Association: Lecturer. Include: Extend: Generalization:		
Normal Flow of Events: 1. The lecturer wants to search history of the student attendance record. 2. The lecturer request to login. 3. The system navigates to login page. 4. The lecturer enters ID & password to the system. 5. The system validates the lecturer ID & password from database. 6. The lecturer begins searching on attendance history. 7. The system navigates to search attendance history page. 8. The lecturer enters required searching details. 9. The system performs searching. 10. The system display result. 11. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F9: Search Attendance History Use-Case Description

Use Case Name: Manually Key-in Attendance	ID: 10	Important Level: High
Primary Actor: Lecturer	Use Case Type: Essential, Detail	
Stakeholders and Interests: Lecturer – wants to key-in the student attendance manually.		
Brief Description: This use case describes how lecturers key-in the student attendance into the system manually.		
Trigger: The lecturer wants to key-in the student attendance manually. Type: External		
Relationships: Association: Lecturer. Include: Update Database. Extend: Generalization:		
Normal Flow of Events: 1. The lecturer wants to key-in the student attendance manually. 2. The lecturer request to manually key-in student attendance. 3. The system navigates to identity verification page. 4. The lecturer password to verify its identity. 5. The system validates the lecturer password from database. 6. The lecturer modifies attendance of selected student. 7. The lecturer provides modification reason for the modification. 8. The lecturer saves the modified attendance record status. 9. The system display result. 10. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F10: Manually Key-In Attendance Use-Case Description

Use Case Name: Manage Report	ID: 11	Important Level: High
Primary Actor: Lecturer	Use Case Type: Essential, Detail	
Stakeholders and Interests: Lecturer – wants to create class attendance record based on report types.		
Brief Description: This use case describes how lecturer creates class attendance report.		
Trigger: The admin wants to create class attendance record based on report types. Type: External		
Relationships: Association: Lecturer. Include: Extend: Daily Report, Individual Report, Tri-semester Report. Generalization:		
Normal Flow of Events: 1. The lecturer wants to create class attendance record based on report types. 2. The lecturer request to login. 3. The system navigates to login page. 4. The lecturer enters ID & password to the system. 5. The system validates the lecturer ID & password from database. 6. The lecturer select type of reports to be generates. 7. If lecturer choose to generate daily class attendance report 8. The system generates daily class report. 9. If lecturer choose to generate individual attendance report 10. The system generates individual report. 11. If lecturer choose to generate tri-semester attendance report 11. The system generates tri-semester report. 12. The system displays the report result. 13. The lecturer analyzes the report. 14. If lecturer print our report 15. The system print out the report. 16. The system end. 17. Else 18. The system end.		
Sub Flows: Not applicable		
Alternative/Exceptional Flows: Not applicable		

Figure 4.4-F11: Manage Report Use-Case Description

4.5 Low-level Class Diagram

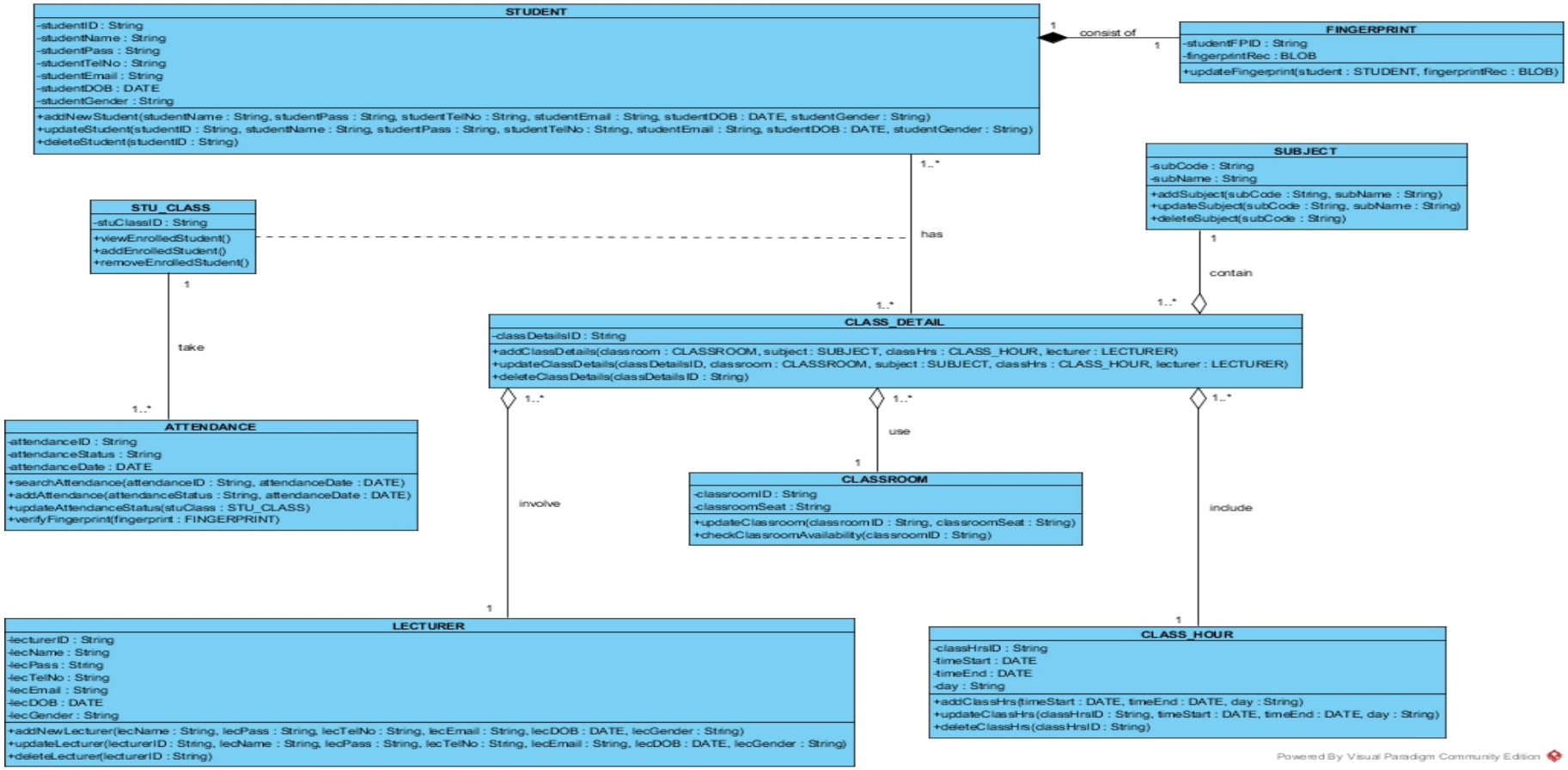

Figure 4.5-F1: Low-level Class Diagram

4.6 Object-Diagram

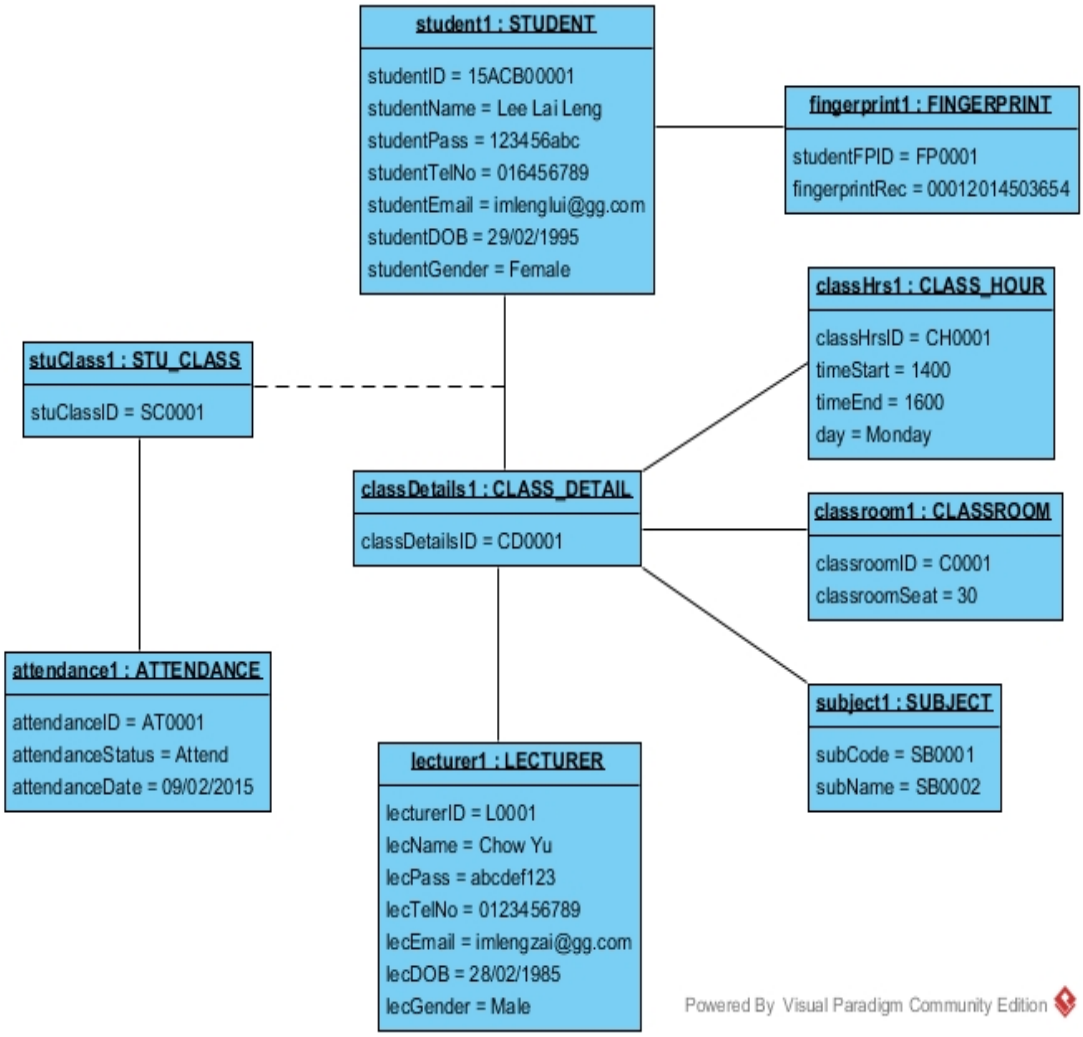

Figure 4.6-F1: Object-Diagram

4.7 CRC Card

Front side of STUDENT class

Class Name: STUDENT	ID: 1	TYPE: Concrete, Domain
Description: An individual that needs to register his/her details or to use the attendance system.		Associated Use Cases: 1, 2, 3, 4, 7, 8, 9

Responsibilities	Collaborators
addNewStudent	FINGERPRINT
UpdateStudent	CLASS_DETAILS
deleteStudent	STU_CLASS

Back side of STUDENT class

Attributes:

studentID	studentEmail
studentName	studentDOB
studentPass	studentGender
studentTelNo	

Relationships:

Generalization (a – kind – of): not applicable

Aggregation (has – parts): not applicable

Other Associations: FINGERPRINT, CLASS_DETAILS, STU_CLASS

Figure 4.7-F1: CRC Card of STUDENT class

Front side of FINGERPRINT class

Class Name: FINGERPRINT	ID: 2	TYPE: Concrete, Domain
Description: A class that used to store all the fingerprint images of students used in attendance verification.		Associated Use Cases: 1
Responsibilities updateFingerprint		Collaborators STUDENT

Back side of FINGERPRINT class

Attributes:	
studentFPID	fingerPrintRec
Relationships: Generalization (a – kind – of): not applicable Aggregation (has – parts): not applicable Other Associations: STUDENT	

Figure 4.7-F2: CRC Card of FINGERPRINT class

Front side of ATTENDANCE class

Class Name: ATTENDANCE	ID: 3	TYPE: Concrete, Domain
Description: A class that used to gain and store the attendance record of all students during every class.		Associated Use Cases: 1, 2, 3, 6, 9, 11

Responsibilities	Collaborators
searchAttendance addAttendance updateAttendanceStatus verifyFingerprint	STU_CLASS

Back side of ATTENDANCE class

Attributes:	
attendanceID	attendanceDate
attendanceStatus	

Relationships:

Generalization (a – kind – of): not applicable

Aggregation (has – parts): not applicable

Other Associations: STU_CLASS

Figure 4.7-F3: CRC Card of ATTENDANCE class

Front side of STU_CLASS class

Class Name: STU_CLASS	ID: 4	TYPE: Concrete, Domain
Description: A class that used to store and list out students that enrolled in which subjects.		Associated Use Cases: 4, 5

Responsibilities	Collaborators
viewEnrolledStudent addEnrolledStudent removeEnrolledStudent	STUDENT ATTENDANCE CLASS_DETAILS

Back side of STU_CLASS class

Attributes:
stuClassID

Relationships:
Generalization (a – kind – of): not applicable Aggregation (has – parts): not applicable Other Associations: STUDENT, ATTENDANCE, CLASS_DETAILS

Figure 4.7-F4: CRC Card of STU_CLASS class

Front side of LECTURER class

Class Name: LECTURER	ID: 5	TYPE: Concrete, Domain
Description: An individual that manage the manually key-in attendance of student and create attendance reports.		Associated Use Cases: 5, 9, 10, 11

Responsibilities	Collaborators
addNewLecturer UpdateLecturer deleteLecturer	CLASS_DETAILS

Back side of LECTURER class

Attributes:

lecturerID	lecEmail
lecName	lecDOB
lecPass	lecGender
lecTelNo	

Relationships:

Generalization (a – kind – of): not applicable

Aggregation (has – parts): not applicable

Other Associations: CLASS_DETAILS

Figure 4.7-F5: CRC Card of LECTURER class

Front side of CLASSROOM class

Class Name: CLASSROOM	ID: 6	TYPE: Concrete, Domain
Description: A class that used to store the classroom details and availability of the classroom.		Associated Use Cases: 3, 10,
Responsibilities updateClassroom checkClassroomAvailability		Collaborators CLASS_DETAILS

Back side of CLASSROOM class

Attributes: classroomID classroomSeat
Relationships: Generalization (a – kind – of): not applicable Aggregation (has – parts): not applicable Other Associations: CLASS_DETAILS

Figure 4.7-F6: CRC Card of CLASSROOM class

Front side of SUBJECT class

Class Name: SUBJECT	ID: 7	TYPE: Concrete, Domain
Description: A class that used to stores and manages the subject details.		Associated Use Cases: 3, 8, 9, 11
Responsibilities addSubject updateSubject deleteSubject		Collaborators CLASS_DETAILS

Back side of SUBJECT class

Attributes:	
subCode	subName

Relationships:

Generalization (a – kind – of): not applicable

Aggregation (has – parts): not applicable

Other Associations: CLASS_DETAILS

Figure 4.7-F7: CRC Card of SUBJECT class

Front side of CLASS_HOUR class

Class Name: CLASS_HOUR	ID: 8	TYPE: Concrete, Domain
Description: A class that used to manage the time start/end and day of every class.		Associated Use Cases: 3, 10,

Responsibilities	Collaborators
addClassHrs updateClassHrs deleteClassHrs	CLASS_DETAILS

Back side of CLASS_HOUR class

Attributes:

classHrsID	timeEnd
timeStart	day

Relationships:

Generalization (a – kind – of): not applicable

Aggregation (has – parts): not applicable

Other Associations: CLASS_DETAILS

Figure 4.7-F8: CRC Card of CLASS_HOUR class

Front side of CLASS_DETAILS class

Class Name: CLASS_DETAILS	ID: 9	TYPE: Concrete, Domain
Description: A class that combined few objects information to form a schedule of a subject to be attends by students.		Associated Use Cases: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Responsibilities	Collaborators
addClassDetails updateClassDetails deleteClassDetails	STUDENT STU_CLASS LECTURER CLASS_HOUR CLASSROOM SUBJECT

Back side of CLASS_DETAILS class

Attributes:
classDetailsID
Relationships:
Generalization (a – kind – of): not applicable Aggregation (has – parts): LECTURER, CLASS_HOUR, CLASSROOM, SUBJECT Other Associations: STUDENT, STU_CLASS

Figure 4.7-F9: CRC Card of CLASS_DETAILS class

4.8 Sequence Diagram

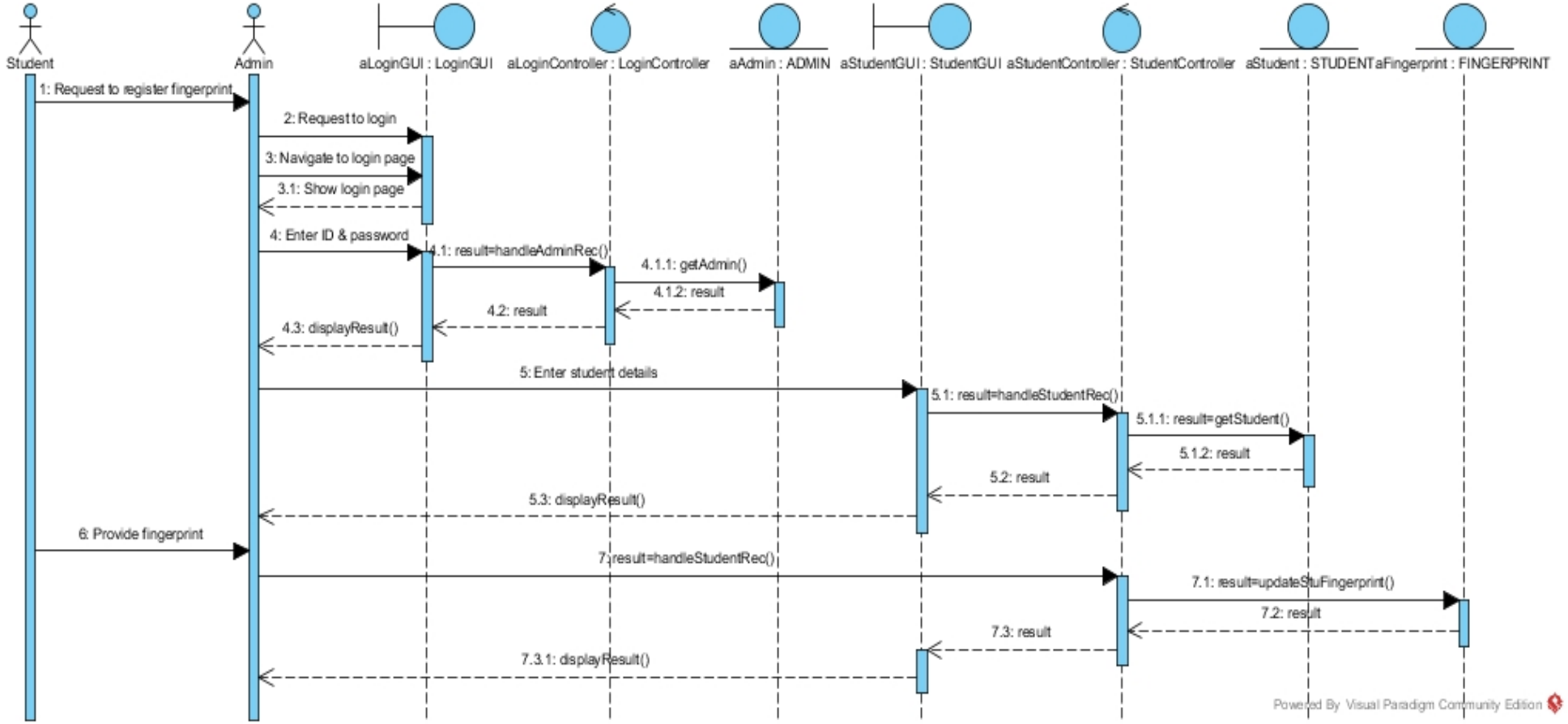

Figure 4.8-F1: Sequence Diagram of Register Fingerprint (Student)

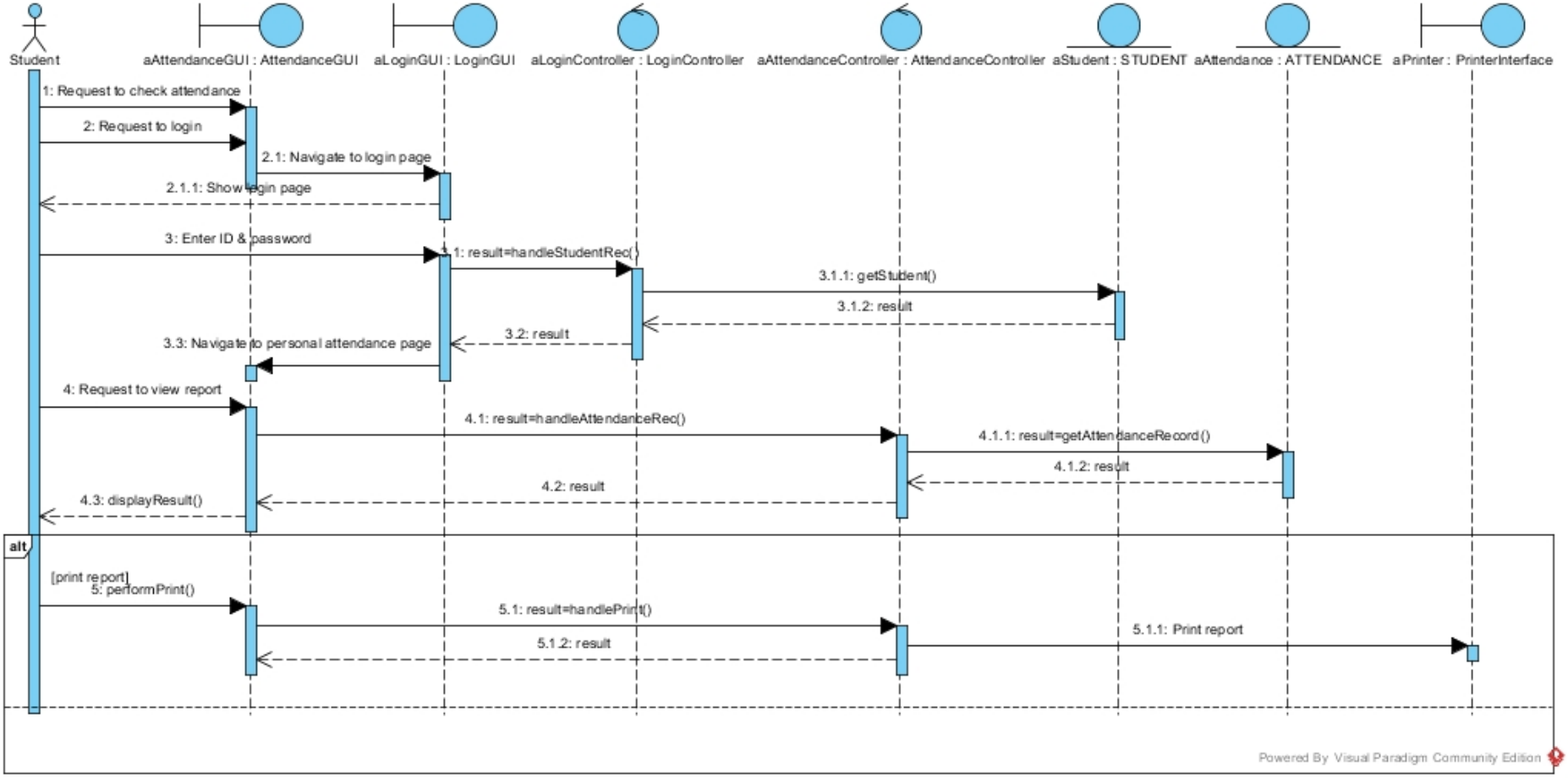

Figure 4.8-F2: Sequence Diagram of Keep Track of Personal Attendance Record (Student)

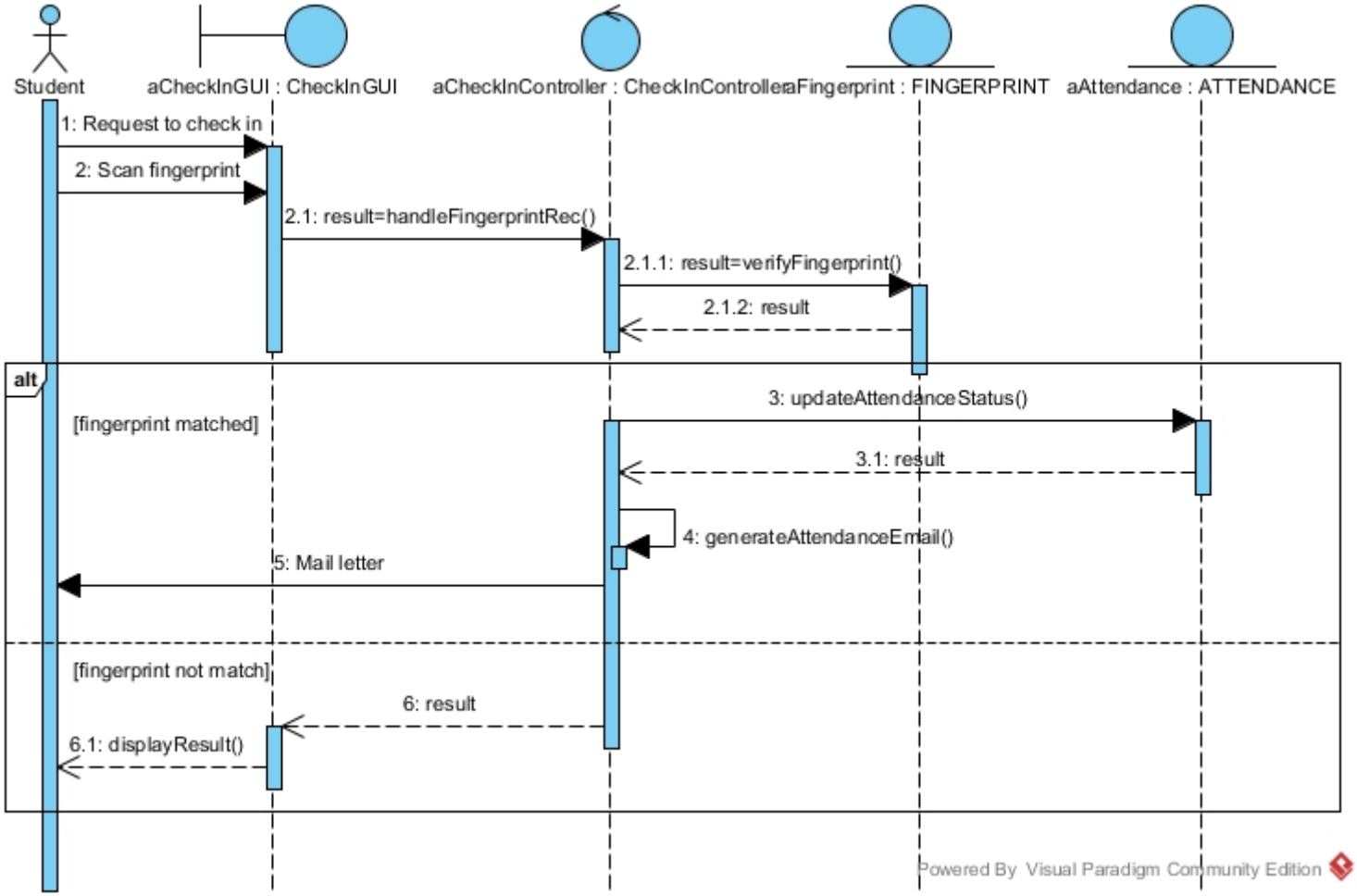

Figure 4.8-F3: Sequence Diagram of Check-in Attendance (Student)

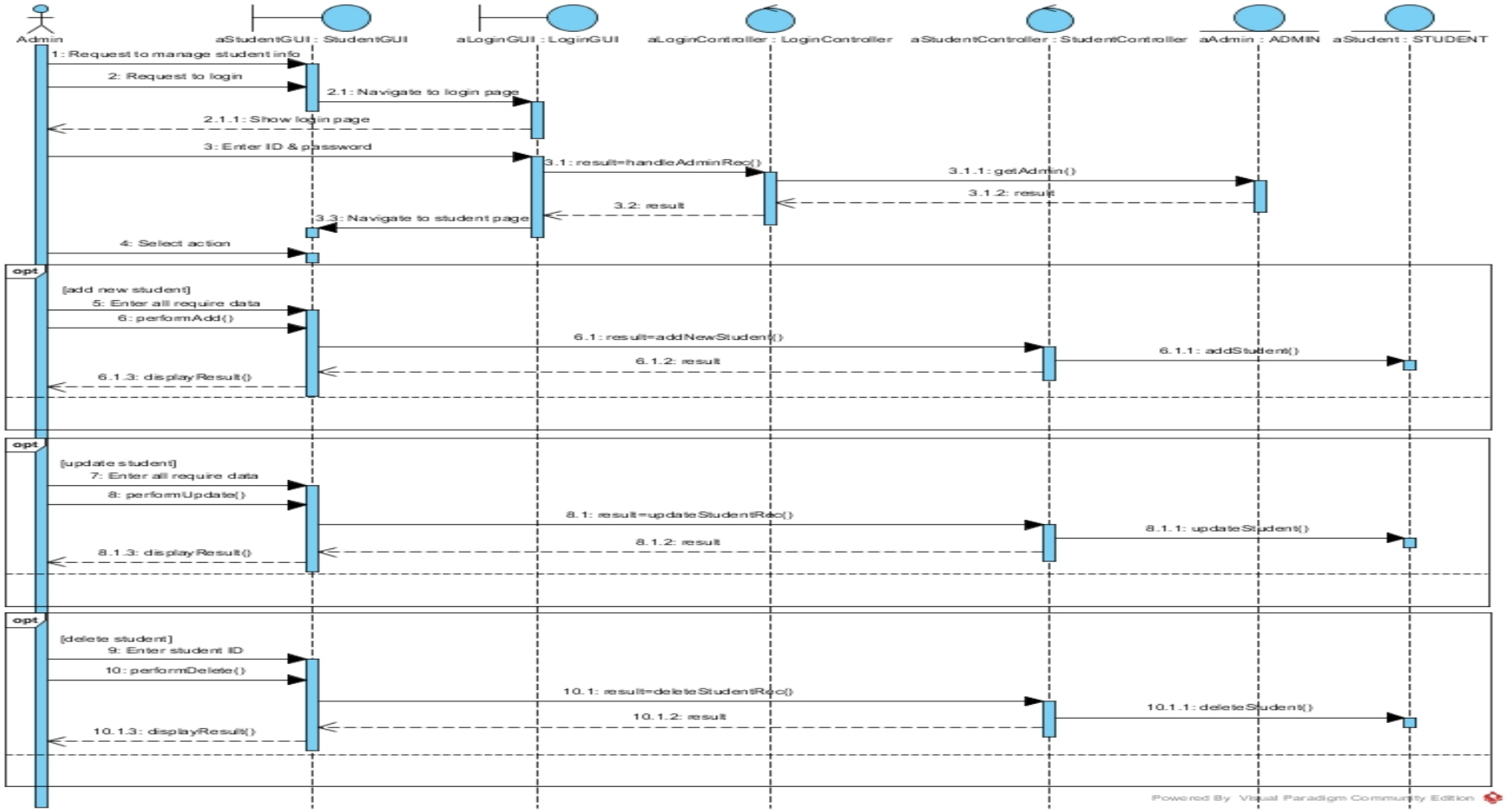

Figure 4.8-F4: Sequence Diagram of Manage Student Information (Admin)

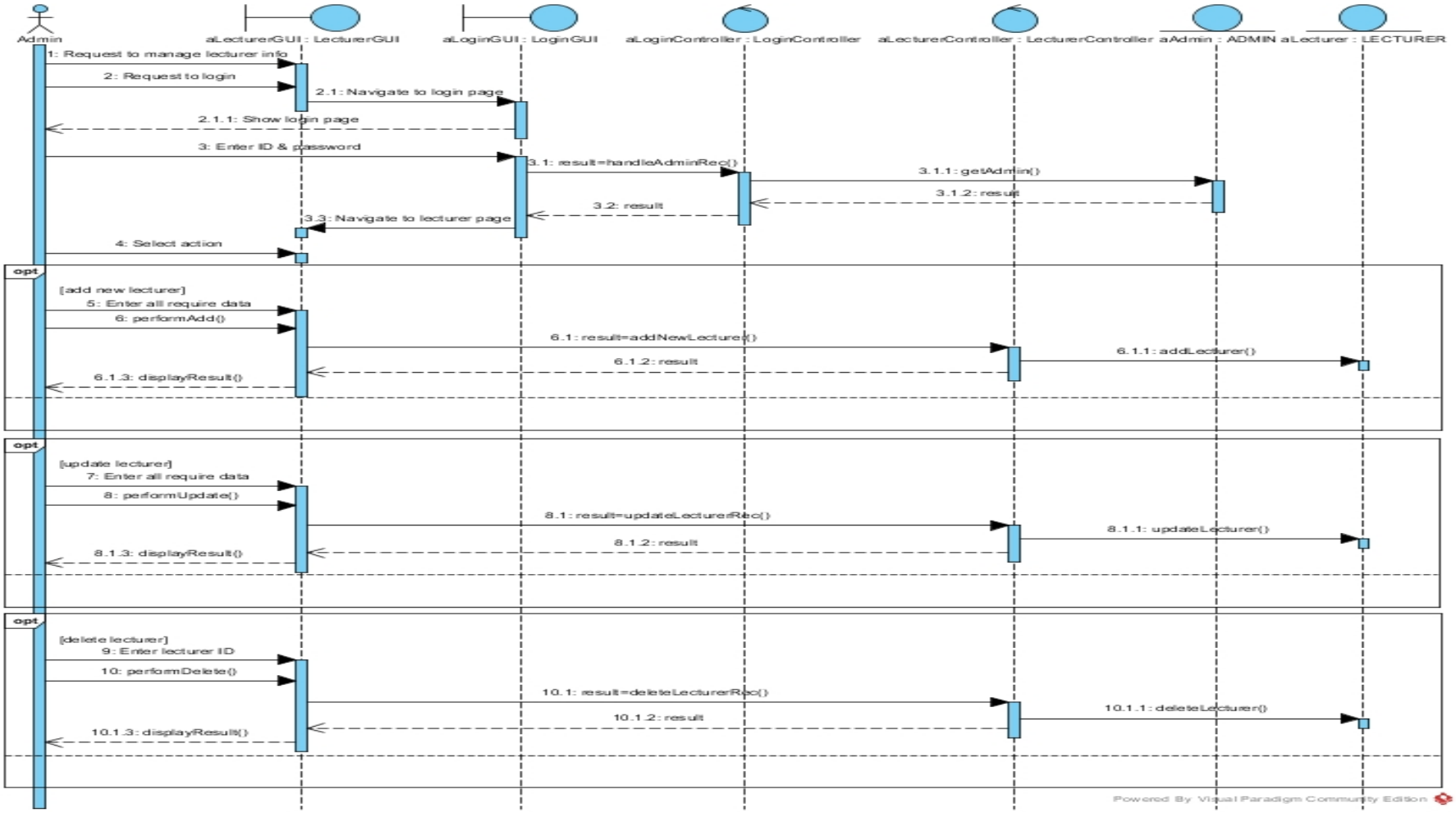

Figure 4.8-F5: Sequence Diagram of Manage Lecturer Information (Admin)

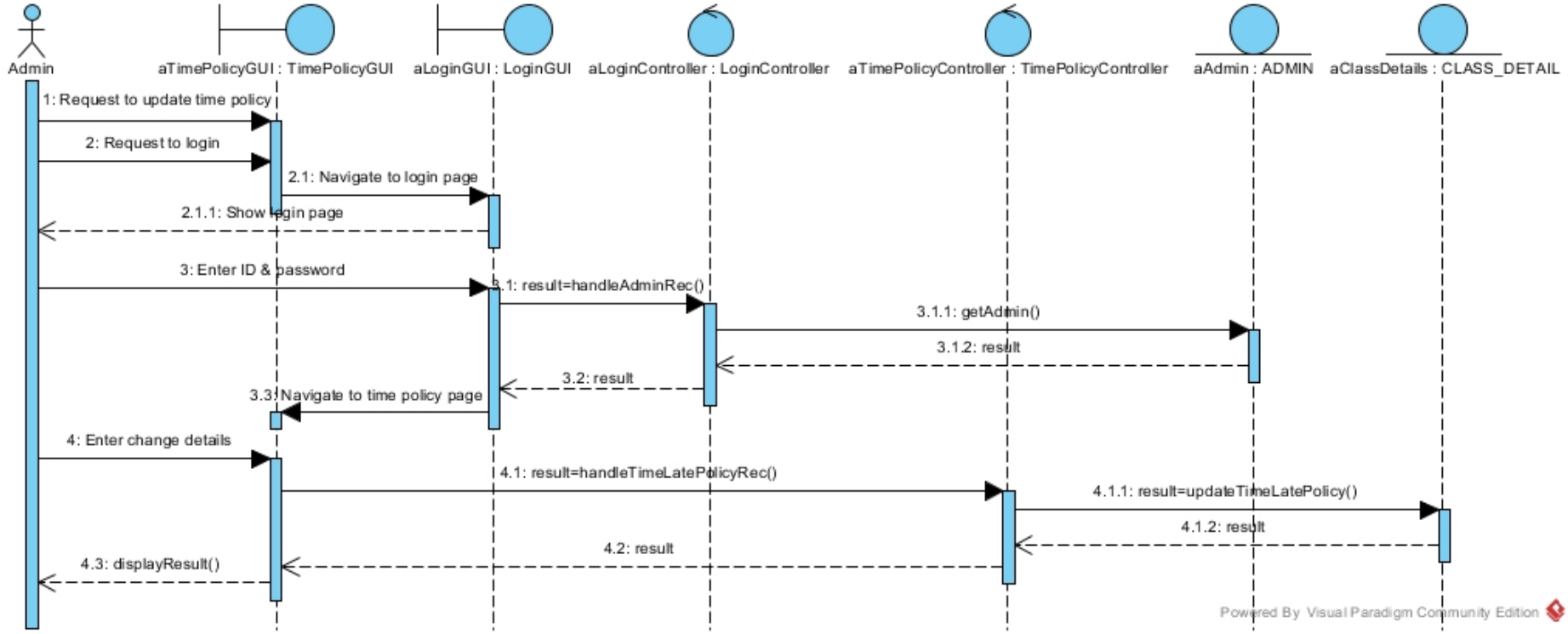

Figure 4.8-F6: Sequence Diagram of Manage Time Late Policy (Admin)

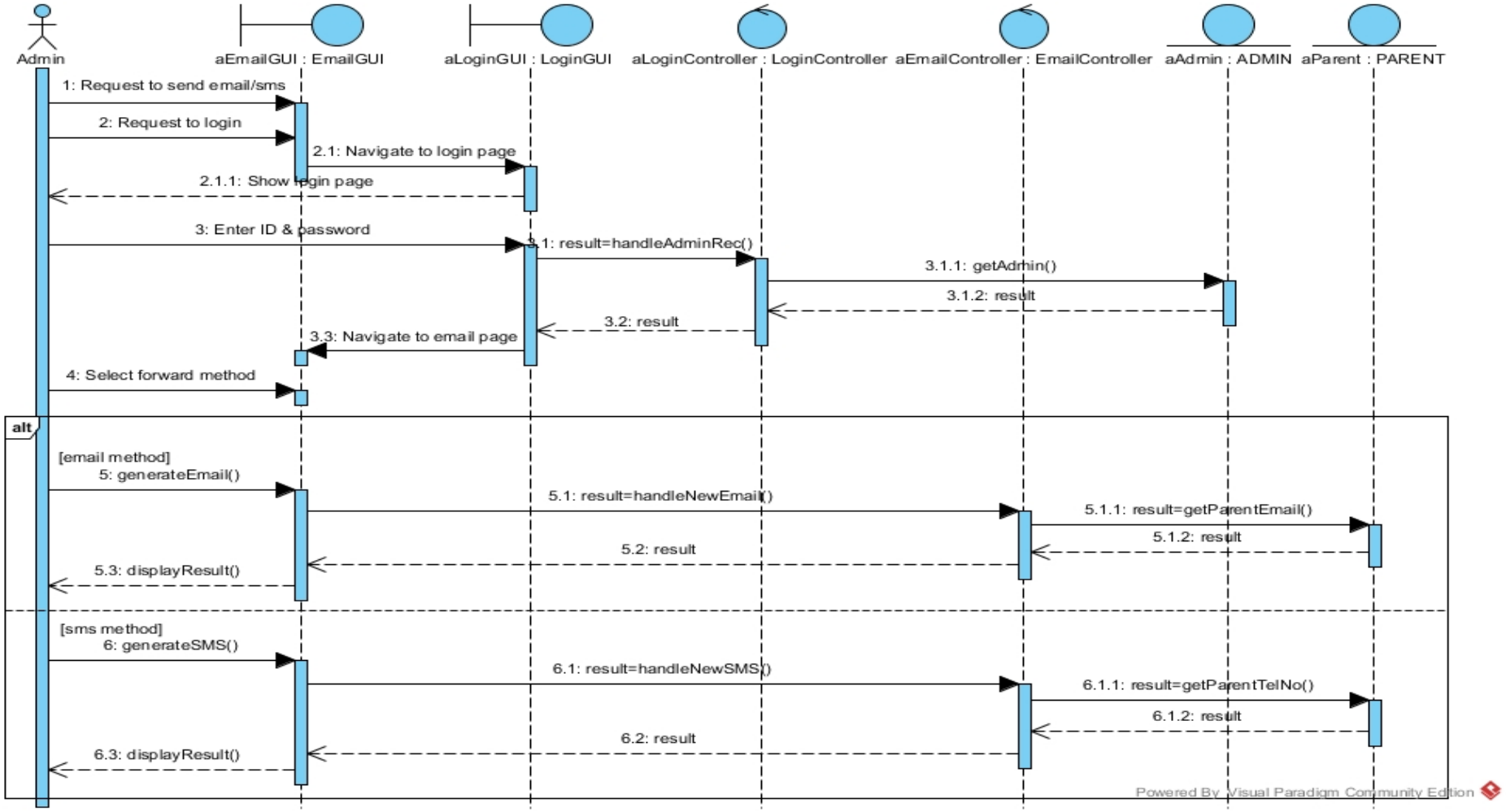

Figure 4.8-F7: Sequence Diagram of Manage Communication Methods (Admin)

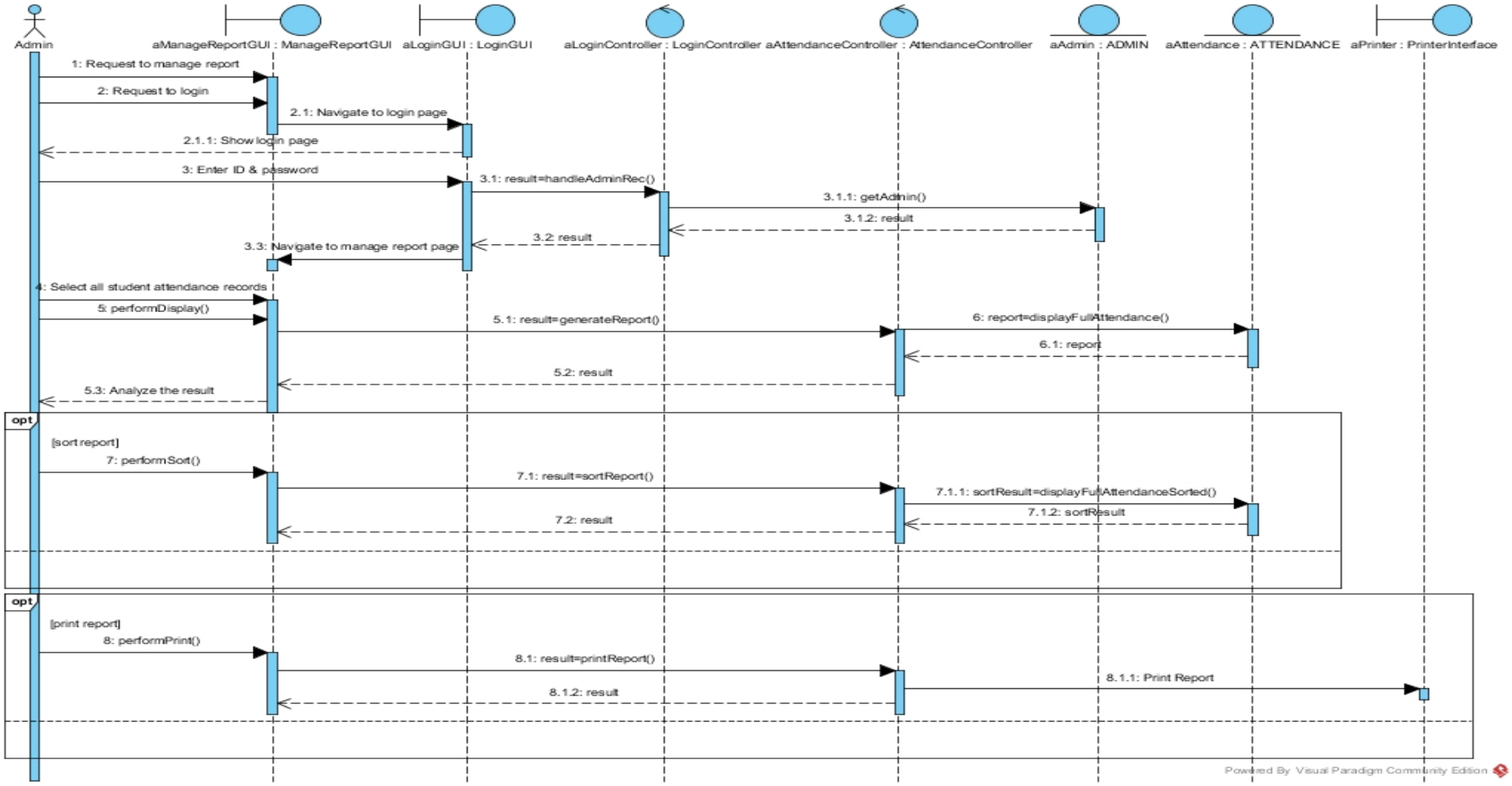

Figure 4.8-F8: Sequence Diagram of Manage Report (Admin)

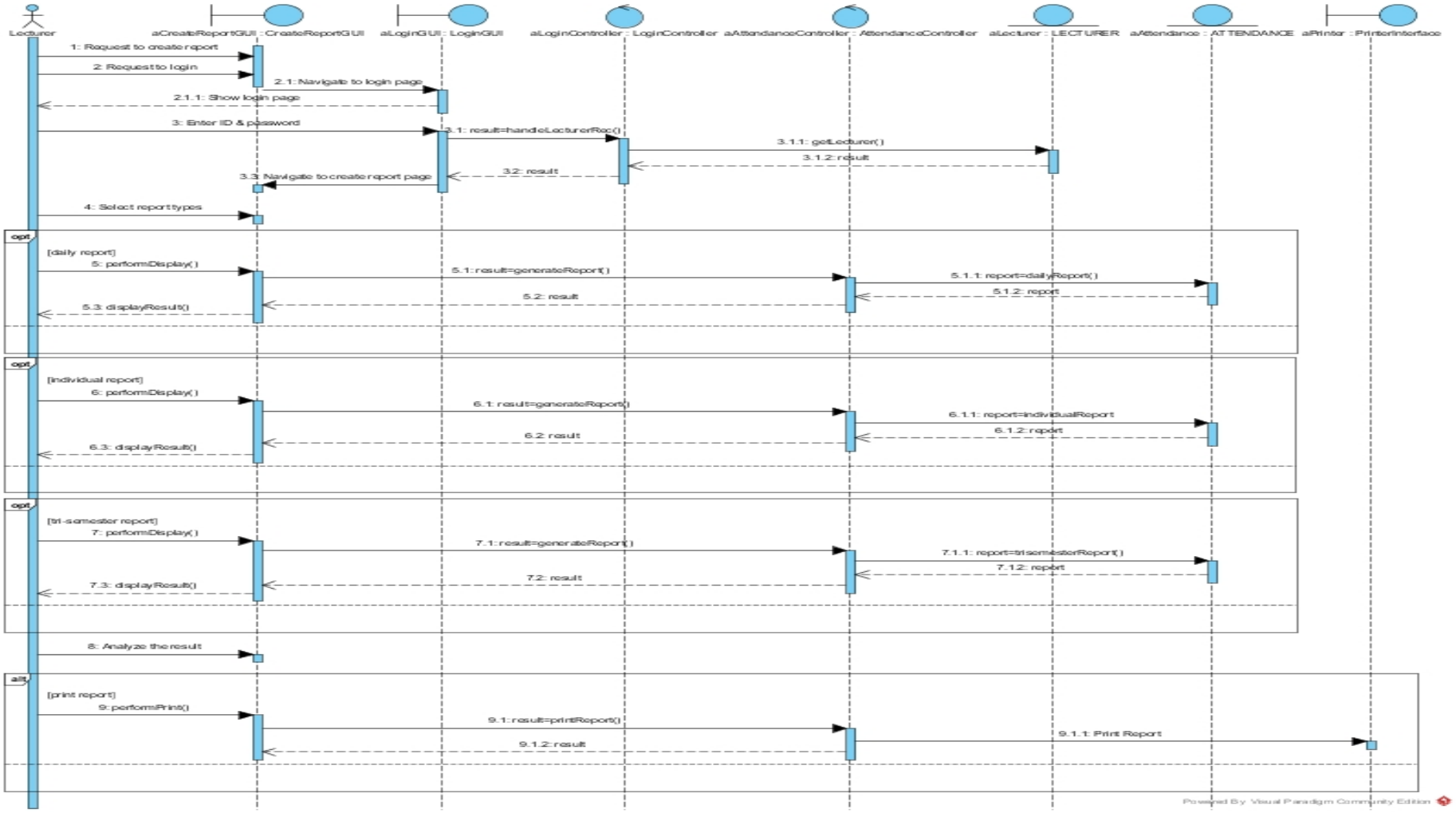

Figure 4.8-F9: Sequence Diagram of Create Attendance Report (Lecturer)

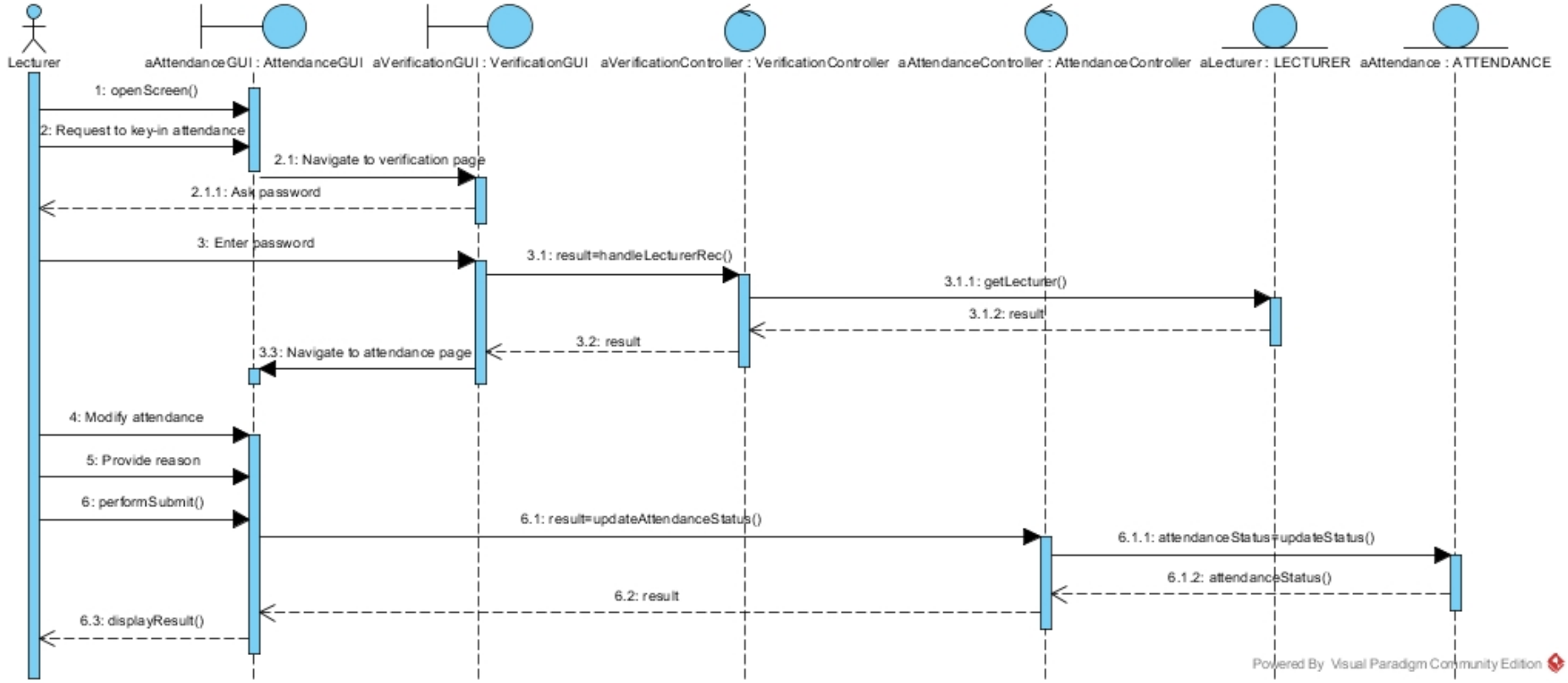

Figure 4.8-F10: Sequence Diagram of Manually Key-in Attendance (Lecturer)

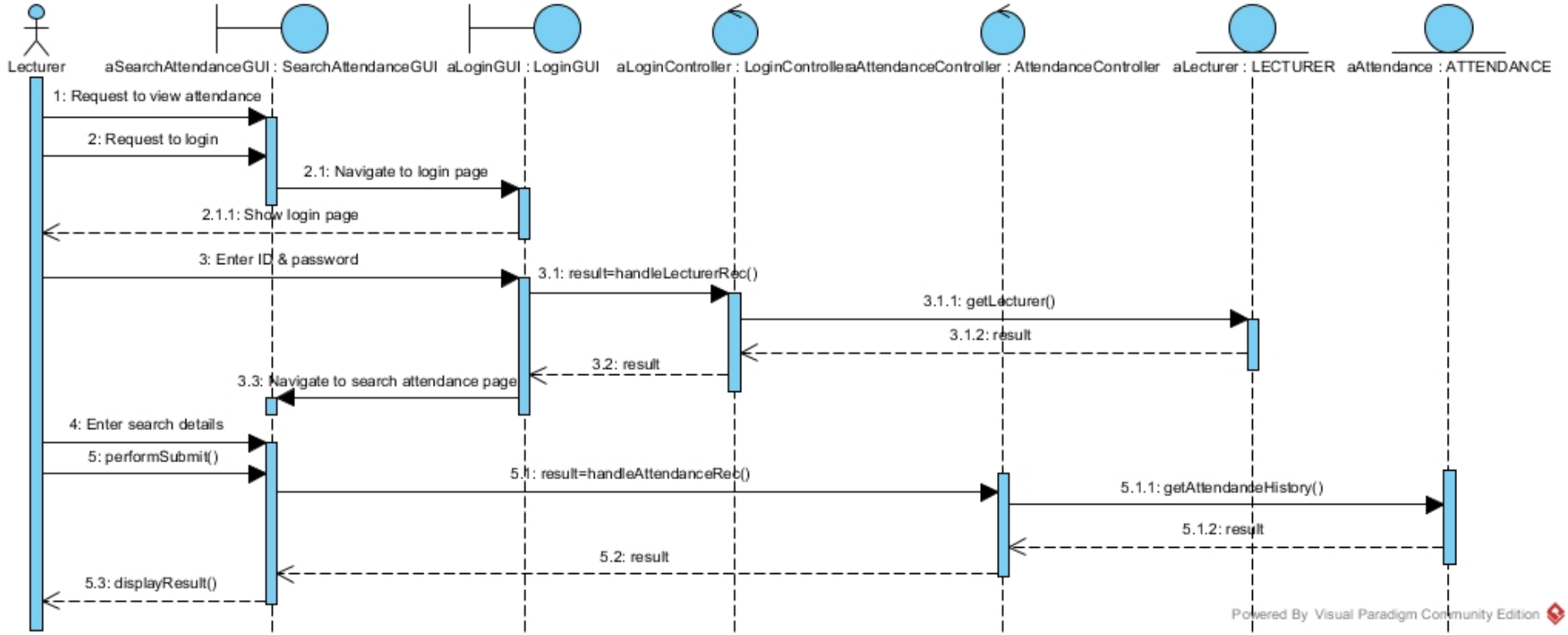

Figure 4.8-F11: Sequence Diagram of Search Attendance History (Lecturer)

4.9 State Machine Diagram

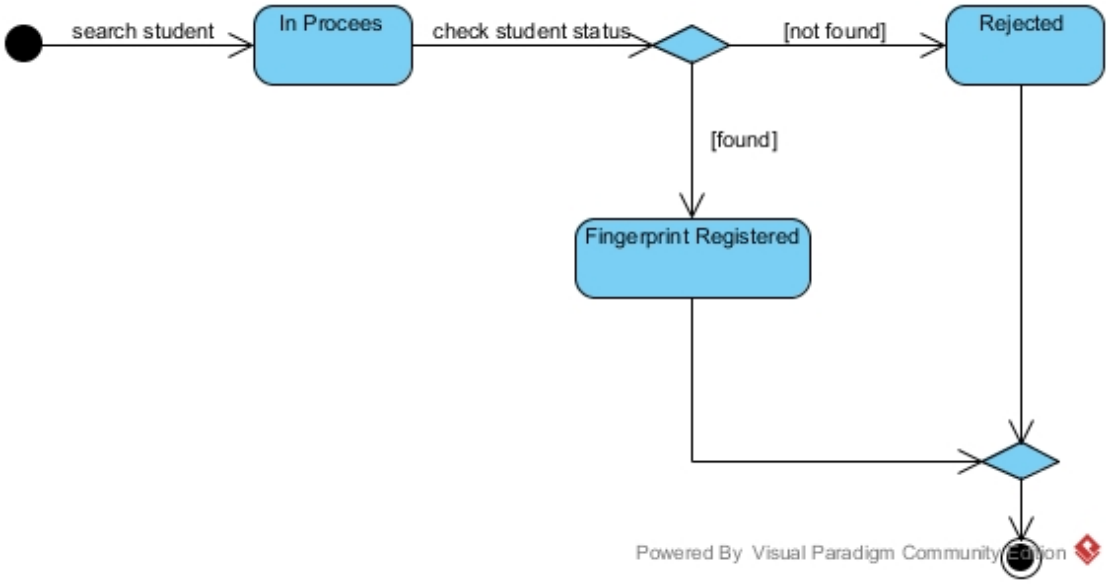

Figure 4.9-F1: State Machine Diagram of Register Fingerprint

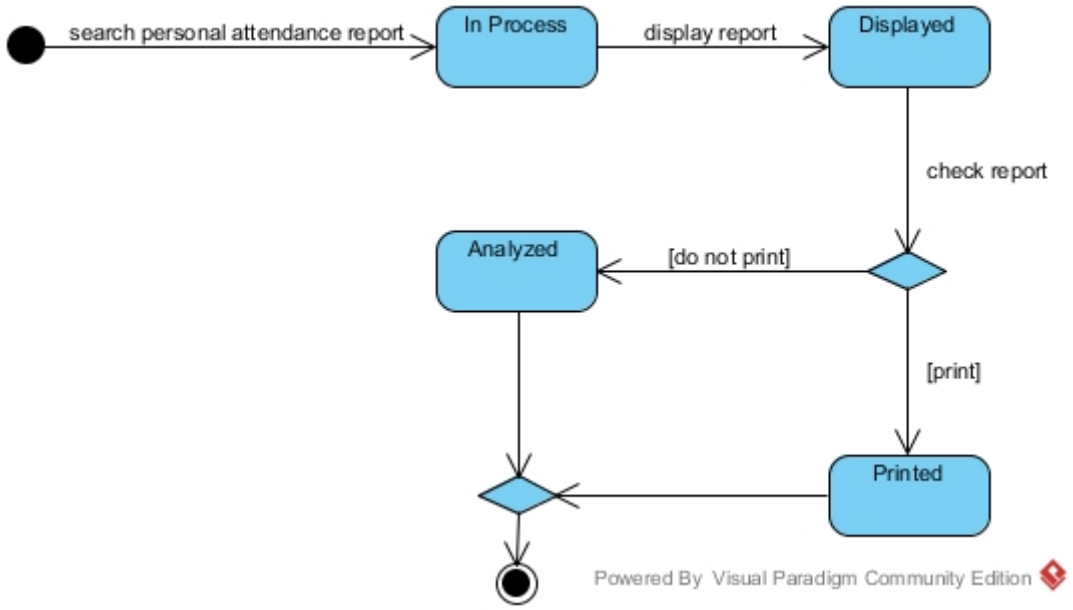

Figure 4.9-F2: State Machine Diagram of Keep Track of Attendance Record

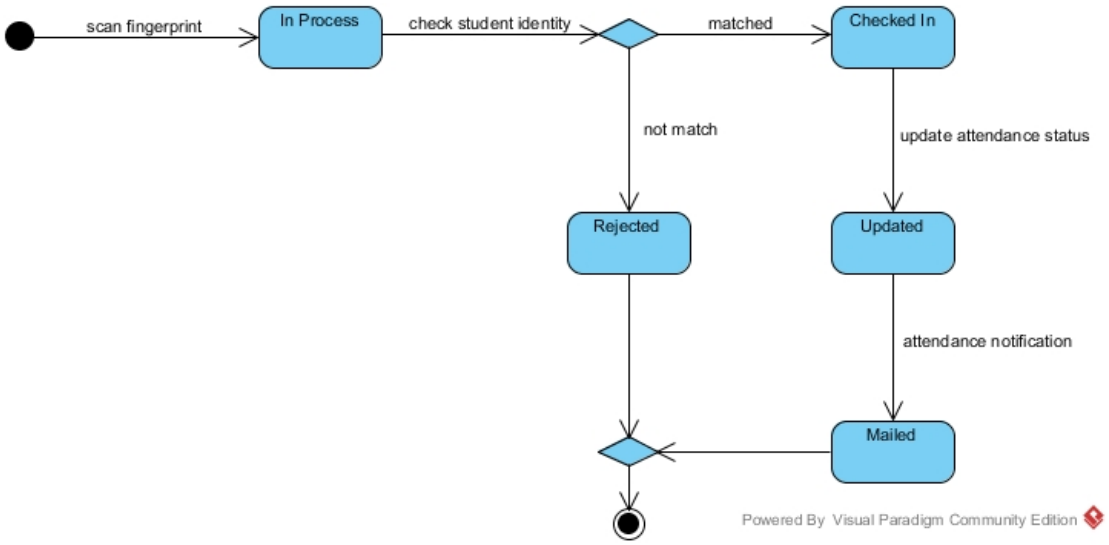

Figure 4.9-F3: State Machine Diagram of Check-in Attendance

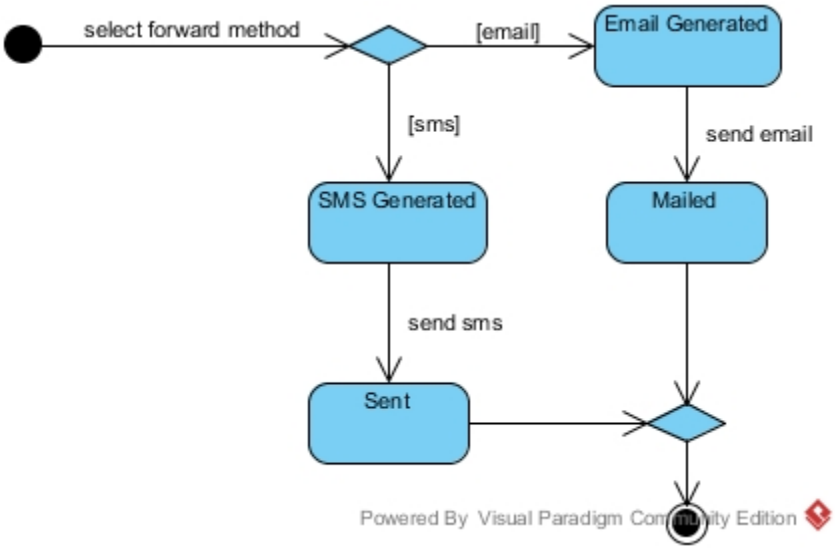

Figure 4.9-F4: State Machine Diagram of Manage Communication Methods

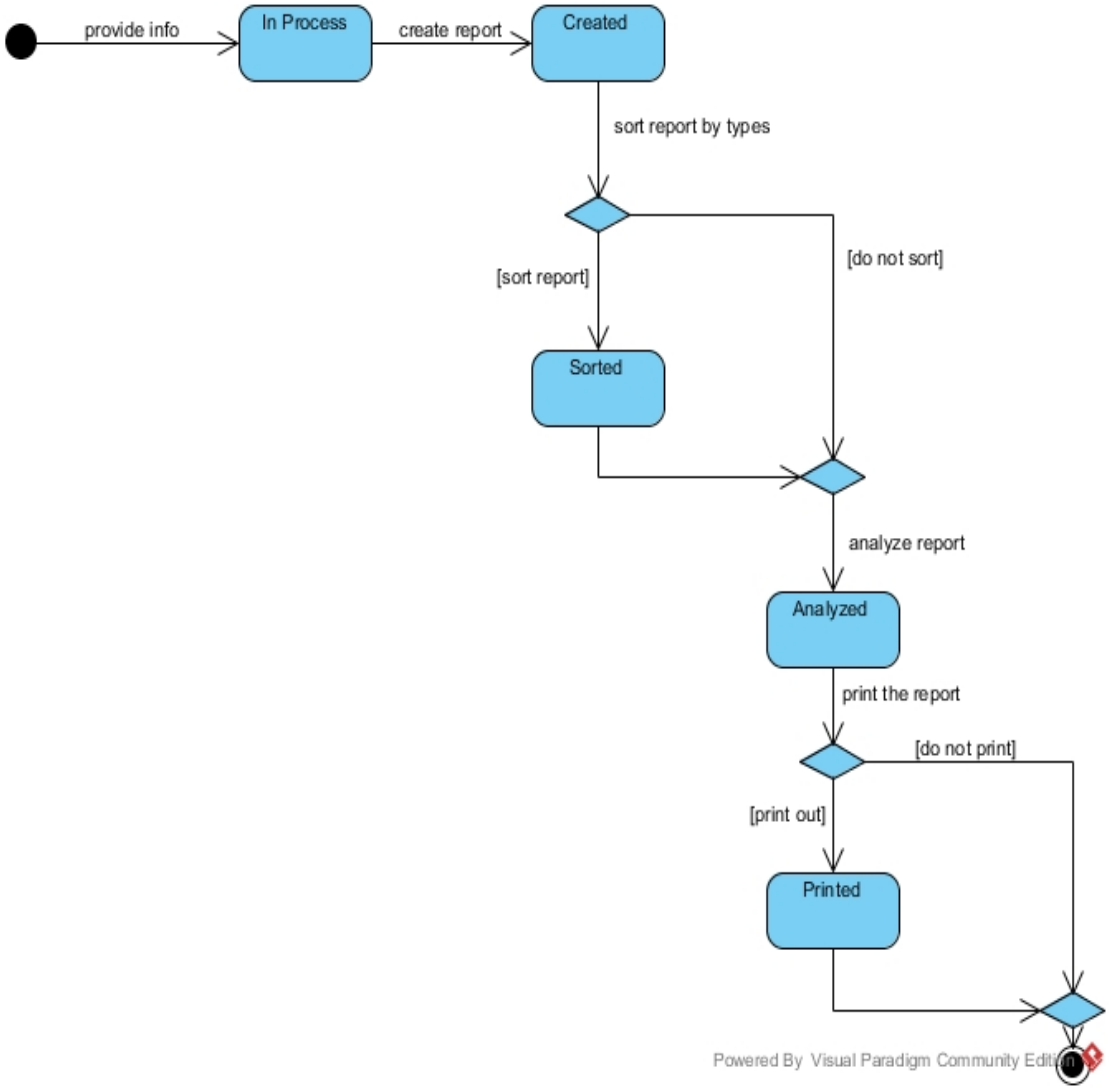

Figure 4.9-F5: State Machine Diagram of Manage Report

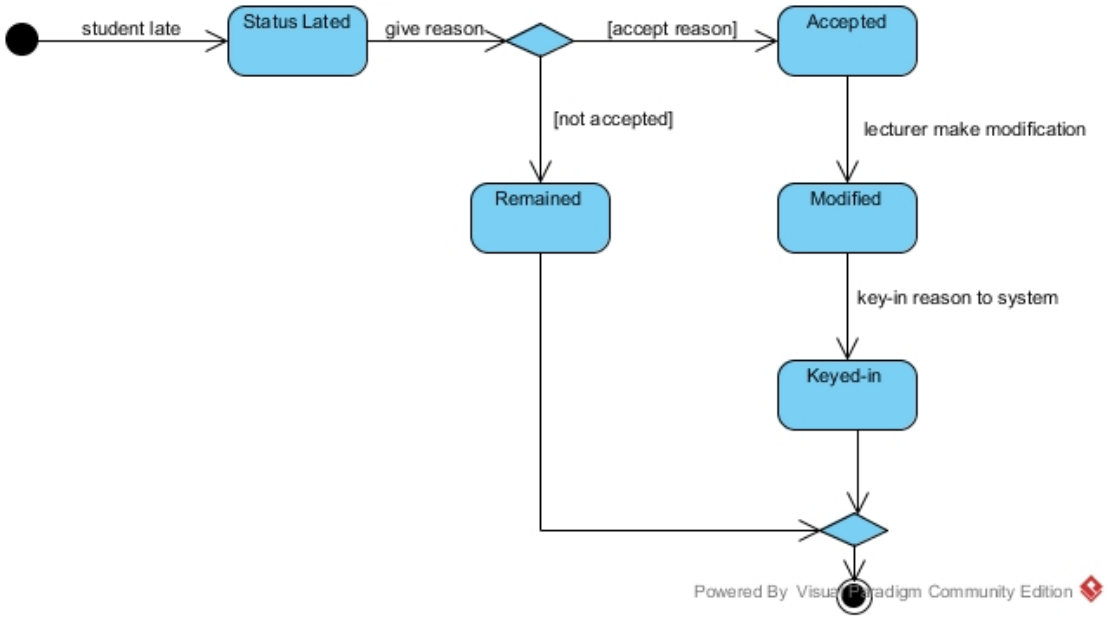

Figure 4.9-F6: State Machine Diagram of Manually Key-in Attendance

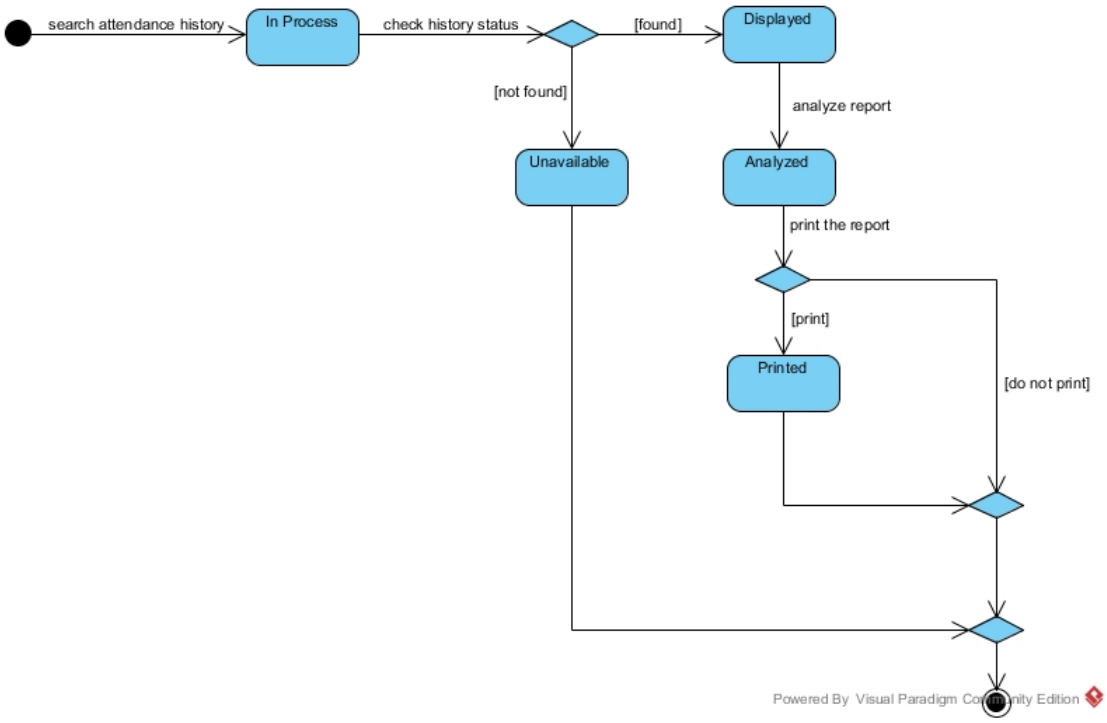

Figure 4.9-F7: State Machine Diagram of Search Attendance History

4.10 CRUDE Analysis

	Student	Fingerprint	Class_Detail	Stu_Class	Attendance	Lecturer	Classroom	Subject	Class_Hour
Student		C, R, U, D	R, E	R, U, E	R, U		R, E		R, E
Fingerprint	R, E				R, U, E				
Class_Detail	R	R, E		C, R, E	R, E	R	R	R	R, E
Stu_Class	R		C, R, U, D		C, U, E				
Attendance	R	R, E		R					
Lecturer			R, E		R, U, E			R, E	
Classroom			R						
Subject			R, U, D			R, U, E			
Class_Hour			C, R, U, D				R, U		

Table 4.10-F1: Table of CRUDE Analysis

**
C – Create R – Read
U – Update D – Delete
E – Execute

4.11 Interaction Overview Diagram

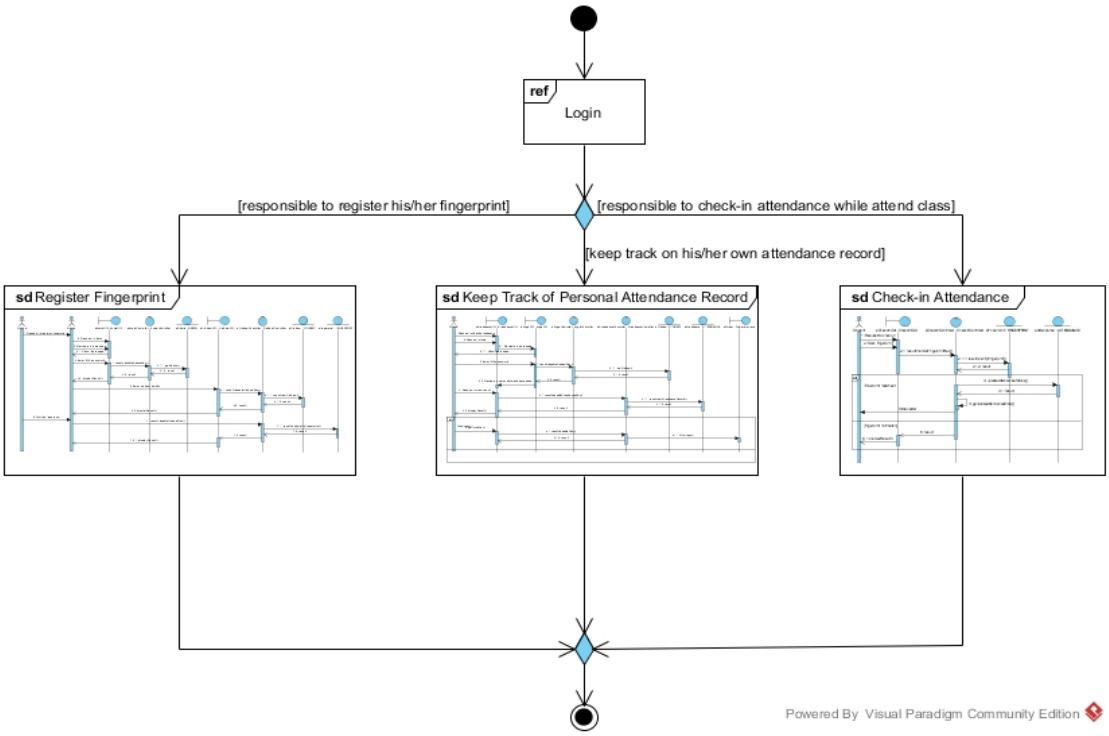

Figure 4.11-F1: Interaction Overview Diagram of Student

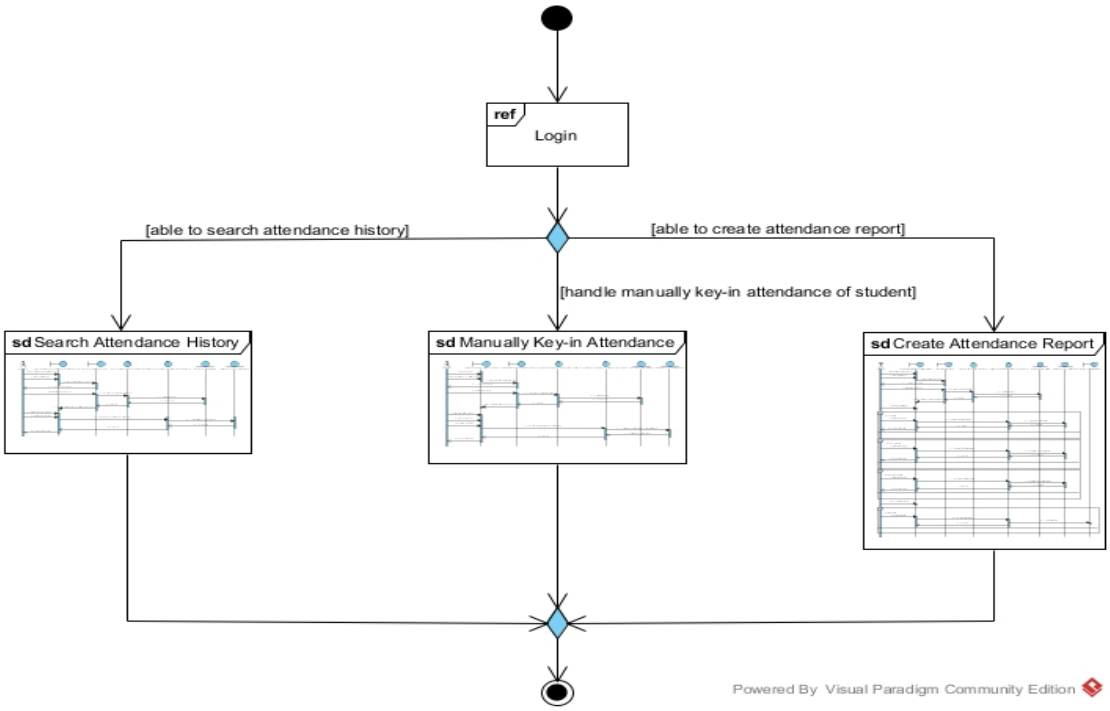

Figure 4.11-F2: Interaction Overview Diagram of Lecturer

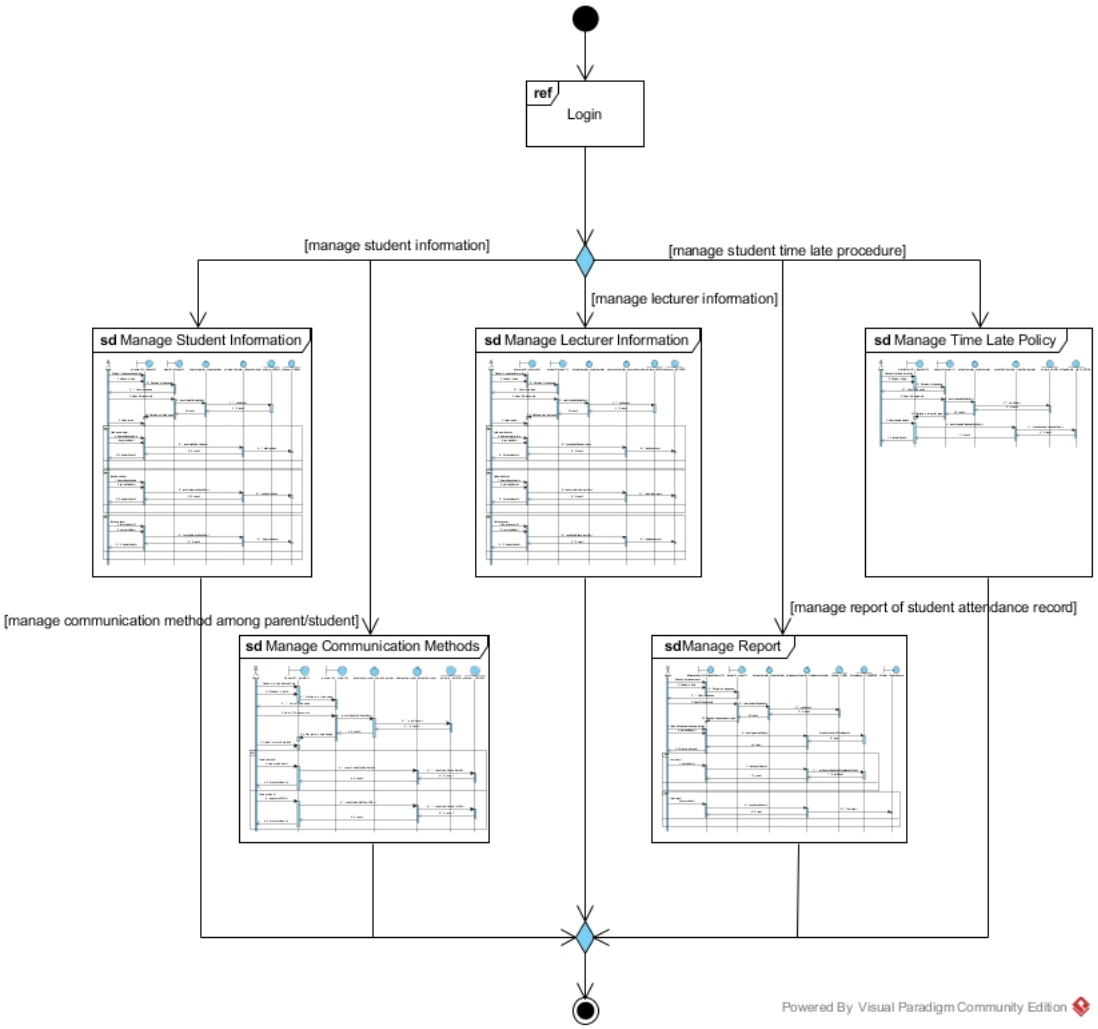

Figure 4.11-F3: Interaction Overview Diagram of Admin

4.12 Low-level Class Diagram with Invariants

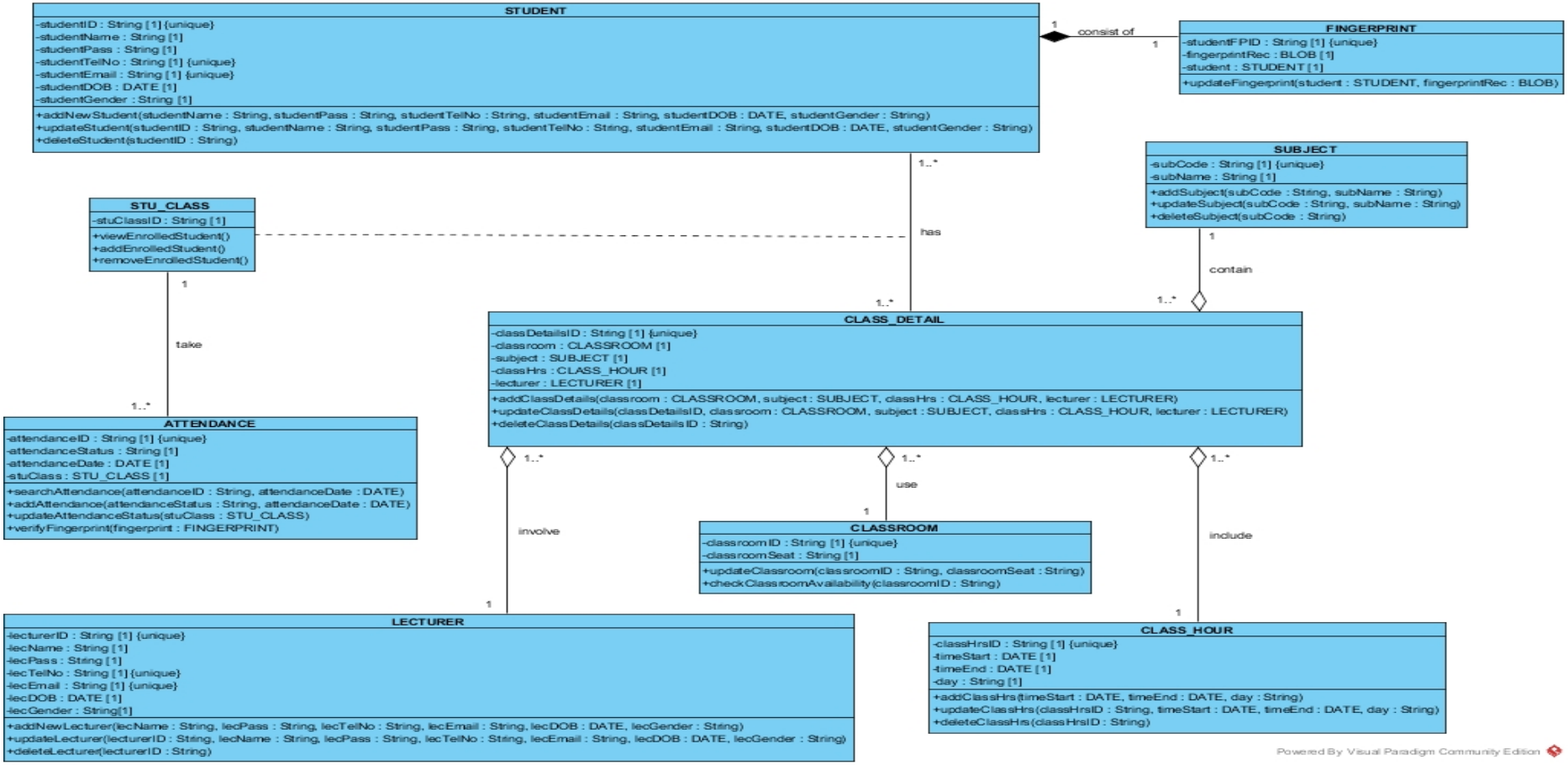

Figure 4.12-F1: Low-level Class Diagram with Invariants

4.13 CRC Card with Invariants

Front side of STUDENT class

Class Name: STUDENT	ID: 1	TYPE: Concrete, Domain
Description: An individual that needs to register his/her details or to use the attendance system.		Associated Use Cases: 1, 2, 3, 4, 7, 8, 9

Responsibilities	Collaborators
addNewStudent	FINGERPRINT
UpdateStudent	CLASS_DETAILS
deleteStudent	STU_CLASS

Back side of STUDENT class

Attributes:

studentID (1…1) (String) {unique}
studentName (1…1) (String)
studentPass (1…1) (String)
studentTelNo (1…1) (String) {unique}
studentEmail (1…1) (String)
studentDOB (1…1) (date)
studentGender (1…1) (String)

Relationships:

Generalization (a – kind – of): not applicable
Aggregation (has – parts): not applicable
Other Associations: FINGERPRINT {1…1}, CLASS_DETAILS {1…*}, STU_CLASS {1…*}

Figure 4.13-F1: CRC Card with Invariants of STUDENT class

Front side of FINGERPRINT class

Class Name: FINGERPRINT	ID: 2	TYPE: Concrete, Domain
Description: A class that used to store all the fingerprint images of students used in attendance verification.		Associated Use Cases: 1
Responsibilities updateFingerprint		Collaborators STUDENT

Back side of FINGERPRINT class

Attributes:
studentFPID (1…1) (String) {unique} fingerPrintRec (1…1) (String) student (1…1) (STUDENT) studentID (1…1) {studentID = student.getStudentID();}
Relationships: Generalization (a – kind – of): not applicable Aggregation (has – parts): not applicable Other Associations: STUDENT (1…1)

Figure 4.13-F2: CRC Card with Invariants of FINGERPRINT class

Front side of ATTENDANCE class

Class Name: ATTENDANCE	ID: 3	TYPE: Concrete, Domain
Description: A class that used to gain and store the attendance record of all students during every class.		Associated Use Cases: 1, 2, 3, 6, 9, 11

Responsibilities	Collaborators
searchAttendance addAttendance updateAttendanceStatus verifyFingerprint	STU_CLASS

Back side of ATTENDANCE class

Attributes:

attendanceID (1…1) (String) {unique}
attendanceStatus (1…1) (String)
attendanceDate (1…1) (date)
stuClass (1…1) (STU_CLASS)
stuClassID (1…1) (String) {stuClassID = stuClass.getStuClassID();}

Relationships:

Generalization (a – kind – of): not applicable
Aggregation (has – parts): not applicable
Other Associations: STU_CLASS (1…1)

Figure 4.13-F3: CRC Card with Invariants of ATTENDANCE class

Front side of STU_CLASS class

Class Name: STU_CLASS	ID: 4	TYPE: Concrete, Domain
Description: A class that used to store and list out students that enrolled in which subjects.		Associated Use Cases: 4, 5

Responsibilities	Collaborators
viewEnrolledStudent	STUDENT
addEnrolledStudent	ATTENDANCE
removeEnrolledStudent	CLASS_DETAILS

Back side of STU_CLASS class

Attributes:
stuClassID (1…1) (String) {unique}

Relationships:
Generalization (a – kind – of): not applicable
Aggregation (has – parts): not applicable
Other Associations: STUDENT {1…*}, ATTENDANCE {1…*}, CLASS_DETAILS {1…*}

Figure 4.13-F4: CRC Card with Invariants of STU_CLASS class

Front side of LECTURER class

Class Name: LECTURER	ID: 5	TYPE: Concrete, Domain
Description: An individual that manage the manually key-in attendance of student and create attendance reports.		Associated Use Cases: 5, 9, 10, 11

Responsibilities	Collaborators
addNewLecturer UpdateLecturer deleteLecturer	CLASS_DETAILS

Back side of LECTURER class

Attributes:

lecturerID (1…1) (String) {unique}
lecName (1…1) (String)
lecPass (1…1) (String)
lecTelNo (1…1) (String) {unique}
lecEmail (1…1) (String) {unique}
lecDOB (1…1) (date)
lecGender (1…1) (String)

Relationships:

Generalization (a – kind – of): not applicable
Aggregation (has – parts): not applicable
Other Associations: CLASS_DETAILS {1…*}

Figure 4.13-F5: CRC Card with Invariants of LECTURER class

Front side of CLASSROOM class

Class Name: CLASSROOM	ID: 6	TYPE: Concrete, Domain
Description: A class that used to store the classroom details and availability of the classroom.		Associated Use Cases: 3, 10,
Responsibilities updateClassroom checkClassroomAvailability	Collaborators CLASS_DETAILS	

Back side of CLASSROOM class

Attributes: classroomID (1…1) (String) {unique} classroomSeat (1…1) (String)
Relationships: Generalization (a – kind – of): not applicable Aggregation (has – parts): not applicable Other Associations: CLASS_DETAILS {1…*}

Figure 4.13-F6: CRC Card with Invariants of CLASSROOM class

Front side of SUBJECT class

Class Name: SUBJECT	ID: 7	TYPE: Concrete, Domain
Description: A class that used to stores and manages the subject details.		Associated Use Cases: 3, 8, 9, 11
Responsibilities addSubject updateSubject deleteSubject	Collaborators CLASS_DETAILS	

Back side of SUBJECT class

Attributes: subCode (1…1) (String) {unique} subName (1…1) (String)
Relationships: Generalization (a – kind – of): not applicable Aggregation (has – parts): not applicable Other Associations: CLASS_DETAILS {1…*}

Figure 4.13-F7: CRC Card with Invariants of SUBJECT class

Front side of CLASS_HOUR class

Class Name: CLASS_HOUR	ID: 8	TYPE: Concrete, Domain
Description: A class that used to manage the time start/end and day of every class.		Associated Use Cases: 3, 10,

Responsibilities	Collaborators
addClassHrs updateClassHrs deleteClassHrs	CLASS_DETAILS

Back side of CLASS_HOUR class

Attributes:

classHrsID (1…1) (String) {unique}
timeStart (1…1) (date)
timeEnd (1…1) (date)
day (1…1) (String)

Relationships:

Generalization (a – kind – of): not applicable
Aggregation (has – parts): not applicable
Other Associations: CLASS_DETAILS {1…*}

Figure 4.13-F8: CRC Card with Invariants of CLASS_HOUR class

Front side of CLASS_DETAILS class

Class Name: CLASS_DETAILS	ID: 9	TYPE: Concrete, Domain
Description: A class that combined few objects information to form a schedule of a subject to be attends by students.		Associated Use Cases: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Responsibilities	Collaborators
addClassDetails	STUDENT
updateClassDetails	STU_CLASS
deleteClassDetails	LECTURER
	CLASS_HOUR
	CLASSROOM
	SUBJECT

Back side of CLASS_DETAILS class

Attributes:

classDetailsID	(1…1) (String) {unique}
lecturer	(1…*) (LECTURER)
lecturerID	(1…*) (String) {lecturerID = student.getLecturerID();}
classHrs	(1…1) (CLASS_HOUR)
classHrsID	(1…1) (String) {classHrsID = classHrs.getClassHrsID();}
classroom	(1…1) (CLASSROOM)
classroomID	(1…1) (String) {classroomID = classroom.getClassroomID();}
subject	(1…1) (SUBJECT)
subjectID	(1…1) (String) {subjectID = subject.getSubjectID();}

Relationships:

Generalization (a – kind – of): not applicable

Aggregation (has – parts): LECTURER {1…1}, CLASS_HOUR {1…1}, CLASSROOM {1…1}, SUBJECT {1…1}

Other Associations: STUDENT {1…*}, STU_CLASS {1…1}

Figure 4.13-F9: CRC Card with Invariants of CLASS_DETAILS class

4.14 Method Specification

Method Name: addNewSudent()	Class Name: STUDENT	ID: 1
Clients (Consumers): StudentController		
Associated Use Cases: Manage Student Information		
Description of Responsibilities: Add the new student record into system.		
Arguments Received: studentName (String), studentPass (String), studentTelNo (String), studentEmail (String), studentDOB (date), studentGender (String)		
Type of Value Returned: String		
Pre-conditions: The student successfully registered to study in the college/university.		
Post-conditions: The student profile is added into STUDENT table.		
Algorithm Specification: 1. Get the data passed in by StudentController. 2. Get the latest student ID from the STUDENT table 3. Create a new student ID by adding one to the latest student ID (e.g. 15ACB00001 -> 15ACB00002) 4. Insert all student profile details into STUDENT table 5. Return the result (e.g. “The Student profile is successfully created”)		

Figure 4.14-F1: addNewStudent() Method Specification

Method Name: updateSudent()	Class Name: STUDENT	ID: 2
Clients (Consumers): StudentController		
Associated Use Cases: Manage Student Information		
Description of Responsibilities: Edit the latest student details into the system if there are any changes.		
Arguments Received: studentID (String), studentName (String), studentPass (String), studentTelNo (String), studentEmail (String), studentDOB (date), studentGender (String)		
Type of Value Returned: String		
Pre-conditions: The student data already exist in the system.		
Post-conditions: The student profile is updated into STUDENT table.		
Algorithm Specification: 1. Get the data passed in by StudentController. 2. Validate entered student record from the STUDENT table. 3. If entered student record exist in the STUDENT table a. Update the latest student details into the STUDENT table. b. Return the result (e.g. “The Student profile is successfully updated”). 4. Else a. Return the result (e.g. “The Student profile doesn’t exist!”).		

Figure 4.14-F2: updateStudent() Method Specification

Method Name: deleteStudent()	Class Name: STUDENT	ID: 3
Clients (Consumers): StudentController		
Associated Use Cases: Manage Student Information		
Description of Responsibilities: Delete the student details from the system.		
Arguments Received: studentID (String), studentName (String), studentPass (String), studentTelNo (String), studentEmail (String), studentDOB (date), studentGender (String)		
Type of Value Returned: String		
Pre-conditions: The student data already exist in the system.		
Post-conditions: The student profile is deleted from the STUDENT table.		
Algorithm Specification: 1. Get the data passed in by StudentController. 2. Validate entered student record from the STUDENT table. 3. If entered student record exist in the STUDENT table a. Delete student details into the STUDENT table. b. Return the result (e.g. “The Student profile is successfully deleted”). 4. Else a. Return the result (e.g. “The Student profile doesn’t exist!”).		

Figure 4.14-F3: deleteStudent() Method Specification

Method Name: addNewLecturer()	Class Name: LECTURER	ID: 4
Clients (Consumers): LecturerController		
Associated Use Cases: Manage Lecturer Information		
Description of Responsibilities: Add the new lecturer record into system.		
Arguments Received: lecName (String), lecPass (String), lecTelNo (String), lecEmail (String), lecDOB (date), lecGender (String)		
Type of Value Returned: String		
Pre-conditions: The lecturer successfully employed to teach in the college/university.		
Post-conditions: The lecturer profile is added into LECTURER table.		
Algorithm Specification: 1. Get the data passed in by LecturerController. 2. Get the latest lecturer ID from the LECTURER table 3. Create a new lecturer ID by adding one to the latest lecturer ID (e.g. 15FICT0001 -> 15FICT0002) 4. Insert all lecturer profile details into LECTURER table 5. Return the result (e.g. “The Lecturer profile is successfully created”)		

Figure 4.14-F4: addNewLecturer() Method Specification

Method Name: updateLecturer()	Class Name: LECTURER	ID: 5
Clients (Consumers): LecturerController		
Associated Use Cases: Manage Lecturer Information		
Description of Responsibilities: Edit the latest lecturer details into the system if there are any changes.		
Arguments Received: lecturerID (String), lecName (String), lecPass (String), lecTelNo (String), lecEmail (String), lecDOB (date), lecGender (String)		
Type of Value Returned: String		
Pre-conditions: The lecturer data already exist in the system.		
Post-conditions: The lecturer profile is updated into LECTURER table.		
Algorithm Specification: 1. Get the data passed in by LecturerController. 2. Validate entered lecturer record from the LECTURER table. 3. If entered lecturer record exist in the LECTURER table a. Update the latest lecturer details into the LECTURER table. b. Return the result (e.g. “The Lecturer profile is successfully updated”). 4. Else a. Return the result (e.g. “The Lecturer profile doesn’t exist!”).		

Figure 4.14-F5: updateLecturer() Method Specification

Method Name: deleteLecturer()	Class Name: LECTURER	ID: 6
Clients (Consumers): LecturerController		
Associated Use Cases: Manage Lecturer Information		
Description of Responsibilities: Delete the lecturer details from the system.		
Arguments Received: lecturerID (String), lecName (String), lecPass (String), lecTelNo (String), lecEmail (String), lecDOB (date), lecGender (String)		
Type of Value Returned: String		
Pre-conditions: The lecturer data already exist in the system.		
Post-conditions: The lecturer profile is deleted from the LECTURER table.		
Algorithm Specification: 1. Get the data passed in by LecturerController. 2. Validate entered lecturer record from the LECTURER table. 3. If entered lecturer record exist in the LECTURER table a. Delete lecturer details from the LECTURER table. b. Return the result (e.g. “The Lecturer profile is successfully deleted”). 4. Else a. Return the result (e.g. “The Lecturer profile doesn’t exist!”).		

Figure 4.14-F6: deleteLecturer() Method Specification

Method Name: addClassHrs()	Class Name: CLASS_HOUR	ID: 7
Clients (Consumers): ClassHrsController		
Associated Use Cases: Add Class Period		
Description of Responsibilities: Add the new class period into system.		
Arguments Received: timeStart (date), timeEnd (date), day (String)		
Type of Value Returned: String		
Pre-conditions: The CLASS_DETAILS must be created first before the CLASS_HOUR.		
Post-conditions: The CLASS_HOUR profile is added into CLASS_HOUR table.		
Algorithm Specification: 1. Get the data passed in by ClassHrsController. 2. Get the latest classHrs ID from the CLASS_HOUR table 3. Create a new classHrs ID by adding one to the latest classHrs ID (e.g. CH0001 -> CH0002) 4. Insert all classHrs profile details into CLASS_HOUR table 5. Return the result (e.g. “The classHrs profile is successfully created”)		

Figure 4.14-F7: addClassHrs() Method Specification

Method Name: updateClassHrs()	Class Name: CLASS_HOUR	ID: 8
Clients (Consumers): ClassHrsController		
Associated Use Cases: Update Class Period		
Description of Responsibilities: Edit the latest classHrs details into the system if there are any changes.		
Arguments Received: classHrsID (String), timeStart (date), timeEnd (date), day (String)		
Type of Value Returned: String		
Pre-conditions: The CLASS_HOUR data already exist in the system.		
Post-conditions: The CLASS_HOUR profile is updated into CLASS_HOUR table.		
Algorithm Specification: 1. Get the data passed in by ClassHrsController. 2. Validate entered classHrs record from the CLASS_HOUR table. 3. If entered classHrs record exist in the CLASS_HOUR table a. Update classHrs details into the CLASS_HOUR table. b. Return the result (e.g. “The classHrs profile is successfully Updated”). 4. Else a. Return the result (e.g. “The classHrs profile doesn’t exist!”).		

Figure 4.14-F8: updateClassHrs() Method Specification

Method Name: deleteClassHrs()	Class Name: CLASS_HOUR	ID: 9
Clients (Consumers): ClassHrsController		
Associated Use Cases: Delete Class Period		
Description of Responsibilities: Delete the classHrs details from the system.		
Arguments Received: classHrsID (String), timeStart (date), timeEnd (date), day (String)		
Type of Value Returned: String		
Pre-conditions: The CLASS_HOUR data already exist in the system.		
Post-conditions: The CLASS_HOUR profile is deleted from CLASS_HOUR table.		
Algorithm Specification: 1. Get the data passed in by ClassHrsController. 2. Validate entered classHrs record from the CLASS_HOUR table. 3. If entered classHrs record exist in the CLASS_HOUR table a. Delete classHrs details from the CLASS_HOUR table. b. Return the result (e.g. “The classHrs profile is successfully deleted”). 4. Else a. Return the result (e.g. “The classHrs profile doesn’t exist!”).		

Figure 4.14-F9: deleteClassHrs() Method Specification

Method Name: addSubject()	CLASS NAME: SUBJECT	ID: 10
Clients (Consumers): SubjectController		
Associated Use Cases: Add Subject		
Description of Responsibilities: Add the new subject into system.		
Arguments Received: subName (String)		
Type of Value Returned: String		
Pre-conditions: The subject is allowed to teach in the college/university.		
Post-conditions: The subject profile is added into SUBJECT table.		
Algorithm Specification: 1. Get the data passed in by SubjectController. 2. Get the latest subjectID from the SUBJECT table 3. Create a new subjectID by adding one to the latest subjectID (e.g. S0001 -> S0002) 4. Insert all subject profile details into SUBJECT table 5. Return the result (e.g. “The subject profile is successfully created”)		

Figure 4.14-F10: addSubject() Method Specification

Method Name: updateSubject()	Class Name: SUBJECT	ID: 11
Clients (Consumers): SubjectController		
Associated Use Cases: Update Subject		
Description of Responsibilities: Edit the latest subject details into the system if there are any changes.		
Arguments Received: subCode (String), subName (String)		
Type of Value Returned: String		
Pre-conditions: The subject data already exist in the system.		
Post-conditions: The subject profile is updated into SUBJECT table.		
Algorithm Specification: 1. Get the data passed in by SubjectController. 2. Validate entered subject record from the SUBJECT table. 3. If entered subject record exist in the SUBJECT table a. Update the latest subject details into the SUBJECT table. b. Return the result (e.g. “The Subject profile is successfully updated”). 4. Else a. Return the result (e.g. “The Subject profile doesn’t exist!”).		

Figure 4.14-F11: updateSubject() Method Specification

Method Name: deleteSubject()	Class Name: SUBJECT	ID: 12
Clients (Consumers): SubjectController		
Associated Use Cases: Delete Subject		
Description of Responsibilities: Delete the subject details from the system.		
Arguments Received: subCode (String), subName (String)		
Type of Value Returned: String		
Pre-conditions: The subject data already exist in the system.		
Post-conditions: The subject profile is deleted from the SUBJECT table.		
Algorithm Specification: 1. Get the data passed in by SubjectController. 2. Validate entered subject record from the SUBJECT table. 3. If entered subject record exist in the SUBJECT table a. Delete subject details from the SUBJECT table. b. Return the result (e.g. “The Subject profile is successfully deleted”). 4. Else a. Return the result (e.g. “The Subject profile doesn’t exist!”).		

Figure 4.14-F12: deleteSubject() Method Specification

Method Name: updateClassroom()	Class Name: CLASSROOM	ID: 13
Clients (Consumers): ClassroomController		
Associated Use Cases: Update Classroom		
Description of Responsibilities: Edit the latest classroom details into the system if there are any changes.		
Arguments Received: classroomID (String), classroomSeat (String)		
Type of Value Returned: String		
Pre-conditions: The classroom data already exist in the system.		
Post-conditions: The classroom profile is updated into CLASSROOM table.		
Algorithm Specification: 1. Get the data passed in by ClassroomController. 2. Validate entered classroom record from the CLASSROOM table. 3. If entered classroom record exist in the CLASSROOM table a. Update the latest classroom details into the CLASSROOM table. b. Return the result (e.g. “The classroom profile is successfully updated”). 4. Else a. Return the result (e.g. “The classroom profile doesn’t exist!”).		

Figure 4.14-F13: updateClassroom() Method Specification

Method Name: checkClassroomAvailability()	Class Name: CLASSROOM	ID: 14
Clients (Consumers): ClassroomController		
Associated Use Cases: Check Classroom Availability		
Description of Responsibilities: Search classroom detail from the system.		
Arguments Received: classroomID (String), classroomSeat (String)		
Type of Value Returned: String		
Pre-conditions: The admin want to search classroom availability.		
Post-conditions: The classroom details successfully show at system.		
Algorithm Specification: 1. Get the data passed in by ClassroomController. 2. Retrieve the detail from user input. 3. Get the classroom detail from CLASSROOM table based on user input. 4. Return the result. (e.g. “classroomID : C0001 Availability: Available”)		

Figure 4.14-F14: checkClassRoomAvailability() Method Specification

Method Name: updateFingerprint()	Class Name: FINGERPRINT	ID: 15
Clients (Consumers): StudentController		
Associated Use Cases: Manage Student Information		
Description of Responsibilities: Edit the latest fingerprint record into the system if there are any changes.		
Arguments Received: studentID (String), fingerprintRec (BLOB)		
Type of Value Returned: String		
Pre-conditions: The student data already exist in the system.		
Post-conditions: The fingerprint image is updated into STUDENT table.		
Algorithm Specification: 1. Get the data passed in by StudentController. 2. Validate entered student record from the STUDENT table. 3. If entered student record exist in the STUDENT table a. Update the latest fingerprint details into the FINGERPRINT table. b. Return the result (e.g. “The fingerprint record is successfully updated”). 4. Else a. Return the result (e.g. “The fingerprint record doesn’t exist!”).		

Figure 4.14-F15: updateFingerprint() Method Specification

Method Name: searchAttendance()	Class Name: ATTENDANCE	ID: 16
Clients (Consumers): AttendanceController		
Associated Use Cases: Search Attendance		
Description of Responsibilities: Search attendance detail from the system.		
Arguments Received: attendanceID (String), attendanceStatus (String), attendanceDate (date)		
Type of Value Returned: String		
Pre-conditions: The lecturer wants to search attendance history.		
Post-conditions: The attendance details successfully show at system.		
Algorithm Specification: 1. Get the data passed in by AttendanceController. 2. Retrieve the detail from user input. 3. Get the attendance detail from ATTENDANCE table based on user input. 4. Return the result. (e.g. “attendanceID : C0001”)		

Figure 4.14-F16: searchAttendance() Method Specification

Method Name: addAttendance()	CLASS NAME: ATTENDANCE	ID: 17
Clients (Consumers): AttendanceController		
Associated Use Cases: Add Attendance		
Description of Responsibilities: Add the new attendance into system.		
Arguments Received: attendanceStatus (String), attendanceDate (date)		
Type of Value Returned: String		
Pre-conditions: The class is going on.		
Post-conditions: The attendance profile is added into ATTENDANCE table.		
Algorithm Specification: 1. Get the data passed in by AttendanceController. 2. Get the latest attendanceID from the ATTENDANCE table 3. Create a new attendanceID by adding one to the latest attendanceID (e.g. A0001 -> A0002) 4. Insert all attendance profile details into ATTENDANCE table 5. Return the result (e.g. “The attendance profile is successfully created”)		

Figure 4.14-F17: addAttendance() Method Specification

Method Name: updateAttendanceStatus()	Class Name: ATTENDANCE	ID: 18
Clients (Consumers): AttendanceController		
Associated Use Cases: Update Attendance		
Description of Responsibilities: Edit the latest attendance record into the system if there are any changes.		
Arguments Received: attendanceID (String), attendanceStatus (String), attendanceDate (date)		
Type of Value Returned: String		
Pre-conditions: The attendance data already exist in the system.		
Post-conditions: The attendance is updated into ATTENDANCE table.		
Algorithm Specification: 1. Get the data passed in by AttendanceController. 2. Validate entered attendance record from the ATTENDANCE table. 3. If entered attendance record exist in the ATTENDANCE table a. Update the latest attendance details into the ATTENDANCE table. b. Return the result (e.g. “The attendance record is successfully updated”). 4. Else a. Return the result (e.g. “The attendance record doesn’t exist!”).		

Figure 4.14-F18: updateAttendanceStatus() Method Specification

Method Name: verifyFingerprint()	Class Name: ATTENDANCE	ID: 19
Clients (Consumers): CheckInController		
Associated Use Cases: Check-in Attendance		
Description of Responsibilities: Check-in attendance into the system.		
Arguments Received: attendanceID (String), attendanceStatus (String), attendanceDate (date)		
Type of Value Returned: String		
Pre-conditions: The student wants to check-in his/her attendance.		
Post-conditions: The attendance status successfully updated into system.		
Algorithm Specification: 1. Get the data passed in by CheckInController. 2. Retrieve the detail from user fingerprint. 3. Update the attendance status into ATTENDANCE table based on user fingerprint. 4. Return the result. (e.g. “attendanceID: C0001 studentID: 15ACB00001 attendanceStatus: Attend”)		

Figure 4.14-F19: verifyFingerprint() Method Specification

Method Name: addClassDetails()	CLASS NAME: CLASS_DETAIL	ID: 20
Clients (Consumers): ClassDetailsController		
Associated Use Cases: Add Class Details		
Description of Responsibilities: Add the new class details into system.		
Arguments Received: classDetailsID (String), CLASSROOM (classroom), SUBJECT (subject), CLASS_HOUR (class_hour), LECTURER (lecturer)		
Type of Value Returned: String		
Pre-conditions: The CLASSROOM, SUBJECT, CLASS_HOUR, LECTURER table must exist.		
Post-conditions: The class details profile is added into CLASS_DETAIL table.		
Algorithm Specification: 1. Get the data passed in by ClassDetailsController. 2. Get the latest classDetailsID from the CLASS_DETAIL table. 3. Create a new classDetailsID by adding one to the latest classDetailsID (e.g. CD0001 -> CD0002) 4. Insert all class details profile into CLASS_DETAIL table 5. Return the result (e.g. “The class details profile is successfully created”)		

Figure 4.14-F20: addClassDetails() Method Specification

Method Name: updateClassDetails()	Class Name: CLASS_DETAIL	ID: 21
Clients (Consumers): ClassDetailsController		
Associated Use Cases: Update Class Details		
Description of Responsibilities: Edit the latest class details record into the system if there are any changes.		
Arguments Received: classDetailsID (String), CLASSROOM (classroom), SUBJECT (subject), CLASS_HOUR (class_hour), LECTURER (lecturer)		
Type of Value Returned: String		
Pre-conditions: The class details data already exist in the system.		
Post-conditions: The class detail is updated into CLASS_DETAIL table.		
Algorithm Specification: 1. Get the data passed in by ClassDetailController. 2. Validate entered class details record from the CLASS_DETAIL table. 3. If entered class detail record exist in the CLASS_DETAIL table a. Update the latest class details into the CLASS_DETAIL table. b. Return the result (e.g. “The class detail record is successfully updated”). 4. Else a. Return the result (e.g. “The class detail record doesn’t exist!”).		

Figure 4.14-F21: updateClassDetails() Method Specification

Method Name: deleteClassDetails()	Class Name: CLASS_DETAILS	ID: 22
Clients (Consumers): ClassDetailsController		
Associated Use Cases: Delete Class Details		
Description of Responsibilities: Delete the class details from the system.		
Arguments Received: classDetailsID (String), CLASSROOM (classroom), SUBJECT (subject), CLASS_HOUR (class_hour), LECTURER (lecturer)		
Type of Value Returned: String		
Pre-conditions: The class details data already exist in the system.		
Post-conditions: The class details profile is deleted from the CLASS_DETAIL table.		
Algorithm Specification: 1. Get the data passed in by ClassDetailsController. 2. Validate entered class details record from the CLASS_DETAIL table. 3. If entered class details record exist in the CLASS_DETAIL table a. Delete class details from the CLASS_DETAIL table. b. Return the result (e.g. “The Class Details profile is successfully deleted”). 4. Else a. Return the result (e.g. “The Class Details profile doesn’t exist!”).		

Figure 4.14-F22: deleteClassDetails() Method Specification

Method Name: viewEnrolledStudent()	Class Name: STU_CLASS	ID: 23
Clients (Consumers): StuClassController		
Associated Use Cases: Search Enrolled Student		
Description of Responsibilities: Search enrolled student detail from the system.		
Arguments Received: stuClassID (String)		
Type of Value Returned: String		
Pre-conditions: The lecturer wants to search student enrolled in the class.		
Post-conditions: The enrolled student details successfully show at system.		
Algorithm Specification: 1. Get the data passed in by StuClassController. 2. Retrieve the detail from user input. 3. Get the enrolled student detail from STU_CLASS table based on user input. 4. Return the result. (e.g. “classDetailsID: CD0001 studentID: 15ACB0002”)		

Figure 4.14-F23: viewEnrolledStudent() Method Specification

Method Name: addEnrolledStudent()	CLASS NAME: STU_CLASS	ID: 24
Clients (Consumers): StuClassController		
Associated Use Cases: Add Student to class.		
Description of Responsibilities: Add the student to enrol in class into system.		
Arguments Received: stuClassID (String)		
Type of Value Returned: String		
Pre-conditions: The student wants to register for the subject.		
Post-conditions: The enrolled student profile is added into STU_CLASS table.		
Algorithm Specification: 1. Get the data passed in by StuClassController. 2. Get the latest stuClassID from the CLASS_DETAIL table. 3. Create a new stuClassID by adding one to the latest stuClassID (e.g. SC0001 -> SC0002) 4. Insert all enrolled student details profile into STU_CLASS table 5. Return the result (e.g. “The enrolled student details profile is successfully created”)		

Figure 4.14-F24: addEnrolledStudent() Method Specification

Method Name: removeEnrolledStudent ()	Class Name: STU_CLASS	ID: 25
Clients (Consumers): StuClassController		
Associated Use Cases: Delete Enrolled Student		
Description of Responsibilities: Delete the enrolled student details from the system.		
Arguments Received: stuClassID (String)		
Type of Value Returned: String		
Pre-conditions: The enrolled student details data already exist in the system.		
Post-conditions: The enrolled student details profile is deleted from the STU_CLASS table.		
Algorithm Specification: 1. Get the data passed in by StuClassController. 2. Validate entered enrolled student details record from the STU_CLASS table. 3. If entered enrolled student details record exist in the STU_CLASS table a. Delete enrolled student details from the STU_CLASS table. b. Return the result (e.g. “The enrolled student details profile is successfully deleted”). 4. Else a. Return the result (e.g. “The enrolled student details profile doesn’t exist!”).		

Figure 4.14-F25: removeEnrolledStudent() Method Specification

4.15 Entity-Relationship Diagram

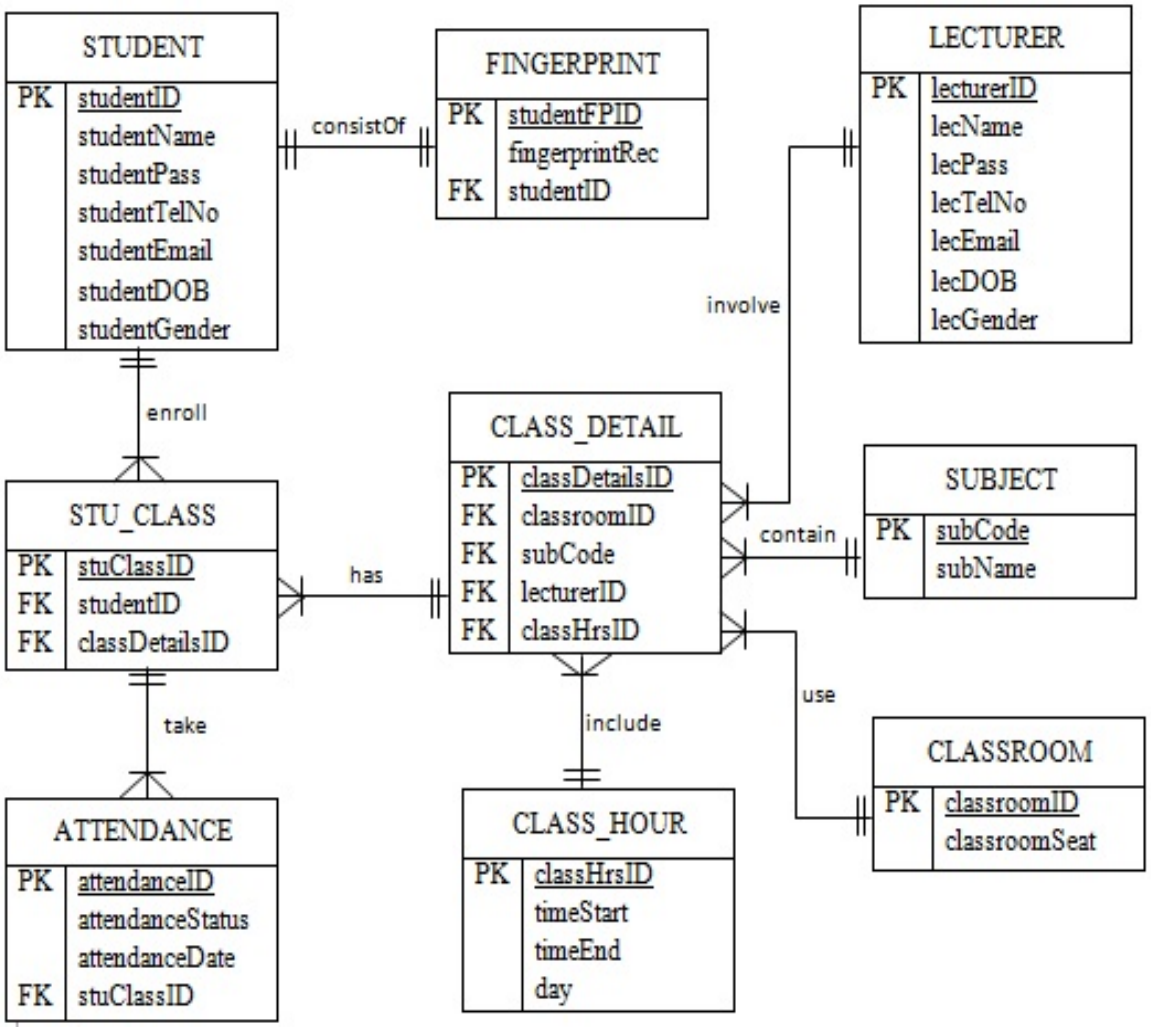

Figure 4.15-F1: Entity-Relationship Diagram

4.16 Data Dictionary

Student Entity

Entity Name	Attributes	Description	Data Type	Null
Student	studentID	Unique identifier for student	varchar(50)	No
	studentName	Name of student	varchar(50)	Yes
	studentPass	Password of student	varchar(50)	Yes
	studentTelNo	Contact no. of student	varchar(50)	Yes
	studentEmail	Email of student	varchar(50)	Yes
	studentDOB	Date of birth of student	date	Yes
	studentGender	Gender of student	varchar(50)	Yes

Table 4.16-F1: Table of Student Entity

Lecturer Entity

Entity Name	Attributes	Description	Data Type	Null
Lecturer	lecturerID	Unique identifier for lecturer	varchar(50)	No
	lecName	Name of lecturer	varchar(50)	Yes
	lecPass	Password of lecturer	varchar(50)	Yes
	lecTelNo	Contact no. of lecturer	varchar(50)	Yes
	lecEmail	Email of lecturer	varchar(50)	Yes
	lecDOB	Date of birth of lecturer	date	Yes
	lecGender	Gender of lecturer	varchar(50)	Yes

Table 4.16-F2: Table of Lecturer Entity

Subject Entity

Entity Name	Attributes	Description	Data Type	Null
Subject	subCode	Unique identifier for subject	varchar(50)	No
	subName	Name of subject	varchar(50)	Yes

Table 4.16-F3: Table of Subject Entity

Classroom Entity

Entity Name	Attributes	Description	Data Type	Null
Classroom	classroomID	Unique identifier for classroom	varchar(50)	No
	classroomSeat	Name of classroom	varchar(50)	Yes

Table 4.16-F4: Table of Classroom Entity

Fingerprint Entity

Entity Name	Attributes	Description	Data Type	Null
Fingerprint	studentFPID	Unique identifier for fingerprint	varchar(50)	No
	fingerprintRec	Templates of fingerprint	Nvarchar(MAX)	Yes
	studentID	Identifier for student	varchar(50)	No

Table 4.16-F5: Table of Fingerprint Entity

Attendance Entity

Entity Name	Attributes	Description	Data Type	Null
Attendance	attendanceID	Unique identifier for attendance	varchar(50)	No
	attendanceStatus	Status of attendance	varchar(50)	Yes
	attendanceDate	Date of attendance	date	Yes
	stuClassID	Identifier for student's class	varchar(50)	No

Table 4.16-F6: Table of Attendance Entity

Stu_Class Entity

Entity Name	Attributes	Description	Data Type	Null
Stu_Class	stuClassID	Unique identifier for student and class enroll	varchar(50)	No
	studentID	Identifier for student	varchar(50)	No
	classDetailsID	Identifier for class	varchar(50)	No

Table 4.16-F7: Table of Stu_Class Entity

Class_Hour Entity

Entity Name	Attributes	Description	Data Type	Null
Class_Hour	classHrsID	Unique identifier for class session	varchar(50)	No
	timeStart	Time start of class	time(7)	Yes
	timeEnd	Time end of class	time(7)	Yes
	day	Day of class	varchar(50)	Yes

Table 4.16-F8: Table of Class_Hour Entity

Class_Detail Entity

Entity Name	Attributes	Description	Data Type	Null
Class_Detail	classDetailsID	Unique identifier for class details	varchar(50)	No
	classroomID	Identifier for classroom	varchar(50)	Yes
	subCode	Identifier for subject	varchar(50)	Yes
	lecturerID	Identifier for lecturer	varchar(50)	Yes
	classHrsID	Identifier for class session	varchar(50)	Yes

Table 4.16-F9: Table of Class_Detail Entity

4.17 Window Navigation Diagram

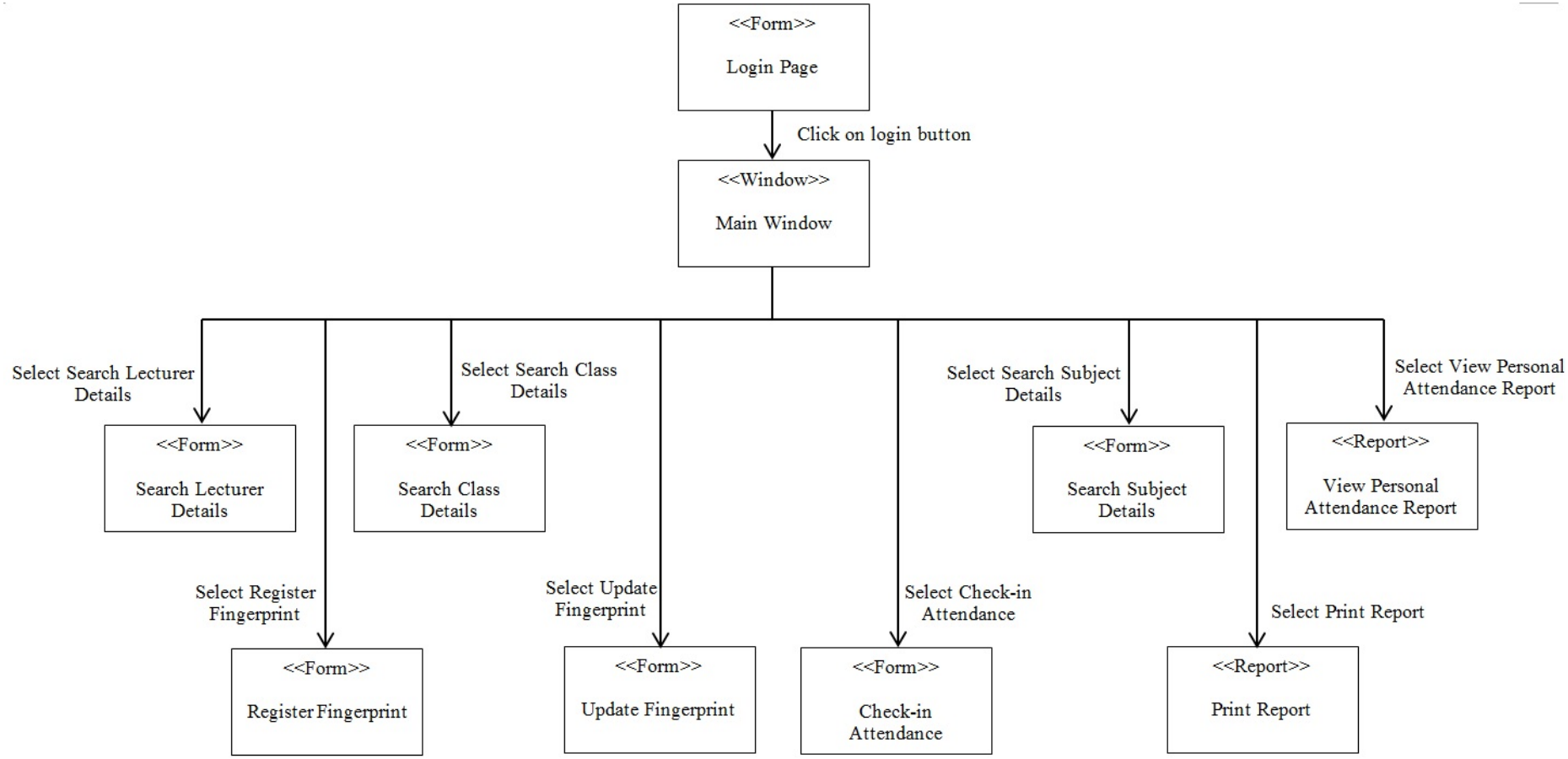

Figure 4.17-F1: Window Navigation Diagram (Student)

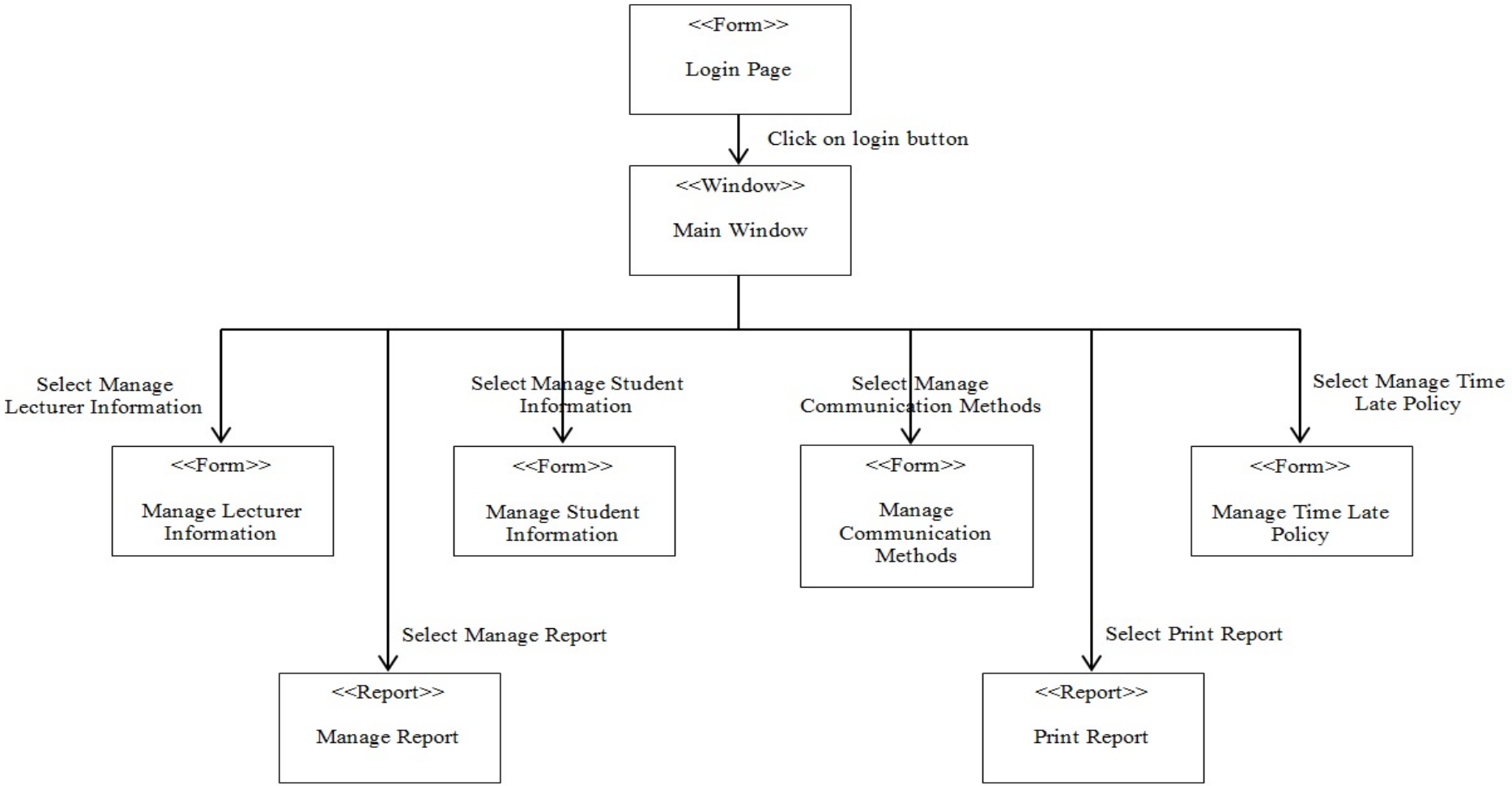

Figure 4.17-F2: Window Navigation Diagram (Admin)

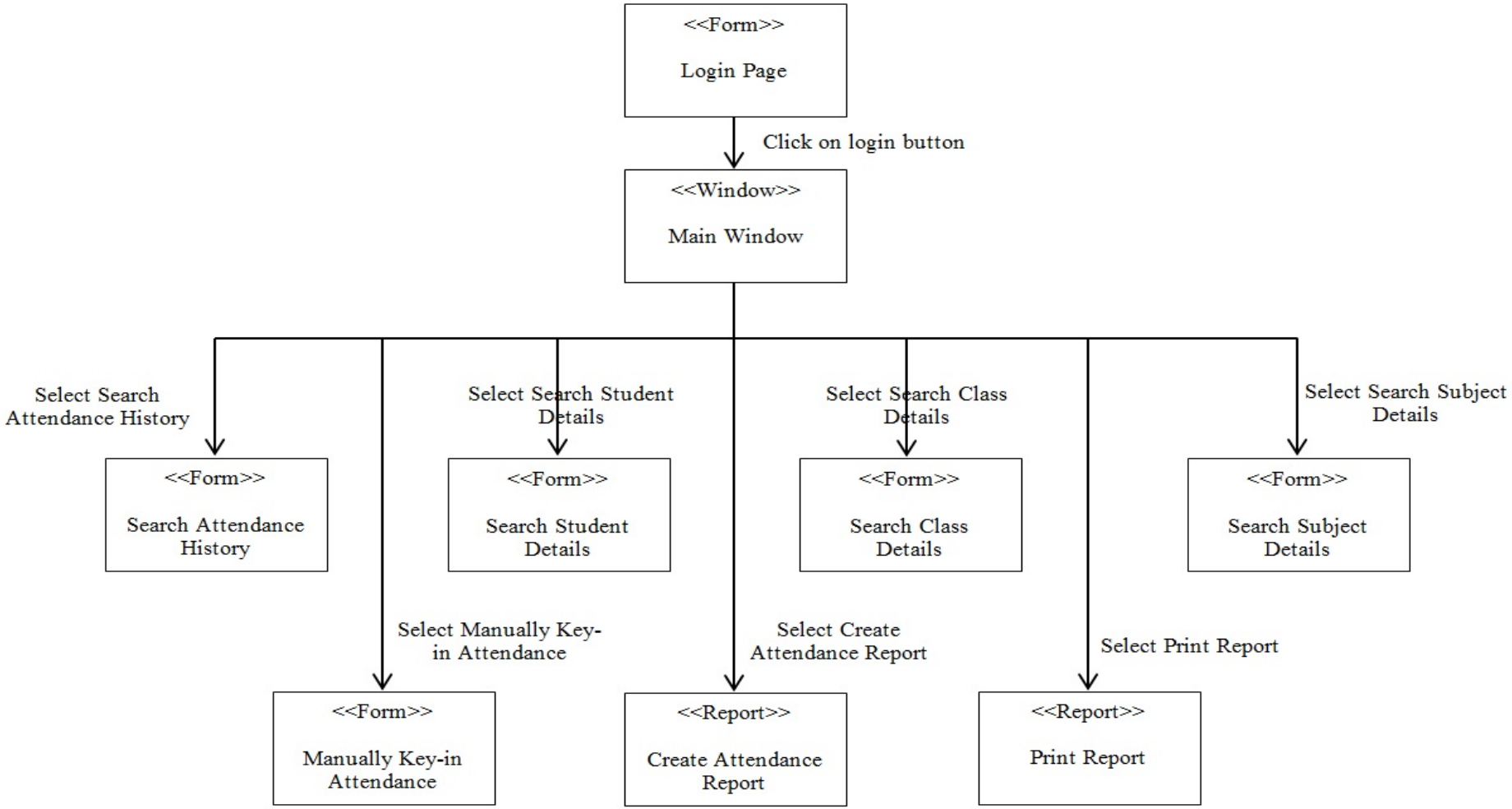

Figure 4.17-F3: Window Navigation Diagram (Lecturer)

4.18 Network Model Diagram

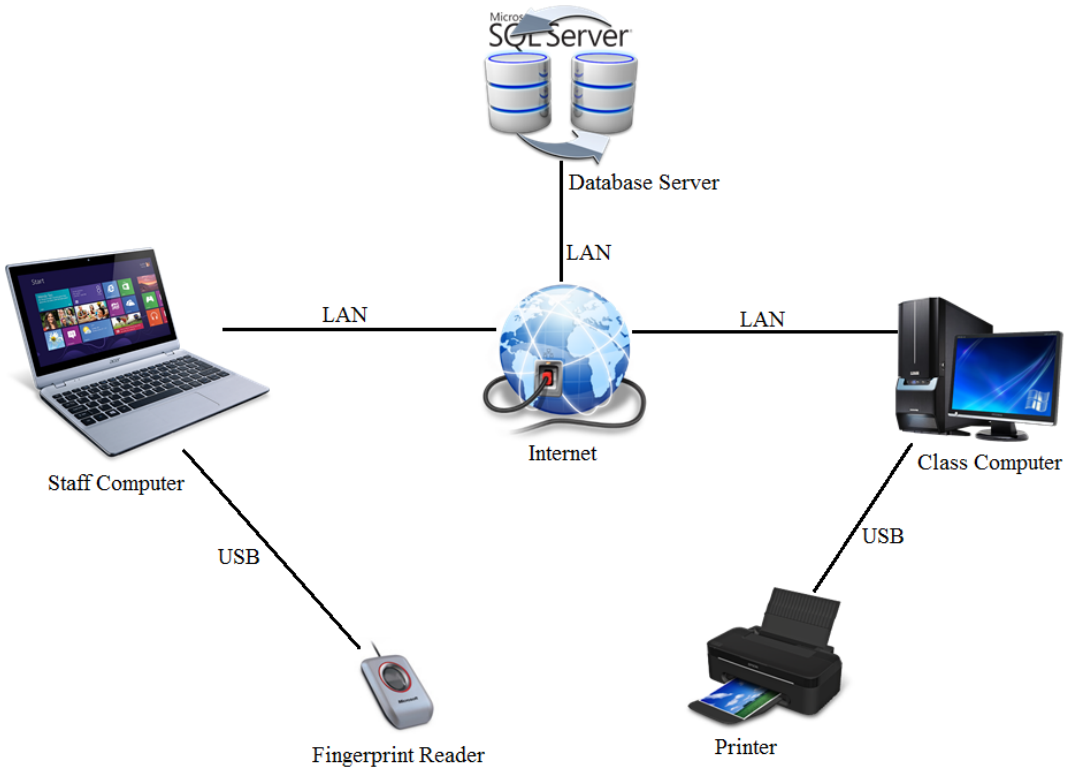

Figure 4.18-F1: Network Model Diagram

Chapter 5: System Implementation and Testing

System implementation and deployment is the part in which the system will be implemented and deployed into real life to be used. The system can only be implemented and deployed when the system design and analysis is completed.

5.1 System Implementation

Before starting to develop the attendance system, the necessarily development tools must be downloaded and installed to the devices used for system development. Therefore, Microsoft Visual Studio 2013, SQL Server 2012, Google Chrome, Flexcode SDK, and etc. was downloaded and installed for system development purpose. The process will be quite time-consuming since every development tools will need to perform the configuration needed during the installation. Although it is quite time-consuming, but this step can be seen as a very important step before the development begins.

After all the development tools installation done, the next thing to do is to configure and create table entity of the database. Based on the system design, the database will be created to be used for later data storage. By using the SQL Server 2012, the database can be easily created since developer just need to key in the queries in order to create a new table. Many features are provided by the SQL Server which ease and reduce the burden of developer in creating database.

After the creation of database, the system development will begin with creation of user interfaces using the Microsoft Visual Studio 2013 which allows us just drag and drop the control elements (button, label, and etc.) from the toolbox provided by the development tools. Visual Studio 2013 have offer a good feature for all developers which is separate the user interface code from the behind-code which make it looks more clean and easy for maintenance later during the development. VB.Net language and HTML 5.0 will be used in this project development.

Besides that, Flexcode SDK will be imported to the Visual Studio 2013 library so that it can be implemented into the system. After all the interfaces designed, the behind-code process will begin to create function on every page.

5.2 System Installation

The development tools needed in order to develop this system are includes Microsoft Visual Studio 2013, Flexcode SDK, SQL Server 2012, and etc. Microsoft Visual Studio 2013 is available for download from the official website of Microsoft. It provides two different development tools version which one is for non-commercialize development and the other one is for commercialize development. However, the features both versions provided are totally same. Therefore, it will not restrict the development of this system.

Another development tool used in the system development is the SQL Server 2012. This development tool mainly purpose is for creating the database to be integrate with the system. In addition, this development tools also can be downloaded for free from the website and without any copyright issues or payment needed. For the Flexcode SDK, the developer will need to purchase the activation code from the provider in order to implement it to the system. It is a tool for register and verifies the fingerprint and converts it into long text to be store into the database. Without the SDK, the system may not be that perfect.

Last but not least, the browser used to test and view the page which can be considered as quite important tool. The main browser used is Google Chrome which to view the page created in the Microsoft Visual Studio 2013.

5.2.1 Hardware Requirements

The minimum hardware requirement in develop this system are listed as below:

Hardware Description	Minimum Requirements
Processor	Intel Pentium D 3.4GHz / AMD Athlon II X2 250 u (Minimum) Intel Core 2 Duo E4400 2.0GHz / AMD Athlon 64 X2 Dual Core 4600+ (Recommended)
Memory	1 GB RAM Recommended, 256 MB RAM (Minimum)
Hard disk space	Up to 3 GB Recommended
Display	65536 colors, set to at least 1024 X 768 resolution

Table 5.2.1-F1: Table of Hardware Requirements

5.2.2 Software Requirements

The minimum software requirement in develop this system are listed as below:

Software Description	Minimum Requirements
Operating System (OS)	All 32-bits Microsoft Windows (95/98/2000/XP/7/8)
Browser	Mozilla Firefox (15.0 & above), Internet Explorer (8.0 & above), Google Chrome (20.0 & above).

Table 5.2.2-F1: Table of Software Requirements

5.3 System Testing

After the system developed, process of system testing must be carry on in order to test if the system is free of bugs. If during the system testing, there are bugs or errors detected, the developer may need to correct and fix the bugs immediately. There are few types of system testing that must be performed which include the unit testing, integration testing, system testing, and acceptance testing. System testing is not a testing that limited only to the development team but it also require the help from specific outsider (beta-tester) to test on the system acceptance.

First of all, unit testing is a testing which requires the developer to test on every single part or component in the system. Unit testing can be kind of time-consuming testing since the tester will need to go through every single component to make sure no bugs or errors occur before the deployment. Every single step of unit testing will be recorded to the test plan for later testing review purposes. In the unit testing, the testing only involves members from the development team which mean beta-tester is not required.

Besides that, the integration testing is a testing that must be conduct in order to test the integration between multiple pages of the system. The purpose of the integration testing is to make sure that there are no defects during the integration of multiple pages or modules. It is usually conducted after the unit testing. During the unit testing, the tester might not found any of the errors but it does not mean that the system will totally free of bugs since the system might not properly integrated which causes errors.

Other than that, the system testing is a testing that must be conduct in order to test the complete system as a whole. The purpose of system testing is to test the whole application after it is considered completed. System testing is a very important testing since it requires the system to meets the requirements and quality set by the users. Last but not least, the final testing is the acceptance testing which will involve the outsider to test the system in order to find out if the system meets their requirements from all perspectives. Once the system successfully goes through all the testing, the system will more likely to be delivered to the real world for use.

5.3.1 Unit Testing

Unit Testing 1: Login as Users (Admin, Lecturer, Student)

Testing Objective: To make sure the login process functioning well.

No	Test Case	Attribute and Value	Expected Result	Result
1	Verify the login ID and password entered by users after click on "Login" button with correct data provided.	User ID: 15L00002 Password: abc123	Login successfully.	Pass
2	Verify the login ID and password entered by users after click on "Login" button with null value.	User ID: Password:	Required Validator validation shown and require users to key-in ID and password.	Pass
3	Verify the login ID and password entered by users after click on "Login" button with invalid data provided.	User ID: 15L02 Password: 123	Login failed. Error message prompt out showing invalid login.	Pass

Table 5.3.1-F1: Table of Login as Users (Admin, Lecturer, Student)

Unit Testing 2: User Personal Profile.

Testing Objective: To make sure data successfully read from database and display to web control. (Admin, Lecturer, Student)

No	Test Case	Attribute and Value	Expected Result	Result
1	Display data to web control based on query string / session.	Query string: lectID Session("lectID")	Personal details successfully displayed on web control.	Pass
2	Direct users to login page when no query string / session found.	Navigate to specific pages without login.	Redirect back to login page.	Pass
3	Retrieve correct details of the users based on query string / session.	Query string: lectID Session("lectID")	Successfully display details of specific users based on query string / session.	Pass

Table 5.3.1-F2: Table of User Personal Profile (Admin, Lecturer, Student)

Unit Testing 3: Edit Personal Profile. (Admin, Lecturer, Student)

Testing Objective: To make sure users successfully edit their personal details. (Admin, Lecturer, Student)

No	Test Case	Attribute and Value	Expected Result	Result
1	Edit all personal details with correct data provided.	Example: Phone No: 0161564817	Personal details successfully edited and saved to database.	Pass
2	Edit all personal details with invalid data provided.	Example: Phone No: 016abc4817	Personal details amendment failed and validation message shown.	Pass
3	Edit all personal details with null data provided.	Example: Phone No:	Personal details amendment failed and required field validation message shown.	Pass

Table 5.3.1-F3: Table of Edit Personal Profile (Admin, Lecturer, Student)

Unit Testing 4: Create User Profile. (Admin)

Testing Objective: To create user profile for lecturer and student.

No	Test Case	Attribute and Value	Expected Result	Result
1	Create lecturer profile by insert their email.	Example: Lecturer ID: 15S00002 Email: p@utar.my	Email contains loginID and temporary password sent to lecturer email.	Pass
2	Create student profile by insert their email.	Example: Lecturer ID: 15S00002 Email: p@utar.my	Email contains loginID and temporary password sent to student email.	Pass
3	Email successfully sent to the lecturer or student based on the email inserted.	User ID: 15S00002 Temp Pass: abc123 Mail Subject: UserID & Temp Pass	Users receive email upon the creation of their personal profiles.	Pass

Table 5.3.1-F4: Table of Create User Profile (Admin)

Unit Testing 5: Create New Classroom. (Admin)

Testing Objective: To create new classroom details.

No	Test Case	Attribute and Value	Expected Result	Result
1	Create new classroom by insert the capacity.	Class ID: C00001 Number of Seats: 60	Classroom successfully created and stored into database.	Pass
2	Classroom ID is auto-generated based on last number stored in the database.	Previous ID: C00001 Current ID: C00002	Classroom ID automatically generated without duplication.	Pass
3	Auto-refresh the page once classroom created to make sure ID of the classroom refreshed.	Auto-refresh after classroom creation.	Page auto refresh after every new classroom created.	Pass

Table 5.3.1-F5: Table of Create New Classroom (Admin)

Unit Testing 6: Create Student and Class Enrollment. (Admin)

Testing Objective: To enroll student to specific class session, subject, and class type.

No	Test Case	Attribute and Value	Expected Result	Result
1	Enroll student to specific class with lecturer, class session, class type, and etc.	ClassID: CD0001 LecturerID: 15L00002 Class Type: Lecture Time Start: 09:00:00 Time End: 11:00:00	Student successfully enrolled into a specific class.	Pass
2	Allow admin to view the details of student on grid view click.	Details fill to web control on row click.	Details of specific student displayed on web control based on the row clicked.	Pass
3	Auto-refresh the page once classroom created to make sure ID of the class enroll refreshed.	Auto-refresh after class enrolls creation.	Page auto refresh after every new class enroll created.	Pass
4	All the web control such as dropdownlist and gridview bind with correct data read from database.	ddlSubject: Subject Name ddlLecturer: Lecturer Name gridviewStudent: Stud Lists	All the web control display the proper data based on the query statement assigned to them.	Pass

Table 5.3.1-F6: Table of Create Student and Class Enrollment (Admin)

Unit Testing 7: Past class details together with student details. (Lecturer)

Testing Objective: To view the details of past class together with student enrolled in the class.

No	Test Case	Attribute and Value	Expected Result	Result
1	Retrieve data of past class from database based on the dropdownlist value.	ddlSubject: Subject Name ddlClassType: Tutorial ddlDate: 12/08/2015	Successfully display the wanted details to gridview based on dropdownlist selection.	Pass
2	Classify the details of attended student and absent student.	Student Attendance Status: Attended Student Attendance Status: Absent	Successfully classify the student attendance based on their attendance status.	Pass
3	Allow lecturer to edit specific student attendance status.	Change absent status of specific student to attended.	Successfully change the attendance status of specific student.	Pass

Table 5.3.1-F7: Table of Past Class Details with Student Details (Lecturer)

Unit Testing 8: View The Bar List. (Lecturer)

Testing Objective: To view bar list of specific class that belongs to the lecturer.

No	Test Case	Attribute and Value	Expected Result	Result
1	Auto-calculate the student attendance rate and display to lecturer on gridview.	ddlSubject: Subject Name ddlClassType: Tutorial ddlDate: 12/08/2015 ddlTime : 09:00:00	Successfully calculate the attendance rate of all students in a class.	Pass
2	View the details of specific student on gridview row click.	Click on gridview to view student details.	Details of specific student displayed on web control.	Pass
3	Classify the student who eligible for exam and who ineligible for exam.	EligibleForExam: studentID IneligibleForExam: studentID	Display studentID to specific gridview based on their attendance rate.	Pass

Table 5.3.1-F8: Table of View the Bar List (Lecturer)

Unit Testing 9: Generate Bar List. (Lecturer)

Testing Objective: To generate bar list report of specific class.

No	Test Case	Attribute and Value	Expected Result	Result
1	Generate bar list based on specific class that belong to the lecturer after week 12 onwards.	Click on generate bar list button.	Generate bar list report with necessarily details.	Pass
2	Bar list only able to generate after week 12 onwards.	>= 12	Bar list is unable to generate before week 12.	Pass
3	Update student to bar list after week 12.	Week 12 -> Week 13	Able to update barred student to report that previously not get barred.	Pass

Table 5.3.1-F9: Table of Generate Bar List (Lecturer)

Unit Testing 10: View Barred Class Lists. (Student)

Testing Objective: To inform student if they are barred from the specific class.

No	Test Case	Attribute and Value	Expected Result	Result
1	View which class has barred the student from taking exam.	studentID: 15L00001 Barred from class: subjectName	Display classes that bar that student from taking exam to gridview.	Pass

Table 5.3.1-F10: Table of View Barred Class Lists (Student)

Unit Testing 11: Record Student Fingerprint Templates. (Student)

Testing Objective: To record student fingerprint templates into database.

No	Test Case	Attribute and Value	Expected Result	Result
1	Record the student fingerprint templates into database for attendance verification purpose.	fingerprintTemplates: Longtext – nvarchar(MAX)	Successfully store the student fingerprint templates into database.	Pass

Table 5.3.1-F11: Table of Record Student Fingerprint Templates (Student)

Unit Testing 12: Take Attendance. (Lecturer)

Testing Objective: To take attendance of student based on specific class.

No	Test Case	Attribute and Value	Expected Result	Result
1	Take attendance of student by matching their fingerprint.	fingerprintTemplates: Longtext – nvarchar(MAX)	Successfully match the student fingerprint from database based on that class.	Pass
2	Send email to student who attended the class on fingerprint matching.	Email: Attendance taken for "studentID"	Send email to those who successfully verify their attendance on fingerprint matching.	Pass
3	All students who late for 15 minutes will automatically considered as late.	Countdown timer: 900s	Student who came to class after 15 minutes will be considered as Absent.	Pass
4	Attendance taken session will be ended if the class attendance full.	Total student = Total attended	Attendance taken session will be ended once the attendance is full.	Pass
5	Those student who had been verified as attended will displayed on gridview for them to view.	Display on gridview.	Student ID of those who verified as attended will displayed on gridview automatically.	Pass

Table 5.3.1-F12: Table of Take Attendance (Lecturer)

Unit Testing 13: Reset Password. (Admin, Lecturer, Student)

Testing Objective: To reset password for those who forgot their login password.

No	Test Case	Attribute and Value	Expected Result	Result
1	Reset password for those who forgot their password.	passwordReset	Reset password successfully.	Pass

Table 5.3.1-F13: Table of Reset Password (Admin, Lecturer, Student)

5.3.2 Functional Testing

Functional Testing 1: Login based on Different Roles (Admin, Lecturer, Student)

Testing Objective: To make sure the login page redirect user to the correct page based on their roles.

No	Test Case	Attribute and Value	Expected Result	Result
1	Verify login user as admin role.	User ID: AD0001 Password: abc123	Redirect login user to the admin page.	Pass
2	Verify login user as lecturer role.	User ID: 15L00002 Password: abc123	Redirect user to the lecturer page.	Pass
3	Verify login user as student role.	User ID: 15S00001 Password: abc123	Redirect user to the student page.	Pass

Table 5.3.2-F1: Table of Login based on Different Roles (Admin, Lecturer, Student)

Functional Testing 2: View and Edit Personal Profile (Admin, Lecturer, Student)

Testing Objective: To make sure users able to view and edit their own personal details.

No	Test Case	Attribute and Value	Expected Result	Result
1	Retrieve own details from database and display to web control.	Textbox Name = "Kenya West"	Display details of login user correctly to the web control.	Pass
2	Click on Edit Profile button to redirect to edit profile page.	Click "Edit Profile" button	Redirect user to the edit profile page.	Pass
3	Click save button to save the personal details that had been amended.	Click "Save" button	Redirect user back to the view personal details page.	Pass
4	Click Change Password button to redirect to change password page.	Click "Change Password" button.	Redirect user to change password page.	Pass
5	Check on validation of the input details.	Check input validation.	Restrict users to simply enter wrong format details.	Pass

Table 5.3.2-F2: Table of View and Edit Personal Profile (Admin, Lecturer, Student)

Functional Testing 3: Create New User Profile, New Classroom, New Subject, and Class Enrollment (Admin)

Testing Objective: To create new user profile, new classroom, new subject, and class and student enrollment.

No	Test Case	Attribute and Value	Expected Result	Result
1	Generate new ID automatically for new user, new classroom, new subject, and class enrollment.	ID: new ID	Automatically generate ID based on last number read from database.	Pass
2	Create profile for new lecturer by insert their email.	Email: email address	Create new lecturer profile on button click provided.	Pass
3	Create profile for new student by insert their email.	Email: email address	Create new student profile on button click provided.	Pass
4	Create new classroom by provide number of seat for the classroom.	Number of Seat: 40	Create new classroom profile on button click provided.	Pass
5	Create new subject by provide the subject name.	Subject Name: subName	Create new subject profile on button click provided.	Pass
6	Create new class enrollment and enroll student into the class.	Class Enrollment with student.	Enroll student into specific subject.	Pass
7	Make amendment to the newly created profile for purpose to re-correct wrongly created profile.	Correction on wrongly created profile.	Successfully make correction on wrongly created profile.	Pass
8	View history of created profile.	View Gridview	View back the created profile details.	Pass

Table 5.3.2-F3: Table of Create New User Profile, New Classroom, New Subject, and Class Enrollment (Admin, Lecturer, Student)

Functional Testing 4: Record Fingerprint (Admin)

Testing Objective: To record the fingerprint of those who not yet registered.

No	Test Case	Attribute and Value	Expected Result	Result
1	Show the student lists who have not register their fingerprint.	FP Record: Not Found.	Display a list of student who have not register their fingerprint to gridview.	Pass
2	Click on register button to start register fingerprint of specific student.	Click “Register” button	Show successful message upon registration success.	Pass
3	Single click on gridview to view details of clicked student.	Single click	Show details of student of the selected row.	Pass
4	Double click on gridview to open the register fingerprint form.	Double click.	Open the register fingerprint windows form.	Pass
5	Remove student from the list once student successfully registered his fingerprint templates.	Remove gridview.	Remove the student id from the gridview list upon fingerprint registration succeed.	Pass

Table 5.3.2-F4: Table of Record Fingerprint (Admin)

Functional Testing 5: Manage Class Attendance (Lecturer)

Testing Objective: To take attendance of student during class.

No	Test Case	Attribute and Value	Expected Result	Result
1	Open the list of class on that day that belongs to the lecturer.	Class List	Redirect to take attendance page based on the class that class lecturer click.	Pass
2	Click on start button to start the timer and allow the students to start take attendance by matching their fingerprint.	Click “Start” button	Verification of student attendance will started upon the start button clicked.	Pass

3	Display the timer showing the time left for student to take their attendance.	Timer	Show countdown timer showing how many time left for student to take attendance to prevent student lateness.	Pass
4	Show the total number of student who attended the class.	Verify attendance	Show the total number of student who attended the class on a textbox.	Pass
5	Email student upon attendance verified.	Email: email address	Email student automatically once his/her attendance verified by the system.	Pass

Table 5.3.2-F5: Table of Manage Class Attendance (Lecturer)

Functional Testing 6: View and Generate Bar List Report (Lecturer)

Testing Objective: To view and generate bar list record for specific class.

No	Test Case	Attribute and Value	Expected Result	Result
1	Retrieve specific class attendance record by provide necessarily information through dropdownlist.	Dropdownliost selection.	Display gridview based on the dropdownlist result provided by users.	Pass
2	Click on generate button to generate the bar list report of specific class.	Click “Generate” button	Generate bar list report for that specific class.	Pass
3	Calculate the attendance rate automatically from the data retrieved from database.	Perform calculation	Accurately calculate the attendance rate of all student enroll in specific class.	Pass
4	Email those student who barred or ineligible for taking the exam.	Click “Email” button	Email student to inform him/her had been barred from taking exam for specific subject.	Pass

Table 5.3.2-F6: Table of View and Generate Bar List Report (Lecturer)

5.4 Future Work

The Fingerprint Recognition Student Attendance Management System is only developed for the use of single faculty. In future, it is assumed that this system will be enhanced to be used by all faculties in a university, school, or college. Other than that, the system developed is more focus on admin role as well as lecturer role which result in fewer features provided to the student role. Student only allowed viewing if they had been barred from certain class. Therefore, student has very limited feature to use in this system.

In future work, student should be able to appeal through the system directly without needing to go to find that lecturer who barred them. Other than that, the report generated will only be in the PDF format. There is no other available option for lecturer to generate the report. Besides that, the system developed only can be used on desktop or laptop but smartphone is not recommended for this system. Therefore, in future work, this system should focus more on smartphone development to ease the attendance process.

Last but not least, Fingerprint Recognition Student Attendance Management System will still have a lot to improve in order to meet every roles requirement. However, current version is good enough to be implemented to the real life to be used.

Chapter 6: Conclusion

In conclusion, fingerprint recognition attendance system will be developed to replace the traditional attendance system that are currently widely using by many colleges and universities. This project will be considered succeed once hybrid student attendance is developed. This system is designed to make the whole attendance taking process to become more reliable, convenient, efficient, and accurate. Besides that, with the implementation of biometric technology will help in reduce errors and attendance data will be able to compile in easier way.

This project is designed to aim in eliminating spotted problems during the initial analysis. The problems spotted are includes buddy-signing, loss of attendance sheet, skip class issue, and hard in analyzing student attendance record from time-to-time. These problems are the major problems faced by most colleges and universities. If view from the Pareto analysis side which also known as 80-20 rule, 80 percent of the problems are always caused by the 20 percent problems. In short, it means that most of the problems faced are mostly because of the usage of traditional student attendance system.

Therefore, this project is designed in effort to eliminate these problems. Some solution had been applied to eliminate these problems which includes the use of biometric technology, change the current system to fully-computerized system, provide easier way to generate report, and student lateness policy to eliminate “last-minute come in take attendance” kind of student. With the proposed solutions, obviously seen not only can eliminate these spotted problems but at the same time also promote a very reliable ways in managing the student attendance record.

On the other hand, from the survey questionnaire data, most of the participants are agree that student attendance in a class to increase their knowledge is very important. Last but not least, college/university with good academy students is also very important as it will affect that college/university reputation.

References

1. Information or article from a website with authors:

- ✓ Ivanilson, F. et al. (2012*). Developing an automatic attendance register system for CPUT.* Available at: <https://www.academia.edu/4079035/Developing_an_automatic_attendance_register_system> [Accessed 10 July 2014].

- ✓ Subramaniam, H. et al. (2013). *Bar Code Scanner Based Student Attendance System (SAS).* Available at: <http://www.researchgate.net/publication/245025631_Bar_Code_Scanner_Based_Student_Attendance_System_(SAS)> [Accessed 11 July 2014].

- ✓ Marvia, D. (2011, June 5). *Review of Literature for Improving Attendance in Secondary Schools.* Available at: <http://intuitiveedesign.wordpress.com/2011/06/05/review-of-literature-for-improving-attendance-in-secondary-schools/> [Accessed 13 July 2014].

- ✓ Hasnieza, M. (2005). *Monitoring Attendance School through SMS.* Available at: <http://library.utem.edu.my/index2.php?option=com_docman&task=doc_view&gid=3756&Itemid=113> [Accessed 16 July 2014].

- ✓ Pallavi, V. et al. (2013, October 10). *Fingerprint Based Student Attendance System Using GSM.* Available at: <http://www.ijsr.net/archive/v2i10/MDkxMDEzMTE=.pdf> [Accessed 16 July 2014].

- ✓ Rishabh, M. et al. (2011, May 9). *Student Attendance System Based on Fingerprint Recognition and One-to-Many Matching.* Available at: <http://ethesis.nitrkl.ac.in/2214/1/thesis.pdf> [Accessed 17 July 2014].

- ✓ Norshidah, K. et al. (2010, April 14). *Development of Attendance System using Biometric Fingerprint Identification.* Available at:

<http://eprints.uthm.edu.my/3297/1/12_Norshidah_Katiran__19Feb2010_.pdf> [Accessed 18 July 2014].

- ✓ Mohd Zamzury, A. (2008, April). *Attendance Management System using Fingerprint Scanner.* Available at: <http://library.utem.edu.my/index2.php?option=com_docman&task=doc_view&gid=5283&Itemid=113> [Accessed 21 July 2014].

- ✓ Abhishek, Jha. (2011). *Classroom Attendance System Using Facial Recognition System.* Available at: <http://klresearch.org/IJMSTM/papers/v2i3_2.pdf> [Accessed 22 July 2014].

- ✓ Mohammad, A. et al. (2013, December). *Integrated System for Monitoring and Recognizing Students during Class Session.* Available at: <http://airccse.org/journal/jma/5613ijma04.pdf> [Accessed 23 July 2014].

- ✓ Broshenka, D. et al. (2012). *Methods of Fact-finding*. Available at: <http://www.fao.org/docrep/q1085e/q1085e07.htm#2.1 written sources> [Accessed 4 Feb 2015].

- ✓ Nanmal, C. (2013). *What are The Advantages and Disadvantages of the Evolutionary Prototyping Technique?* Available at: <http://www.answers.com/Q/What_are_the_advantages_and_disadvantages_of_the_evolutionary_prototyping_technique> [Accessed 15 Feb 2015].

- ✓ Shin, L. S. (2013, Jan). *Online Final Year Project System for FICT, UTAR using AMP technologies.* Available at: <http://eprints.utar.edu.my/950/1/IB-2013-0903525-1.pdf> [Accessed 27 Feb 2015].

- ✓ LaTonya P. (2013, July). *The Four Levels of Software Testing.* Available at: < http://www.seguetech.com/blog/2013/07/31/four-levels-software-testing> [Accessed 15 Aug 2015].

2. Information or article from a website without author:

- ✓ Microtonics Technologies. (2013, June 4). *RFID Based Attendance Management System.* Available at: <http://www.projectsof8051.com/rfid-based-attendance-management-system/> [Accessed 11 July 2014].

- ✓ University of Strathclyde. (2013). *Advantages and Disadvantages of Observation*. Available at: <http://www.strath.ac.uk/aer/materials/3datacollection/unit5/advantagesanddisadvantagesofobservation/> [Accessed 2 Feb 2015].

- ✓ HN Computing. (2007). *Evolutionary Prototyping.* Available at: <http://www.sqa.org.uk/e-learning/IMAuthoring01CD/page_10.htm> [Accessed 17 Feb 2015].

Appendices

APPENDICES

User Login ID and Password (For Testing Purpose)

i. Admin Role

User ID : AD0001
Password : admina

ii. Lecturer Role

User ID : 15L00002
Password : abc123

iii. Student Role

User ID : 15S00010
Password : abc123

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 2**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

Review back the document that had been created during the final year project one to refresh the mind of what had to be done and achieve at the end of the final year project two.

2. WORK TO BE DONE

Review the mistake that accidentally made in fyp one and fix the mistake. Start doing the final year project two documentation.

3. PROBLEMS ENCOUNTERED

Mistake that been concede during final year project cause some parts of the document critically need changes.

4. SELF EVALUATION OF THE PROGRESS

An early start of the final year project which can help in finish the job and not a last minute job.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 3**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

Half part of the documentation had been done and corrected and still working on it for the diagram parts.

2. WORK TO BE DONE

Finish the diagrams that need to be included in the final documentation by this week.

3. PROBLEMS ENCOUNTERED

Expiration of drawing software due to the trial version. Need to re-download, uninstall and reinstall again the software in order to continue the work.

4. SELF EVALUATION OF THE PROGRESS

Still on schedule, all progress still able to catching up.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 4**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

Complete the whole documentation except for the test case and due to the program incompletion.

2. WORK TO BE DONE

Create user interface of the program completely before start the coding section.

3. PROBLEMS ENCOUNTERED

Unable to complete the test case since program not yet developed.

4. SELF EVALUATION OF THE PROGRESS

Final year documentation is completed except for the test case parts which show the progress still on track.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 6**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

All required user interface had been designed and created.

2. WORK TO BE DONE

Start the code behind section in order to create the function for the program.

3. PROBLEMS ENCOUNTERED

Difficulties in designing the user interface due to weak design skills.

4. SELF EVALUATION OF THE PROGRESS

All user interfaces which include three roles had been successfully created and the following days will be focus on the code behind of the program.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 8**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

Half of the code behind of the program had already done and tested with bug free.

2. WORK TO BE DONE

Finish the rest of the code behind of the program and show it to the supervisor for feedback.

3. PROBLEMS ENCOUNTERED

Some modules are quite time consuming due to it complexities but still be able to completed on time.

4. SELF EVALUATION OF THE PROGRESS

Half of the program had been completed which show a very good progress with bug free.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 10**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

Complete parts of the test case and provide user manual guide in the documentation in order to guide the users.

2. WORK TO BE DONE

Complete the full documentation of the final year project report.

3. PROBLEMS ENCOUNTERED

Coding parts had been stopped awhile due to other subject's midterms and tests.

4. SELF EVALUATION OF THE PROGRESS

There are still 4 more weeks to go for the VIVA presentation.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: 1 , 3	**Study week no.: 12**
Student Name & ID: Liew Ken Nam 1304584	
Supervisor: Mr. Sohail Safdar	
Project Title: Biometric Fingerprint – Student Attendance Management System	

1. WORK DONE

80% percent of the behind code had been completed which mean only 20 % of the code left to go.

2. WORK TO BE DONE

Complete the rest of the program code before the VIVA presentation.

3. PROBLEMS ENCOUNTERED

Not much time left for coding parts but still be able to complete the code before the VIVA presentation.

4. SELF EVALUATION OF THE PROGRESS

80% of the project completed and only 20% not yet completed.

Supervisor's signature

Student's signature

Originality Report

8/31/2015 Turnitin Originality Report

Turnitin Originality Report

FYP2 Final Report by liew ken nam

From FYP May 2015 (FYP2015)

Processed on 28-Aug-2015 01:16 MYT
ID: 564108301
Word Count: 24569

Similarity Index	Similarity by Source	
20%	Internet Sources:	4%
	Publications:	0%
	Student Papers:	20%

sources:

1. 16% match (student papers from 02-Mar-2015)
   Submitted to Universiti Tunku Abdul Rahman on 2015-03-02

2. 1% match (student papers from 25-Aug-2015)
   Submitted to Universiti Tunku Abdul Rahman on 2015-08-25

3. < 1% match (student papers from 31-Jul-2014)
   Submitted to Universiti Tunku Abdul Rahman on 2014-07-31

4. < 1% match (Internet from 13-Dec-2014)
   http://eprints.utar.edu.my/1178/1/IA-2013-1100590.pdf

5. < 1% match (student papers from 22-Jul-2014)
   Submitted to Universiti Tunku Abdul Rahman on 2014-07-22

6. < 1% match (Internet from 25-Feb-2011)
   http://www.wiha-design.nl/download%20patronen/845001%20Zeeschip.phd