FTMK INTERNSHIP LOG BOOK SYSTEM

CHAN WILSON

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

JUDUL: FTMK INTERNSIP LOG BOOK SYSTEM

SESI PENGAJIAN: 2016

Saya CHAN WILSON

Mengaku membenarkan tesis (PSM/ Sarjana/ Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat syarat kegunaan seperti berikut:

1. Tesis dan project adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat Salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat Salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_____ SULIT (Mengandungi maklumat yang berdarjah Keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

_____ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/ badan di mana penyelidikan dijalankan)

/ TIDAK TERHAD

(TANDATANGAN PENULIS)
Alamat tetap: B1-08-07, Taman Nibong Indah, 11900, Bayan Lepas, P. Pinang
Tarikh: 26/8/2016

(TANDATANGAN PENYELIA)

CATATAN:* Tesis dimaksudkan sebagai Laporan Akhir Project Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa

FTMK INTERNSHIP LOG BOOK SYSTEM

CHAN WILSON

This report is submitted in partial fulfillment of the requirements for the
Bachelor of Computer Science (Database Management)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2016

DECLARATION

I hereby declare that this project report entitled

FTMK INTERNSHIP LOG BOOK SYSTEM

Is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : _____ DATE: 26/8/2016

(CHAN WILSON)

SUPERVISOR: _____ DATE: 26/8/2016

(PN. MAS AINA)

ACKNOWLEDGEMENTS

I would like to dedicate my gratitude to my warmest family who gave me supports and encouragements along the way in completing my Project Sarjana Muda. They have inspired and encouraged me to do my best in completing this project.

I wish to express my since gratitude to my supervisor, Pn. Mas Aina, for her guidance during the process of developing this project, FTMK Internship Log Book System. She has helped and guided me inside out of how to build a system and this has encouraged and inspired me with ideas to be implemented into the system.

Last but not least, thanks to the Faculty of Information and Communication Technology as well as Universiti Teknikal Malaysia Melaka for all the materials and services that has been provided in completing my Project Sarjana Muda.

ABSTRACT

FTMK Internship Log Book System is a web-based system that helps students, faculty advisors and industry advisors to manage their daily log records during internship period, such as recording daily logs and commenting log records. Currently, Fakulti Teknologi Maklumat dan Komunikasi (FTMK) is using paper-based with physical book for managing log book system. It would be tedious and inconvenient for students and advisors as managing the log records require the presence of physical log book or by sending photos of the records which may causes extra work load for advisors to comprehend the contents of log records. Thus, FTMK Internship Log Book System is developed for replacing this inefficient system to one that is comfortable and easy to use system. The system is developed based on the unstructured interviewed session and observation of the log book. Thus, functional and non-functional requirements are constructed. Therefore, FTMK Internship Log Book System will be developed to improve the management of current log book system of FTMK in a more systematically and evolve the management of information from manual works to computerized system in order to solve the current problems those are occurring in the management.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF QUERIES

CHAPTER I

INTRODUCTION

1.1 Project Background

Writing log books during internship session has always been a part of the evaluation process for students by their respective faculty advisors and being checked by industry advisors. It is a compulsory process for students to act their part and record their daily activities, training, and experiences during the whole internship session. After the recording them, the records will turn into proofs of students for having the internship at the company.

Fakulti Teknologi Maklumat dan Komunikasi (FTMK) Internship Log Book System is a project developed primarily for recording students' daily logs during the entire internship session via a web browser. To this day, FTMK is still practicing hand written method to record logs manually on log books. This impractical and tedious process is making students hard to write them every day as it would rather too cumbersome to write

with a pen. This method would bring more varied issues such as illegible handwriting by students, who may or may not procrastinate and write the whole session within a few hours, are giving hard to for advisors to comprehend the content. Besides that, checking the log books is a time consuming for advisors as they require the look at the physical book and to some extent, students send pictures to their advisors so that they can read and verify them. There are times when the log books are not complete as students may forget to write it after a hard and long working hour in the office. As a result, it is very time-consuming for students and both faculty and industry advisors to carry out their tasks.

Thus, FTMK Internship Log Book System is proposed to help and reduce all these varied factors so that writing log books would turn into a very simple task for students and checking them by advisors would be easier. This system facilitates to access and manage students' daily logs. Students record their daily logs and, optionally, to upload a picture as an attachment. If students are having medical leaves, they are required to submit a photo of the medical leave letters and advisors are able to verify them remotely. Both parties of advisors will be able to check and review students' logs and comment at designated text area respectively. After that, advisors are allowed to change or modify their comments. Plus, advisors are able to know the actual date when students record their logs as some students are not recording their logs daily. Hence, this system provides assistances to both students and advisors in avoiding human errors and simplify the whole process of internship log books.

1.2 Problem Statement

According to the observation, there is no available internship log book system available in Fakulti Teknologi Maklumat dan Komunikasi (FTMK) yet. FTMK is still practicing the manual way for students to record their logs on books. Every student is given a log book to record their daily log during their internship period. Students are required to record daily and occasionally would let both industry and faculty advisors to check. Disadvantage of recoding them manually is illegible hand-writing by students. This causes faculty and industry advisors hard to read and comprehend the logs recorded by students. Consequently, advisors would lose interest in continuing to read and even simply care less about the significances brought by the log book.

Besides that, the process of checking students' log books is totally impractical and time-consuming using the old method. Both advisors are required to get hold of the physical book in order to review and comment students' logs. Advisors are not able to check students' log books remotely. There are times when students are required to meet with their respective advisors to hand in the book for them to check. This method would be a tiresome and troublesome for both students and advisors too. This method is not only time-consuming but also tedious for students and advisors who busy with their daily life tasks.

In addition, faculty advisors are having hard to check photos of their student's log book. For students who are not be able to reach their faculty advisors, they would send photos of the log they wrote and send to them via social app or email to them. This would cause discomfort to faculty advisors to review as there might some varied factors that causes them unable to review. Those factors are image quality, illegible hand writing and

various application students use, such as email or social chat applications. All this extra work is giving faculty advisors a hard time despite their daily work.

1.3 Objective

The objectives of developing FTMK Internship Log Book System are identified based on the review of the problem statements stated at **Section 1.2**. The objectives are listed as below:

i. **To develop an effective system that is able to improve the management of log books and reducing human errors.**
All students are able to write in logs systematically with the aid of calendar that provides indication of logging process. Both faculty and industry advisors are able to view respective students' status, whom they are in-charged of, based on the calendar, whether they have record their logs or not during the whole internship process. Both advisors are also able to view the actual date when students record it.

ii. **To simplify and reduce work load students, industry and faculty advisors.**
All students are able to type their logs with ease by going to the website and insert the title and content of the log with the option of uploading photos. If students are on medical leave, they are required to upload medical check (MC) as prove of absence. Industry and faculty advisors are able to review and update their remarks about students' logs content remotely and easily. This would reduce the difficulties faced by advisors who try to get hold of the physical book or looking at photos sent by students on phone or email.

iii. **To eliminate all hand-written log records with information system.**

Manual records are perishable and liable to spoiled, such as, water spilled all over would damage the log books. Hand written records would cause loss of data and uncomprehend-able hand-writings would make wonders, especially both advisors. Digitized records are easier to access with promising integrity and readable for student and advisors.

1.4 Scope

Scope that involved in FTMK Internship Log Book System is divided into two(2) parts, which are involvement of user and types of module. The scope is described as below:

i. User

a. Student

Students of FTMK is one the main users for FTMK Internship Log Book. Students can use the system to record their daily logs during the internship period set by university. The system allows them to carry out their tasks primarily based on the requirements of the subject BITU 3946.

b. Faculty Advisor

Faculty advisor of FTMK is also one of the main users for this system. The system enables them to observe and review students' log books during their internship remotely and effectively in order to make their job easier.

c. Industry Advisor

Industry Advisors of respective students is also one of the main user for this log book system. Just like faculty advisor, they are able to view and observe the records of students they are assigned with by using this system, making the process to check the log book much faster and simpler.

ii. Module

a. Login Module

This module is used to authenticate the validity and eligibility of each users before he or she can use the system. Each users are required to login using their username and password. If either username or password is inserted wrongly, the system will prompt out a login error message. Only the correct combination of username and password keyed-in by users allows users to access the system. This is very important as it protects the confidentiality of the students, faculty and industry advisors, log records and to prevent any unauthorized user from using, accessing and manipulate the data inside the system.

b. Calendar Module

This module allows users to pick a valid date from the interactive calendar interface and to view, insert and update the log data for that selected date. The calendar has a few color indicators to describe the metadata of those dates. For example, yellow indicates students have record their logs but haven't review by either one of the advisors; green indicates either one of students' advisors have commented students' logs; dark blue indicates dates that are outside range of the internship period; gray indicates weekends of respective companies that students intern; black indicates public holidays; red indicates the date when students are having medical leaves; beige color indicates no log is recorded by students yet. This module is future-proof which means that it is able to show dates, months

and years automatically and no maintenance needed. Overall, this calendar is the main core for the whole internship log book as recording a log starts from picking a date.

c. Record and Review Logs Module

This module allows users to manipulate the data of the selected date from the Calendar Module. After selecting a date from the previous module, students are able to record their daily logs accordingly by inserting the title and content. Students are allowed to upload a picture (.jpg for now) to show more interactive media for own self or advisors to see. If students were on leave, they would just simply click on the On Leave action button (red color) to automate the record with prior uploading letter for medical leave so that it provides conveniences to sick students. On the other hand, advisors are only allowed to review dates with yellow or green indicator (students have recorded their logs). They are allowed to make new comments about students' logs (yellow color) or update their comments (green color). An error message will be shown to advisors if the selected date has no log recorded by students (beige color). For a plus point, advisors are able to view the actual date when students recorded for the selected date, indicating if students did it on time or late let faculty advisors to determine the overall performance of their students. This part is not shown to students.

d. Users' Overview Module

This module is the first page when users log into their own respective account. For students, they will be able to see a message (red color) showing total number of dates that they did not record. For advisors, they

are able to view their own respective students' information that they are in-charged with, such as name, identity number/passport number, matric number, mobile number, company they are having intern at and position that they are having. This module is to allow users to see the first information they would like to look at when they use the system.

e. Generate Documents Module

This module allows all users to print out the log of selected date (with record), allowing users to keep it as physical record for external usage such as proof of medical leave. As good as a system is, physical record is still essentials for good.

1.5 Project Significance

The primary benefactors, which there are two, that will get advantages from FTMK Internship Log Book System are the faculty and industry advisors. The system provides a user-friendly interface with simple yet familiar user experience for them to manage students' records. The system allows them to retrieve, view and manage their respective students' information and records during the internship periods. Besides, the system allows faculty and industry advisors to be able to check and read students' record remotely, anytime and anywhere. Compare to the old method by sending photos using social applications or email, this system provides them simplicity and ease with their work. Besides that, complaints from industry advisors about illegible hand-writing would be no longer an issue as digitized records comes in.

With the thorough and correct configuration and database design, the system is able to store large about of data efficiently and effectively, including storing photos upload by students. Besides that, the system is reliable and monitor how diligently the students are by recording the actual date when students record their logs and acknowledge both advisors. Furthermore, the system ensure all data stored safely and ready to be displayed accordingly.

1.6 Expected Output

The expected outcomes from FTMK Internship Log Book System are as listed as below:

i. New web application to replace the manual method of recording and processing the log books
ii. To enable advisors to keep track and review students' records easily and effectively
iii. Remove human errors such as bad hand-writing with typing
iv. To store and ensure records are safe and complete.

1.7 Conclusion

Chapter I has introduced the basic idea of the project, FTMK Internship Log Book System, briefly. This chapter is the preliminary chapter to describe why the system is essential to be carry out and how it would bring benefits and changes the old method to record logs by replacing them with newer technology for faster, simpler and effective

execution by all users. FTMK Internship Log Book System is able to reduce tedious workload by advisors, increase simplicity in recoding log books by students, making the overall process to be more effective and efficient.

Introduction of the project is a good start to coordinate the work and explanatory to readers effectively. Hence, the next chapter will be Chapter II - Project Methodology and Planning, which will discuss how the project is planned and what methodology will be used to attain the aims as stated in this chapter.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

Last chapter had discussed about the basic idea on how the system would work, who are the users, etc. In this chapter will be focusing on about the method and plans to make this project (mainly about database). In this project, 5 phases of Database Life Cycle (DBLC) have been used as shown in Figure 2.1 Diagram of DBLC:

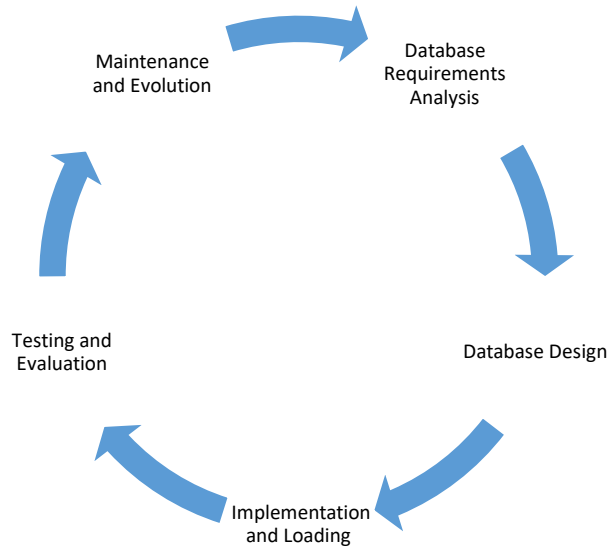

Figure 2.1: Diagram of DBLC

In the next section, a more detailed descriptions and tasks about each phase involved in the DBLC methodology used to create FTMK Internship Log Book System will be discussed.

2.2 Project Methodology

As mentioned above in Figure 2.1, the Database Life Cycle (DBLC) that is used in the process of developing FTMK Internship Log Book System compromises 5 crucial phases as follows:

i. Database Requirements Analysis

During this phase, user requirements are collected by consulting the users mentioned from Chapter 1, student, faculty advisor and industry advisor, and reviewing some log books from previous years. As mentioned, the consulted industry advisor would like not to disclose her identity to without company's permission. User requirements are very important for this system as one of objectives of this system is to make the whole process easier for users. So precise and accurate interpreted requirements are required, for example, advisors would like to know the actual date that students record their logs, and this would add an attribute to logs table. Another example is that students would like to show some pictures with their advisors or requiring students to upload photo of medical cheque as a proof of medical leave.

All type users are interviewed using unstructured and spontaneous method to get detailed and basic understanding of each users point of view of current manual log book system. The process of recording logs and letting advisors to check is based on experiences of each user. This method is preferable as it allows users to be more comfortable, subsequently detailed elements and attributes would be obtained, for example, advisors would sometimes call

students that they are in-charged. So students' contact number should be recorded into database, student table to be more precise.

Furthermore, referring to log books is also one of the method to review and understand the data that are required to be stored into database, such as title, date, and content. There are times that advisors would like to comment some of the logs recorded by students. So, advisors' comments, as attributes, are required in the table too. Combining these two requirements, an overall understanding of the whole system is produced.

Looking back at Malaysia's weekends and public holidays for some states or company policies are different, one of the requirements included the manipulations of weekends based on company instead of state law. This is very crucial as the system should disallowed students to record logs during weekends and public holidays. In order to maintain data integrity, this point of requirements should be take accounted.

Besides that, the period of internship session is set by the faculty and changeable depending on circumstances. This should be recorded in the database and students are only allowed to record logs on dates within the range of the internship session. This is also crucial to ensure there's no mistakes between log dates and internship sessions.

With all this overwhelmed data and requirements, writing down on a piece of paper would be the best way to review and analyze the whole system. Thus, the next step is to produce the Entity Relational Diagram (ERD) of the system based on all the requirements recorded.

ii. Database Design

In this phase, or referred as Database Logical Design, is to come out an overall Entity Relational Diagram (ERD). By all means, this phase focuses on the design of the database model that will support FTMK Internship Log Book System operations and objectives.

Based on the requirements collected from phase I, the system consists of a few strong entities such as students (table student), faculty advisors (table lecturer), industry advisors (table company_employee), companies (table company) too, not to forget internship session (table session) and weekends (table weekend) too. A table that act as calendar utilities is created as well to act as public holiday records.

After sorting out the strong entities, other entities are determined based requirements such as, table dailylog (for students to record their daily logs), company_location (for companies with multiple branches), company_weekends (to record company weekends), etc. This process is one of the most tedious process to ensure that less data redundancy and maintain data integrity. After that, normalization of the database model is carried out.

To further reduce data redundancy, normalization process is carried out. This step is to take out attributes that may contain repetitive data in the same attributes. For example, postcode, name of city and name of state can be repetitive for addresses. That is why, the creation of table malaysia_poscode, malaysia_city and malaysia_state is done as there are companies has the same postcode, city and state as part of their addresses.

'Bridge' formed between the entities as normalization process occurs., such as table company_weekend (repetitive of same weekends as other companies). Thus, the basic model of utem_intern (database name for FTMK Internship Log Book System, mentioned on 2.1 (Introduction) has be formed. Instead of thinking and wondering if the ERD is workable or not, might as well create, implement and test it with real database which will be discussed on phase III.

iii. Implementation and Loading

Implementation and Loading is basically the phase where creation the physical database is being carried out. In this system, XAMPP is being installed that comes with MySQL Relational Database Management System (RDBMS). To make it easier to be used, MySQL Workbench is installed into the machine as well. Later on, a new connection between MySQL and the workbench is created. After the successfully connected, utem_intern database is created. From here on wards, all tables such as table student, table lecturer, table company, etc. is created by using Data Dictionary Language (DDL) tables. Of course, if there are any constrains such as primary key, foreign key, unique, check, etc., it should be included using DDL. For example, students' Matric Number is the primary of table student (student_id) with their respective Identity Card (IC) number (student_nric) with a constraint of unique constraint as each student must be different.

Besides that, stored procedures and triggers are created as well at this phase. Store procedures are great as it is able to provide better performances. For example, retrieving dates when students recorded their logs is a heavy process as FTMK students would normally have 6 months of internship which sum up approximate 150 records for each student. Therefore, store procedures is the ideal solutions in this case. Triggers automate the checking constrains or business logic before and after insert, update and even delete. For example, before inserting new log, the database would check if is the selected date is inside the range of the session or not. If it isn't, error message will be shown.

Later on, some data sample is stored in to each table just to ensure there are some data to for the next phase, which is Testing and Evaluation of utem_intern database.

iv. Testing and Evaluation

During the second (2^{nd}) phase, database is designed with decisions that were made to ensure integrity, security and performance. In third (3^{rd}) phase, those design are put into place with some dummy data.

In this phase, the system's database is tested and fine-tuned to ensure that it performs as expected result from phase I, such as follows:

a. Test the database

While testing the database, both valid and invalid data are inserted into each table accordingly to test the data integrity and security of the data. If there's any process that does not fulfilled the requirements from phase I, the database should be tuned in order to fix it. For example, inserting same value of students' Matric Number (student_id) into table student, will resulting primary key constrain error. Besides that, getting the relevant data to show users is a crucial part to. This test case is the most important part as privacy and closure of data is important. Although the system is meant for university usage, unauthorized users are still not allowed to view unauthorized data. For example, advisors are only able to view students' log records that are assigned to them.

b. Evaluate Database and Its Application Programs

This phase is the most exciting phase where the database and applications are tested spontaneous. Based user requirements and objectives of

FTMK Internship Log Book System, a complete Test Plan (Chapter VI) is produced with various test cases to test the proposed system. It needs to be able to pass the test case in order to ensure user satisfactions and par with the requirements that were collected at the early stage of the process. If some of the test case failed, it's time to go next in order to maintain and evolve the database so that it can be a better system.

v. Maintenance and Evolution

In this phase, maintenance of the database is done by using MySQL Workbench. In the process of maintaining the database and the data, backup and recovery is very crucial so that when altering tables or any changes to the constrains of each tables, these data are backup and recover after the altering the tables. Loss of data is a huge hassle for project development.

However, in this project, this phase has not been carried out as the whole process would require much time and it is a long-term process. Therefore, this process is not included in this project.

2.3 Project Schedule and Milestones

Table 2.1: Project Milestone

Milestones	Expected Documents	Dates (Week)
1. Requirements Analysis	1. Flow chart of the system 2. DFD of the proposed system 3. Requirement of the proposed system (Functional, non-functional)	11th March (Week 3)
2. Conceptual Design of the proposed system	1. A completed ERD	18th March (Week 4)
3. Physical Design of the proposed system		29st April (Week 10)
4. Testing and Evaluation of the proposed system	1. Test Plan 2. Test Cases 3. Test Result and Analysis	6th May (Week 11)

5. Maintenance and Evolution of the proposed system		27th May (Week 14)

In Table 2.1, it shows the milestones of development of FTMK Internship Log Book System. In each milestone, there are documents that are expected to be produced as deliverables or documentations for this project for better and smoother development process. This project milestone also acts as a timeline for development process to ensure all process are on par with the expected timeline and should be on time with relevant documents.

Table 2.2: Gantt Chart

ACTIVITIES / WEEK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Title Discussion																
Proposal Correction and Improvement																
Requirement Analysis																
Designing ERD																
Constructing Physical Database																
Mid Semester Break																
Test and Evaluate the System																
Maintenance and Evolution of the System																
Final Presentation																
Revision Week																

In Table 2.2 Gantt Chart, it shows the overall development timeline that this project has carried out. Aligning the activities and number of week keeps the project development on time and set targets to be reached on each week.

2.4 Conclusion

Chapter II has provided the methodology and planning for this project, FTMK Internship Log Book System. This chapter is the chapter to give a technical breakthrough on how to implement and carry out the process of developing this project from idea into a reality.

A more detailed on technical analysis of overall system will be discussed on the next chapter which is Chapter III Analysis.

CHAPTER III

ANAYLYSIS

3.1 Introduction

This chapter will be discussing about the analysis phase of the project. Analysis is basically a process of breaking a whole system into modules and collect factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system which includes studying the current log book processes of all users, gathering operational data of the system, understand the flow of all required data, finding bottlenecks from all users and finding solutions to ease their workloads. Thus, this phase majorly will be focusing on the understanding and documenting the current Fakulti Teknologi Maklumat dan Komunikasi (FTMK) internship log book system (the as-is system) and identify improvements for the new system (to-be system) with the fulfillment of user requirements.

In this phase, the users of the new system (students, industry advisors and faculty advisors) are required to collaborate in order to be familiar with the details and their needs and wants from the new system in their respective point of view. Collected information are analyzed and will be featured in the to-be system.

3.2 Problem Analysis

Some interview sessions and observations have been conducted in order to identify the hard knocks and weak points of the current internship log book system in FTMK. It appears that the current log book system is still done manually with papers and pens. Instead of fully utilize the available technology, like Microsoft Word, lots of papers are used. The internship log book system requires students to write for each day per page using a booklet that are given by the faculty. This booklet is bind and it is unable to tear out the unused ones to make another booklet. This situation would cause a lot of paper wastage, not eco-friendly and resource usage is inefficient. Besides that, these hand-written logs recorded daily by students are troublesome as may be seen on Appendix A. Illegible handwritings are hard to read and understand that will cause extra work and time for faculty and industry advisors. Difficulties to comprehend the content would cause advisors to lose interest in continuing to read and understand, a burden for every both parties of advisors. Plus, if it is a medical leave, students are required to attach the medical cheque to the log book with staples. If there is any mishap, it would crumple the medical cheque or even tear apart and causing the loss of alibi for medical leave.

In addition, faculty advisors who wish to check students' log books would take up a lot of time just to get hold of the book. Students are required to book a date and time with advisors just to show the book which is totally impractical way of checking process. Advisors who are away from their students would not be able to obtain the physical book to review and comment students' log books. Besides that, advisors who have a tight schedule would not be able to meet up with their respective students as they are busy with their jobs and tasks in office or faculty. These situations would cause the lack of communications between both students and advisors which could lead to picturized the incapability of both parties to work or in short, causing both

parties to be care less about the log book. In the end, the significance that can be brought by log books would definitely deserted and unkempt.

Currently, the solution only solution for faculty advisors who are away from students to check their records are students take photos of their records and send to their respective advisors. Students are to send these photos to their advisors via social app or email. This sort of situation is very irksome and inconvenient for faculty advisors as there might be some varied factors that causes them unable to review. Those factors are image quality, illegible hand writing and various application students use, such as email or social chat applications. All this extra work is giving faculty advisors a hard time despite their daily work.

The following figures illustrate the overall current manual internship log book process in FTMK from students', faculty advisors' and industry advisors' point of view.

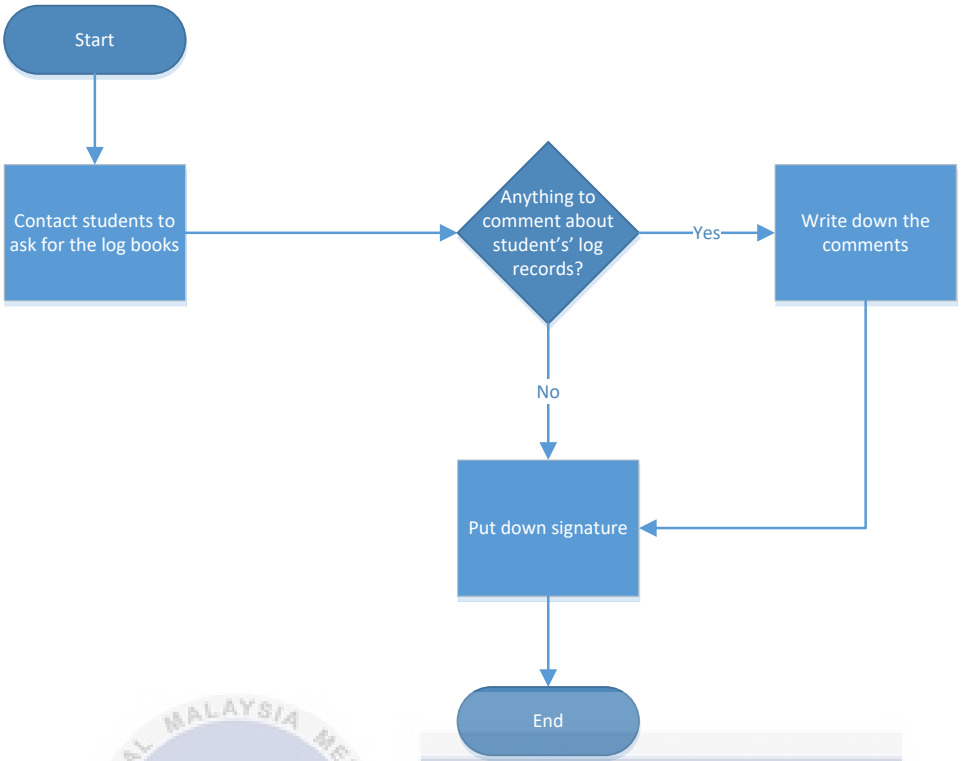

Figure 3.1: Current Internship Log Book System Flow Chart (Industry Advisor)

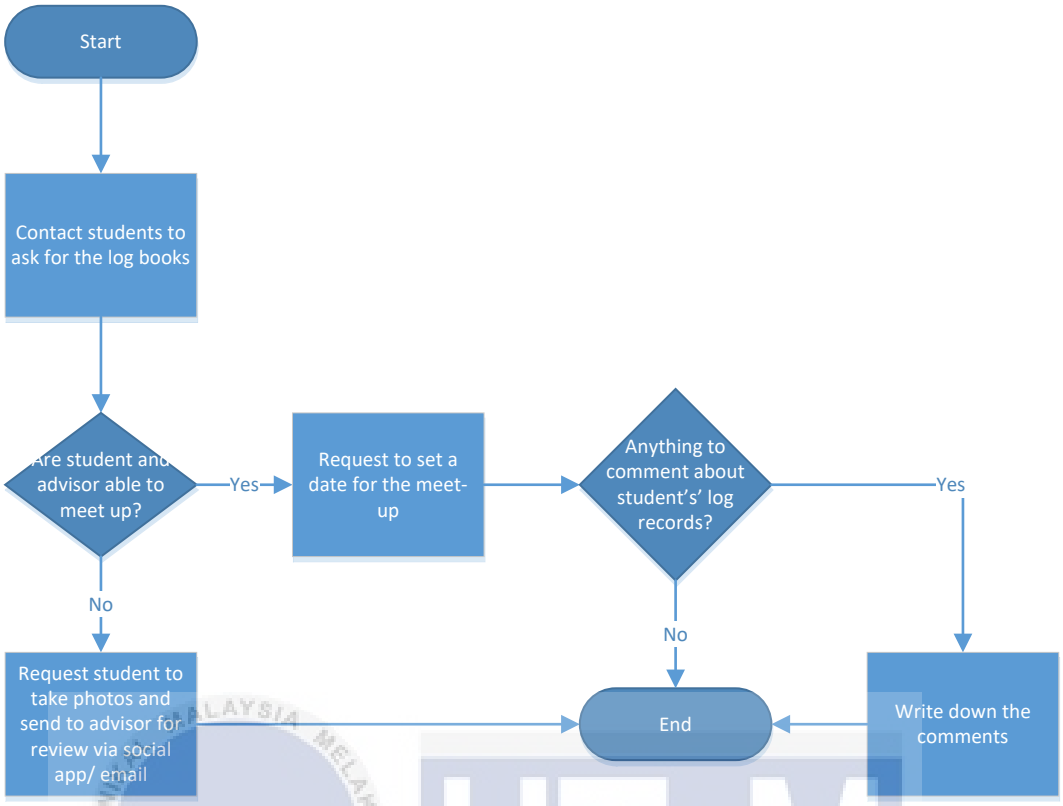

Figure 3.2: Current Internship Log Book System Flow Chart (Faculty Advisor)

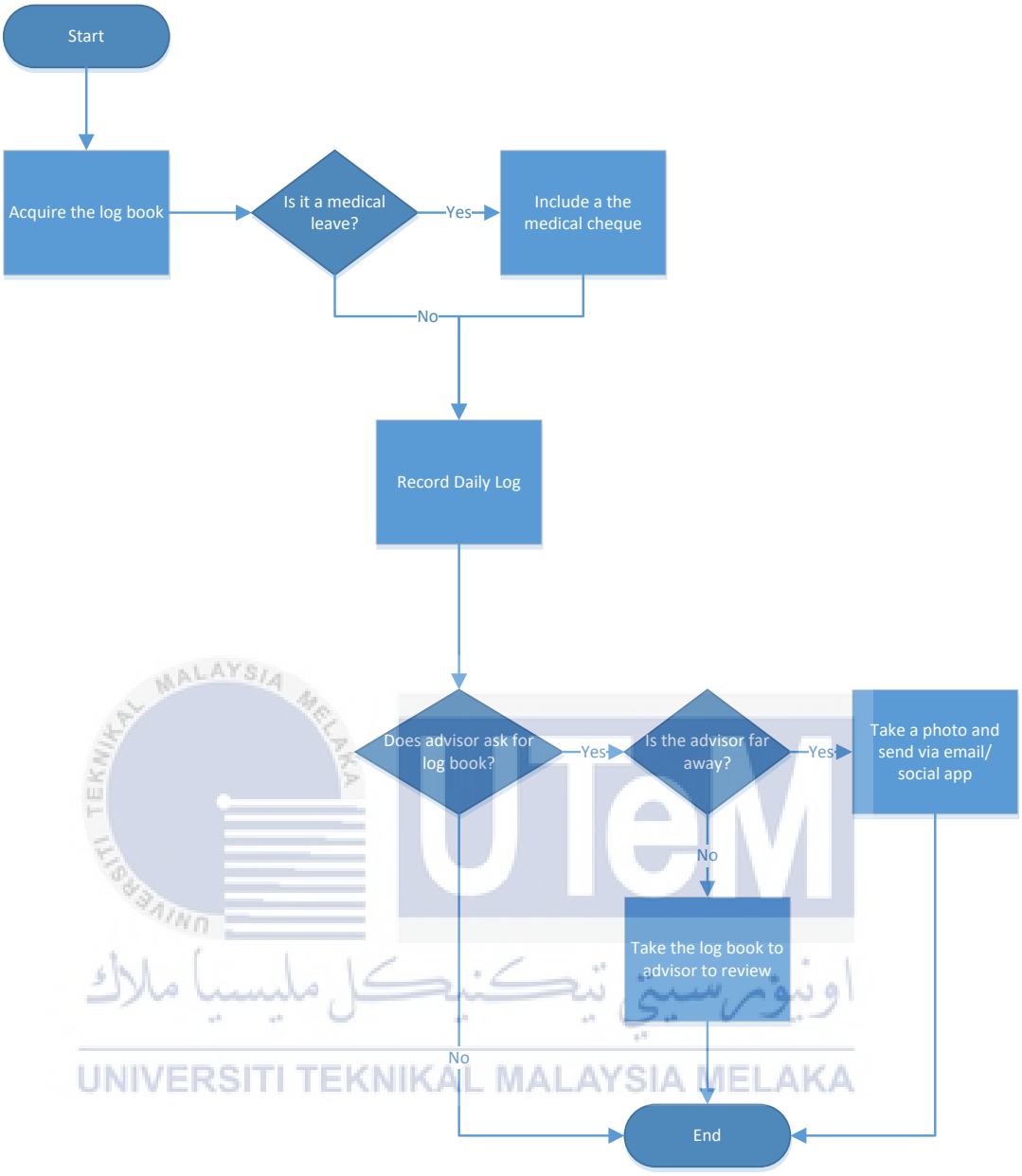

Figure 3.3: Current Internship Log Book System Flow Chart(Student)

3.3 The Proposed Improvements / Solutions

The proposed system must be an evolutionary, simple yet sophisticated in order to overcome the flaws and drawbacks of the current internship log book system used in FTMK. The proposed system is necessary to reduce paper work and save time for both parties, students and advisors, to carry out their corresponding tasks and roles. This can be done by storing all the data into database through a user friendly web based system. The user interface must be created with simplicity as priority so that users are able to understand and easily pick up with the system. User experience must the same concept as the current manual system, such as recording logs so that users would not feel alienated from the current system to the proposed system. The proposed system should also be able to retrieve and store data in a fast and accurate way. Therefore, the proposed system should be an information system which works perfectly for storing and managing data. Thus, there is no need to record logs on papers anymore.

Moreover, the overall process of current internship log book system should be transformed from manual procedures to computerized, to be more precise; a web-based system. This is because the chance of any human error can be lowered by using digital process. Besides that, a centralized data that can be view and retrieved remotely by both faculty and industry advisors would lessen their job process as advisors. The proposed system need to be capable to differentiate medical leave and normal logs with indicators so that users are able to differentiate which date students had medical leave.

The proposed system also should be able to upload and view images for the medical cheques. This is to overcome the problem of attaching the medical cheques on log books which possess high possibility of losing if neglected. The proposed system should require students to upload the image of the medical cheque if students are having medical leave on certain date. By doing so, the data recorded would not be counterfeit and can be ensured to be shown as proofs upon any questions from both

advisors. The following figures are the overall process for each user involved in the proposed system.

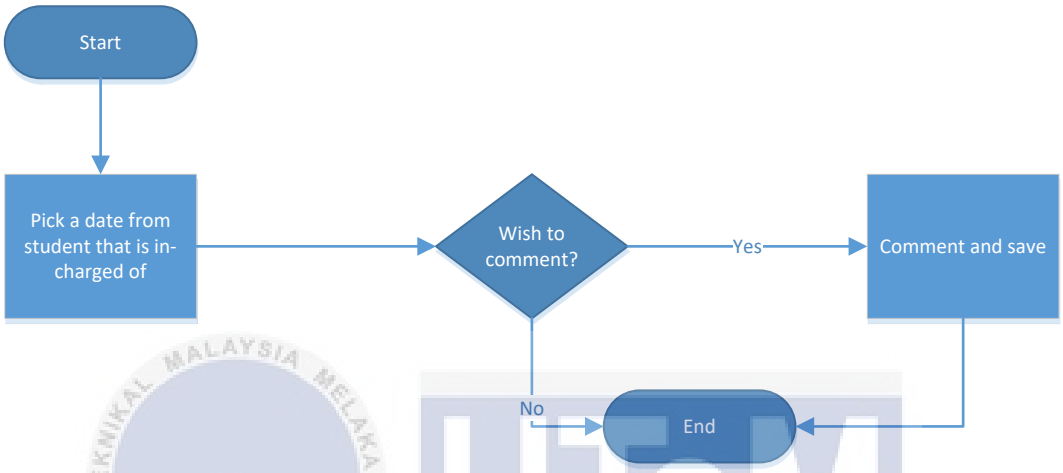

Figure 3.4: Proposed System Process Flow Chart (Industry and Faculty Advisor)

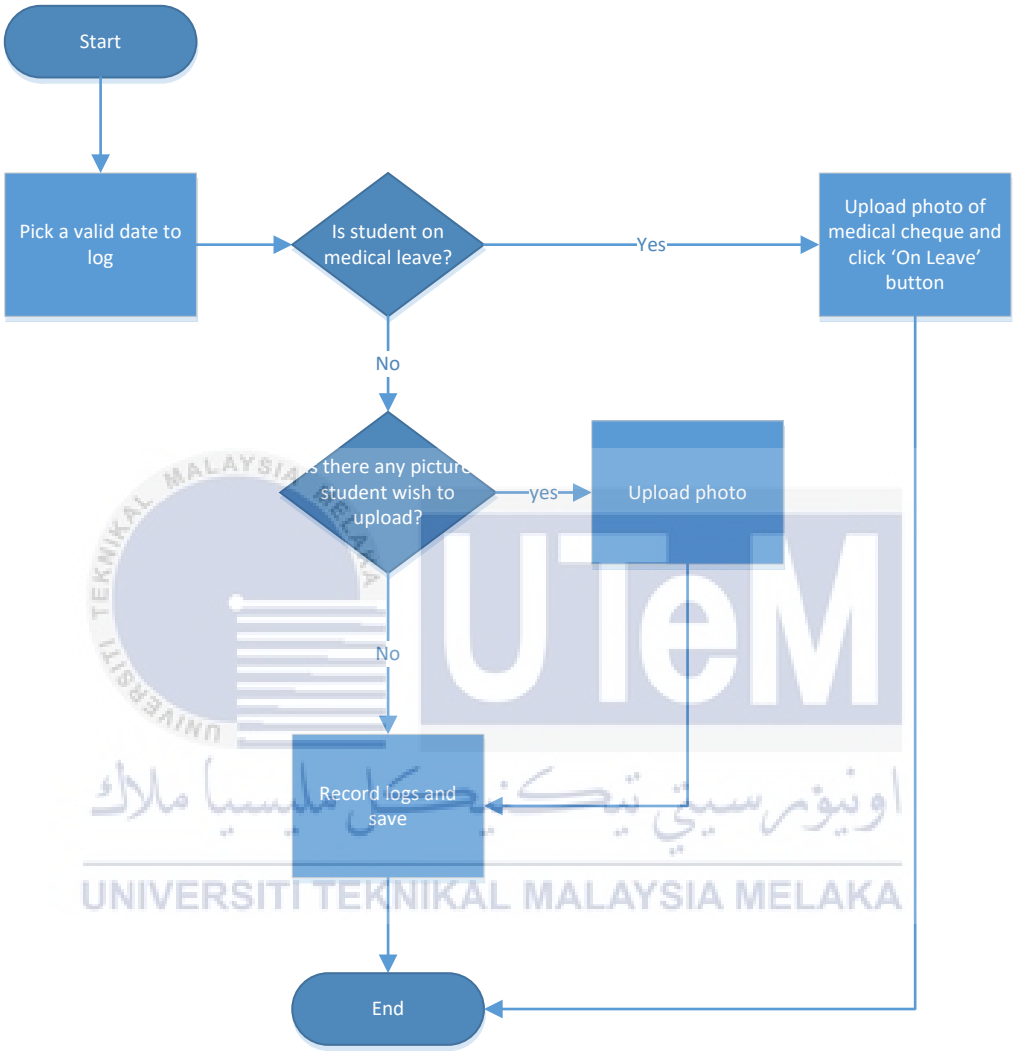

Figure 3.5: Proposed System Process Flow Chart (Student)

3.4 Requirement Analysis of The to-be System

Requirement analysis is used to convert the system requirements into refined and precise list of details on how the proposed system should behave and what the

system should do to provide valuable data. There are three categories of requirement in this section, which are functional requirement, non-functional requirement and other requirement such as software and hardware requirement.

3.4.1 Functional Requirement

Functional requirements are identified by the main user of the system, which are students, faculty advisors and industry advisors. The functional requirements state the services that would be provided by the system the data the system must store during the process. Table 3.1 lists out all the functional requirements and its descriptions for FTMK Internship Log Book System.

Table 3.1 Functional Requirements

No	Functional Requirement	Description
1	User Authentication	1. The system enables user to login and logout 2. The system is able check the validity of the user's id and password and redirect to respective page.
2	Calendar Management	1. The calendar is able show indications and status of each students' log records 2. The calendar is shown with the range of internship of each students 3. The calendar is able to indicates weekends of companies where students intern and public holidays. 4. The calendar is able to control dates of students' choice, including invalid dates and future dates.

3	Record Logs Management	1. The system allows students to record their daily logs according to the date they pick by inserting title and content of the log. 2. The system allows students to upload pictures to the system 3. The system requires students to upload medical cheque upon on medical leave. 4. The system is able retrieve and present logs saved in the database
4	Review Logs Management	1. The system allows advisors to review and records comments of students' log records 2. The system is able to store comments by advisors 3. The system is able to retrieve and present comments by advisors 4. The system allows advisors to update and change their comments.
5	Student Management	1. The system allows advisors to view list of students and their biodata that they are in-charged of, 2. The system is able retrieve and present that data from database.
6	Generate Documents	1. The system allows users (students and both advisors) to generate and print log records saved in the database. 2. The system enables printing of document in Portable Document Format (PDF).

3.4.1.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) is one of the graphical way to illustrate the processes or activities that are performed and how data are move among them. In this project, context diagram is first to be built to show the external entities involved and the data flows that start or end in them. Afterwards, DFD fragment is created based on the project requirements to show how the data is flowing through every processes, external entities and data stores. The maximum level the DFD is decomposed in this project is level 2 DFDs.

i. Context Diagram

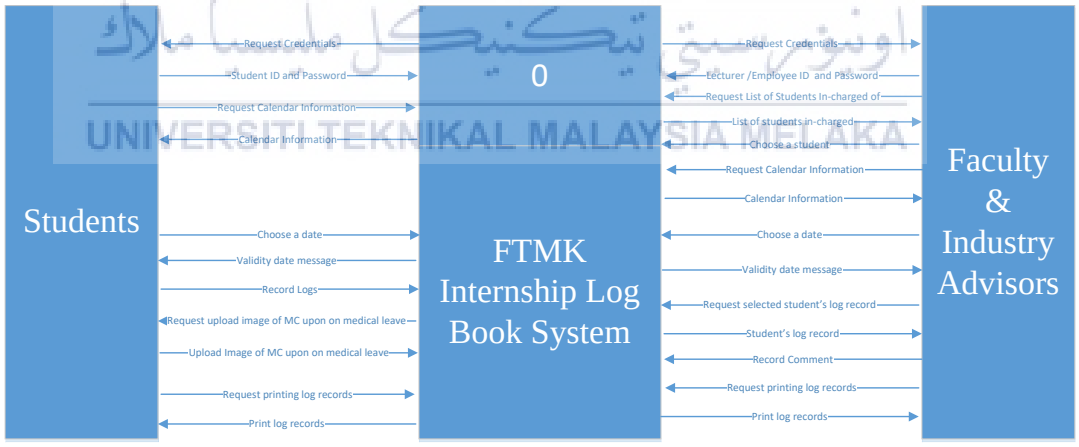

Figure 3.6 Context Diagram

ii. Level 0 DFD

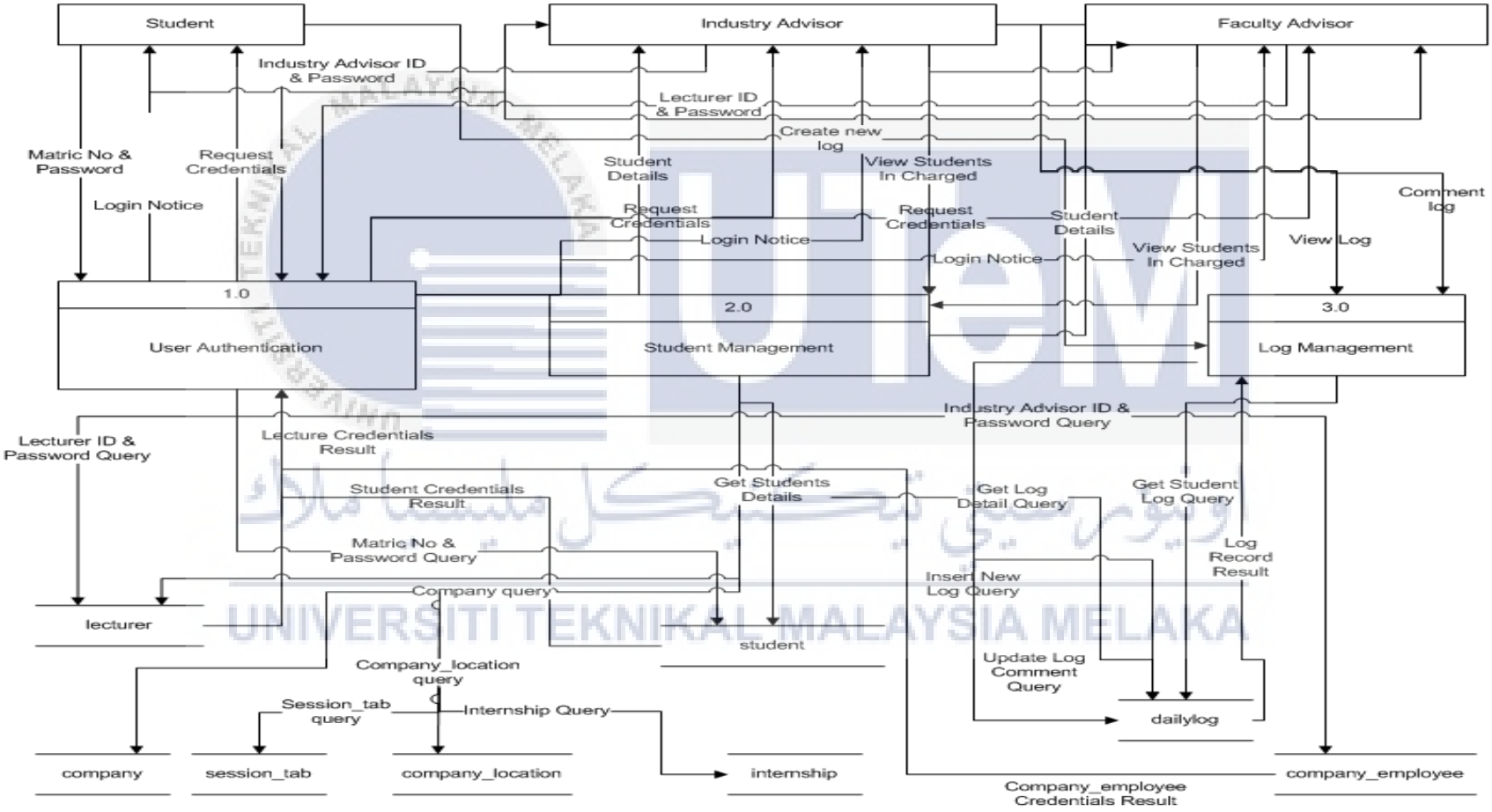

Figure 3.7 FTMK Internship Log Book System Level 0 DFD

iii. Level 1 DFD

a. Process 1.0 – User Authentication

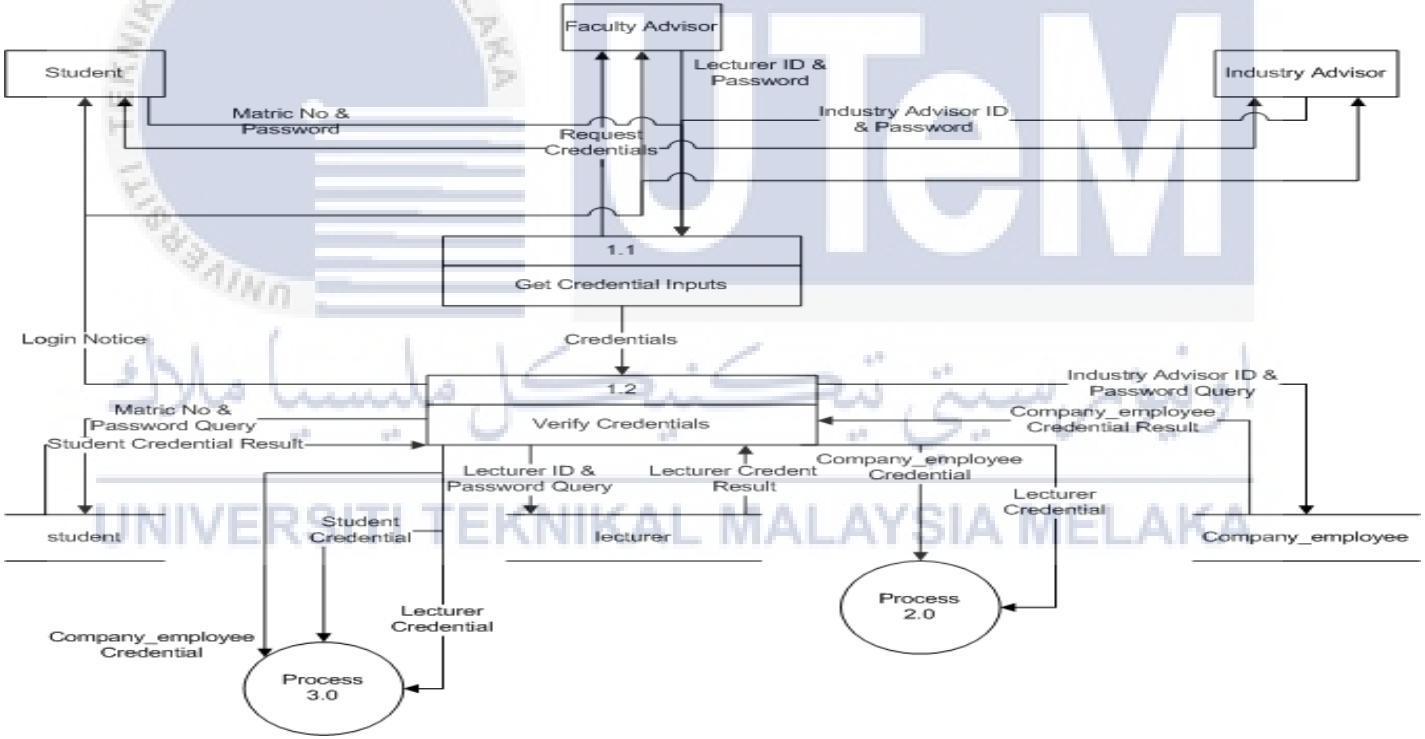

Figure 3.8: FTMK Internship Log Book System Process 1 (User Authentication) Level 1 DFD

b. Process 3.0 – Log Management

i. Student

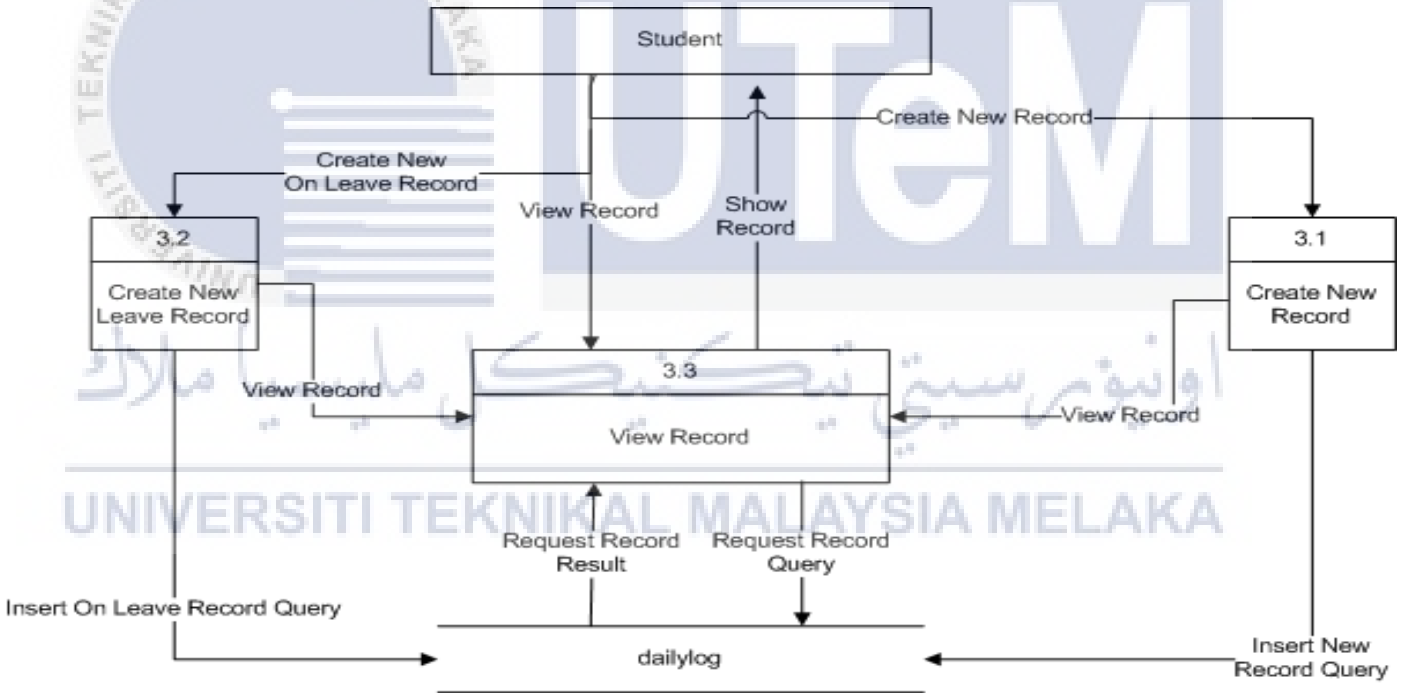

Figure 3.9: FTMK Internship Log Book System Process 3 (Log Management) Level 1 DFD (Student)

ii. Faculty Advisor

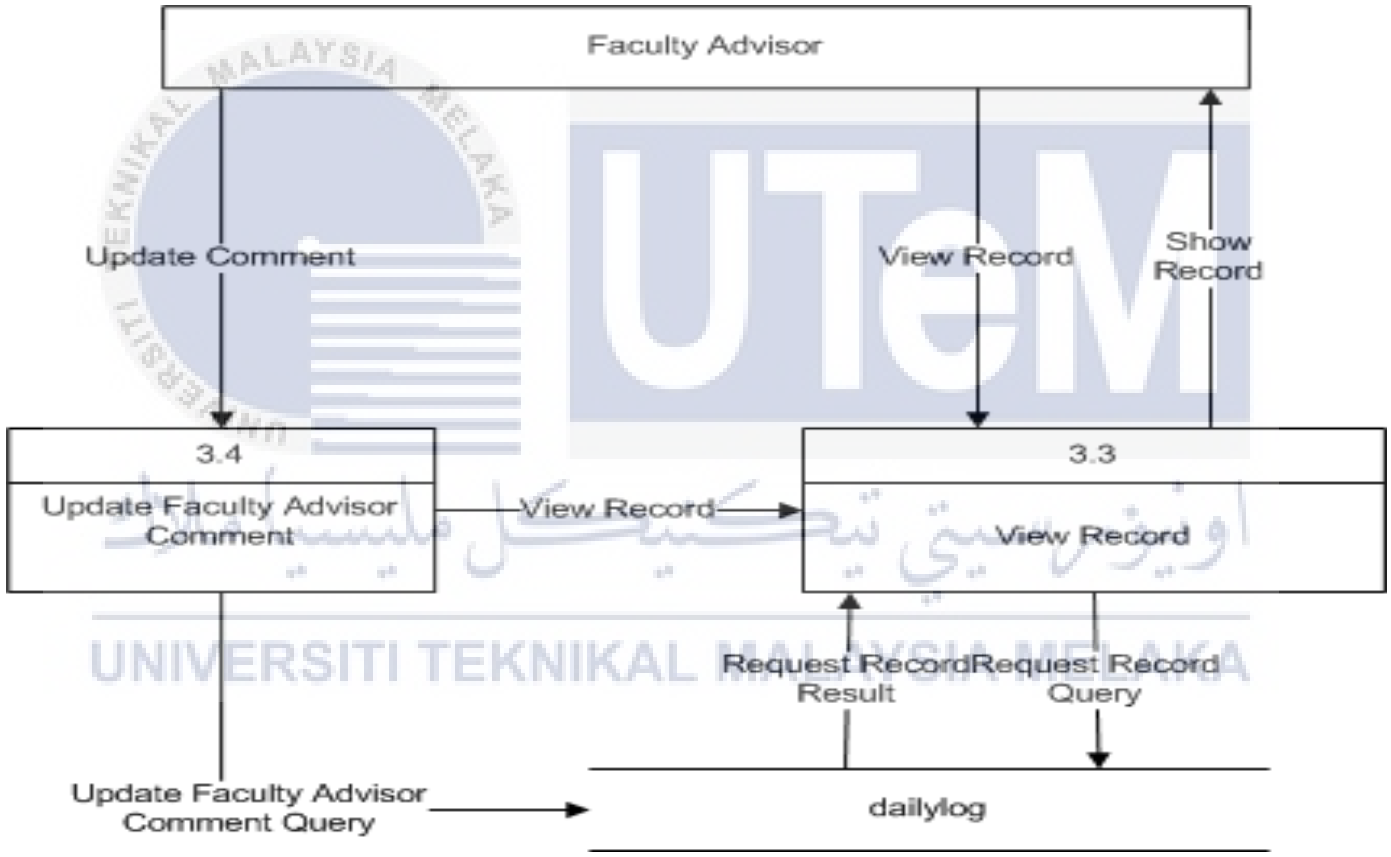

Figure 3.10: FTMK Internship Log Book System Process 3 (Log Management) Level 1 DFD (Faculty Advisor)

iii. Industry Advisor

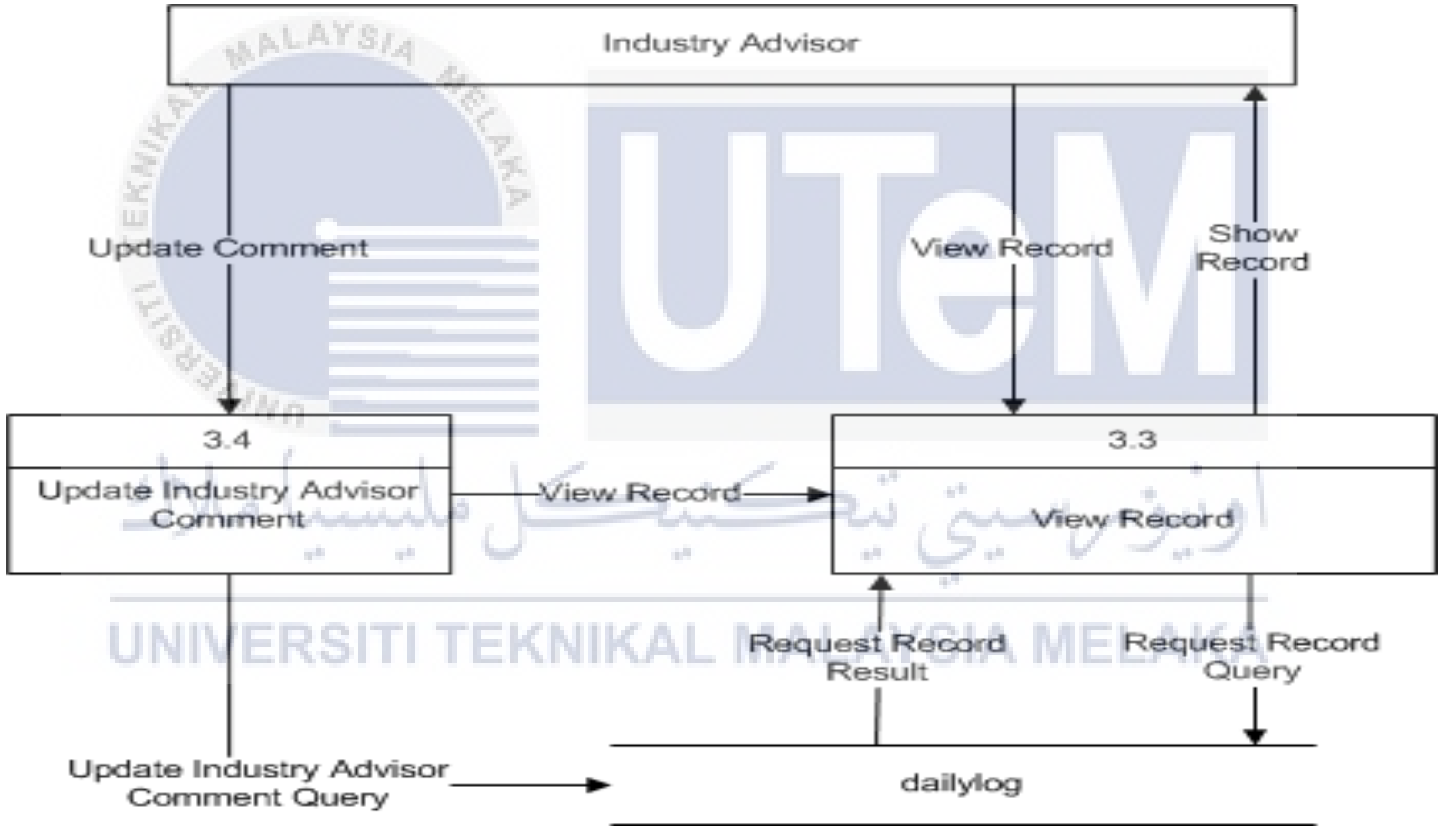

Figure 3.11: FTMK Internship Log Book System Process 3 (Log Management) Level 1 DFD (Industry Advisor)

3.4.2 Non-functional Requirement

Non-functional requirement is a group of requirement that describes a variety of system characteristics to attain user's higher satisfaction toward the system. Table 3.2 lists out all the non-functional requirements and its descriptions for Student Attendance Management System.

Table 3.2: Non-functional Requirement

No	Non-functional Requirement	Description
1	Security	Only authorized lecturers can use and access to the system.
2	Data Integrity	Accuracy – Data stored and displayed shall be 100% accurate at all times in accordance with the data that is inserted into the database.
3	Performance	The system reduces the overall log records process time.
4	Operational	The system should run on desktop or laptop to be used by users.

3.4.3 Other Requirement

Other requirements that are required in order to complete the system are categorized into 2 types, which are software and hardware.

3.4.3.1 Software Requirement

Software requirement and specification that will be used in FTMK Internship Log Book System is as listed in Table 3.3

Table 3.3 Software Requirement

No	Software	Advantage	Disadvantage
1	XAMPP	It is a free and most popular Apache distribution for PHP development environment	Instead of MySQL server, it uses Maria Server which slightly incompatible with MySQL Workbench
2	MySQL Workbench	It is a UI application that enables DBA, developer to design, model, generate and manage databases rapidly.	It has several bugs while developing the system which requires to restart the software sometimes.
3	PHP Storm	It is a smart IDE for PHP development with full features, such as frameworks, tools, refactoring, etc. which enables faster development.	It is not a free software which requires yearly subscriptions.
4	Microsoft Visio 2013	Diagramming application that will be used to draw diagrams such as data flow diagram (DFD), context diagram and entity relationship diagram (ERD).	It has a harder UI/UX for diagramming applications
5	Microsoft Word 2016	Word processing program that will be used to type and save documents.	-
6	Window 10 Single Language	This is the latest Windows OS that is compatible with most of the	

		software available in the market as it is backward compatible	-

3.4.3.2 Hardware Requirement

The list of hardware components that will be used in the FTMK Internship Log Book System is as shown in the Table 3.4

Table 3.4: Hardware Requirement

No	Hardware	Description
1	Laptop	Laptop with specification Intel Core i5 or higher, 4GB RAM and higher will be used to build the system
2	Printer	Device that will be used to print documentation.
3	USB Drive	Data storage that will be used to transfer files from one computer to another.

3.5 Conclusion

Chapter III summarizes the investigation of the current system, identifies solution or improvement opportunities and develops a concept for the new system. It is found out that the current attendance process is done manually by the lecturer. Therefore, the probability of human error to occur in the attendance process is higher. Hence, a user friendly and convenient system is required to minimize the error besides easing the overall attendance process. All the functional and non-functional requirements of the new system are first being identified in this chapter before it is implemented. Data flow diagram is also being constructed to know how the data should flow throughout the system before moving to the design phase of the project.

The deliverable from the analysis phase is both an analysis and a high-level initial design for the system. The coming chapter that will be explored in depth is Chapter IV – Design. The following chapter will decide how the system going to operate based on the analysis done in this chapter.

CHAPTER IV

DESIGN

4.1 Introduction

After analyzing the internship log book system (Chapter III Analysis), design phase is the phase to carry out and realize the ideas and outcomes from that phase. In this phase will be discussing about designing the system, majorly in database design which compromises starting from drawing out Entity Relational Diagram (ERD) in Conceptual Design to normalization and SQL statements in Logical Design, follow by implementation in Physical Design. Not to mention, this will chapter will be further discussing about Graphical User Interface (GUI) Design as well as one of the objectives of Fakulti Maklumat dan Komunikasi (FTMK) Internship Log Book System is to let users have a hassle-free user experience.

4.2 Introduction Preview of this Chapter

FTMK Internship Log Book System is designed based on 2-Tier design architecture. “2-Tier architecture or client-server model architecture separates the logical tier on the client side and data tier on the server side. 2-Tier architecture has 3 layers, which are layer to store data, a layer to deliver the data, and a layer to display the data (Gustavo, A *et al*., 2004).” The following Figure 4.1 illustrates the 2-tier architecture design002E

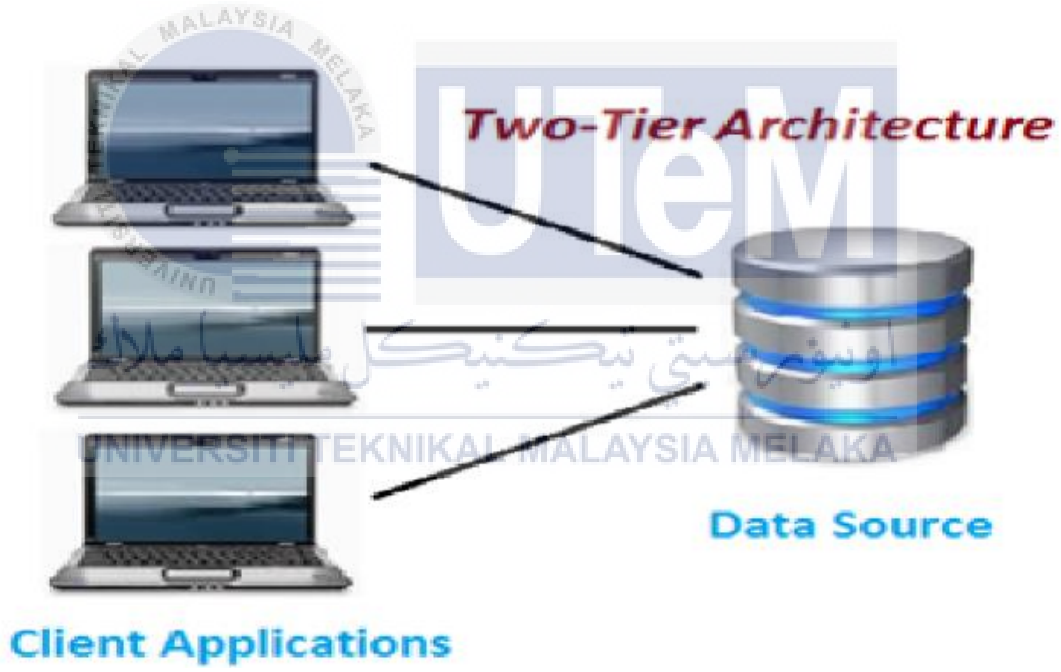

Figure 4.1: 2-Tier Architecture

Logical tier is responsible for to carry out logical operations request by users while preserving the business rules that are found and recorded during the previous phase, Analysis phase. This tier is to control the presentation of data to authenticated users based on their respective privileges and authorization so that the requested operations or data are carried out appropriately. Besides that, logical tier is also responsible to carry out process of typical dynamic web-based system; create, retrieve,

update, delete (CRUD), by processing inputs from users while using the internship log book system.

Data tier is responsible for database storage and programming such as schema, tables, stored procedures, etc. of the internship log book system. All inputs from users by using this system is stored in this tier through operation requests which compromises triggers, stored procedures, functions, etc. that are created in the database during Physical Design.

4.2.1 Conceptual Design

Figure 4.2 shows the Entity Relational Diagram of FTMK Internship Log Book System. The system consists of 15 entities, as shown, to store all the data that are used in the log book system. The ERD consists of a few noticeable strong entities which are crucial to the whole internship log book system, which are student, lecturer, company, company_employee, company_location, internship, dailylog and sesstion_tab entities. Other entities are to support and make the whole system to be more robust and complete in order to handle uncertainty situations that may happened.

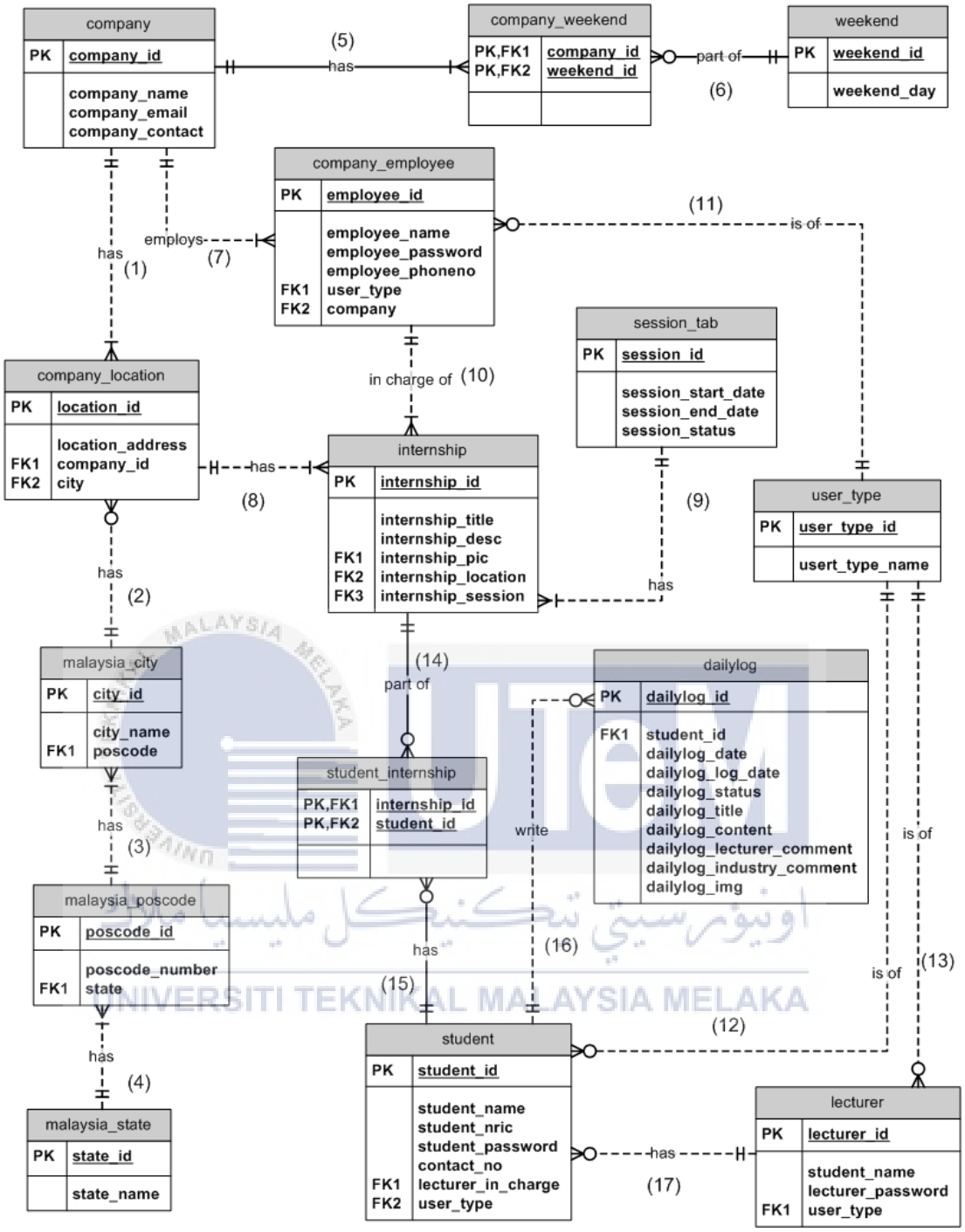

Figure 4.2: ERD of FTMK Internship Log Book System

The following Table 4.1 contains the overview of business rule of the whole internship log book system. The numberings of the business rules are mapped according to the one on Figure 4.2.

Table 4.1: Business Rule of FTMK Internship Log Book System

No.	Business Rule Description
1	One company has one or many company_location and one company_location belongs to one and only one company
2	One malaysia_city may have zero or many company_location and one company_location located at one and only one city
3	One malaysia_poscode has one or many malaysia_city and one malaysia_city has one and only one malaysia_poscode
4	One malaysia_state has has one or many malaysia_poscode and one malaysia_poscode belongs to one and only one malaysia_state
5	One company has one or two company_weekend and one company_weekend belongs to one and only one company_weekend
6	One weekend is part of zero or many company_weekend and one company_weekend belongs to one and only one weekend
7	One company employs one or more company_employee and one company_employee belongs to one and only one company
8	One company_location offers one or many internship and one internship is offered by one and only one company
9	One internship belongs to one and only one session_tab and one session_tab has one or many internship
10	One company_employee is in charge of one or many internship and one internship is in charged by one and only one company_employee
11	One company_employee is one and only one user_type and one user_type belongs to zero or many company_employee
12	One student is one and only one user_type and one user_type belongs to zero or many student
13	One lecturer is one and only one user_type and one user_type belongs to zero or many lecturer
14	One student_internsip is part of one and only one internship and one internship is part of zero or many student_internship
15	One student has zero or one student_internship and one student_internship belongs to one and only one student

16	One student writes one or many dailylog and one dailylog is written by one and only one student
17	One lecturer is in charge of zero or many students and one student is in charged by one and only one lecturer

4.2.2 Logical Design

Logical Design is the second part of database design. In this phase, relations between local logical data are defined, normalized and validated against user transactions. Besides that, constraints for each local logical data are reviewed.

4.2.2.1 Data Dictionary for Entity Relational Diagram

Table 4.2: Table Company

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Company_id	ID number for company	Varchar (30)	PK	RC001
Company_name	Name of company	Varchar (255)	Unique / Not Null	Infineon
Company_contact	Contact number of company	Varchar (10)	Unique / Not Null	0644434343
Company_email	Email of company	Varchar (50)	Unique / Not Null	contact@infineon.com.my

Table 4.3: Table company_location

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Location_id	ID for location of company	Varchar (30)	PK	Texas06
Location_address	Address for location of company	Varchar (255)	Unique / Not Null	Texas Instrument Melaka, Jalan Bandar Raya
Company_id	ID for company	Varchar (30)	FK references table company	RC001
City	ID for City	Varchar (5)	FK references table malaysia_city	MC7

Table 4.4: Table company_employee

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Employee_id	ID for employee	Varchar (20)	PK	T031310166
Employee_name	Name of employee	Varchar (255)	Not Null	Lee Kwon Hoe
Employee_password	Password of employee	Varchar (30)	Not Null	Texas123
Employee_phoneno	Phone number of employee	Varchar (11)	Unique / Not Null	012-3456789

Company	ID of company	Varchar (30)	FK references table company	RC001
User_type	ID of user type	Varchar (30)	FK references table user_type	UT004

Table 4.5: Table company_weekend

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Company_ id	ID for company	Varchar (30)	PK & FK references table company	RC001
Weekend_ id	ID for weekend	Varchar (3)	PK & FK references table weekends	WK6

Table 4.6: Table weekends

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Weekend_ id	ID for weekend	Varchar (3)	FK	WK6
Weekend_ day	Day for weekend	Varchar (45)	Unique / Not Null	SATURDAY

Table 4.7: Table malaysia_city

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
City_id	ID for city	Varchar (5)	PK	MC1
City_name	Name for city	Varchar (100)	Not Null	Bandar Hilir
Poscode	ID of poscode	Varchar (5)	FK references table malaysia_poscode	MP1

Table 4.8: Table malaysia_poscode

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Poscode_id	ID for poscode	Varchar (5)	PK	MP1
Poscode_number	number for poscode	Varchar (6)	Not Null	75000
State	ID for state	Varchar (5)	FK references table malaysia_state	MS1

Table 4.9: Table malaysia_state

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
State_id	ID for state	Varchar (5)	PK	MS1

State_name	Name for state	Varchar (50)	Unique / Not Null	MELAKA

Table 4.10: Table internship

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Internship_id	ID for internship	Varchar (15)	PK	I001
Internship_title	Title for internship	Varchar (45)	Not Null	Software Engineer
Internship_desc	Description for internship	Varchar (255)	Not Null	To document each software process
Internship_salary	Salary for internship	Number (5,2)	Not Null	900.00
Internship_location	ID for location	Varchar (10)	FK references table company_location	Texas06
Internship_session	ID for session	Varchar (7)	FK references table session_tab	S2Y1516
Internship_pic	ID for employee	Varchar (20)	FK references table company_employee	T031310166

Table 4.11: Table session_tab

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Session_id	ID for session	Varchar (7)	PK	S2Y1516

Session_start_date	Start date for session	Date	Date must be bigger than Session_end_date	2015-05-01
Session_end_date	End date for session	Date	Date must be smaller than Session_start_date	2016-05-01
Session_status	Status for session	Varchar (20)	Must be either “ACTIVE” or “EXPIRED”	ACTIVE

Table 4.12: Table student_internship

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Student_id	ID for student	Varchar (10)	PK & FK references table student	B031310166
Internship_id	ID for internship	Varchar (15)	PK & FK references table internship	I001

Table 4.13: Table student

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Student_id	ID for student	Varchar (10)	PK	B031310166
Student_name	Name for student	Varchar (100)	Not Null	CHAN WILSON

Student_nric	NRIC for student	Varchar (12)	Unique / Not Null	930612075363
Contact_no	Contact number for student	Varchar (11)	Unique / Not Null	012-5923325
Student_password	Password for student	Varchar (8)	Not Null	123456
User_type	ID for user type	Varchar (30)	FK references table user_type	UT002
Lecturer_in_charge	ID for lecturer	Varchar (10)	FK references table lecturer	L031310166

Table 4.14: Table dailylog

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Dailylog_id	ID for daily log	Varchar (100)	PK	LOG001
Student_id	ID for student	Varchar (10)	FK references table student	B031310166
Dailylog_date	Date for this log	Date	Not Null	2016-05-10
Dailylog_log_date	Datetime for student record this log	Datetime	Current_timestamp	2016-05-26 16:21:29
Dailylog_status	Status for this log	Varchar (45)	Must be either “LOGGED” or “SICK_LEAVE”	LOGGED
Dailylog_title	Title for this log	Varchar (45)	Not Null	First Log Title

Dailylog_content	Content for this log	Longtext	Not Null	First Log Content
Daily_lecturer_comment	Comment from lecturer	Longtext		First Lecturer Comment
Daily_industry_comment	Comment from company employee	Longtext		First Industry Comment
Dailylog_img	Image uploaded for this log	Longblob		BLOB Object

Table 4 15: Table lecturer

Attribute	Description	Data Type	Primary Key (PK)/ Foreign Key (FK)/ Constraint	Sample Data
Lecturer_id	ID for lecturer	Varchar (15)	PK	L031310166
Lecturer_name	Name for lecturer	Varchar (45)	Not Null	Mas Aina
Lecturer_password	Password for lecturer	Varchar (8)	Not Null	123456
User_type	ID for user type	Varchar (30)	FK references table user_type	UT001

Table 4.16: Table session_tab

Attribute	Description	Data Type	Primary Key (PK)/	Sample Data

			Foreign Key (FK)/ Constraint	
Session_id	ID for session	Varchar (45)	PK	S2Y1516
Session_start_date	Start date for the session	Date	Not Null	2016-05-01
Sessioon_end_date	End date for the session	Date	Not Null	2016-09-30
Session_status	Status for the session	Varchar (45)	Not Null	ACTIVE

4.2.2.2 ERD Validation

Entity Relational Diagram (ERD) Validation is a process of validating ERD that is designed for the system. This process is to ensure that all data, entities and relationships between entities are valid and necessary for the success of the development of the project. This process can be done by using Structured Query Language (SQL) in stored procedure and triggers. Followings are the example of validating ERD.

i. To get list of students' transaction.

```
SELECT
  student.student_name,
  student.student_id,
  internship.internship_title,
  company_location.location_address,
  company.company_name,
  session_tab.session_id,
```

```
  malaysia_poscode.poscode_number,
  malaysia_city.city_name,
  malaysia_state.state_name
FROM student
  LEFT JOIN student_internship ON
student_internship.student_id = student.student_id
  LEFT JOIN internship ON internship.internship_id =
student_internship.internship_id
  LEFT JOIN session_tab ON session_tab.session_id =
internship.internship_session
  LEFT JOIN company_location ON
company_location.location_id =
internship.internship_location
  LEFT JOIN malaysia_city ON company_location.city =
malaysia_city.city_id
  LEFT JOIN malaysia_poscode ON malaysia_city.poscode =
malaysia_poscode.poscode_id
  LEFT JOIN malaysia_state ON malaysia_poscode.state =
malaysia_state.state_id
  LEFT JOIN company ON company.company_id =
company_location.company_id
WHERE student.lecturer_in_charge = 'L031310166'
ORDER BY session_tab.session_status DESC;
```

Query 4.1: Get student list query

Query 4.1 is to select list of students in charged by faculty advisor of 'L031310166'. The data retrieved is the students' names, students' matric number, internship title, address of the internship location, companies' name, session's id, postcode number, name of the city and name of the state of the internship location. Following Figure 4.3 is the transaction path that is required to achieve such transaction.

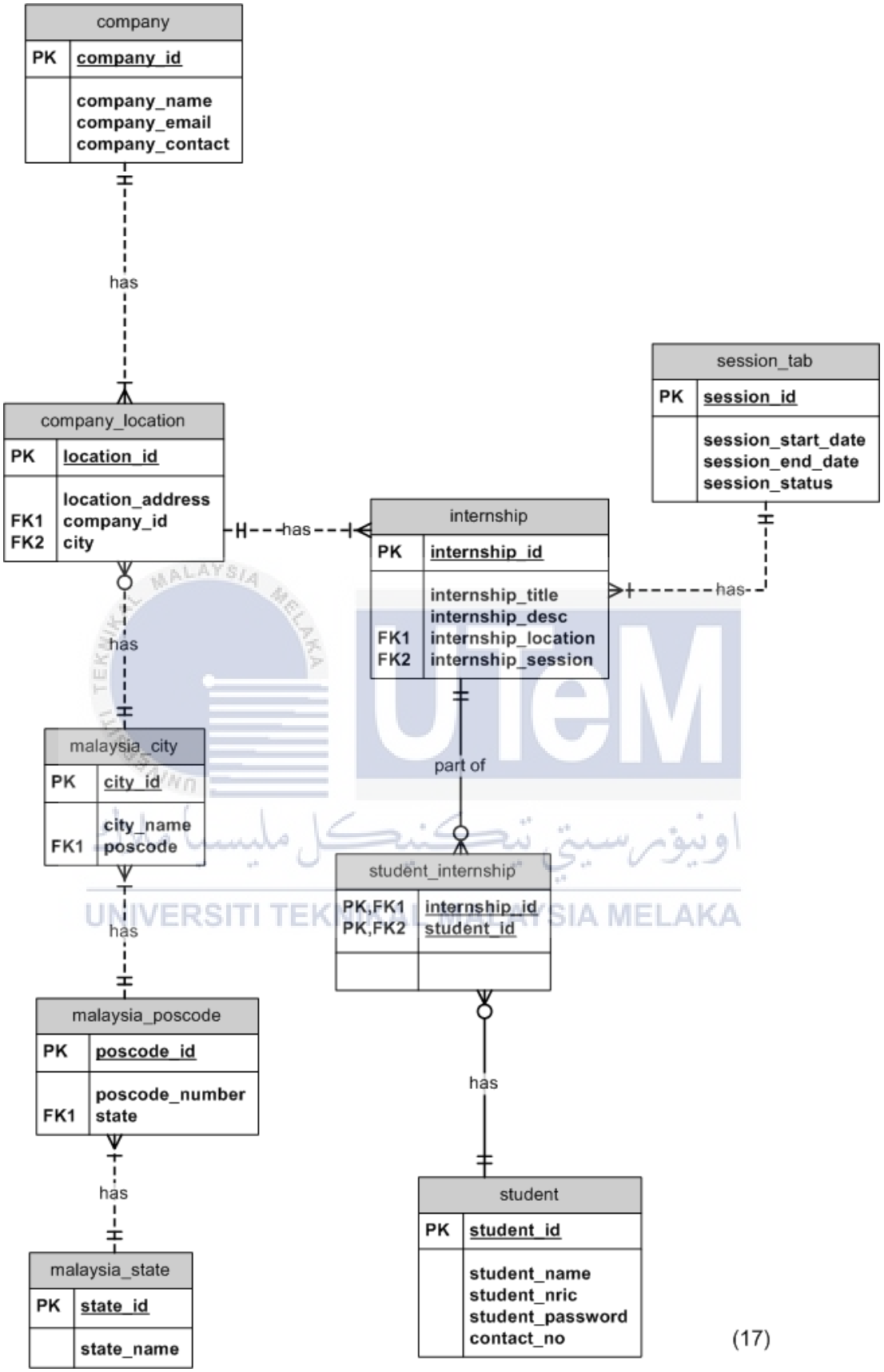

Figure 4.3: Get list of student transaction path (Faculty Advisor)

ii. To get company's weekend for calendar details

```
SELECT weekends.weekend_day
FROM weekends, company, company_weekend
WHERE company.company_id = company_weekend.company_id
      AND company_weekend.weekend_id =
weekends.weekend_id
      AND company.company_id = comp_id;
```

Query 4.2: Get company weekends query

Query 4.2 show the query to select weekends of company where student is having internship at. This is for the purpose to show weekends on the calendar User Interface (UI). The following Figure 4.4 shows the transaction path.

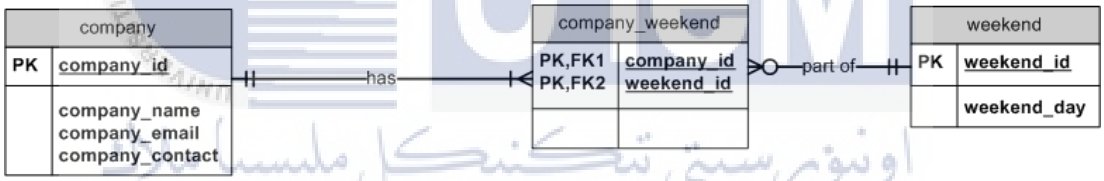

Figure 4.4: Get company's weekends transaction path

iii. To get user's login credentials

```
SELECT COUNT(*)
INTO student_count
FROM
  student
WHERE
  student_id = matricNo
  AND student_password = password_input;
SELECT COUNT(*)
INTO lecture_count
```

```
FROM
  lecturer
WHERE
  lecturer_id = matricNo
  AND lecturer_password = password_input;

SELECT COUNT(*)
INTO company_emp_count
FROM
  company_employee
WHERE
  employee_id = matricNo
  AND employee_password = password_input;

IF(student_count > 0) THEN
SELECT u.user_name
INTO usertype
FROM student s, user_type u
WHERE s.user_type = u.user_type_id
      AND s.student_id = matricNo AND s.student_password
= password_input;
ELSEIF (lecture_count > 0) THEN
SELECT u.user_name
INTO usertype
FROM lecturer l, user_type u
WHERE l.user_type = u.user_type_id
      AND     l.lecturer_id     =     matricNo     AND
l.lecturer_password = password_input;
ELSEIF (company_emp_count > 0) THEN
SELECT u.user_name
INTO usertype
FROM company_employee ce, user_type u
WHERE ce.user_type = u.user_type_id
      AND     ce.employee_id     =     matricNo     AND
ce.employee_password = password_input;
ELSE
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Username or
Password in incorrect';
END IF;
```

Query 4.3: User Login query

Query 4.3 shows the query for user login. This query requires the users input of credentials and search for respective table. Following Figure 4.5 shows the transaction path of this query.

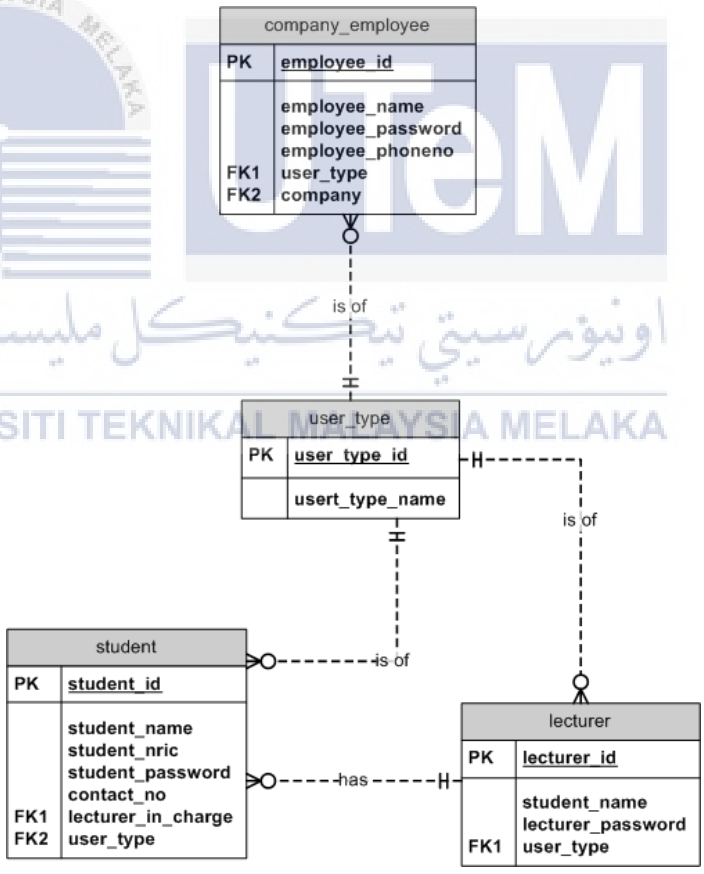

Figure 4.5: Get users' credentials transaction path

iv. To insert students daily log records

The main transaction of FTMK is the daily log records. In order to insert new records, not on leave records, following Query 4.4 is needed by students.

```
INSERT INTO dailylog (student_id, dailylog_date,
dailylog_title, dailylog_content, dailylog_img)
VALUES (stud_id, date_log, title, content, img);
```

Query 4.4: Insert daily log query

The following Figure 4.6 shows the transaction path for inserting daily log query.

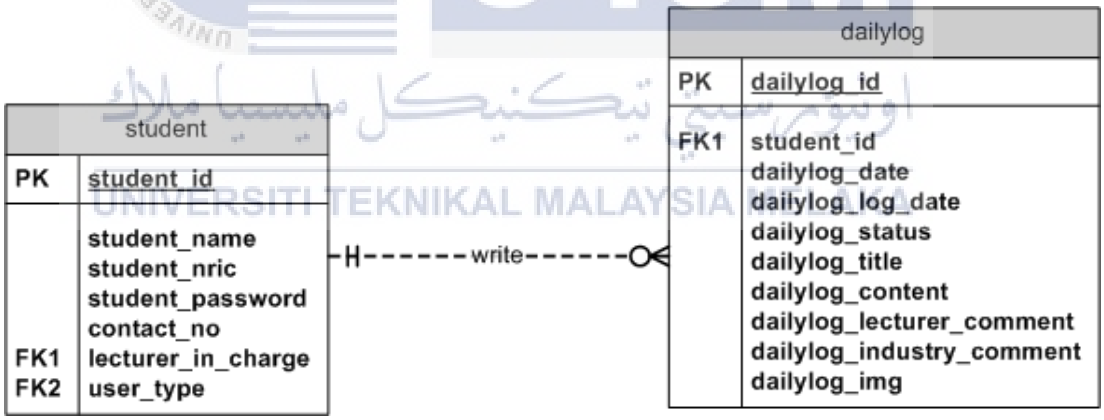

Figure 4.6: Insert daily log transaction path

As a conclusion, all the entities are being validated as per each transaction required. Thus, the ERD shown in Figure 4.2 is apparently correct and necessary for the example in Query 4.1, 4.2, 4.3 and 4.4 above.

4.2.2.3 Query Design

There are various types of query design will be used in the proposed system which is FTMK Internship Log Book System in the form of relational algebra and SQL statements. Below are the query designs:

i. Unary Operation (Selection)

Display the information from table dailylog where the dailylog_date is 2016-05-12 and student's ID is B031310166.

a. SQL Statement

```
SELECT *
FROM dailylog
WHERE dailylog_date = '2016-05-12' AND student_id
= 'B031310166';
```

b. Relational Algebra

$\Pi_{\text{dailylog\_date} = \text{'2016-05-12'} \wedge \text{student\_id} = \text{'B031310166'}}(\text{dailylog})$

ii. Unary Operation (Projection)

Retrieve dailylog_date, dailylog_date, dailylog_industry_comment, dailylog_lecturer_comment and dailylog_status from table dailylog of student ID of B031310166.

a. SQL Statement

```
SELECT
  dailylog_date,
  dailylog_industry_comment,
  dailylog_lecturer_comment,
```

```
  dailylog_status
FROM dailylog
WHERE student_id = 'B031310166';
```

b. Relational Algebra

Π dailylog_date, dailylog_industry_comment,dailylog_lecturer_comment, dailylog_status (σ student_id = 'B031310166') (dailylog);

iii. Aggregation Operation

To get the number of days missed to log since last log date until present date

a. SQL Statement

```
SELECT 5 * (DATEDIFF(now(), start_date) DIV 7) +
  MID('0123444401233334012222340111123400012345
001234550', 7 *
 WEEKDAY('2016-05-12') + WEEKDAY(now()) + 1, 1) -
  COUNT (*) FROM dailylog WHERE student_id =
'B031310166';
```


b. Relational Algebra

Π 5 * (DATEDIFF(now(),'2016-05-12') DIV 7) +
MID('0123444401233334012222340111123400012345001234550', 7 * WEEKDAY('2016-05-12') + WEEKDAY(now()) + 1, 1) - $G_{count(*)}$ (σ student_id = 'B031310166') (dailylog)

4.2.3 Physical Design

In physical design, discussion of selecting the correct Database Management System (DBMS), usage of stored procedures, triggers, etc., security of mechanism and database contingency of the FTMK Internship Log Book System will be carried out.

Firstly, talk about selecting the appropriate DBMS that will be used for developing this system. MySQL version 5.7.9, integrated with MySQL Community Server (GPL) and MySQL Workbench 6.3 CE, is used for this project. The main reason for this selection is because MySQL is an open source DBMS that is easily available on internet and can be simply installed on computer that comes along with server service provider such as XAMPP or even standalone server. Comparing to other DBMS, certain steps and procedures are required in order to enable the server to integrate with the DBMS. For example, to connect Oracle DBMS, server need to be configured in such a way that enable oracle string (oic8) and required Oracle Instant Client packages to do that, which required to download another package separately. By using MySQL, it is also suitable for this potential system with massive data sets as it is scalable and able to handle large amount of data especially media objects. For example, Facebook, one of the most famous social media applications that handles billions of users including media streaming, is currently using MySQL for video and image storage. FTMK Internship Log Book System has one of the feature that is to store images from students. This would come in handy for image storage and retrieval.

Onwards to the usage of stored procedures, triggers and other database objects, such as views and function. Stored procedures are widely used in the logical operations in data tier (4.2 Introductory preview to this chapter), no doubt that there are still some logical operations in the logical tier. Logical operations that involved in the systems are mainly basic CRUD operations and

checking condition if it is suitable for that operation. For example, getting a particular student's daily log records by using a stored procedure with parameters; get_log_students ('@studID'), @studID should be replaced with the student ID number. (Refer to Table 4.16) To handle large about of daily log records of each and every students, stored procedures is one of the most suitable method as stored procedures, "if used correctly, are stored programs that be used to can improve the performance, security, maintainability, and reliability of system applications (Guy Harrison *et al*., 2006)".

```
CREATE DEFINER =`root`@`localhost` PROCEDURE
`get_log_student`(studId VARCHAR(10))
  BEGIN
    SELECT
      dailylog_date,
      dailylog_status,
      dailylog_lecturer_comment,
      dailylog_industry_comment
    FROM dailylog
    WHERE student_id = studId;
  END
```

Query 4.5: Sample code of stored procedure

Triggers, "named program units that are executed in response to events that occur in the database (Arup Nanda *et al*., 2005)". In this project, most of the triggers will be used for creating Primary Key (PK) before inserting new data into tables. For example, before students start inserting new daily logs, a new PK would be created to represent the new records. This is to ensure that all data could be identified with a distinct key to represent the data and to maintain data integrity. For example, before inserting new log records by students, a new dailylog_id is created by database automatically. (Refer to Table 4.)

```
CREATE DEFINER =`root`@`localhost` TRIGGER
`utem_intern`.`dailylog_BEFORE_INSERT` BEFORE INSERT ON
`dailylog` FOR EACH ROW
  BEGIN
    INSERT INTO dailylog_seq VALUES (NULL);
    SET new.dailylog_id = concat('LOG',
LPAD(LAST_INSERT_ID(), 3, '0'));
    IF (new.dailylog_lecturer_comment IS NULL)
    THEN
      SET new.dailylog_lecturer_comment =
'NOT_COMMENTED';
      SET new.dailylog_industry_comment =
'NOT_COMMENTED';
    END IF;
  END
```

Query 4.6: Sample code of Trigger

4.3 Graphical User Interface (GUI) Design

A graphical user interface design defines the overall process of designing how a user will be able to interact with a system and the nature of the inputs that the system accepts and produces. It includes the screen displays, which provide navigation through the system, the screens and forms that capture data. In graphical user interface design consists of three parts which are navigation design, input design and output design.

4.3.1 Navigation Design

The navigation component of the interface design enables users to navigate through the system and perform actions, which available to the system A consistent and predictable set of navigation buttons give the user a sense of system's organization and make the logic and order of the system visually explicit. The following Figure 4.7 illustrates the navigation design and path for FTMK Internship Log Book System.

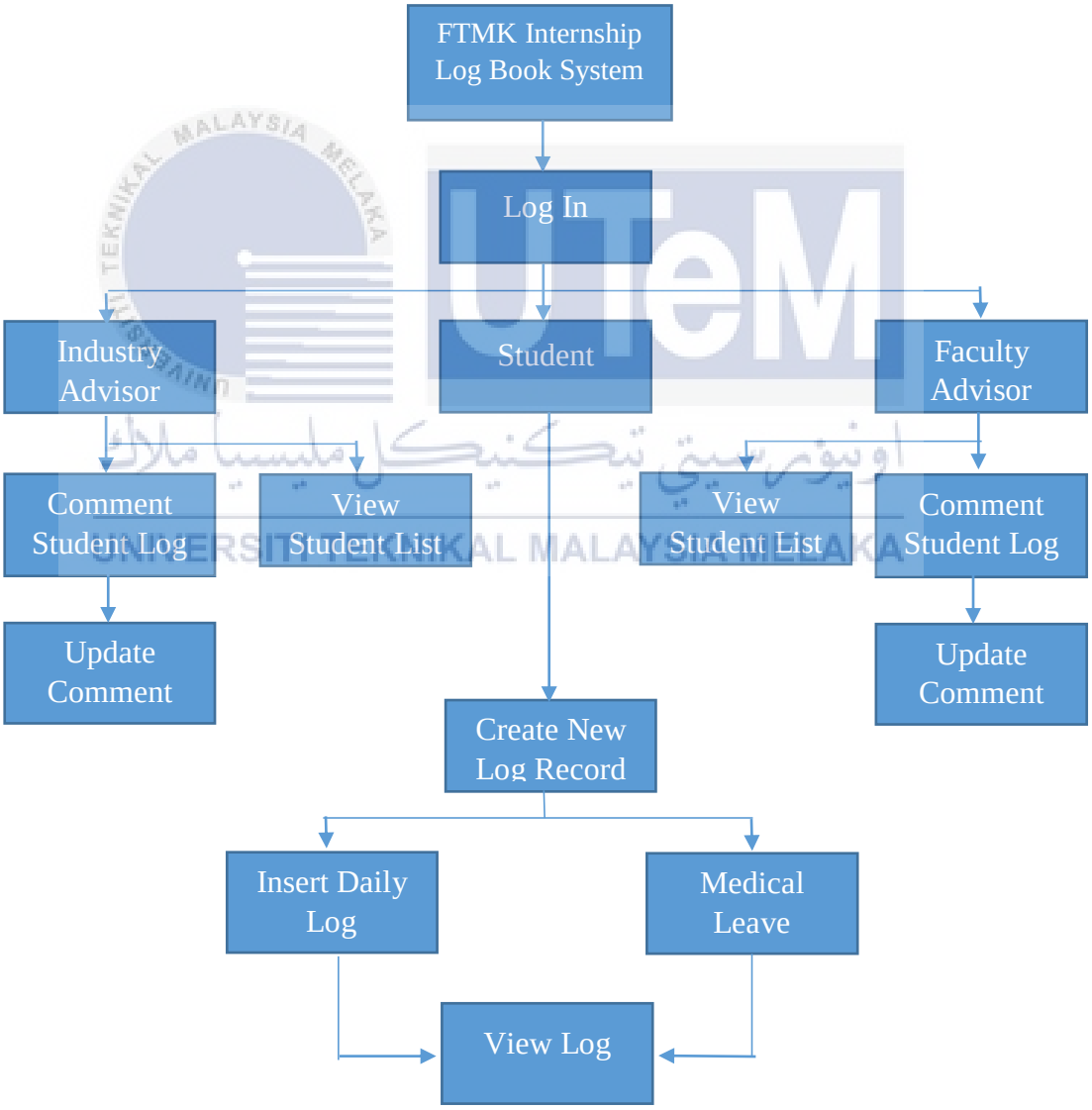

Figure 4.7 Navigation Flow of FTMK Internship Log Book System

4.3.2 Input Design

The input design focuses on the entry of data into the system by users in the form of structured and unstructured data. The screen and forms are designed and used to store information for the system for an action performed. Following Figure 4.3 to Figure 4.8 illustrates the input design.

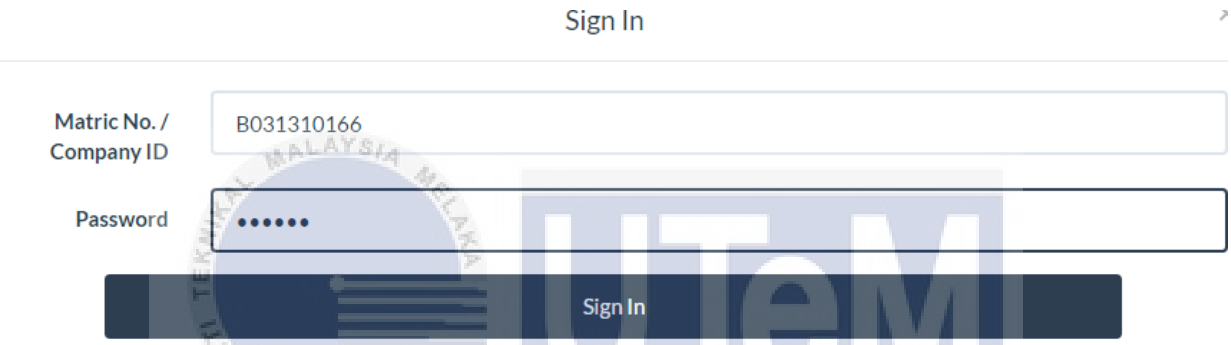

Figure 4.8: User Login Form

Figure 4.8 shows the input form for user to insert credentials in order to login into system.

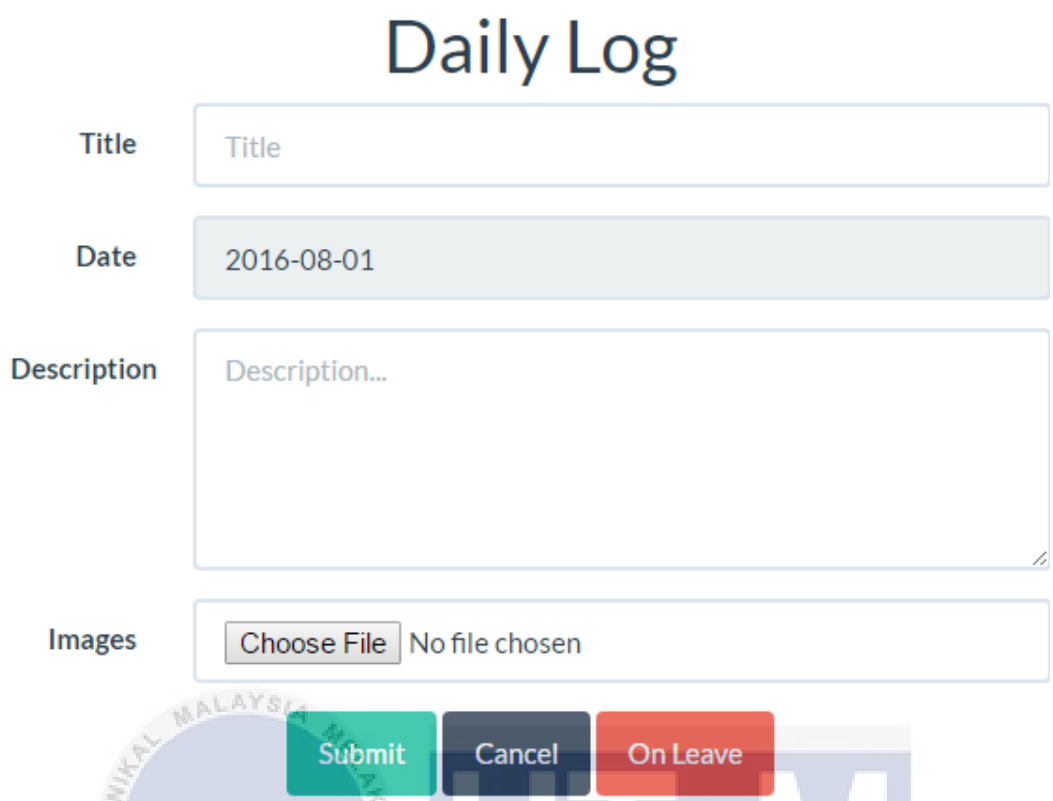

Figure 4.9: Student Create New Log Record

Figure 4.9 shows the input form for students to create new daily log records.

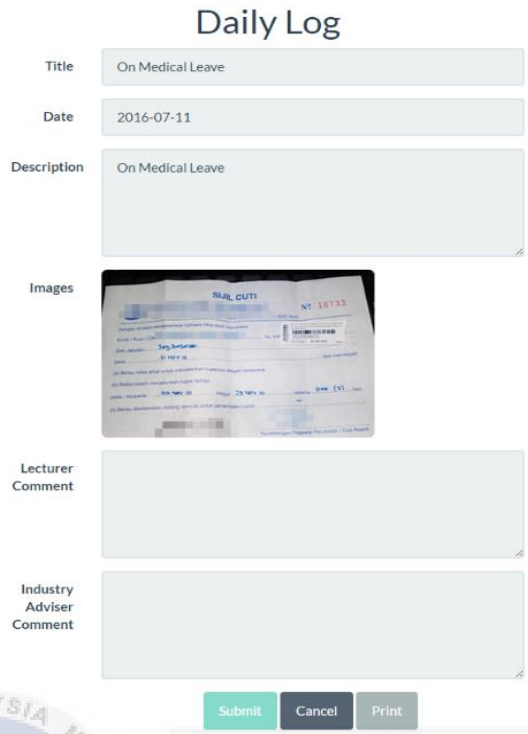

Figure 4.10: Student Create New Medical Leave Record

Figure 4.10 shows the result of creating medical leave record by clicking "On Leave" button.

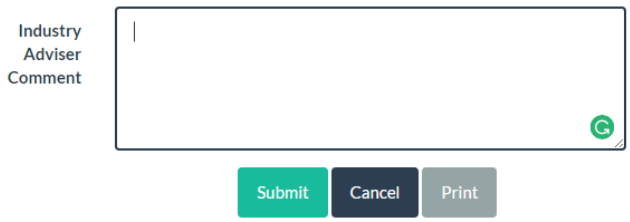

Figure 4.11: Industry Advisor Comment Student's Log

Figure 4.11 shows the form for industry advisor to create or update comments on studentss' log records.

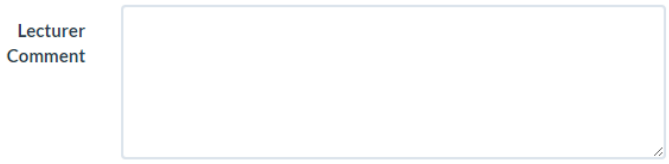

Figure 4.12: Faculty Advisor Comment Student's Log Record

Figure 4.12 shows the input form for faculty advisors to insert/update comments of students.

4.3.3 Output Design

The output design concerns on presenting retrieved information of the system on the screen or form. The output design in shown at Figure 4.8 to 4.14

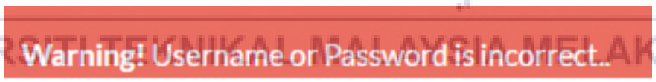

Figure 4.13: Error Message when User login failed

Figure 4.13 shows the error/ warning message that users will receive if login credentials are not match to the one in database.

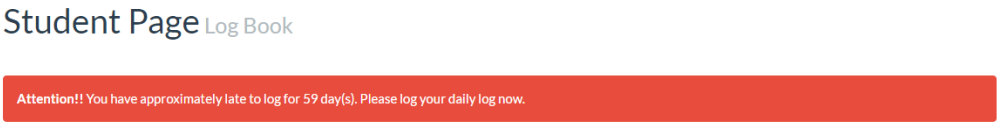

Figure 4.14: Student Main Page

Figure 4.14 illustrates the number of days students that haven not record any logs on students' main page.

Figure 4.15: Faculty Advisor Main Page that shows Student List

Figure 4.15 shows the list of students in charged by faculty advisor on the main page.

Figure 4.16: Industry Advisor Main Page that shows Student List

Figure 4.16 shows the list of students in charged by industry advisor on the main page.

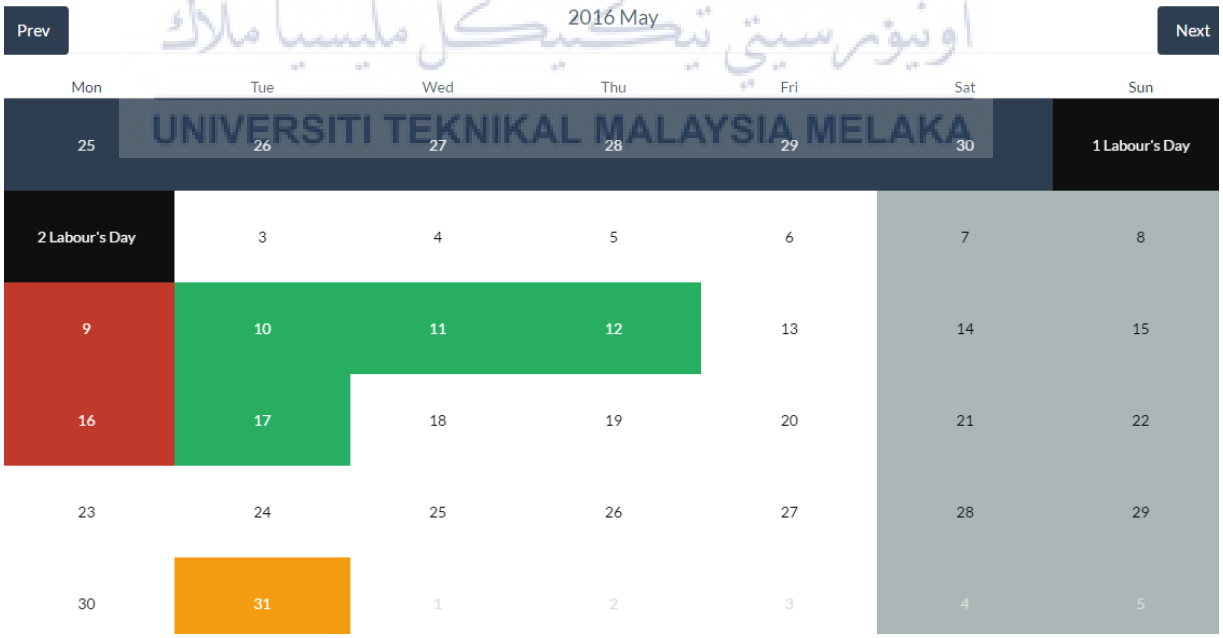

Figure 4.17: A calendar that shows student's internship logs

Figure 4.17 shows example of collections of log records by students. Advisors are able to view this particulars too.

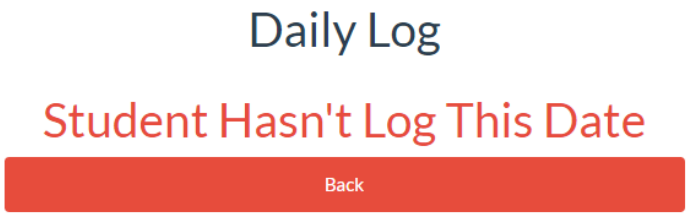

Figure 4.18: A warning to show advisors that student has not record log

Figure 4.18 shows the warning message to advisors when they click on a date that students have not record daily logs yet

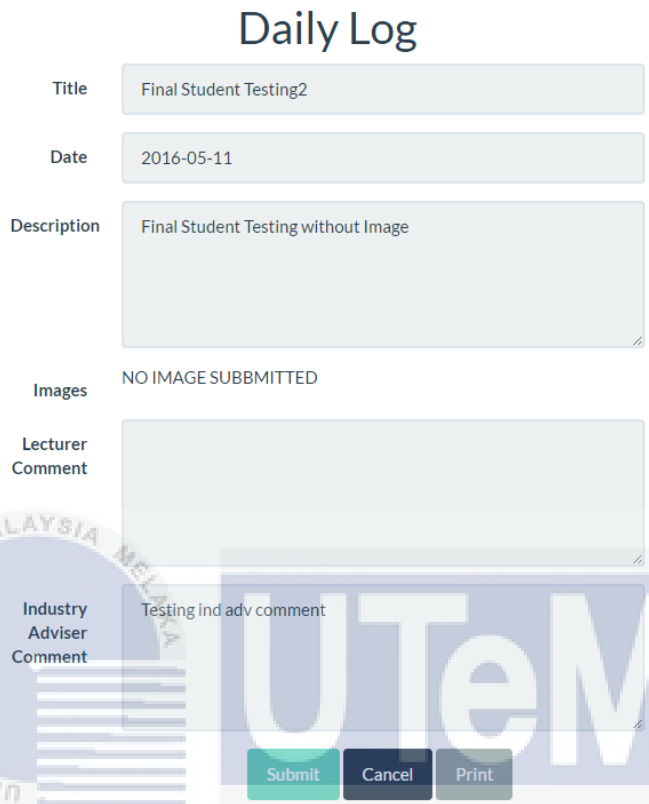

Figure 4.19: A page showing a particular date's log record

Figure 4.19 shows an example of daily log records.

4.4 Conclusion

This chapter is concluded with the knowledge on designing the whole FTMK Internship Log Book System in a more systematic way while fulfilling requirements (functional and non-functional requirements) and features that are resulted from Chapter III (Analysis). Besides that, the purpose of the design phase is to seek a solution of the problem specified by the requirement document in the analysis phase. This phase is the first of the step in change from problem domain to the solution

domain. The output of this phase is the design document. This design document is used later during the next other phase included implementation, testing and maintenance.

The next chapter is about the implementation phase will introduces the detail about how the system development environment setup included the installation step and database services. Besides, database implementation also will introduce in implementation phase to detail about Data Definition Language (DDL) in the chosen for all database on objects.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

This chapter will be discussing about the implementation phase of FTMK Internship Log Book System, where it will provide a detailed description of each and every tasks that are undertaken during the process.

The implementation of FTMK Internship Log Book System starts from system development environment setup, followed by implementation of database. The system development environment setup will be discussing regarding the tasks included installation step, assigning admin login, the step to starting the database services and the creation and objects of the database. Database implementation will focus on the present of Data Definition Language (DDL) and Data Control Language (DCL) in the chosen DBMS and the main process using the stored procedures and triggers. Besides, database implementation will also discuss the process of data loading

5.2 System Development Environment Setup

Development environment determines the speed of developing a system. This section will describe on how to configure the FTMK Internship Log Book System development environment in order for the system to be compiled and running it.

5.2.1 Installation Step

i. Windows 10 as Operating System

During the development of this system, Windows 10, Microsoft's latest and best windows operating system that ever made, is served as the platform for deployment for FTMK Internship Log Book System. Thus, all installation and configuration of software packages are done base on this version of operating system.

ii. Install and configure WAMP Server Provider

FTMK Internship Log Book system requires WAMP server provider. This server provider installation includes MySQL database. This would come in handy as one installation includes 2 services. After installing WAMP, a folder named "PSM", which signifies "Project Sarjana Muda", is created within the directory "C:\wamp64\www" and all the Hypertext Preprocessor (PHP) files of this system is stored inside. Besides that, there are subdirectories created to organize all the PHP files to ensure faster development and to maintain development principle of "Don't Repeat Yourself", (DRY) with a well-organized

environment. Besides that, this method would help to beautify the URLs.

iii. Install MySQL Workbench 6.2 CE

Installing MySQL Workbench is crucial as it contains user interface (UI) for easier and faster database development and progress. After installing the workbench, it will automatically be connected to MySQL database that comes within the WAMP that was previously installed. Thus, the foundation setup for developing FTMK Internship Log Book System has been done.

5.2.2 Assigning Admin Login

While implementing the system, configuration of the database and WAMP have been configured upon installation. Thus, administrator of the database the system will be using, in this case is utem_intern, will be assign. This user will be granted with the privileges as Database Administrator (DBA) to have full control over the database. For FTMK Internship Log Book System, the developer is assigned as the DBA and the responsibilities are as followed:

i. **Planning and development of the database**
ii. **Integrity and security of the database**
iii. **Database and backup recovery**
iv. **Designing conceptual, logical and physical design of the database**

5.2.3 Starting the Database Service

WAMP server provider has been a great help in starting up database service. This installation comes with the help of Guided User Interface (GUI) to reduce the hassle of starting up server and database services using command lines. By double clicking the WAMP executables, server and database server is booted up instantly.

5.2.4 Database Creation and Database Objects

In MySQL, CREATE DATABASE statement is used to create databases. This is the most fundamental step that needs to be completed before creating other database objects, including creating views, tables, stored procedures and functions, users and etc.

Database objects can be created swiftly by using Graphical User Interface (GUI) or by simply executing Structured Query Language (SQL) statements. Features and functionality of databases should be fundamentally observed and reviewed in order to create a better and smoother development process so that any yielding during the process can be executed quickly and accurately. The followings are the database objects created during the developments of FTMK Internship Log Book System:

i. **Database**
ii. **Tables**
iii. **Constraints**
iv. **Triggers**

v. **Stored Procedures**
vi. **Views**
vii. **Functions**

5.3 Database Implementation

In the database implementation phase, Database Definition Language (DDL), Database Control Language (DCL), triggers, procedures, and data loading will be explained in a very detailed manner in order to demonstrate the functionality of the FTMK Internship Log Book System

5.3.1 Data Definition Language

The Data Definition Danguage (DDL) is used to manipulate databases and respective database objects. DDL is used primarily by Database Administrator (DBA) during the development of databases. Following are the example of DDL that is used in FTMK Internship Log Book System.

5.3.1.1 Create Database

'CREATE DATABASE' command is used before creating any database objects. The name of the database is 'utem_intern'.

```
CREATE DATABASE IF NOT EXISTS `utem_intern`
```

5.3.1.2 Create Table and Constraint

'CREATE TABLE' commands are used for creating tables and its attributes, including defining the data types, field length, default values, constraints. Primary and foreign keys can be created by using this command.

i. Table company

```
CREATE TABLE `company` (
  `company_id`      VARCHAR(30)  NOT NULL DEFAULT '0',
  `company_name`    VARCHAR(100) NOT NULL,
  `company_contact` VARCHAR(10)  NOT NULL,
  `company_email` 0076archar(50) NOT NULL,
  PRIMARY KEY (`company_id`),
  UNIQUE KEY `company_name_UNIQUE` (`company_name`),
  UNIQUE KEY `company_contact_UNIQUE`
(`company_contact`),
  UNIQUE KEY `company_email_UNIQUE` (`company_email`)
);
```


ii. Table company_employee

```
CREATE TABLE `company_employee` (
  `employee_id`       VARCHAR(20) NOT NULL,
  `employee_name`     VARCHAR(45) NOT NULL,
  `employee_password` VARCHAR(45) NOT NULL,
  'employee_phoneno` varchar(45) NOT NULL,
  `company` VARCHAR (45
```

```
) NOT NULL,
  `user_type`          VARCHAR(45) NOT NULL,
  PRIMARY KEY (`employee_id`),
  KEY `employee_company_fk_idx` (`company`),
  KEY `employee_usertype_fk_idx` (`user_type`),

CONSTRAINT `employee_company_fk` FOREIGN KEY
(`company`) REFERENCES `company` (`company_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

iii. Table company_location

```
CREATE TABLE `company_location` (
  `location_id`      VARCHAR(10) NOT NULL,
  `location_address` VARCHAR(45) NOT NULL,
  `company_id`       VARCHAR(30) NOT NULL,
  `city`             VARCHAR(5)  NOT NULL,
  PRIMARY KEY (`location_id`),
  UNIQUE KEY `location_address_UNIQUE`
(`location_address`),
  KEY `location_company_fk_idx` (`company_id`),
  KEY `location_city_fk_idx` (`city`),
  CONSTRAINT `location_city_fk` FOREIGN KEY (`city`)
REFERENCES `malaysia_city` (`city_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `location_company_fk` FOREIGN KEY
(`company_id`) REFERENCES `company` (`company_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

iv. Table company_weekend

```
CREATE TABLE `company_weekend` (
  `company_id` VARCHAR(30) NOT NULL,
  `weekend_id` VARCHAR(3)  NOT NULL,
  PRIMARY KEY (`company_id`, `weekend_id`),
  KEY `weekend_id_fk_idx` (`weekend_id`),
  CONSTRAINT `company_id_fk` FOREIGN KEY (`company_id`)
REFERENCES `company` (`company_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `weekend_id_fk` FOREIGN KEY (`weekend_id`)
REFERENCES `weekends` (`weekend_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```


v. Table dailylog

```
CREATE TABLE `dailylog` (
  `dailylog_id`               VARCHAR(100) NOT NULL
DEFAULT '0',
  `student_id`                VARCHAR(10)  NOT NULL,
  `dailylog_date`             DATE         NOT NULL,
  `dailylog_log_date`         DATETIME     NOT NULL
DEFAULT CURRENT_TIMESTAMP,
  `dailylog_status`           VARCHAR(45)  NOT NULL
DEFAULT 'LOGGED',
  `dailylog_title`            VARCHAR(45)  NOT NULL,
  `dailylog_content`          LONGTEXT     NOT NULL,
  `dailylog_lecturer_comment` LONGTEXT,
  `dailylog_industry_comment` LONGTEXT,
```

```
  `dailylog_img`              LONGBLOB,
  PRIMARY KEY (`dailylog_id`),
  KEY `dailyLog_student_fk_idx` (`student_id`),
  CONSTRAINT `dailylog_student_fk` FOREIGN KEY
(`student_id`) REFERENCES `student` (`student_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

vi. Table internship

```
CREATE TABLE `internship` (
  `internship_id`       VARCHAR(15) NOT NULL DEFAULT
'0',
  `internship_title`    VARCHAR(45) NOT NULL,
  `internship_desc`     VARCHAR(45) NOT NULL,
  `internship_salary`   VARCHAR(45) NOT NULL,
  `internship_location` VARCHAR(10) NOT NULL,
  `internship_session`  VARCHAR(45) NOT NULL,
  `internship_pic`      VARCHAR(20)          DEFAULT
NULL,
  PRIMARY KEY (`internship_id`),
  KEY `intern_location_fk_idx` (`internship_location`),
  KEY `intern_session_fk_idx` (`internship_session`),
  KEY `intern_pic_fk_idx` (`internship_pic`),
  CONSTRAINT `intern_location_fk` FOREIGN KEY
(`internship_location`) REFERENCES `company_location`
(`location_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `intern_pic_fk` FOREIGN KEY
(`internship_pic`) REFERENCES `company_employee`
(`employee_id`)
```

```
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `intern_session_fk` FOREIGN KEY
(`internship_session`) REFERENCES `session_tab`
(`session_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

vii. Table lecturer

```
CREATE TABLE `lecturer` (
  `lecturer_id`       VARCHAR(15) NOT NULL,
  `lecturer_name`     VARCHAR(45) NOT NULL,
  `lecturer_password` LONGTEXT    NOT NULL,
  `user_type`         VARCHAR(45) NOT NULL DEFAULT
'UT001',
  PRIMARY KEY (`lecturer_id`),
  KEY `lec_userType_fk_idx` (`user_type`),
  CONSTRAINT `lec_userType_fk` FOREIGN KEY
(`user_type`) REFERENCES `user_type` (`user_type_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

viii. Table malaysia_city

```
CREATE TABLE `malaysia_city` (
  `city_id`   VARCHAR(5)   NOT NULL,
  `city_name` VARCHAR(100) NOT NULL,
  `poscode`   VARCHAR(5)   NOT NULL,
  PRIMARY KEY (`city_id`),
  KEY `city_poscode_fk_idx` (`poscode`),
  CONSTRAINT `city_poscode_fk` FOREIGN KEY (`poscode`)
REFERENCES `malaysia_poscode` (`poscode_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

ix. Table malaysia_poscode

```
CREATE TABLE `malaysia_poscode` (
  `poscode_id`     VARCHAR(5)  NOT NULL DEFAULT '0',
  `poscode_number` VARCHAR(45) NOT NULL,
  `state`          VARCHAR(5)  NOT NULL,
  PRIMARY KEY (`poscode_id`),
  UNIQUE KEY `poscode_number_UNIQUE`
(`poscode_number`),
  KEY `poscode_city_fk_idx` (`state`),
  CONSTRAINT `poscode_city_fk` FOREIGN KEY (`state`)
REFERENCES `malaysia_state` (`state_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

x. Table malaysia_state

```
CREATE TABLE `malaysia_state` (
  `state_id`   VARCHAR(5)  NOT NULL DEFAULT '0',
  `state_name` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`state_id`),
  UNIQUE KEY `state_name_UNIQUE` (`state_name`)
);
```

xi. Table session_tab

```
CREATE TABLE `session_tab` (
  `session_id`         VARCHAR(45) NOT NULL DEFAULT
'0',
  `session_start_date` DATE        NOT NULL,
  `session_end_date`   DATE        NOT NULL,
  `session_status`     VARCHAR(45) NOT NULL DEFAULT
'"ACTIVE"',
  PRIMARY KEY (`session_id`),
  UNIQUE KEY `session_start_date_UNIQUE`
(`session_start_date`),
  UNIQUE KEY `session_end_date_UNIQUE`
(`session_end_date`)
);
```


xii. Table student

```
CREATE TABLE `student` (
  `student_id`         VARCHAR(10)  NOT NULL,
  `student_name`       VARCHAR(100) NOT NULL,
  `student_nric`       VARCHAR(12)  NOT NULL,
```

```
  `contact_no`          VARCHAR(45)  NOT NULL,
  `student_password`    LONGTEXT     NOT NULL,
  `user_type`           VARCHAR(5)   NOT NULL DEFAULT
'UT002',
  `created_date`        DATETIME     NOT NULL DEFAULT
CURRENT_TIMESTAMP,
  `updated_date`        DATETIME     NOT NULL DEFAULT
CURRENT_TIMESTAMP,
  `lecturer_in_charge` VARCHAR(15)  NOT NULL,
  PRIMARY KEY (`student_id`),
  UNIQUE KEY `student_nric_UNIQUE` (`student_nric`),
  KEY `user_type_fk_idx` (`user_type`),
  KEY `student_lectuer_fk_idx` (`lecturer_in_charge`)
);
```

xiii. Table student_internship

```
CREATE TABLE `student_internship` (
  `student_id`    VARCHAR(10) NOT NULL,
  `internship_id` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`student_id`, `internship_id`),
  KEY `intern_id_fk_idx` (`internship_id`),
  CONSTRAINT `intern_id_fk` FOREIGN KEY
(`internship_id`) REFERENCES `internship`
(`internship_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `stud_id_fk` FOREIGN KEY (`student_id`)
REFERENCES `student` (`student_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

xiv. Table user_type

```
CREATE TABLE `user_type` (
  `user_type_id` VARCHAR(30) NOT NULL,
  `user_name`    VARCHAR(45) DEFAULT NULL,
  PRIMARY KEY (`user_type_id`)
);
```

xv. Table weekends

```
CREATE TABLE `weekends` (
  `weekend_id`  VARCHAR(3)  NOT NULL,
  `weekend_day` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`weekend_id`),
  UNIQUE KEY `weekend_day_UNIQUE` (`weekend_day`)
);
```


5.3.1.3 Create Sequence Table

a. Sequence ID for table company

```
CREATE TABLE `company_seq` (
  `int` INT(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`int`)
);
```

b. Sequence ID for table dailylog

```
CREATE TABLE `dailylog_seq` (
  `int` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`int`)
);
```

5.3.1.4 Trigger

Database triggers are executed automatically when there a data manipulation event happens such as Data Manipulation Language (DML). By using database triggers, it is guaranteed that specifically defined operation(s) will be performed before or after the main action is performed. Note that database triggers are fired each time the DML happens. Following is the example for trigger.

Description:

- When student insert new daily log record, new dailylog_id will be created.
- Faculty and industry advisors' comment section will be set as 'NOT_COMMENTED';

```
CREATE TRIGGER `dailylog_BEFORE_INSERT` BEFORE INSERT
ON `dailylog` FOR EACH ROW BEGIN
  INSERT INTO dailylog_seq VALUES (NULL);
  SET new.dailylog_id = concat('LOG',
LPAD(LAST_INSERT_ID(), 3, '0'));
  IF (new.dailylog_lecturer_comment IS NULL OR
new.dailylog_industry_comment IS NULL)
```

```
  THEN
    SET new.dailylog_lecturer_comment =
'NOT_COMMENTED';
    SET new.dailylog_industry_comment =
'NOT_COMMENTED';
  END IF;
END;
```

5.3.1.5 Stored Procedures

A stored procedure is a structured block of code that can accept parameter(s) and can be invoked. Stored procedure is used to perform a specific action. Stored procedure is able to improve system performances by reducing process time on server and promote usability and maintainability. The followings are example of stored procedures used in FTMK Internship Log Book System.

i. Select Statement Stored Procedure

Description: Display students' info and info of internship location in charged by faculty advisor.

`CREATE OR REPLACE PROCEDURE get_list_students(lect_id VARCHAR (30))`
`BEGIN`
`SELECT`
`  lecturer.lecturer_id,`
`  student.student_name,`
`  student.student_id,`
`  internship.internship_title,`
`  company_location.location_address,`
`  company.company_name,`

```
  session_tab.session_id
FROM lecturer
  LEFT JOIN student ON lecturer_id =
student.lecturer_in_charge
  LEFT JOIN student_internship ON
student_internship.student_id = student.student_id
  LEFT JOIN internship ON internship.internship_id =
student_internship.internship_id
  LEFT JOIN session_tab ON session_tab.session_id =
internship.internship_session
  LEFT JOIN company_location ON
company_location.location_id =
internship.internship_location
  LEFT JOIN company ON company.company_id =
company_location.company_id
  AND student.lecturer_in_charge = lect_id;
END;
```

Description: Stored procedure for user to login.

```
CREATE OR REPLACE PROCEDURE user_login(matricNo VARCHAR
(20), password_input VARCHAR (30))

BEGIN
DECLARE student_count INT;
DECLARE lecture_count INT;
DECLARE company_emp_count INT;

SELECT COUNT(*)
INTO student_count
FROM student
WHERE student_id = matricNo
      AND student_password = password_input;

SELECT COUNT(*)
```

```
INTO lecture_count
FROM lecturer
WHERE lecturer_id = matricNo
      AND lecturer_password = password_input;

SELECT COUNT(*)
INTO company_emp_count
FROM company_employee
WHERE employee_id = matricNo
      AND employee_password = password_input;

IF(student_count > 0) THEN
SELECT u.user_name
INTO usertype
FROM student s, user_type u
WHERE s.user_type = u.user_type_id
      AND s.student_id = matricNo AND
s.student_password = password_input;

ELSEIF (lecture_count > 0) THEN
SELECT u.user_name
INTO usertype
FROM lecturer l, user_type u
WHERE l.user_type = u.user_type_id
      AND l.lecturer_id = matricNo AND
l.lecturer_password = password_input;

ELSEIF (company_emp_count > 0) THEN
SELECT u.user_name
INTO usertype
FROM company_employee ce, user_type u
WHERE ce.user_type = u.user_type_id
      AND ce.employee_id = matricNo AND
ce.employee_password = password_input;

ELSE
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Username or
Password in incorrect';
     END IF;
END
```

ii. Insert Stored Procedure

Description: Insert daily log record by student.

```
CREATE OR REPLACE PROCEDURE dailylog_log_proc(stud_id
VARCHAR (30), date_log DATE, title VARCHAR (30),
content LONGTEXT, img LONGBLOB )

BEGIN
DECLARE start_date DATE;
DECLARE end_date DATE;

SELECT
  session_tab.session_start_date,
  session_tab.session_end_date
INTO start_date, end_date
FROM session_tab, internship, student_internship
WHERE student_internship.student_id = stud_id AND
student_internship.internship_id =
internship.internship_id AND
      internship.internship_session =
session_tab.session_id;

IF selected_date > now() OR selected_date > end_date OR
selected_date < start_date THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Selected Date is not in the range
of the session or/and cant be future date';
ELSE
```

```
INSERT INTO dailylog (student_id, dailylog_date,
dailylog_title, dailylog_content, dailylog_img)
VALUES (stud_id, date_log, title, content, img);

END
```

Description: Insert daily log type sick leave by student

```
CREATE OR REPLACE PROCEDURE
dailylog_sick_leave_proc(user_id VARCHAR, selected_date
DATE, title VARCHAR, log_content LONGTEXT, img LONGBLOB
)
BEGIN
DECLARE start_date DATE;
DECLARE end_date DATE;

SELECT
  session_tab.session_start_date,
  session_tab.session_end_date
INTO start_date, end_date
FROM session_tab, internship, student_internship
WHERE student_internship.student_id = user_id
      AND student_internship.internship_id =
internship.internship_id
      AND internship.internship_session =
session_tab.session_id;

IF selected_date > now() OR selected_date > end_date OR
selected_date < start_date THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Selected Date is not in the range
of the session or/and cant be future date';
ELSE
```

```
INSERT INTO dailylog (student_id, dailylog_status,
dailylog_date, dailylog_title, dailylog_content,
dailylog_img)
VALUES (user_id, 'SICK_LEAVE', selected_date, title,
log_content, img);
END IF;
END
```

iii. Update Stored Procedure

Description: Stored procedure for faculty advisor to comment students' daily log records

```
CREATE OR REPLACE insert_lect_comment(lect_comment
LONGTEXT, stud_id VARCHAR (12), log_date DATE )

BEGIN
UPDATE dailylog
SET dailylog_lecturer_comment = lect_comment
WHERE student_id = stud_id AND dailylog_date =
log_date;
END
```


Description: Stored procedure for industry advisor to comment students' daily log records

```
CREATE OR REPLACE insert_ind_comment(ind_comment
LONGTEXT, stud_id VARCHAR (12), log_date DATE )

BEGIN
UPDATE dailylog
SET dailylog_industry_comment = ind_comment
```

```
WHERE student_id = stud_id AND dailylog_date =
log_date;
END
```

5.3.2 Data Control Language

Data Control Language (DCL) is one of the component of in SQL that is used to create users and grant them privileges in order constrict and constraint their authorization of resource usage. For example, 'utem_intern' database can be used only by user 'webmaster' to use. The following is the example to create user and granting privileges.

```
Step 1: mysql.exe -u root
Step 2: Create user 'webmaster' identified by 'abc123';
Step 3: Grant all privileges on utem_intern.* to
'webmaster'@ 'localhost' with grant option;
Step 4: mysql.exe -u webmaster -p
Step 5: Enter password: abc123
```


5.3.3 Data Loading

The functional data of FTMK Internship Log Book System can be loaded in two different ways as:

i. The data can be entered using the FTMK Internship Log Book System User Interface (UI).
ii. The data can be entered using MySQL Workbench.

5.4 Conclusion

The implementation phase refers to the final process of moving the solution from development status to production status. The FTMK Internship Log Book System is using the WAMP Server as a server-side scripting language and MySQL as the storage to keep and save the required data. Besides, triggers and procedures are using to manage and maintain the flow of the system where it based on business logic.

For the next step is the testing phase, the FTMK Internship Log Book System is tested by the user to validate and verify the requirements and to find the error and debug it. The testing phase will be explained in Chapter 6. The testing phase involves test plan, test strategy, test design and test results and analysis.

CHAPTER VI

TESTING

6.1 Introduction

System testing, in generally speaking, is the phase of verification and validation of a system before handing in the complete software to customer or pushing it as a live system.

There are two main goals in the process of testing FTMK Internship Log Book System:

i. To demonstrate to end-users that FTMK Internship Log Book System has met its user requirements.

ii. To discover any faults and defects in the FTMK Internship Log Book System by testing the behavior of the system which includes any incorrect, undesirable or unconfirmed specifications or features.

System testing is one of the most important phase in software development lifecycle (SDLC). Activities that are included in this phase are test plan, test strategy, test design and test results and analysis.

6.2 Test Plan

There are a few submodules will be discussing in test plan which includes test organization, individual or organization who will be involved in the software testing; test environment, location and setup for the testing to be carried out; and test schedule, timeline of test for review, tracking and approval.

6.2.1 Test Organization

In the development and testing phase of FTMK Internship Log Book System, the test organization includes 3 users, which are student, faculty supervisor and industry supervisor. These users will be testing the system's functional and non-functional requirements as well as detecting any defects or faults that exist in the system. All users will be testing their testing based on their responsibilities that described in the table below:

Table 6.1: Users and Responsibilities of Involved Users

Users	Responsibilities
Student	• Testing the system based on test script given • Testing the students' module • Defect and fault detection
Faculty Advisor	• Testing the system based on test script given • Testing the faculty advisors' module • Defect and fault detection
Industry Advisor	• Testing the system based on test script given

	• Testing the industry advisors' module • Defect and fault detection

The information of each user who have tested the system can be referend in Appendix A.

6.2.2 Test Environment

Test environment in this system includes hardware requirements, software requirements and documentation, as stated below detailed information.

6.2.2.1 Hardware Requirements

Table 6.2: Hardware Requirements

Suggested Configuration	Minimum Configuration
i5 Intel Processor and above	i3 Intel Processor
10GB SSD and above	5GB HDD
Monitor	Any Monitor
Keyboard and Mouse	Any Keyboard and Mouse
Network Card	Any Network Card

6.2.2.2 Software Requirements

Table 6.3: Software Requirement

Item	Description
Microsoft Windows 10	Operating System/ Platform
Google Chrome	Web Browser
WAMP Server Provider	Web Server

6.2.2.3 Documentation

Each and every test that users will be performing, depending on their respective responsibilities, in each module are documented. The outcomes of tests performed by each user will be explained in the section 6.5 (Test Results and Analysis) in a detailed manner.

6.2.3 Test Schedule

Test schedule describes the time line of the test by tracking the test revision and approval.

Table 6.4: FTMK Internship Log Book System Test Schedule

Task	Start Date	End Date
1. Unit Testing	8 MAY 2016	17 MAY 2016

2. Integration Testing	18 MAY 2016	31 MAY 2016
3. Acceptance	16 AUG 2016	19 AUG 2016

6.3 Test Strategy

Test strategy explains the strategy for users to test out the systems in order to ensure that the system is full-fledge able to be used and to find defects in an organized and orderly mannered. There are a few classes of test included, which will be explain on Section 6.3.1.

6.3.1 Classes of Tests

Below are the descriptions of test classes that are being conducted on FTMK Internship Log Book System:

i. Accessibility Testing

This is to test if the users are able to view or access their respective and related data.

ii. Graphical User Interface (GUI) Testing

The aim of this test is to test if the GUI used in the system suites general choices and tastes as required in the non-functional requirements.

iii. Security Testing

This test basically is to check if data are shown to authenticated and authorized users correctly.

iv. Usability Testing

Application flow, user experience and usability of using the system are tested in this test to ensure users are able get hold and understand the system well and easily.

6.4 Test Design

There are two primary test designs that are involved in testing FTMK Internship Log Book System, which are test description and test data. Each test is based on user type; student, faculty advisor and industry advisor

6.4.1 Test Description

i. Login Test Description

Table 6.5: Login Test Description

Purpose of Test	To ensure that authorized user can login to the system successfully.
Test ID	A001

a. Student

Table 6.6: Login Test Script (Student)

Test Case ID	Test Cases	Expected Result
A001_001	Login button is clicked with all field is empty	System will show alert box showing specify field cannot be empty
A001_002	Login button is clicked with either matric no or password field is empty	System will show alert box showing specify field cannot be empty
A001_003	Login button is clicked with enter wrong either matric no or password	Error message indicate that user enter wrong username and password
A001_004	Login button is clicked with enter correct matric no and password	Successful login to the system

b. Faculty Advisor

Table 6.7: Login Test Script (Faculty Advisor)

Test Case ID	Test Cases	Expected Result
A001_005	Login button is clicked with all field is empty	System will show alert box showing specify field cannot be empty
A001_006	Login button is clicked with either lecturer id or password field is empty	System will show alert box showing specify field cannot be empty

A001_007	Login button is clicked with enter wrong either lecturer id or password	Error message indicate that user enter wrong username and password
A001_008	Login button is clicked with enter correct lecturer id and password	Successful login to the system

c. Industry Advisor

Table 6.8: Login Test Script (Industry Advisor)

Test Case ID	Test Cases	Expected Result
A001_009	Login button is clicked with all field is empty	System will show alert box showing specify field cannot be empty
A001_010	Login button is clicked with either employee id or password field is empty	System will show alert box showing specify field cannot be empty
A001_011	Login button is clicked with enter wrong either employee id or password	Error message indicate that user enter wrong username and password
A001_012	Login button is clicked with enter correct employee id and password	Successful login to the system

ii. Inserting Log Record Test Description

Table 6.9: Inserting Log Record Test Description

Purpose of Test	To ensure that student is able to insert new log record successfully.
Test ID	A002

a. Student

Table 6.10: Inserting Log Record Test Script (Student)

Test Case ID	Test Cases	Expected Result
A002_001	Submit button is clicked with all field is empty	System will show alert box showing specify field cannot be empty
A002_002	Submit button is clicked with either title or content field is empty	System will show alert box showing specify field cannot be empty
A002_003	Submit button is clicked with enter title and content	Successful inserting new log record
A002_004	Submit button is clicked with no image upload	Successful inserting new log record
A002_005	Submit button is clicked with no non-image upload	Warning Message will be shown regarding the type of uploaded file
A002_006	Submit button is clicked with image size more than 500KB upload	Warning Message will be shown regarding the size of uploaded file must not more than 500KB

` A002_007	Submit button is clicked with image size less than 500KB upload	Successful inserting new log record

iii. Inserting On Leave Log Record Test Description

Table 6.11: Inserting On Leave Log Record Test Description

Purpose of Test	To ensure that student is able to insert new on leave log record successfully.
Test ID	A003

a. Student

Table 6.12: Inserting On Leave Log Record Test Script (Student)

Test Case ID	Test Cases	Expected Result
A003_001	On Leave button is clicked with all field is empty	System will show alert box showing picture of medical leave or check must be submitted
A003_002	On Leave button is clicked with sick leave image uploaded with less than 500KB	Successfully inserting new on leave log record
A003_003	On Leave button is clicked with non-image type uploaded	Warning Message will be shown regarding the type of uploaded file
A003_004	Submit button is clicked with image size more than 500KB upload	Warning Message will be shown regarding the size of uploaded image

iv. Inserting Faculty/ Industry Advisor Comment Test Description

Table 6.13: Inserting Faculty/ Industry Advisor Comment Test Description

Purpose of Test	To ensure that faculty and industry advisor is able to comment student's log records successfully
Test ID	A004

a. Faculty Advisor

Table 6.14: Inserting Faculty Advisor Comment

Test Case ID	Test Cases	Expected Result
A004_001	Clicked submit button with empty comment	Warning Message is shown to ensure no empty comment is submitted
A004_002	Clicked submit button with non-empty comment	Comment successfully inserted

b. Industry Advisor

Table 6.15: Inserting Industry Advisor Comment

Test Case ID	Test Cases	Expected Result
A004_003	Clicked submit button with empty comment	Warning Message is shown to ensure no empty comment is submitted
A004_004	Clicked submit button with non-empty comment	Comment successfully inserted

v. Updating Faculty/ Industry Advisor Comment Test Description

Table 6.16: Updating Faculty/ Industry Advisor Comment Test Description

Purpose of Test	To ensure that faculty and industry advisor is able to update their comment on student's log records successfully
Test ID	A005

a. Faculty Advisor

Table 6.17: Updating Faculty Advisor Comment Test Description

Test Case ID	Test Cases	Expected Result
A005_001	Clicked submit button with empty comment	Warning Message is shown to ensure no empty comment is submitted

A005_002	Clicked submit button with non-empty comment	Comment successfully updated

b. Industry Advisor

Table 6.18: Updating Industry Advisor Comment Test Description

Test Case ID	Test Cases	Expected Result
A005_003	Clicked submit button with empty comment	Warning Message is shown to ensure no empty comment is submitted
A005_004	Clicked submit button with non-empty comment	Comment successfully updated

6.4.2 Test Data

In FTMK Internship Log Book System, test data are used to make confirmation with test plan in order to reach expected results. These confirmations are done by testing, validate and verify the software behavior. Due to high sensitivity of the data, these test cases, tested with dummy date, are preserved and used only for testing with no commercial value at all.

6.4.3 Integration Testing

Integration testing is to test the aggregates of each unit perform accurately together. In this section, prove of integration of calendar management, student management and record logs management, as stated in Chapter III, Analysis. Following Figures 6.1 to 6.4 are the example for this integration testing.

Figure 6.1: Integration Image I

Figure 6.2: Integration Image II

Figure 6.1. show that total date that student did not log and Figure 6.2 shows the log indicator in calendar of student. In this testing, date of 4th will be chosen to test the integration testing.

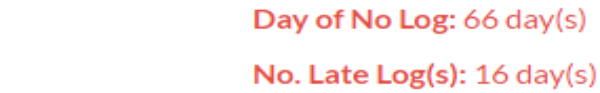

Figure 6.3: Integration Image III

Following Figure 6.3 is showing the total days of student did not log. This number should be the same as Figure 6.1 when student log into their respective index page.

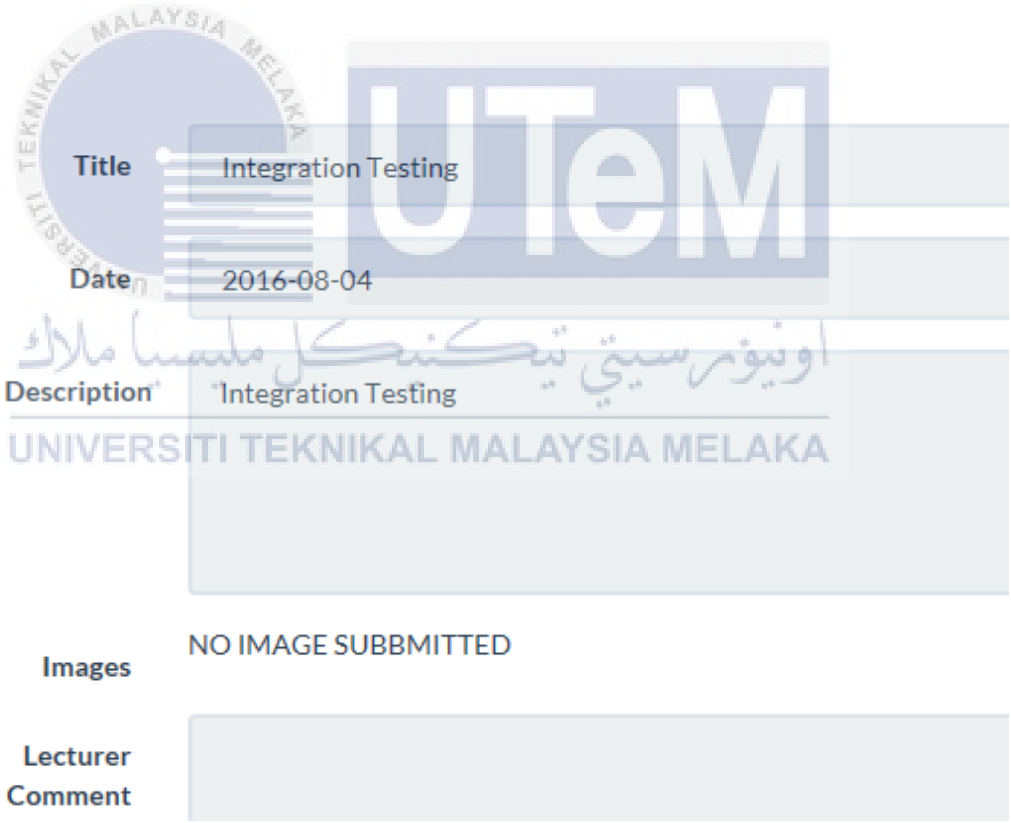

Figure 6.4: Integration Image IV

Figure 6.4 is showing student has recorded a log on the date of 4th. After recording, number of no logs recorded should be decreased and the calendar should show indicator on the selected date too.

Figure 6.5: Integration Image V

Figure 6.5 shows the logged but uncommented indicator on the calendar to show that none of the advisors has commented on this date's log record.

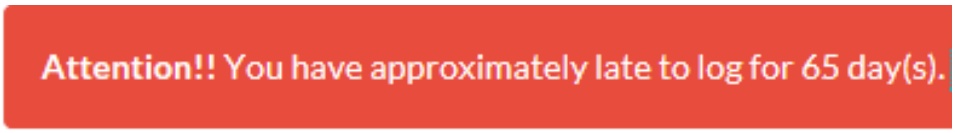

Figure 6.6: Integration Image VI

Figure 6.6 shows the number has been reduced since there is one new date student has recorded.

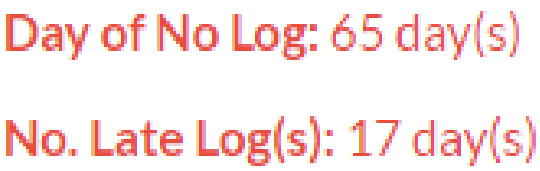

Figure 6.7: Integration Image VII

Figure 6.7 shows that new total number of no log records of student on advisors' index page on Student Management Module.

This concludes that each module is able to integrate with one another to prove that the integration between units is working and successfully passed the integration test.

6.4.4 Acceptance Testing

In acceptance testing, functional requirements, based on Chapter III Analysis, have been fulfilled. Following are the detailed explanation of the this testing phase.

6.4.4.1 Functional Requirements

i. User Authentication

In this requirement, users are required to login to be able to use the system. This is proven when student is trying to access own page without logging in. When user is logout, they are also required to login to be able to use the system

ii. Calendar Management

For fulling this requirement, it has been tested with data inside the database. This is tested when users are accessing the calendar module in the system. Figure 6.8 to 6.10 shows that this system has fulfilled the requirements of sowing indication and status of daily logs, range of internship, weekends of companies, public holidays, and date control.

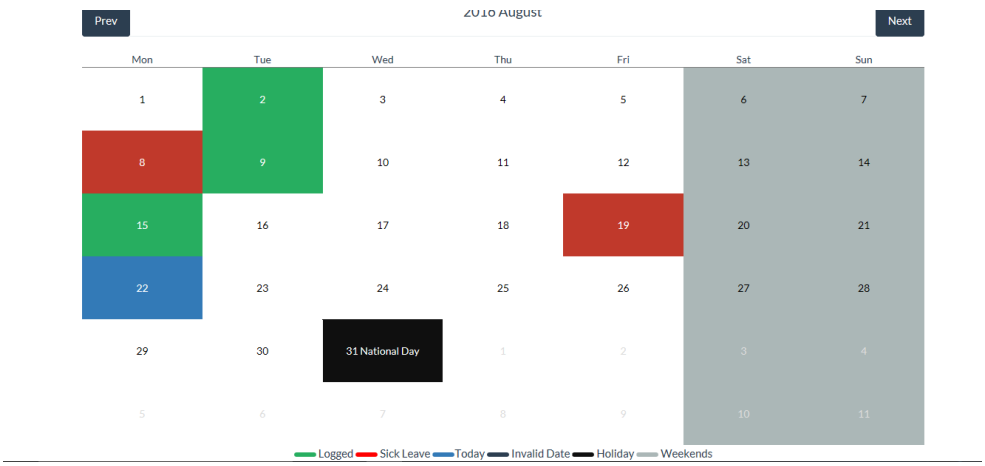

Figure 6.8: Calendar I

Figure 6.8 shows that the calendar is shown to users with legend and indicators which fulfilled one of the requirements of Calendar Management. The grey sections shows the weekends of company that student having internship at. The calendar also shows public holidays.

Figure 6.9: Calendar II

Figure 6.9 shows that calendar consists of range of the internship period.

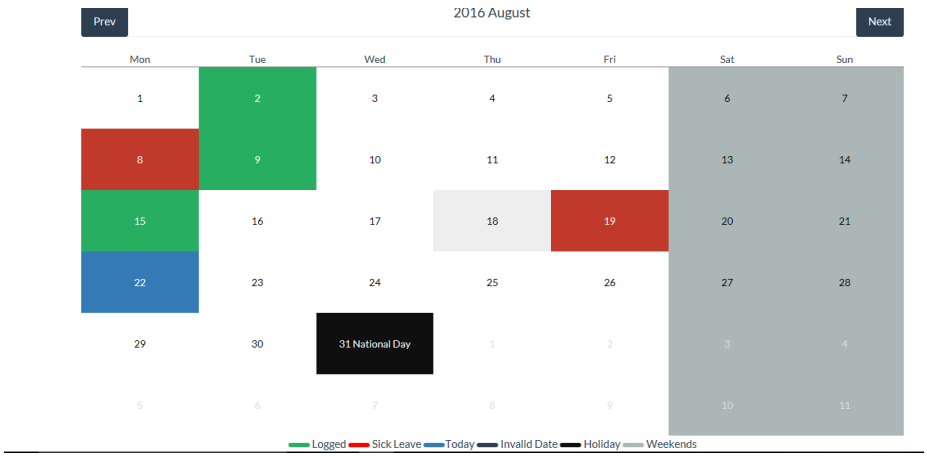

Figure 6.10: Calendar III

Figure 6.10 shows that the calendar is able to control date of accessing by users. If the date is available, the pointed date will float.

Therefore, acceptance testing of calendar management has been successfully fulfilled.

iii. Record Logs Management

In this phase, recording of logs by student has been tested by inserting new daily log and on leave records to fulfilled the functional requirements. Following Figure 6.9 to 6.10 shows the test by student.

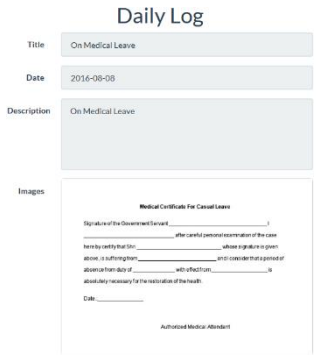

Figure 6.11: Log Record I

Figure 6.11 shows that student has recorded an on leave records on the system. This function can only completed if student submited an image of medial leave as a prove.

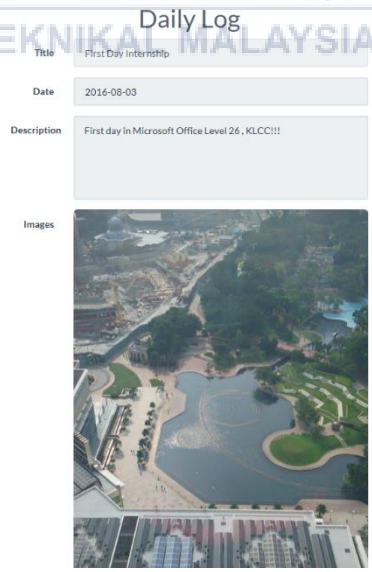

Figure 6.12: Log Record II

Figure 6.12 shows students who have successfully recorded a daily log. This figure also show proves that an image is able to be submitted through the system. As a conclusion. Record Logs Management has be proven fulfilled.

iv. Review Logs Management

In this phase, log review by advisors is tested. Following Figure 6.13 shows that advisors are able to comment student's log records. This is proven by comparing Figure 6.13 with Figure 6.5 for the difference indicator on date 3rd.

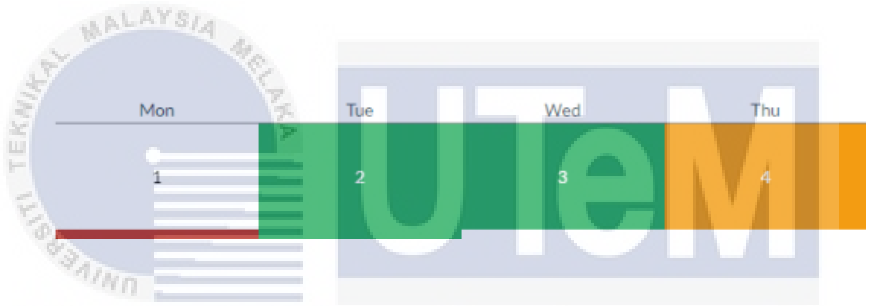

Figure 6.13 Review Logs I

Figure 6.14 Review Logs II

Thus, reviewing students log records by advisors are proven functional in this test.

v. Student Management

In this test, student management module has been proven fulfilled by showing students that are in charged by supervisors. Figure 6.15 and 6.16 shows list of students of faculty advisor and industry advisor respectively. There is a difference of student in the list, showing that the system is showing the related students in charged by advisors.

Figure 6.15: Student Management I

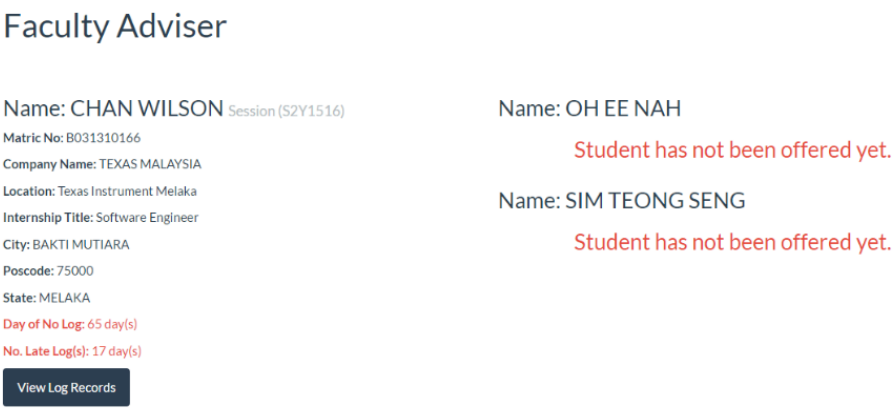

Figure 6.16: Student Management II

vi. Generate Documents

In this test, users are able to print out daily log as paper-based records. Figure 6.17 shows the document that can be printed by users through this system.

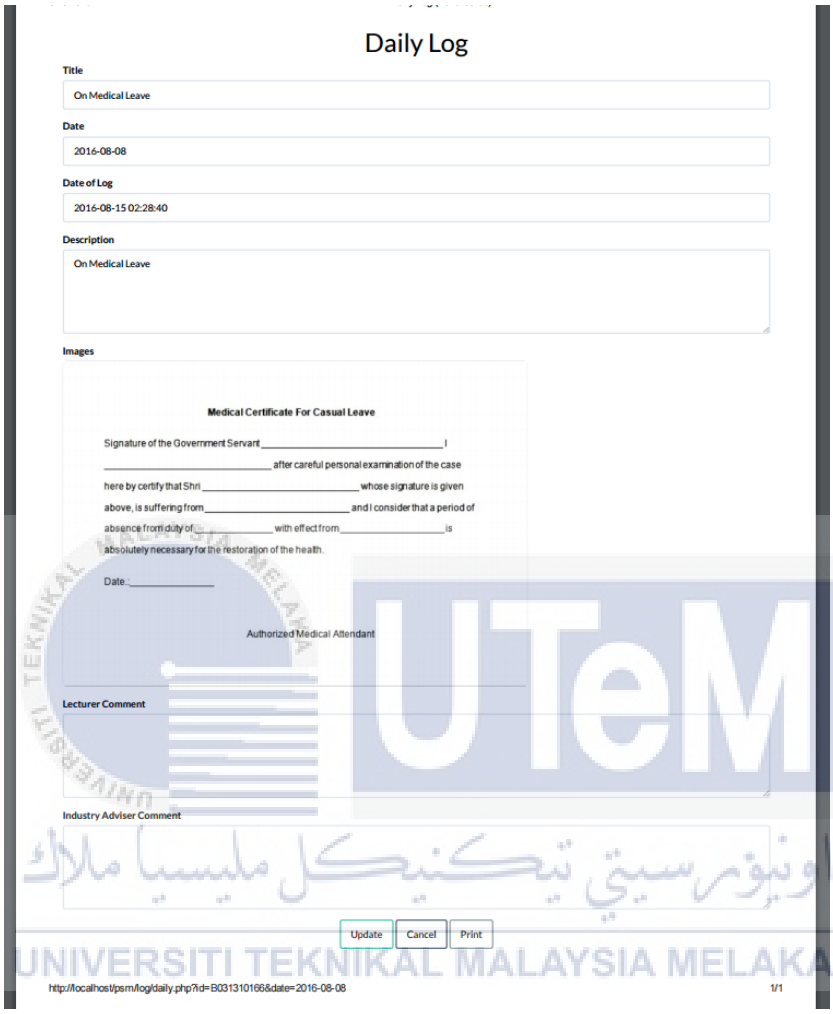

Figure 6.17 Generate Document I

6.5 Test Results and Analysis

Test results are the results after using FTMK Internship Log Book System in various possible test cases with temporary data. In this result, each test case that has been tested and the result are as shown below table:

Table 6.19: Test Result

Test Case Identification	Tester Identification	Result (Passed/ Failed)
A001 – Login Test Case		
A001_001	OK	Passed
A001_002	OK	Passed
A001_003	OK	Passed
A001_004	OK	Passed
A001_005	OK	Passed
A001_006	OK	Passed
A001_007	OK	Passed
A001_008	OK	Passed
A001_009	OK	Passed
A001_010	OK	Passed
A001_011	OK	Passed
A001_012	OK	Passed
A002 – Creating New Log Record Test Case		
A002_001	OK	Passed
A002_002	OK	Passed
A002_003	OK	Passed
A002_004	OK	Passed
A002_005	OK	Passed

A002_006	OK	Passed
A002_007	OK	Passed
A003 – Creating New On Leave Log Record Test Case		
A003_001	OK	Passed
A003_002	OK	Passed
A003_003	OK	Passed
A003_004	OK	Passed
A004 – Inserting Faculty/ Industry Comment Test Case		
A004_001	OK	Passed
A004_002	OK	Passed
A004_003	OK	Passed
A004_004	OK	Passed
A005 – Updating Faculty/ Industry Comment Test Case		
A005_001	OK	Passed
A005_002	OK	Passed
A005_003	OK	Passed
A005_004	OK	Passed

Based on Table 6.19, it is proven that the system has fulfilled the objectives that are stated in Chapter 1.

6.6 Conclusion

System testing is the one of the most important phase in Software Development Lifecycle (SDLC). Testing is a vigorous activity of analyze and gauge a function or feature of a program or system in order to reach its user requirements. All the sub section of testing phase had been explained in detail in this chapter. The test form, test data and the results also attached here. Throughout this phase, FTMK Internship Log Book System has been further improved and bug fixing in order to reach users' satisfactions and produce usable and functional system.

CHAPTER VII

CONCLUSION

7.1 Introduction

This chapter will be discussing about the overall conclusion for FTMK Internship Log Book System, which will consist of strength and weaknesses of the system. Besides that, propositions on improvements and contributions to this project will be defined and elaborate in this chapter as well.

7.2 Observation on Weaknesses and Strengths

Every system has its own strengths and weaknesses, including FTMK Internship Log Book System. FTMK Internship Log Book System is developed based on the needs and requirements of student, industry advisor and faculty advisor. This is a web-based system that is designed for these three users. As mentioned earlier, some of the strengths and weaknesses of the system have been summarized as following.

7.2.1 Strengths

vii. Human Error Reduction

As this is a web-based system, most of the human errors, such as spelling errors and illegible handwritings will greatly reduce. Comparing to the older system, spelling errors and illegible handwritings have always been the norm in recording log books, especially students who wrote it last minute. Thus, this system will be able to reduce such human errors and increase the significant of this log book system to faculty.

viii. Convenience

A web-based system is able to used and reached out to others who are far away and remain contacted. FTMK Internship Log Book System has the ability to let users to use the system conveniently, anytime and anywhere. This gives advantages to the all three users the conveniences to management the log book, as per the internship needs. Advisors who are far away from students and unable to reach out the physical log book is a disastrous moment for them as they would require students to send in photos for them to check. This is not convenient at all for all users. Thus, FTMK Internship Log Book System allows advisors to view students' log books remotely and easily to process the log book management. This will reduce the hassle of using third party applications to manage the log records as there may loss or missing of data during the process. All data are stored in one single database and users are all access them, provided that they are authorized and authenticated users.

ix. Efficiency Improvement

Comparing to the older system, log book management has always been taking tremendous time to write and check. For students, writing is not fond as much as to typing. Recording logs by hand is very consuming. In addition, illegible handwritings and spelling errors will cause more time for faculty and industry advisors to understand and comprehend the content of students. This may also cause dissatisfaction to advisors as they may feel frustrated during the process. This system is able to eliminate these factors. By typing on a modern web browser, these two factors is no longer an issue as all wordings are constant and even spelling errors are able to be detected and letting students to know that. Plus, free browser plugin for English Language is available to be installed in order to improve their language. Besides that, advisors are able to view and remark students' log records comfortably through web browser. Comparing the process of snapping pictures of the records and send them via mobile application or email with cause efficiency to be reduced. Fast and easy are the main factors for this system development in order to reduce time consumption and the system has that ability.

v. Usability

As much as a new system to be introduced, there are not much to learn how to use the system as the system can be easily be used by the students, faculty advisors and industry advisors. The system is developed based on user requirements and experiences in managing log books. Thus, the users are able to use the system remarkably easily.

7.2.2 Weaknesses

i. Lack of Reporting Features

This system has one of the document feature to print out students' daily log records of a single day. The system has no features for printing all records of students but instead printing everyday record manually.

ii. Testing with dummy data

This system is tested with dummy data and as prove of concept. Due to some confidential data and issues, the system tested users are based on the dummy data. This is one of the most important step to really test and detect any defects or fault existed in the system.

7.3 Propositions for Improvement

As mentioned earlier in the upper section of this chapter, the system requires improvements to ensure this project can be truly implemented in order for students and advisors to use. Following are the propositions for improving the system to a better state and use.

7.3.1 Enhancing Reporting Features

The system should consist of better reporting features allowing advisors to have a better overview of the log records of students. This system should allow advisors to view and print log records of student easily and instantly, in order for physical records.

7.3.2 Testing with Real Data

Before implementation on a faculty's server for usage, it is suggested that FTMK Internship Log Book System is tested on the server with real data to test and check the true performances of the system before letting students and advisors to use it.

7.4 Contribution

As mentioned at earlier chapter, FTMK Internship Log Book System is designed and developed based on 3 type of users, which are student, faculty advisor and industry advisor.

i. Student

Student is one of the main users to use this system. This system is able help them in managing their log book and remotely as well. The system is very useful for them as there are factors for them not able to get hold of the physical book and record logs daily. But instead, using this system, those factors will be eliminated.

ii. Faculty Advisor

Faculty advisor is able to view, manage and comment students log records daily and remotely without requiring the presence of the physical book or photos sending through third party applications. Faculty advisors are able to accomplish their basic role in log book management easily and remotely.

iii. Industry Advisor

Same goes to industry advisors who has the similar role with faculty advisor in managing the log records. Industry advisor is able to view, manage and comment students log records daily and remotely too. Comparing to older system, industry advisors are to require the book in order to do so. Industry advisors are able to accomplish their basic role in log book management easily and remotely.

7.5 Conclusion

Finally, for the conclusion of this chapter and as well as the rest of the other chapters, it is hereby FTMK Internship Log Book System has successfully meet the objectives of the project. Both functional and non-functional requirements of the project have been fulfilled.

In a nutshell, this project has been successfully finished in partial fulfilment of the requirements for the Bachelor of Computer Science (Database Management).

REFERENCES AND BIBLIOGRAPHY

1. Gustavo, A., Fabio, C., Harumi K., Vijay M. (2004); "Web Services: Concepts, Architecture and Applications" 1st Ed. Germany: Springer
2. Guy, H., Steven, F, (2006); "MySQL Stored Procedure Programming", 1st Ed. USA: O'Reilly Media
3. Arup, N., Steven, F, (2005); "Oracle PL/SQL for DBAs", 1st Ed. USA: O'Reilly Media

APPENDIX A

Questionnaires

Final Year Project 2016

FTMK INTERNSHIP LOG BOOK SYSTEM

The purpose of this questionnaire is to evaluate the overall experience and status of the log book system. This questionnaire is made for the purpose of gathering and recording the information and feedback testers.

For each question, please tick the boxes that best match your answer.

1. Gender:
   - ☐ Male
   - ☐ Female

2. Role:
   - ☐ Student
   - ☐ Faculty Advisor
   - ☐ Industry Advisor

3. Guided User Interface (GUI)

   I. Overall GUI Design:

   Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

   II. Calendar GUI:

   Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

   III. Log Book GUI:

   Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

4. Usability

I. Calendar:

Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

II. Calendar Indicator (Legend):

Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

III. Log Book:

Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

IV. Main Page:

Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

V. Overall Usability:

Very poor ☐ Poor ☐ Average ☐ Good ☐ Very Good ☐

5. Overall Experience

I. Would use the system to handle log book management:

Very Unlikely ☐ Unlikely ☐ Unsure ☐ Likely ☐ Very Likely ☐