LEAVE MANAGEMENT SYSTEM

MOHD FAREEZUL NAIM BIN RAHMAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: LEAVE MANAGEMENT SYSTEM (LMS)

SESI PENGAJIAN: 2015/2016

Saya MOHD FAREEZUL NAIM BIN RAHMAN

mengaku membenarkan tesis (PSM) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis 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)

________________________ (TANDATANGAN PENYELIA)

Alamat tetap: No 16, Jln Seksyen 3/25,
Tmn Kajang Utama, 43000
Kajang, Selangor

En.Yahya Bin Ibrahim

Tarikh: 24/8/2016

Tarikh: 25/8/2016.

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

LEAVE MANAGEMENT SYSTEM

MOHD FAREEZUL NAIM BIN RAHMAN

This report is submitted in partial fulfilment of the requirement 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

LEAVE MANAGEEMENT SYSTEM

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

STUDENT : ______________________ Date : 24/8/2016

(MOHD FAREEZUL NAIM BIN RAHMAN)

I hereby that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Database Management) With Honors.

SUPERVISOR : ______________________ Date : 25/8/2016.

(EN YAHYA BIN IBRAHIM)

DEDICATION

To my beloved parents,
my supervisor and lectures
and also to all my friends.

ACKNOWLEDGEMENTS

First of all, I would like to thank En. Yahya Bin Ibrahim because of his kindness to accept me as one of the students under his supervision. Not forget to mention I would like to thank him because of his brilliant idea, comment, guidance and assistant that really help me in completing the system. I also want to thank my parents because of their understanding and support that give me strength in doing this system. Appreciation also goes to my friend that always help me by giving their opinion and help when needed. Last but not least, thanks to all that have been involved during the development of this system.

ABSTRACT

Leave Management System is a web-based database system. It was developed to replace the existing paper based system used by student to apply leave. Leave Management System have many feature that will not only help student, but also will ease and reduce the staff workload. As for student, this system will save their time and work when applying leave. Student only need to open the site and fill the required information. The application will be sent to the admin through the system. And for the admin section, admin can manage student leave application more easily, just click the update button and finish. If admin accept student application, an email will be sent to the student saying that leave application have been accepted and an email also will be sent to lecturer that teach the student on the leave day to notify them about student absent. Analysis also can be made just by entering the desired information and it will be presented in form of pie chart, bar graph and table. This system will use Oracle 11g as the database and also Google Chrome as the default browser. It was develop based on three-tier architecture and agile methodology in order to give user full satisfaction and deliver a complete system to them.

TABLE OF CONTENTS

CHAPTER II PROJECT METHODOLOGY AND PLANNING

CHAPTER III ANALYSIS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Project Background

Every people need some leave from what they are doing right now. Leave can be categorized according to the person job. Different job might have different leave. For example, student, a student can apply for emergency leave, medical leave, license test and family business. Can you imagine a person working every day without any leave even when him or her is sick? How will that particular person life be? How much quality work can be produce by that person? All the question show that how important leave can be in a person life.

According to George, Robbin (2013), not taking leave isn't good for your health and it's only a matter of time before something goes wrong with your health. Lack of leave can lead to stress, tiredness and cardiovascular disease. Furthermore, studies have shown that if people don't take regular vacations, they are eight times more likely to suffer from depression and have a 50% higher chance of heart disease. Student also included in this term. Student also human that need some rest. Rest here can be in term of emergency, medical leave, license test or even family business or vacation. Student need some leave to maintain their brain creativity and make release them form the study stress.

But the problem is, how the student wants to apply their leave? Student particularly university student, facing problem when applying their leave because it involves many work and take time to process. Many universities only have the computerized system for the staff to apply leave but for the student, they need to manually take the leave application from the administrative office and follow the steps and regulations. Same goes Universiti Teknikal Malaysia Melaka, currently they are still using the old and paper based system for the student to apply leave. This is not tally with the motto of the university itself which is always a pioneer and always ahead. That’s where the idea of implementing this project comes from.

Nowadays, the student uses the manual system when applying leave. By using the old system, student need to go to the administrative office and take the leave application form. Next, student need to fill in the form and meet the lecturer for the subject that due on the day student want to apply the leave. Student need to get their permission as well as their signature as an approval sign for student to not attending the class on that day. After that, student need to meet head of department to take his or her signature. The application form is then need to be sent back to administrative office for approval before the student can take their leave. Not forget to mention student also need to make photocopies of the application form and give it to all his or her lecturer as a prove. All this process mostly will take about 3 days to complete. It is very time consuming and involving many unnecessary work.

Meanwhile from the admin side, it is hard to keep track student leave because of in a faculty, there are many student and the staff that supposed to take care all the student welfare is only about 5 to 10 people. The staff will need to manually key in everything into the system. This where data loss tends to happen because when human doing all the job and suddenly they get tired, the quality and concentration will be reduced. They will do it as fast as they can, as long as the job done without concerning about the quality of their job.

Besides, when using manual paper based system, the statistic of student leave is very hard and time consuming to do. Statistic here mean record of how many days a student has applied for leave, type of leave that student always applied and those kind of analysis that need to be include in faculty record.

Leave Management System (LMS) is a computerized and systematic system that can help the staff as well as the admin to keep track, store and manage all the leave application made by student in the faculty. The system will ease the student when applying leave and also the staff when monitoring student leave. Through the system, all activities that refers to student leave can be automatically and systematically. This will give satisfaction not only to student but also admin and staff.

As a conclusion, Leave Management System (LMS) will gives the user best new experience with user-friendly as well as interactive interfaces to support student when applying leave and assist staff while monitoring student leave. It helps to reduce time taken to apply for leave as student may access the system 24/7 anywhere and anytime according to their available time. Activities done will be recorded and statistics is generated automatically as a references for the faculty and staff record without any careless mistakes or data loss occurred.

1.2 Problem Statements

The leave management system developed to ease the user which is the student, staff and admin to manage the leave application. Some problem has been identified as the guideline on what should be include in the system to overcome existing problem. The first problem is hard to keep track of how many time student has applied for leave. If there is no computerized record, there is no mean to keep track on every student leave as there is many students in the faculty. Later, more students will be registered. How staff could manage the data manipulation with only depends on paper forms and how they are able to maintain the data to make sure there is no data loss and careless mistakes occur throughout the implementation of manual method?

Next, slow processing time. If paper based system is still used, sometimes it takes days before the admin can grant the leave application, it because the admin can simply say he or she just receive the application of leave because there is no evidence indicating that the letter has arrive earlier. It also time consuming because of too many work need to be done just to apply for leave.

Last but not least the problem that arise when using the old system is the loss of data. It a normal problem when using the paper based system. Sometimes the paper might be thrown away and causing loss of record. The data that we loss might be crucial data that we need as references in the future. If there is some incident, the loss of data is not recoverable because there is no backup and recovery method if using paper based system.

1.3 Objective

According to problem statements mentioned before, some objectives are figured out as a guidelines and target of what the system implemented should achieved.in order The objectives of this system are listed in the next page:

i. **To replace the paper based system with computerized, reliable and convenience system.**

The paper based system should not be used anymore as it not reliable and can be compromised. Leave Management System (LMS) helps staff to monitor student leave. All the data about student leave will be stored and managed through this system. All activities such as student apply, view status, accept or reject leave and leave analysis can be done using the system. Hence, no paper-based forms involve in these activities and data loss tendency could be reduced.

ii. **To increase or to speed up the process time to apply leave.**

Leave Management System (LMS) can be access 24/7 anywhere and anytime. This reduce the task and time taken to apply leave. When request made by student, the request will directly send to the admin and admin will get notification about it. This help to make sure that the admin is notified with the request and can respond to it faster.

iii. **To generate a report and analysis in order to keep track on the student leave.**

Leave Management System (LMS) can make analysis of student leave and generate report automatically. Analysis made will be presented to the user using graph, ordered list and pie chart. This report is reliable enough to be used as references in the future.

1.4 Scope

The scopes of the Leave Management System (LMS) will be focused on two major points of view which are firstly focused on users and secondly focused on the system itself. Focused on user is based on the range of users that will use this system internally and externally which is the student and the admin. Meanwhile, the focused on systems are divided into several modules that are related the system that will be developed. The scopes are:

i. User Scope

There are two main of users will use this system internally and externally. The internal users are the student who will used the system to apply for leave. Student also can view the status of their leave application whether it accepted or rejected by the admin. The external users are the admin itself who act as administrator for this system. The admin will use this system to process the student leave application updated the status whether accept or reject the application, view all list of student application, generate report and statistics based on each student, type of leave and report by month for each leave type.

ii. System Scope

There are several modules involve in Leave Management System (LMS) system scope which are described as below:

a. Login Module

The objective of this module is to manage the student and admin login. This module is important because the system need to differentiate between the student and the admin. The purpose is to give different approach to different user. Different user will have different menu. As a student, student can only view student menu and only have privileges to use the student menu. Same goes to the admin, admin only can use admin menu in this system. To use this module, the user need to enter their matric number or staff id and password. Login is an important part for Leave Management System (LMS) to avoid illegal user from accessing the system.

b. Leave Application Module

The objective of this module is for the student to apply for leave. This module will be used to record all information needed to apply the leave. The information is the start date and the end date for the leave, reason why they apply it and who apply it. In the database side, the system will automatically calculate the number of day the student applies for leave which does not include Saturday and Sunday as it is a weekend. In this module student also can view their application status and print the application that have been accepted by the admin.

c. Notification Module

The objective is for the admin to process the student leave application. Admin will view all leave application that is not process yet and update the status. If the admin accepts the application, two email will be sent. The first email is for the student to notify them that their leave application is accepted. The second email will be sent of the lecturer that teach the student on the day them apply for leave to notify the lecturer that the student will be absent on the particular day. This will use the student timetable as the platform to know what subject and who teach the student on that day.

d. Analysis module

This module provides an analysis for the staff or admin to view. Analysis made is based on the number of student taking leave per month for a year, student versus type of leave and also type of leave student always apply. The analysis is presented in term of pie chart, table and also bar graph.

1.5 Project Significance

The main purpose if developing this system is to help the student to have a better management while applying a leave. Using this system, the data or information is secured because all the information will be save into the database system. Student also does not need to waste time to meet all the lecturer to get their signature anymore. Moreover, this system will help the administrator to decrease the use of papers and files to keep the data. All the data inserted will automatically be saved into the database and it is easier for admin to manage it. Admin also can process each student leave application faster and able to generate report and statistic more easily.

1.6 Expected Output

The system concentrates to meet the outcomes where all the manual paper-based forms are disposed. Moving towards the Green Technology City, reducing the usage of paper is one of the way showed in supporting the campaign held by government. Computerized system where it can be controlled only by the fingertips is much better way to be used in this modern technology nowadays.

Manual system which can be view as traditional system should not be used anymore because the number of student will always be increased from year to year. If we still depend on traditional system, its cannot be deny that we will faced problem in the future. Normally, manual system in managing a lot of items could lead to data loss. By implementing Leave Management System (LMS), monitoring and managing can be done more easily and more effectively. Data can be stored safely in the database and it can be retrieved upon request by the user through this system.

Besides, it is expected that the time consumed in applying leave would be reduced considerably. This would be a great opportunity not only for student but also for the staff as their work load can be reduced.

1.7 Conclusion

Chapter 1 gives the brief introduction of Leave Management System (LMS) where it is generally known that existing system which is the traditional system is still using paper-based method. It gives a method to solve student and staff problem, that is computerized system that can support all student applying leave activities. Primarily, this system is planned to transform the manual method to more systematic computerized system. The objectives of the newly develop system is to overcome large amount of student leave data that will grow in number from year to year. The system contains several modules that will be use by two users: student and staff.

In the next chapter Project Methodology and Planning will be discussed. Type of methodology that the system used will be shown on the next chapter. This chapter will be used as guide in completing the next chapter.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

In the previous chapter, the introduction to Leave Management System (LMS) are discussed. The overview of the system has been presented clearly which will act as the guideline in completion of this chapter. In this chapter, the planning on how the system will be completed will be discussed. It is important to define the methodology to be used in order to avoid failure while conducting the system implementation to achieve targeted objectives. For Leave Management System (LMS), the Database Development Life Cycle (DBLC) will be used is Agile method.

2.2 Project Methodology

As mentioned earlier, DBLC applied in developing Leave Management System (LMS) is Agile method. Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile methods break the product into small incremental builds. These builds are provided in iterations. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders. In this term, the customer and the stakeholder are the supervisor who will supervise the system development and add requirements where needed.

Agile development model each sprint has phases of requirements, design, development and testing. In the development phase, development team concentrates on new features to be developed and unit testing around it. In Database Life Cycle (DBLC), there are several processes need to be done in order to develop Leave Management System (LMS).

Figure 2.1 below shows the phases involved in Agile method and flow of them where it consists of four major phases:

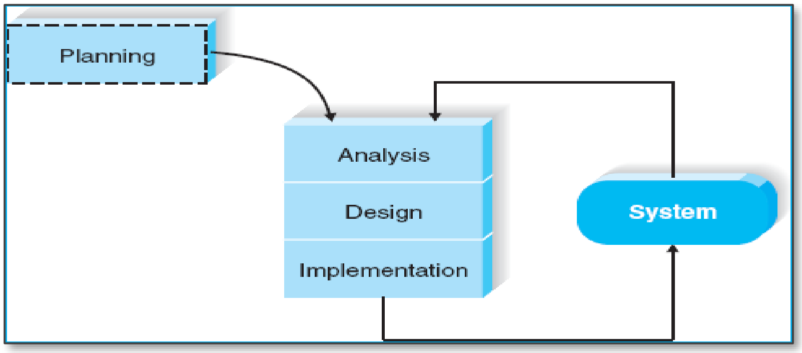


Figure 2.1: Agile Method

Agile methodology will be adopted for the development of Leave Management System (LMS) because it will give the advantage web-based-system and encourage the development of large, free standing systems that are capable of providing benefit rapidly. Agile method quickly delivers a working product and is considered a very realistic development approach. The model produces ongoing releases, each with small, incremental changes from the previous release. At each iteration, the product is tested. This help to detect defect and correct it before moving to other requirements.

Agile method is one of the methodologies that concentrate on producing working code. The purpose of choosing the Agile method is because of there will be consequences that will having changes will be implementing the system. With Agile method, defect can be found and corrected over several iterations. The advantage of using agile method is that the development process can be done in a short time rather than others methodologies, has good schedule visibility and produce a high reliable system. Functionality of the system also can be developed rapidly and demonstrated to the supervisor so that if there are any changes that the supervisor wants to suggest, the developer can change its easily. In simple word, its gives flexibility to developers to add or change the requirements when necessary.

It is important in determining the actions that will take place in every phase to ensure that the implementation of Leave Management System (LMS)will run smoothly. The actions involved are described in Figure 2.2 and Table 2.1 on the next page :

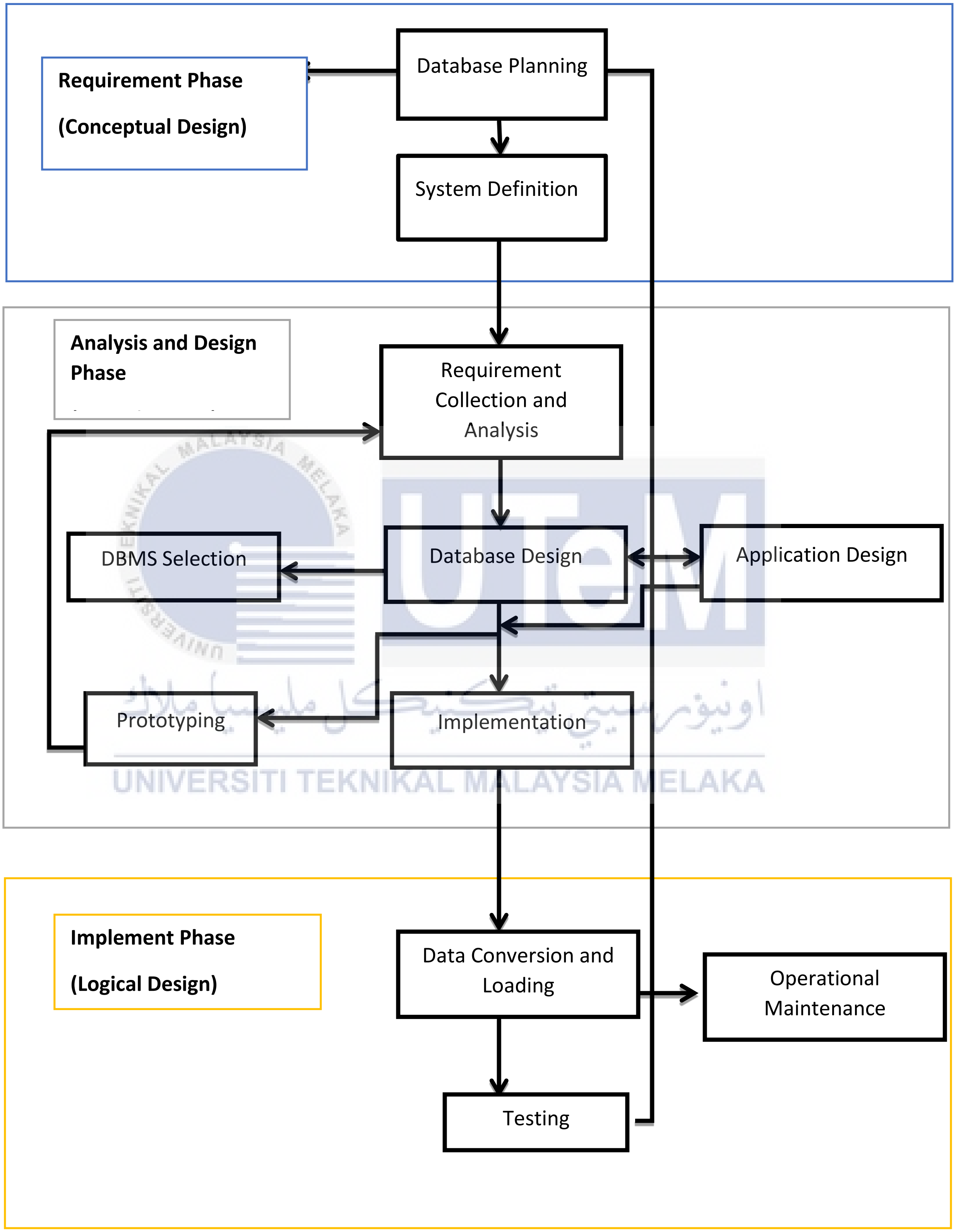


Figure 2.2: Agile method process flow

Table 2.1: Action involved in Agile method

Phase	Phase Description	Action Involved
Requirement	Process of identifying user requirement and also identifying the system objectives as well as the problem faced by user. This phase will include database planning and system definition process.	i. Define problems faced and state the objectives based on the problems. ii. Study the ability of proposed planning for the system.
Analysis	Process of examining and analyzed the detail of elements involved in developing system and also the user requirement.	i. Analyze user requirements. ii. Evaluate existing system.
Detailed system design	Process of taking out the idea of expected system outcomes by showing how the system will be made. This phase involve several stages : i. Database Design ii. Application Design iii. DBMS Selection iv. Prototyping v. Implementation	i. Define the elements, modules, architecture of the system and also system database. ii. Determine the system will look like and the functional of the system. iii. Develop the system based on the collected requirement.

Implementation	Process of executing the plans in order to make the system actually happen. This phase will include the : i. Data conversion and Loading ii. Testing iii. Operational maintenance	i. Installation of WAMP as system platform and Oracle 11g as system database. ii. Coding, testing and debugging the system. iii. Compare the actual outcome with the expected outcome.

2.3 Project Requirement

2.3.1 Software Requirement

Software	Purpose
Microsoft Project 2007	Project Management
Microsoft Word 2016	Project Documentation
Notepad++	Interface Designing and Implementation
Apache	Server Connection
Oracle 11g, WampServer	Database application
Adobe Dreamweaver	Interface Designing and Implementation

2.3.2 Hardware Requirement

Hardware	Purpose
Personal Computer	Project development and documentation
LAN	Connection of the system
Cable	Intranet Connection
Printer	Print out report

2.4 Project Schedule and Milestones

This section is very important so that it can be the guidance so that Leave Management System (LMS) can be finished according to the datelines given.

2.4.1 Project Milestones

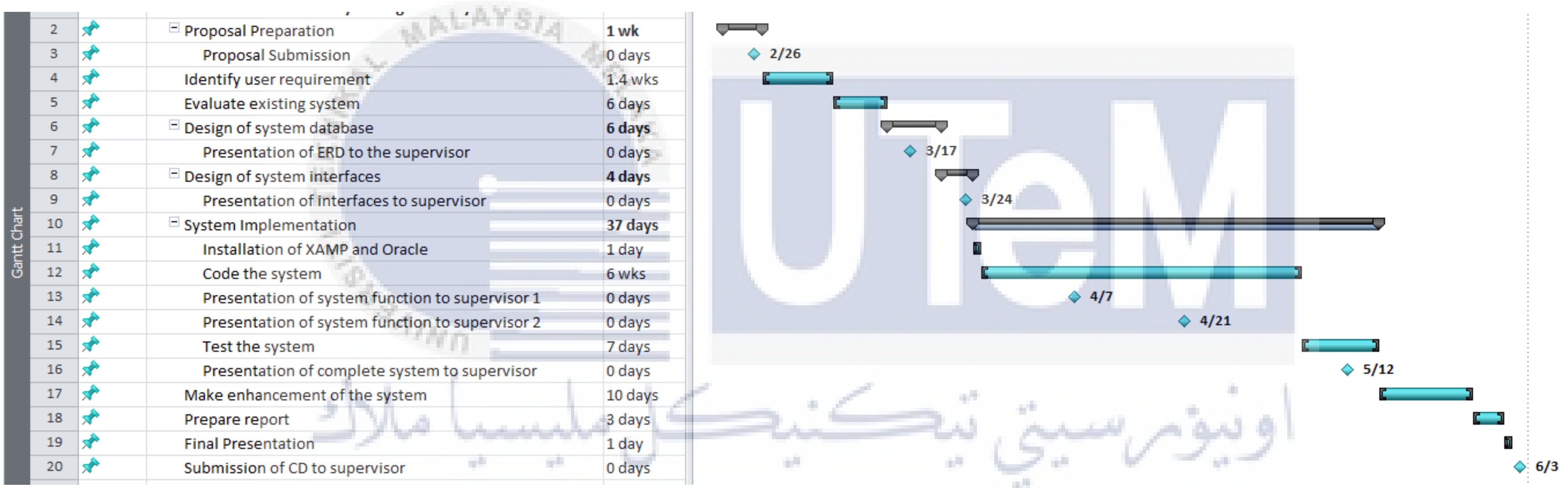
Milestones contributed to this system will help to ensure that the project done is staying on track without delivering any unnecessary or unimportant deliverable. It is good to have these milestones to show the progress of this system implemented. The milestones of Leave Management System (LMS) is explained in Table 2.2 below:

Table 2.2: Project Milestones

Milestone	Expected Document	Date
Proposal Submission	Proposal with brief explanation about Leave Management System (LMS).	26th February 2016
Presentation of ERD to supervisor	Entity-Relationship Diagram (ERD), Data Definition Language (DDL).	17th March 2016
Presentation of interfaces to supervisor	System interfaces and flow chart.	24th March 2016

First presentation of system function to supervisor	Functional requirement of the system and Data Manipulation Language (DML).	7th April 2016
Second presentation of system function to supervisor	Non-functional requirement with addition of functional requirement based on previous presentation.	21st April 2016
Presentation of complete system to supervisor	Complete functional system.	12th May 2016
Submission of CD to supervisor	Full system codes, logbook, ERD, DDL, DML and presentation slides.	3rd June 2016

2.4.2 Project Gantt Chart



2	Proposal Preparation	1 wk
3	Proposal Submission	0 days
4	Identify user requirement	1.4 wks
5	Evaluate existing system	6 days
6	Design of system database	6 days
7	Presentation of ERD to the supervisor	0 days
8	Design of system interfaces	4 days
9	Presentation of interfaces to supervisor	0 days
10	System Implementation	37 days
11	Installation of XAMP and Oracle	1 day
12	Code the system	6 wks
13	Presentation of system function to supervisor 1	0 days
14	Presentation of system function to supervisor 2	0 days
15	Test the system	7 days
16	Presentation of complete system to supervisor	0 days
17	Make enhancement of the system	10 days
18	Prepare report	3 days
19	Final Presentation	1 day
20	Submission of CD to supervisor	0 days

Figure 2.3: Gantt chart of Leave Management System (LMS)

2.5 Conclusion

As a conclusion for Chapter II, Leave Management System (LMS) is developed by following Agile methodology in DBLC as system methodology. It will ensure that the system implemented meets the objectives defined and also the expected outcomes. It also will act as the baseline in developing the system in order to make sure the develop system is complete. A well-defined planning in conducting this project plays an important role in making the system done on time. 15 weeks is prepared in order to complete this full system by illustrated it in a Gantt chart where it also consists of a few milestones for progress purposes.

The next chapter is Chapter III – Analysis. Analysis session will guide in developing the system by analyzing problem faced with proposed solution, functional requirements, non-functional requirements and also other related requirements.

Chapter III

Analysis

3.1 Introduction

This chapter is prepared in accordance with the user requirement. Analysis chapter can be divided into two that is the analysis of existing or current system and new system to be develop. It is a process of producing a detailed and complete analysis of the current system and the flow of the system that will be develop. This chapter will show how the system is illustrated to make the idea of the newly develop system. Through the analysis made, the weakness of current system can be identified and can be overcome in the system to be develop. All the problems that have been identify will be used as the bench mark in developing the new system in order to make sure that the new system is more reliable compared to current system. The analysis will be described using Context Diagram, and Data Flow Diagram and Flow Chart.

3.2 Problem Analysis

Problem analysis is an approach to software requirement analysis. It provides a view of understanding to the problems when developing the system. It produces system improvement objectives that address the problems. It includes the task of studying the problem domain, analyzing problem and establishing system improvement objective. Some problems that can be collected from the analysis of the current system are :

i) Difficult to search data.

All the information's are more difficult to search using the current system. The staff needs to search data from one file to another. It is time consuming and not systematic. Sometimes, the record might be lost due to external factor such as the paper have been thrown away.

ii) The probability data will lose is high

From the current system, the probability data will lose is high because the data only saved into the manual file system. The data will be lost from the file or taking by other people. The data also tend not be reliable enough because they will be many redundant data.

iii) The security of the information is not secure

Everybody can access the information because all the information was only saved into the manual file system. They are no mean to control the file access because there is no security measure.

Below is the flow of the current system :

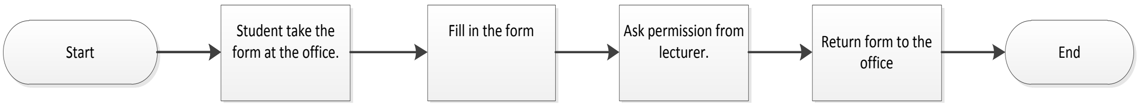

Start
Student take the form at the office.
Fill in the form
Ask permission from lecturer.
Return form to the office
End

Figure 3.1: Flow of current system for student.

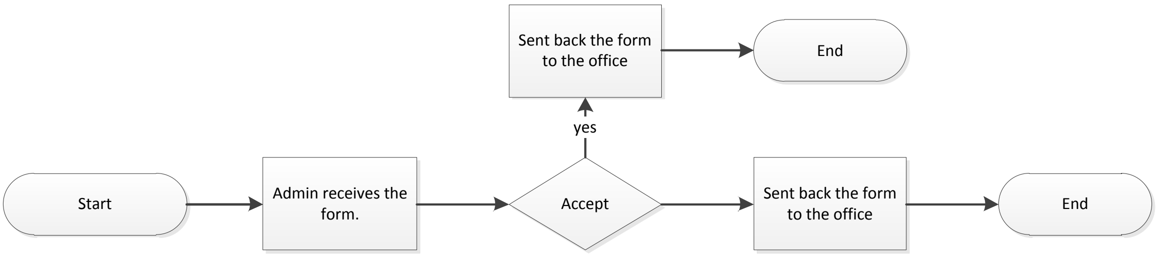

Sent back the form to the office
End
yes
Start
Admin receives the form.
Accept
Sent back the form to the office
End

Figure 3.2: Flow of current system for admin.

3.3 The Proposed Improvement

To overcome the problem that faces by the current system, a web based system is develop to replace the existing paper based system. The newly develop system should be able to support the weaknesses of old system. In order to do this, some analysis and planning have been made to make sure that the new system is good enough to replace the old one. The functional and non-functional requirement of the system have been identified to act as guidance in developing the system.

3.3.1 Functional Requirement

Functional requirement defines the internal working of the system such as the calculations, technical details, date manipulation and other processing functionality that define what the system should accomplish.it capture the intended behaviors of the system.

The functionalities of Leave Management System (LMS) will be shown in the form of flow chart shown below. The flow chart will show what the system should include and who is the user. The functionalities will also be shown using the context diagram and data flow diagram.

a. Flow Chart of Proposed System

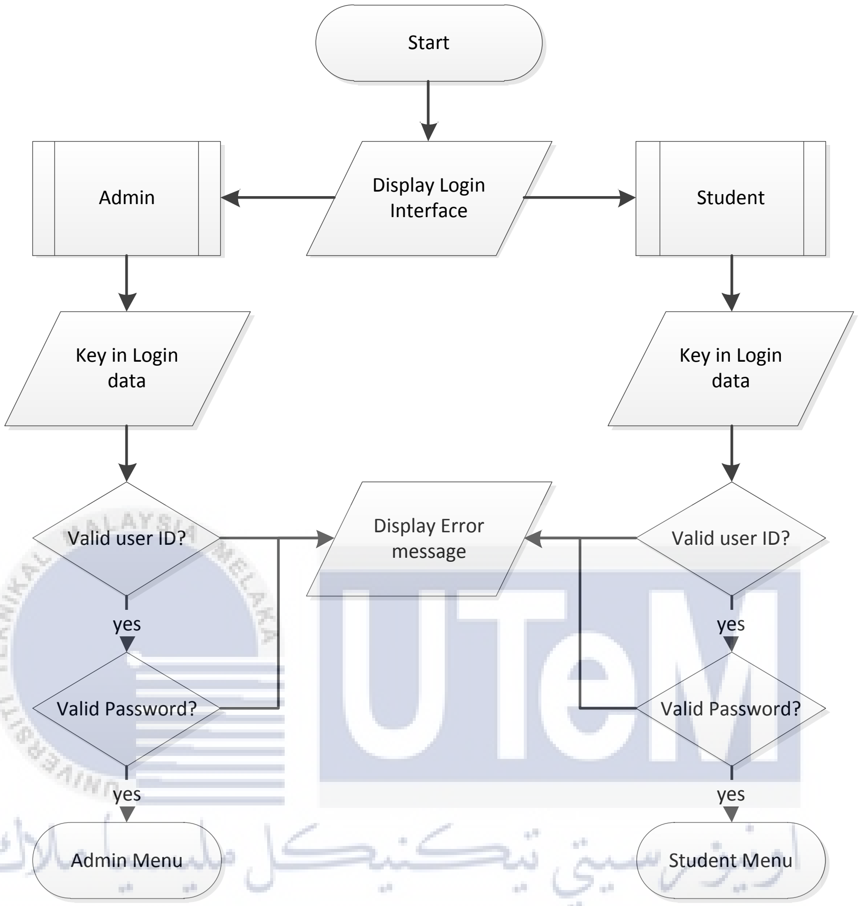


Figure 3.3: Login process for both user: Student and Admin.

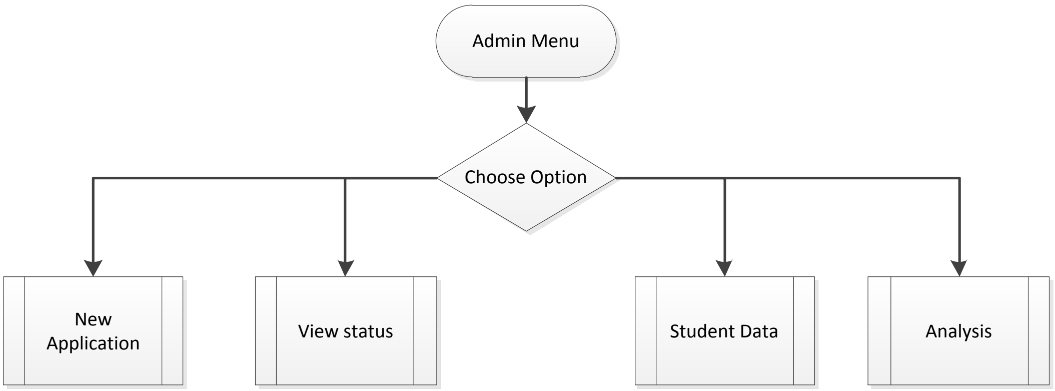


Figure 3.4: Admin flow chart.

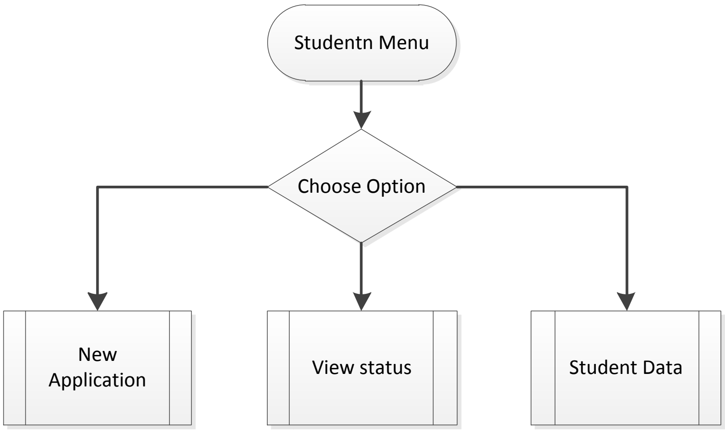


Figure 3.5: Student flow chart.

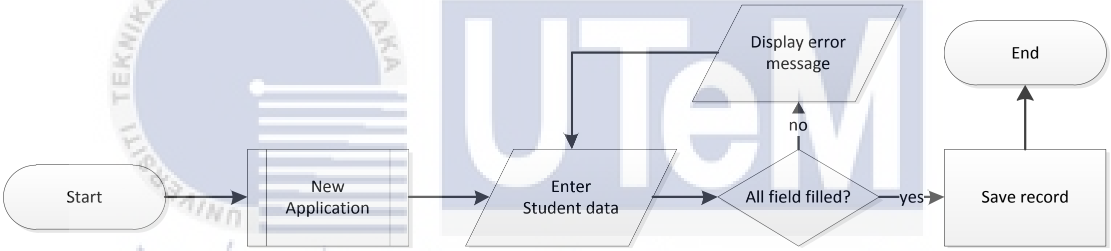


Figure 3.6: Apply leave flow chart.

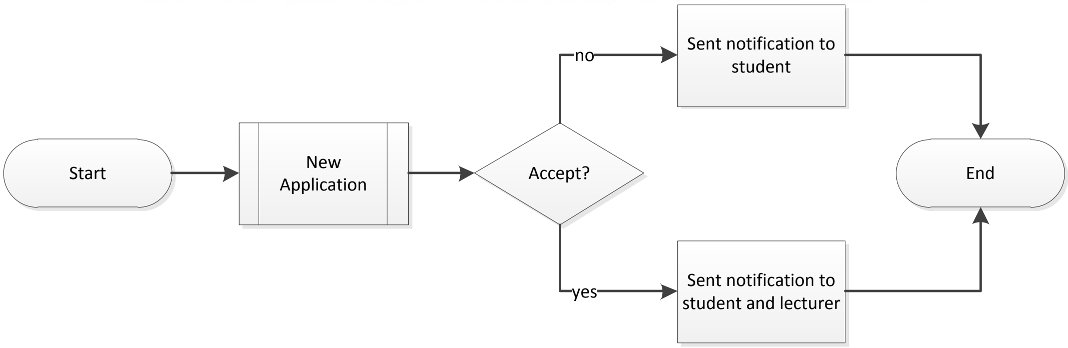

Start
New Application
Accept?
no
Sent notification to student
yes
Sent notification to student and lecturer
End

Figure 3.7: Process leave flow chart.

b. Context Diagram

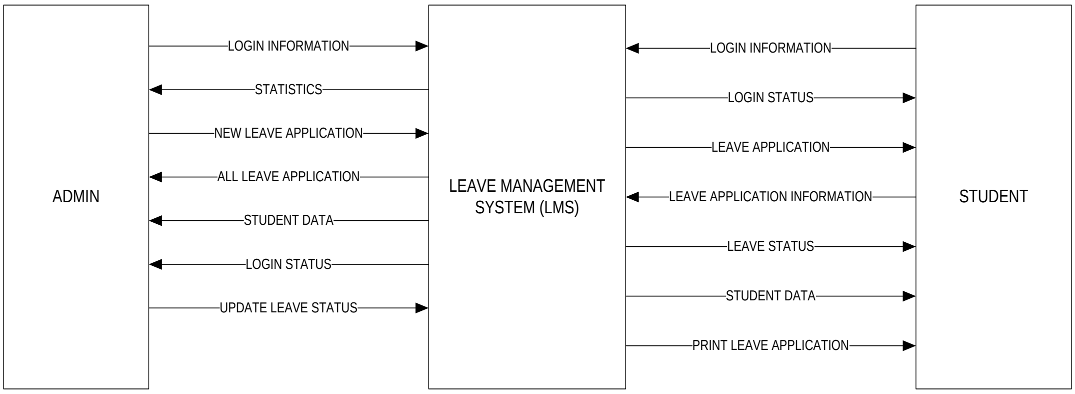


Figure 3.8: Context diagram for Leave Management System (LMS)

c. Data Flow Diagram

i. Level 0

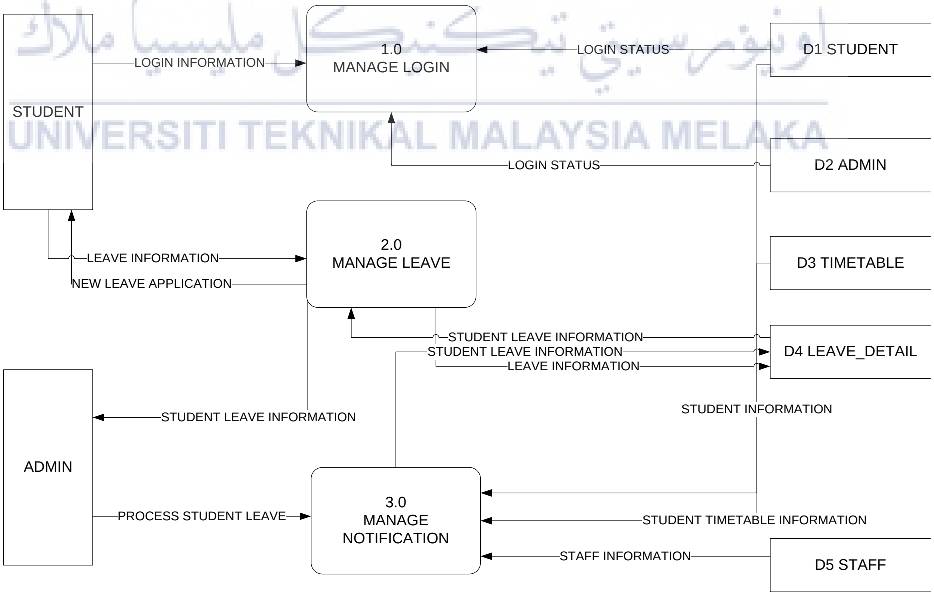


Figure 3.9: Data flow diagram level 0

ii. Level 1 for Student

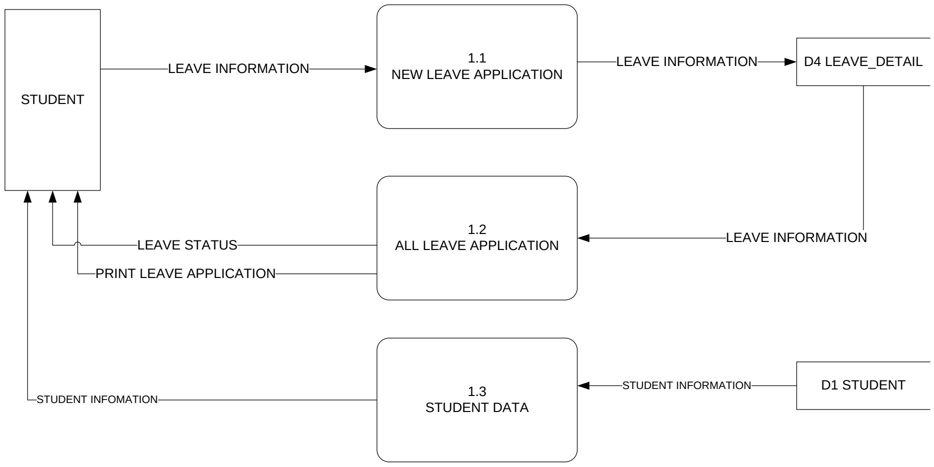


Figure 3.10: Data flow diagram for student

iii. Level 1 for Admin

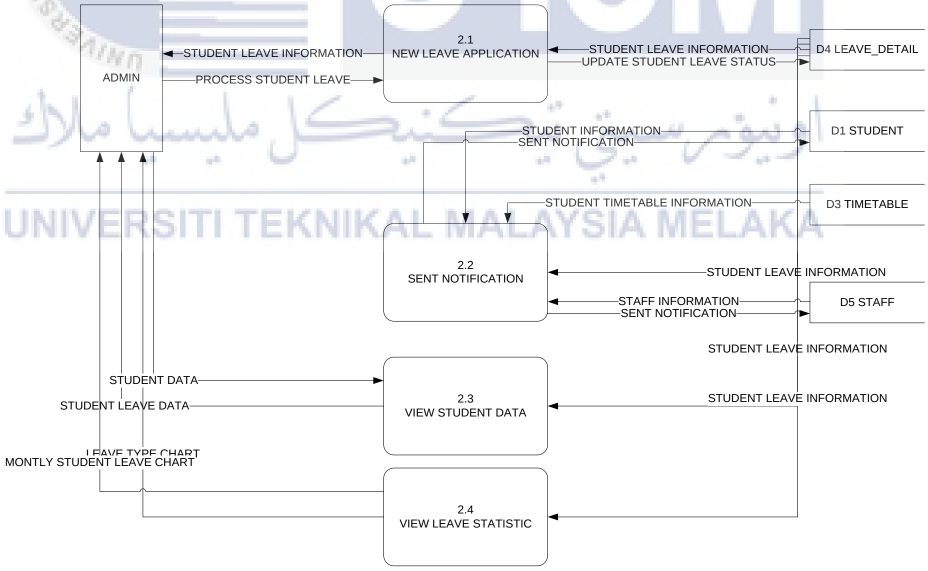


Figure 3.11: Data flow diagram for admin

3.3.2 Non-functional Requirement

Non-functional requirement of the system depends on the system architecture, implementation strategy and also the operational scenario. Non-functional requirement describes how well or to what standard should the system performed. The non-functional requirement of Leave Management System (LMS) are shown as below.

Table 3.1: Non-functional requirement of Leave Management System (LMS).

System Quality	Description
Usability	Graphical and easy to use interface to ensure smooth navigation as work with application. The system shall provide timely and helpful assistance with usage and operation.
Reliability	The system should store and retrieve information accurately as provided by the user.
Security	The system shall be accessible and usable to authorized user only.
Performance	The system should able to response to every transaction as fast as possible without any immediate delays.

3.4 Conclusion

In conclusion, the problem faced by the existing system have been identify and ways to overcome them have been proposed. With the usage of flow chart, context diagram and data flow diagram, the functional requirement of the proposed system have been explained. The non-functional requirement also has been underlined so that the user will have better understanding about the proposed system. This system is proposed to increase user satisfaction and reduce their workload.

Next chapter will be focusing on the system design where the architecture, physical design, logical design and also the conceptual design will be explained in detail.

CHAPTER IV

DATABASE SYSTEM DESIGN

4.1 Introduction

Database design has three main phases which are conceptual database design, logical database design, and physical database design. Conceptual database design is the first phase in database design methodology. Process of constructing a model of information used in an enterprise, independent of all physical considerations. In this phase, the entity types, relationship type and the attributes are identified and associated. The attribute domains, candidate and primary key attributes are also determined in this phase, not forget that the consideration to use enhanced modelling concepts are also made in this phase. At the same time, the model is check again and again for redundancy and validate local conceptual model against user transactions and review local conceptual data model with user.

Logical database design is the second phase in the database design methodology. The process of constructing a model of information used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations. In this phase, relations for local logical data is derived, validation of relations using normalization and validation of relations against user transactions are also conducted. Definition of integrity constraints and review local logical data model with user is also a must in this phase. Next, physical database design is the third and last phase of database design methodology. Process of producing a description of the implementation of the database on secondary storage. Describes the base relations, file organizations, and indexes design used to achieve efficient access to the data, and any associated integrity constraints and security measures. In this phase, developer code Database Designation Language (DBDL) for the system.

4.2 System Architecture Design

Leave Management System (LMS) is develop using the three-tier web based architecture. According to Bradley Morgan (2007) in his article 'An overview of common Database Architecture' states that the three tier architecture for the web based database system allows user to interact with the database through applications that a hosted on a web-server and displayed through a web-browser such as Google Chrome, Mozilla Firefox and Internet Explorer. The three tiers in a three-tier architecture are:

i. Presentation Tier:
Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.

ii. Application Tier:
Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

iii. Data Tier:
Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

The application tier connects to data tier to pass along query which in turn used to access, view and modify data. The data tier then passes back the requested data which will formatted by the application tier to be displayed on presentation tier for the user. Figure 4.1 shows the three-tier web-based architecture for Leave Management System (LMS).

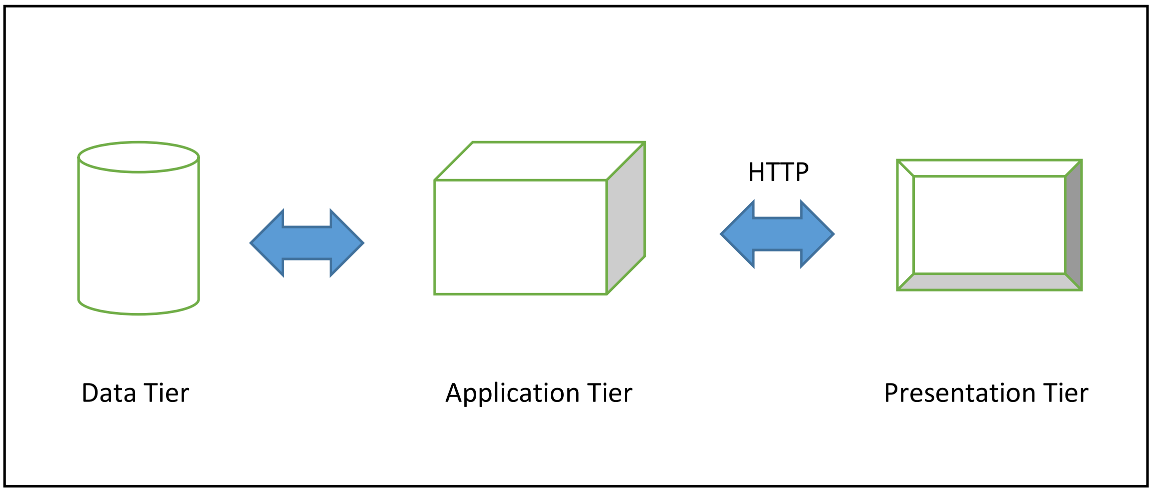


Figure 4.1: System Architecture Design For Leave Management System (LMS)

4.3 Database Design

Database design requires the understanding both the operational and business requirement of the system that are going to be develop as well as the ability to model visualize those requirement using organized database. This involves creating and constructing suitable model for the requirement. Database design has three main phases which are conceptual database design, logical database design, and physical database design.

4.3.1 Conceptual Design

Conceptual design is a process of constructing a model that most reliable and suitable to support all the requirement of Leave Management System (LMS). In the end of this process, business rule that indicate the system and entity relationship diagram to represent the relationship between entities and attributes will be produced.

Entity relationships diagram is a modelling concept that will help to organize the requirement of Leave Management System (LMS) into entities and define the relation between the entities. Figure 4.2 below will show the entity relationship diagram for Leave Management System (LMS). All together there are seven (7) entities which comprise of five (5) parent entities and two (2) bridge entities.

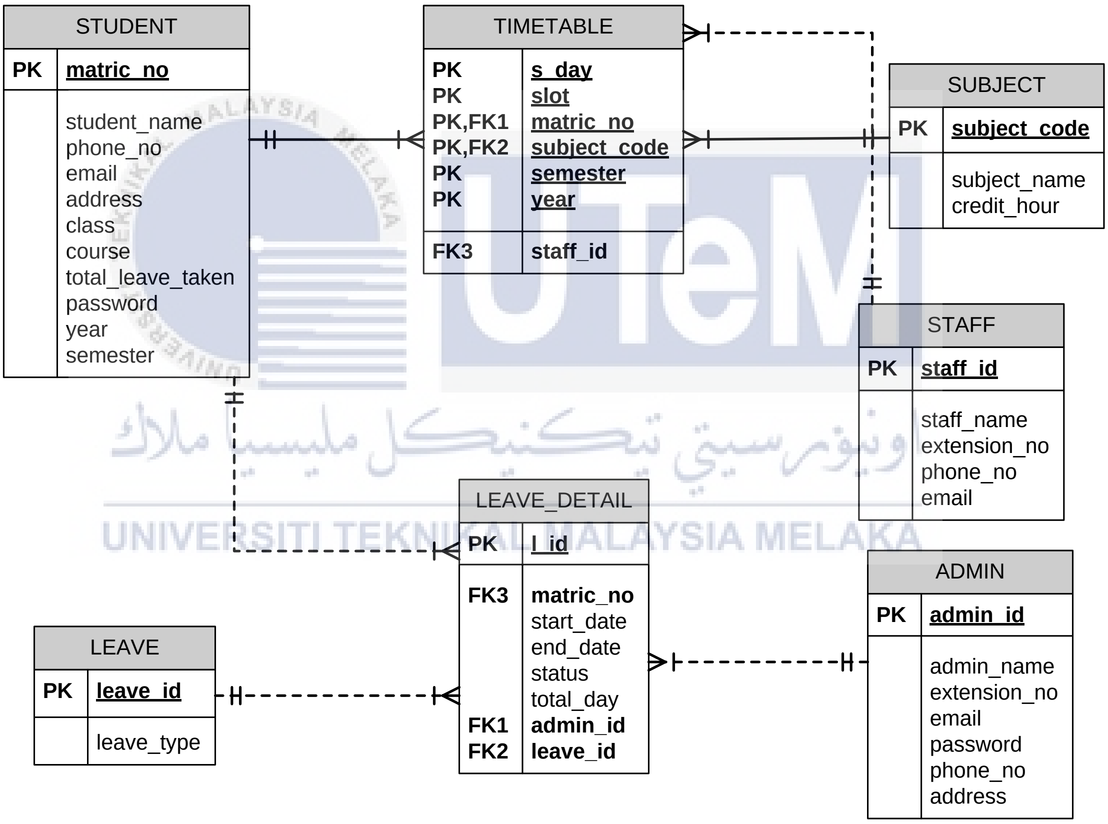


Figure 4.2: Entity Relationship Diagram for Leave Management System (LMS).

Business Rule

1. Student is allowed to apply more than one leave.
2. Student can apply many type of leave.

3. Student can view total leave taken.
4. Student can print many leave applications.
5. Admin can view many students leave application.
6. One admin can manage, update and delete many students leave application.
7. Admin can make many analysis based on student leave.
8. System can send many notifications to student.
9. System can send many notifications to lecturer.

4.3.2 Logical Design

The conceptual design is mapped into logical database design once it is completed and perfected. Basically, logical design is a process of constructing a model on information used in this project based on specific data model but independent of a particular Database Management System (DBMS) and other physical considerations. The result of this process is data dictionary. Data dictionary is used to explain further detail about the entity relationship model constructed in conceptual design phase. Table 4.1 in the next page show the data dictionary used in Leave Management System (LMS).

Table 4.1 : Data dictionary of Leave Management System (LMS)

Entity	Attribute Name	Data Type	Primary Key	Foreign Key	Unique	Not Null	Description	Example
Student	matric_No	varchar2(15)	Yes		Yes	Yes	Student matric number	B031310203
	student_Name	varchar2(50)					The name of student	Ali bin Abu
	phone_No	varchar2(15)					Student contact number	0122341234
	email	varchar2(30)					Student email address	ali@gmail,com
	address	varchar2(50)					Student personal address	No 16, Taman Murai Jaya,Melaka
	class	varchar2(10)					Student class	S1G2
	course	varchar2(10)					Student course	BITD
	total_Leave_Taken	number					Number of leave applied	20
	password	varchar2(10)					Student personal password	Abcd123
	year	number					Student current study year	2016
	semester	number					Student current semester	3
	staff_ID	varchar2(15)	Yes		Yes	Yes	Staff matric number	S001
	staff_Name	varchar2(50)					Staff full name	Kamil bin Ali

Staff	extension_No	varchar2(6)					Staff office extension number	2312
	phone_No	varchar2(15)					Staff personal telephone number	01212234345
	email	Varchar2(30)					Staff email address	kamil@utem.com
Admin	admin_ID	varchar2(15)	Yes		Yes	Yes	Admin matric number	A0012
	admin_Name	varchar2(50)					Admin full name	Ali bin Abu
	extension_No	varchar2(6)					Admin office extension number	4534
	email	varchar2(30)					Admin email address	ali@yahoo.com
	password	varchar2(10)					Admin personal password	123asd
	phone_No	varchar2(15)					Admin personal phone number	0196784123
	address	varchar2(30)					Admin address	No 14, Tmn Melati,Melaka
Leave	leave_ID	varchar2(15)	Yes		Yes	Yes	Leave identification number	L001
	leave_Type	varchar2(30)					Leave name	Emergency
Subject	subject_Code	varchar2(15)	Yes		Yes	Yes	Subject identification code	BLHHW 2321
	subject_Name	varchar2(50)					Subject name	Statistics

	credit_Hour	number					Subject credit hour	3
Leave_detail	l_ID	varchar2(15)	Yes		Yes	Yes	Leave detail identity number	LD00123
	start_Date	date					Leave start date	12-03-2016
	end_Date	date					Leave end date	15-03-2016
	status	varchar2(10)					Leave status	Accepted
	total_Day	number					Total day leave	3
	matric_No	varchar2(15)		Yes			Student matric number	B031310203
	admin_ID	varchar2(15)		Yes			Admin matric number	A0012
	leave_ID	varchar2(15)		Yes			Leave identification number	L001
Timetable	s_Day	date	Yes		Yes	Yes	Slot day name	12-03-2016
	slot	varchar2(15)	Yes		Yes	Yes	Slot time	9-11
	matric_No	varchar2(15)	Yes		Yes	Yes	Student matric number	B031310203
	subject_Code	varchar2(15)			Yes	Yes	Subject identification code	BLHHW 2321
	semester	number	Yes		Yes	Yes	Subject offered semester	3
	year	number	Yes		Yes	Yes	Year subject offered	2016
	staff_ID	varchar2(15)		Yes			Staff matric number	S001

4.3.2.1 Query Design

There is a few type of query that will be used to develop the Leave Management System (LMS). The example are :

1. Restricting and Sorting Data

a) Retrieving the leave application that still not been process yet.

```
select l.l_id, l.start_date, l.end_date, l.status,l .total_day,
b.leave_type, l.matric_no
 from leave_detail l, leave b
 where status ='In_Process' and
 l.leave_id = b.leave_id
 order by l_id desc;
```

b) Retrieving the leave application that still have been process.

```
select l.l_id, l.start_date, l.end_date, l.status,l .total_day,
b.leave_type, l.matric_no
 from leave_detail l, leave b
 where status !='In_Process' and
 l.leave_id = b.leave_id
 order by l_id desc;
```

c) Retrieving student application form data.

```
select l.l_id, l.start_date, l.end_date, l.status, l.total_day,
l.matric_no, t.leave_type
from leave_detail l, leave t
where matric_no=m_n and l.leave_id=t.leave_id
order by l_id desc;
```

2. Single-Row Function

a) Display all data of student leave application for specific student.

```
select * from leave_detail where matric_no=m_no;
```

b) Deleting data about student leave application.

```
delete from leave_detail where l_id=LD_id
```

c) Display all student data based on the matric number.

```
select * from student where matric_no=m_no
```

d) Updating student leave status.

```
update leave_detail set status ='Accepted',
admin_id = a_id where l_id=LD_id
```

e) Validating student matric number and password.

```
select count(*) into cc from student
where matric_no = p_id and password = p_pass ;
```

f) Validating admin number and password.

```
select count(*) into cv from admin
where admin_id = p_id and password = p_pass ;
```

3. Retrieving Data from Multiple Table

a) Retrieving student matric number and leave start and end date from table leave_detail and student.

```
select s.matric_no,l.start_date,l.end_date
into v_matric,s_date,e_date
from leave_detail l,student s
where l.l_id=v_l_id and
l.matric_no = s.matric_no;
```

b) Retrieving subject code, lecturer id and also lecturer email address from table staff, timetable, leave_detail and student based on student personal timetable.

```
select t.subject_code,a.staff_id,a.email
from leave_detail l,student s,timetable t,staff a
where l.l_id=v_l_id and
t.matric_no = v_matric and
l.matric_no = s.matric_no and
t.matric_no=s.matric_no and
t.staff_id=a.staff_id and
v_day = to_char(s_day,'day');
```

c) Retrieving the total leave number of leave based on student matric number group by the leave type from table leave_detail, leave and student.

```
select count(  l.leave_id) as
total,b.leave_type,s.student_name
from leave_detail l,leave b,student s
where l.matric_no=m_no
and l.matric_no=s.matric_no
and l.leave_id=b.leave_id
and l.matric_no LIKE '%' || m_no || '%'
group by s.student_name,b.leave_type;
```

d) Retrieving student application form data.

```
Select l.l_id, l.start_date, l.end_date, l.status, l.total_day,
l.matric_no, t.leave_type
 from leave_detail l, leave t
 where matric_no=m_no
 and l.leave_id=t.leave_id
order by l_id desc;
```

e) Calculate number of leave type taken by student.

```
select count(  l.leave_id) as total,b.leave_type
from leave_detail l,leave b
where l.leave_id=a_id
and l.leave_id=b.leave_id
and l.leave_id LIKE '%' || a_id || '%'
group by leave_type;
```

f) Retrieving student name, matric number and leave type from table student, leave detail and leave

```
select a.matric_no,a.student_name,b.leave_type
from   student a, leave b,leave_detail c
where c.matric_no=a.matric_no and
c.leave_id=b.leave_id
order by matric_no desc
```

g) Retrieve the leave id, leave type, total day, student matric number, status and also the name of the admin from leave, leave detail and admin table.

```
select l.l_id,l .start_date, l.end_date, l.status, l.total_day,
b.leave_type, l.matric_no,d.admin_name
from leave_detail l,leave b,admin d
where l.leave_id=b.leave_id and l.admin_id=d.admin_id and
l_id=l.l_id;
```

4. Aggregating Data Using Group Function

a) Display data about total number student have taken the leave type based on month and year input.

```
select count(c.l_id) as total,b.leave_type
from leave b,leave_detail c
where c.leave_id=b.leave_id and
to_char(start_date,'MM/YYYY') ='".$rl_mnth_yr."'
group by (b.leave_type) order by total desc";
```

b) Grouping data retrieve based on the student name and leave type.

```
select count(  l.leave_id) as
total,b.leave_type,s.student_name
from leave_detail l,leave b,student s
where l.matric_no=m_no
and l.matric_no=s.matric_no
and l.leave_id=b.leave_id
and l.matric_no LIKE '%' || m_no || '%'
group by s.student_name,b.leave_type;
```

c) Grouping data retrieve from leave detail and leave based on the leave type.

```
select count(  l.leave_id) as total,b.leave_type
from leave_detail l,leave b
where l.leave_id=a_id
and l.leave_id=b.leave_id
and l.leave_id LIKE '%' || a_id || '%'
group by leave_type;
```

4.3.3 Physical Design

The purpose for the physical design of a database to be written is to help configure the software and hardware for optimal performance and space utilization. It also important to help developer to understand more about the database used. This section will cover the translation and conversation of the logical design discus earlier to specific DBMS using Data Definition Language (DDL). This section will also include in details all the triggers, function and stored procedures that used in the system.

4.3.3.1 Data Definition Language (DDL)

DDL statements are used to create and modify the tables structure of a system in a database. The DDL statement differ depending on the type of database used. For Leave Management System (LMS) the database used is Oracle 11g. the database are created with the name of the system with username “naim” and database name “lms”.

There are seven (7) tables that will make up this system. The table are design using Oracle 11g statement which are mapped from the entity relationship diagram. The full code of the entity creation will be included in chapter 5 – Implementation.

Other than that, Leave Management System (LMS) is also developed using trigger, function and Stored Procedure. All this oracle Object is very important to make sure a system is reliable and consistent.

a) Trigger

A trigger is like a stored procedure that Oracle Database invokes automatically whenever a specified event occurs. A trigger can be invoking repeatedly but cannot be invoke explicitly. For Leave

Management System (LMS), several type of trigger is used to make the system more logical and make some event happen automatically. Type of trigger used and its name in the database are listed as below. The code example will be included in the Appendix A.

i. Before insert trigger.

This type of trigger is used to generate an ID for every row record automatically. Every time user insert new data in the database, this trigger will be invoked before the data successfully inserted into the database. Below is the name for the before insert trigger used in Leave Management System (LMS).

pk_student, pk_staff, pk_admin, pk_leave, pk_leave_detail

ii. After insert trigger.

This trigger will only be executed after and insert statement is successfully executed. It can only be done if the trigger is used to insert data to other table automatically after insert statement run successfully. Trigger example is :

after_insert_leave

iii. After delete trigger.

This trigger is used to insert the old record automatically to another table after a delete statement is run. Trigger example is:

after_delete_leave

iv. After update trigger.

This trigger is used to change the value of the existing record after an update executed on another table. The example is:

after_update_status

v. Before delete trigger.

Before delete trigger can be used in variety of way in condition that the event should occur before a delete statement runs. The example is :

before_delete_leave

b) Function Syntax

Oracle Function is a subprogram that returns a single value. The declaration and definition must be done before invoking the function. It can be done simultaneously or declare it first and later define it within the same block. Example of function used is to calculate the number of day.

calculateday_try

c) Stored Procedure Syntax

Stored procedure is a group of SQL statements that form a logical unit and perform a particular task, and they are used to encapsulate a set of operations or queries to execute on a database server. A stored procedure can be used in many ways such as update process, insert process and many more. Below is the name of stored procedure used to develop Leave Management System (LMS). The full code will be included in appendix section.

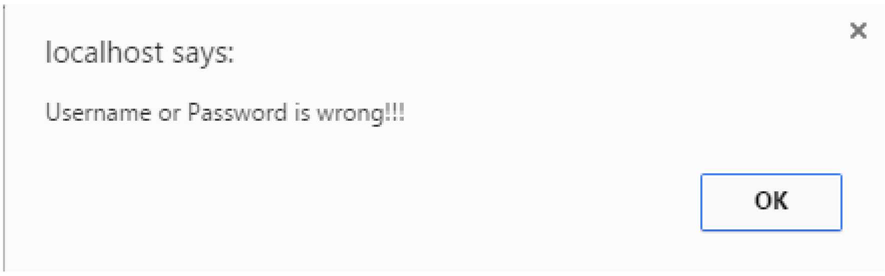
loginuser, check_date, deletedata_proc, leave_proc_try, list_student_leave3, listdata_proc, listleave_proc, liststudent1_proc, student_proc, updateleave_proc1, updateleave_proc2, updatestatus_proc, updatestudent

4.3.3.2 Security Mechanism

In the Leave Management System (LMS), security mechanism is used to prevent unauthorized user from accessing the system. Besides protect the system integrity and confidentiality, security mechanism also helps to avoid data lost. This will help to protects the system from any security challenges such as hackers or unauthorized user.

Security mechanism can be categorized that is physical security, authentication, authorization, accounting and data encryption. When all the categories implemented on a system, the system can be said well protected. For Leave Management System (LMS), authentication, authorization (security-level) and data encryption are used to protect the system.

Authentication is used in the login process where it requires staff number or student matric number and password before accessing to the system. Wrong staff number, matric number or password inserted in the field provided will force the system to pop out an error message. Same goes to if user does not complete either one of the fields provided.

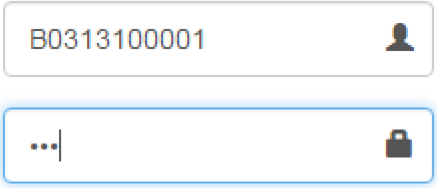

localhost says:

Username or Password is wrong!!!

Figure 4.3: Error mesaage if the user insert wrong detail needed to login.

Other than that, different user will be prompt different menu. This means different user will be assign different task. The two user of this system is student and also the admin. Admin can be dean, head of department and also deputy dean. The admin will be grant task such as accepting, rejecting and make analysis based on student leave while student will be grant task to apply leave application.

Lastly, data encryption is done in presentation level where password entered in the password field at the login page encrypted. It protects the password from being read by other people. It may lead to being hacked by other people if they know the valid staff number or matric number and password to access the system.


B0313100001

•••

Figure 4.4: Password encryption

4.3.3.3 Navigation Design

Navigation Design show the flow of the user interface in the Leave Management System (LMS) for each module. It gives an idea about how the user can navigate from one interface to another interface. It also helps user to fully understand the system flow. The navigation is divided into two that is for student and admin. The navigation design is shown in figure below.

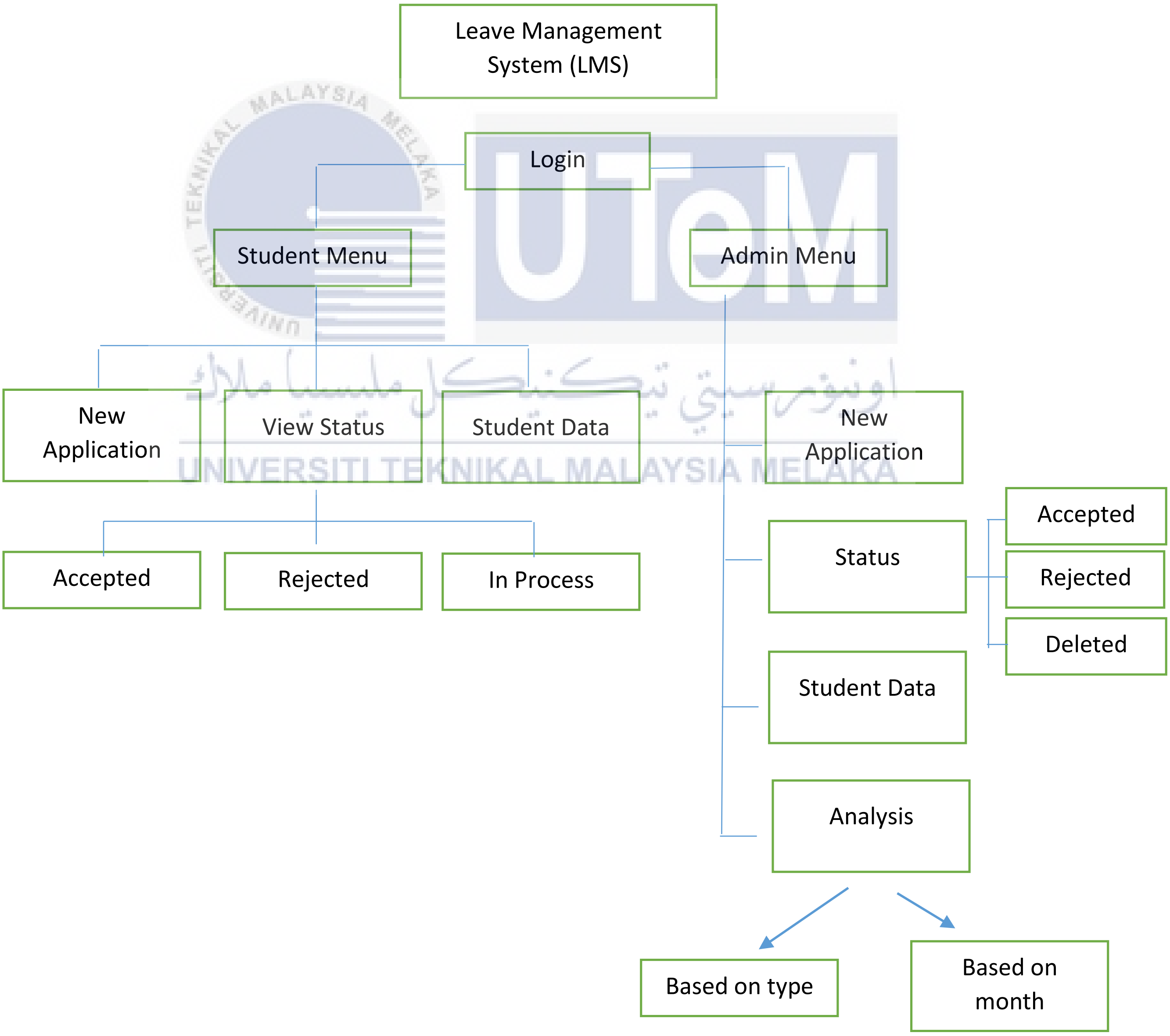


Figure 4.5: Navigation Design for Leave Management System (LMS)

4.4 CONCLUSION

In conclusion, this chapter describes overall design phase which it will be used in implementation of Leave Management System (LMS). The system architecture for this proposed system will use three-tier system architecture as it fulfills the criteria needed for this system development. It provides application layer that will manipulate data to be used or stored in the system. There are three part in designing the database which is conceptual design, logical design and physical design. These good database design will help in ensuring that the system run well with minimal mistakes occurred. The planning for the best user interface is one of important parts done as it will retrieve first impression from the user regarding the implemented system.

Activities has been done in design phase will play an important role in implementing the proposed system.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

A complete system must be implement correctly with the useful functionality. This chapter is about implementation phase which involve the description of all activities that undertaken in Leave Management System (LMS). The purpose of this system implementation is to transform the planned system design in previously chapter into a run able executable file.

It will meet few strategies which are performance, methods, debugging for error, libraries used to ensure that the Leave Management System is deliverables based on specifications and requirement identified earlier.

The implementation of Leave Management System (LMS) its software development environment setup, software configuration management, database server implementation and the implementation status for each module.

5.2 System Development Environment Setup

Before the system can be start to implement, it is important to setup the system implementation development environment first. For the system architecture, Leave Management System (LMS) is using the web based three-tier architecture as presented in the previous chapter. Three-tier architecture allow user to interact with database through application that are based on a web server and displayed via a web browser. In simple words, it means our personal laptop will need three things to operate this system which is the database, web server and also web browser as shown in the Figure 5.1 below. Database used is Oracle 11g, web server used is WampServer and the web browser used is Google Chrome.

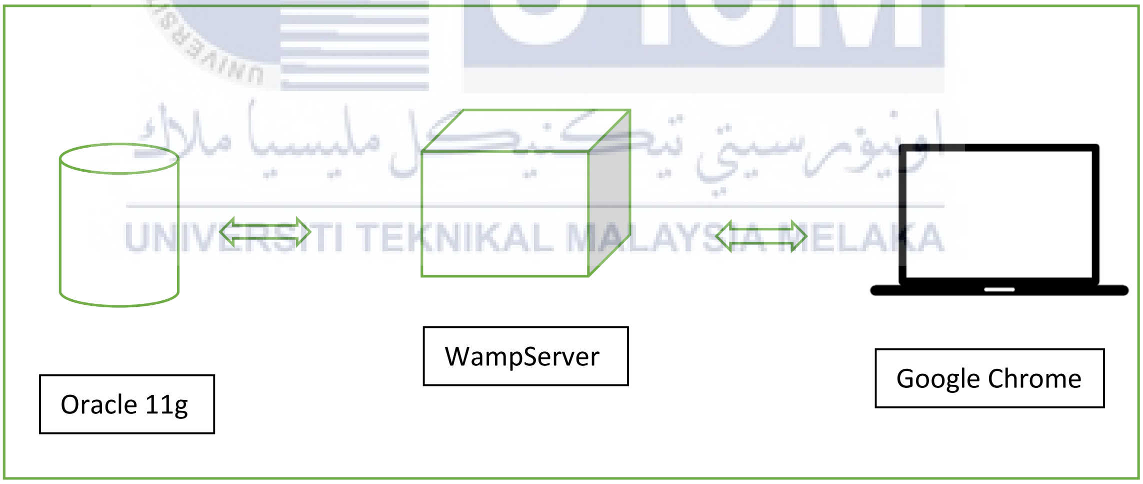


Figure 5.1 Web Based, Three-tier Client Server Architecture Design of Leave Management System (LMS).

5.2.1 Environment Setup

i. Database

The database used to develop Leave Management System (LMS) is Oracle 11g. Download the installation step from Google and simply follow the step given.

Table 5.1 : Database used in Client and Developer side.

Client	Developer
Oracle 11g	Oracle 11g

ii. Server

The server that used to implement the system is a localhost server that is WampServer. The full installation step will be included in Appendix B.

iii. Browser

Web browser which used to develop Leave Management System (LMS) is Google Chrome. But this system also compatible with the following as shown on table 5.2.

Table 5.2 Supported Browser

Client	Developer
Google Chrome (version 10 and above)	Google Chrome (version 10 and above)
Mozilla Firefox (version 4 and above)	Mozilla Firefox (version 4 and above)
Internet Explorer (version 6 and above)	Internet Explorer (version 6 and above)

iv. Computer Requirement

Hardware specification that used to develop Leave Management System (LMS) is stated as below.

Table 5.3 Hardware Requirement

Name	Description
Operating System	Microsoft Window 7 Ultimate
RAM	4 GB
Processor	Intel® Core™ i3-3210M
CPU	2.50Hz

5.2.2 Database Creation

After the database installation, the database for Leave Management System (LMS) is created with the name of “lms”. The statement used to create the database is shown as below.

```
Create database lms;
```

When the database has been created, all the entity in the Entity Relationship Diagram (ERD) is created inside the database. The primary key and foreign key are also assign during the entity creation in the database. Any constraint needed also included in this step.

5.3 Database Implementation

Database implementation refers to the database infrastructure that is developed for the system. The creation of entities as database objects will be presented through Data Definition Language (DDL). Meanwhile, overall processes of the system will be using triggers and stored procedure in order to make sure the system is consistent and reliable.

5.3.1 Data Definition Language (DDL)

Data Definition Language (DDL) is an interpretation process of data dictionary to the computer language in order to create and modify database objects in the existed database. All DDL used in creation of entities of Leave Management System (LMS) is shown as below.

a. Table Student

```
create table student(
matric_no varchar2(15) primary key,
student_name varchar2(50),
phone_no varchar2(15),
email varchar2(30),
address varchar2(50),
class varchar2(10),
course varchar2(10),
total_leave_taken number(3),
password varchar2(10),
semester number,
year number );
```

b. Table Staff

```
create table staff(
staff_id varchar2(15) primary key,
staff_name varchar2(50),
extension_no varchar2(6),
phone_no varchar2(15),
email varchar(30));
```

c. Table Admin

```
create table admin(
admin_id varchar2(15) primary key,
admin_name varchar2(20),
extension_no varchar2(5),
email varchar2(20),
phone_no varchar2(15),
password varchar2(15),
address varchar2(50));
```

d. Table Leave

```
create table leave(
leave_id varchar2(15) primary key,
leave_type varchar2(15));
```

e. Table Subject

```
create table subject(
subject_code varchar2(15) primary key,
subject_name varchar2(50),
credit_hour number(2));
```

f. Table Timetable

```
create table timetable(
s_day date,
slot varchar2(15),
matric_no references student(matric_no),
subject_code references
  subject(subject_code),
staff_id references staff(staff_id),
semester number,
year number,
primary key(s_day, slot, matric_no,
subject_code, semester,year));
```

g. Table Leave Detail

```
create table leave_detail(
l_id varchar2(10),
start_date date,
end_date date,
status varchar2(10) DEFAULT 'In_Process',
total_day number(2),
leave_id references leave(leave_id),
matric_no references student(matric_no),
primary key(l_id,leave_id),
constraint c_check_date
check (end_date >= start_date) );
```

5.3.2 Implementation of Main Process

Implementation of main process is an important part in order to make the system functions well and maximize user satisfaction. Some activities are using trigger in order to create auto generate ID. This trigger will use sequence to count the number. This step will reduce data required to be inserted by the user as it also may cut the time cost while running any activity. There are several types of trigger are used in the development of the system which are before insert, after insert, after update, after delete and before update trigger. Below is some example of trigger use in Leave Management System (LMS).

a. Trigger below is used to create ID for new leave application automatically.

```
create or replace trigger pk_leave_detail
before insert on leave_detail
for each row
declare ld_id leave_detail.l_id%TYPE;
begin
select leavedetail_seq.nextval into ld_id from dual;
:new.l_id := 'LD0'||ld_id;
end;
```

b. Trigger below is used to update the number of leave student have applied.

```
create or replace trigger after_insert_leave
after insert on leave_detail
for each row
begin
update    student    set    total_leave_taken    =
total_leave_taken + :new.total_day
where matric_no= :new.matric_no;
```

c. Trigger below is used to automatically insert the data into other table before the data is deleted.

```
create or replace trigger before_delete_leave
before delete on leave_detail
for each row
begin
update student set total_leave_taken=
total_leave_taken - :new.total_day
where matric_no = :new.matric_no;
end;
```

Moreover, all of main activities through this system is managed in database level where insert, update, delete and view is done with the usage of stored procedure. The data will be directly store in the database provided and it could be retrieve according to user request also by the projection of stored procedure. Usage of stored procedure ensure that the system is more secure and reliable as well help to lower the response time of a system. Several examples of stored procedure used is shown below as below.

a. Stored Procedure to calculate number of day based on start date and end date as well insert the data into database.

```
create or replace procedure leave_proc_try(
start_date in out varchar2,
end_date in out varchar2,
leave_id in leave.leave_id%TYPE,
matric_no in student.matric_no%TYPE )
IS
total_day number;
begin
total_day:=
calculateday_try(to_date(start_date,'YYYY-MM-
DD'),to_date(end_date,'YYYY-MM-DD'));
```

```
insert into leave_detail
(start_date,end_date,total_day,leave_id,matric_
no)
Values(to_date(start_date,'YYYY-MM-
DD'),to_date(end_date,'YYYY-MM-
DD'),total_day,leave_id,matric_no);
end;
```

b. Stored Procedure below is used to delete data from the database.

```
create or replace procedure
deletedata_proc(LD_id in leave_detail.l_id%TYPE)
as
begin
delete from leave_detail where l_id=LD_id;
commit;
end;
```

c. Stored Procedure below is used to view data of leave that is have been process.

```
create or replace procedure listdata_proc(
rc out sys_refcursor)
as
begin
open rc for
select  l.l_id,l.start_date,l.end_date,l.status,
l.total_day,b.leave_type,l.matric_no
from leave_detail l,leave b
where status !='In_Process' and
l.leave_id=b.leave_id
order by l_id desc;
end;
```

5.4 Conclusion

The implementation phase refers to the final process of moving the idea from development status to production status. Leave Management System (LMS) is implement using several tool such as database and server.

The system evolves from just an idea into a full system after this phase is completed. The system developed will going to be tested in the testing phase where it is discussed in the next chapter, Chapter IV – Testing. Full implementation of testing activities is defined where it consists of test plan, strategy to run the test and test design. The test results and analysis is provided after going through the system testing.

CHAPTER VI

TESTING

6.1 Introduction

In this chapter, the testing phase of Leave Management System (LMS) is discussed. Testing is a process to ensure that the system specification is fully developed and fulfills the user requirement. It is important as it helps developer to understand the limitation and error of the system and find way to handle it. Testing also help developer to measure the value of the developed system. Testing can be divided into several categories that is unit testing, integration testing, system testing and also user acceptance testing.

For Leave Management System (LMS), the unit testing was conduct to test each of the modules included and identify the software failures for each of the function in each module. Integration testing is done to check the flow of the system whether it was properly integrated together. System testing was done to validate the functional and non-functional requirement of user was successfully included in the system. The user acceptance testing is a testing to measure end-user satisfaction when using the system.

Testing activities in Leave Management System (LMS) start with creating a test plan which describe how the system delivers the testing. It involves the testing process, location or the environment for the testing to be carried out and the duration of the testing. Next step is prepared a test strategy which contains classes of test, test design that will include the test description to show the result of the testing. Last but no least is the test result analysis. The test result analysis will be made by the developer to make sure that the developed system is complete and have high potential value.

6.2 Test Plan

Test plan is a detailed document that state how a system should be test. It describes the overall testing strategy and defines the methodology that will use to conduct an effective test. Test plan is used to test the system to ensure that the system developed meets its specifications and requirement including functional and non-functional requirements. Besides, the test organization, test environment and test schedule will be defined in test plan. It also identifies the hardware, software and tools that required for testing and defines the features and function that will be tested. A good test plan help developer to do an effective test to the developed system as it be a guide to conduct the test.

6.2.1 Test Organization

Test organization describes personnel who are involved in testing the system. Mostly for every testing, they usual personnel involves are system developer, software tester and end-user such as the admin who will be using the system in future. Each personnel involve will have different role and different responsibilities to test the functionality of the system. The results which is bug, error or defect will be detected and reported to be fixed. Table 6.1 will show and describes the personnel involve in the testing.

Table 6.1: Personnel involved in testing Leave Management System (LMS) and their role.

No	Personnel	Role and Responsibility
1	System Developer	- -responsible for overall project scheduling functional and non-functional requirement of the system. - person who is in charge for unit testing, integration testing and system testing to ensure the success of the system.

2	Software Tester	- Responsible to detect and track the system error or bug. - Person who implement test strategy and prepares test script .
3	End-User	- Test the completed system and signals the approval of the developed system. - Person who provides comment toward system whether it fulfill their need and whether the system give them benefits.

6.2.2 Test Environment

This section will explain the test environment set-up for the system testing. The set-up will include the hardware used, software needed and other supporting tool in order the test to be done. As for the location, it can be done anywhere as a long as there is power supply. The system will be running on localhost server that is Wampserver. Wampserver will connect the Php and the Oracle 11g database. Table below will give the detail of the test environment set-up for testing phase.

Table 6.2: Test environment set-up for Leave Management System (LMS)

System Configuration	Specification
Operating System	Windows 7 Ultimate
Processor	Intel® Core™ i3-3210M
RAM	4 GB
Hard Disk Space	500 GB or above
Input Device	Mouse, Keyboard
Web Browser	Google Chrome, Internet Explorer

6.2.3 Test Schedule

The test schedule is prepared to efficiently manage the testing process of the system. The test strategy involved, date, duration and personnel involve will be included in the test schedule. The test schedule for the system is depicted on the table below.

Table 6.3 show the Test Schedule for Leave Management System (LMS).

No	**Test Strategy**	**Date**	**Duration**	**Personnel Involved**
1	Unit Testing	4 April 2016	3 days	System Developer
2	Integration Testing	13 May 2016	5 hours	System Developer
3	System Testing	13 May 2016	5 hours	Software Tester
4	User Acceptance Testing	14 May 2016	4 hours	End-user

6.3 Test Strategy

Test strategy is a section where the strategy used to test the system will be explained. It is a strategic document used to guide the personnel who involved in the system testing. There are many testing strategies but for Leave Management System (LMS), the strategy used is Bottom-up, White Box and Black Box testing.

Bottom-up strategy is a testing in which the order of the testing is reversed. In this type of testing strategy, the lower module of the system is test first followed by the main module. If there are problems occur at the lower module testing, it might affect the progression of the testing strategy. Figure below show the bottom-up strategy for Leave Management System (LMS) where the login is the first module to be test follow by the rest of module include in the system.

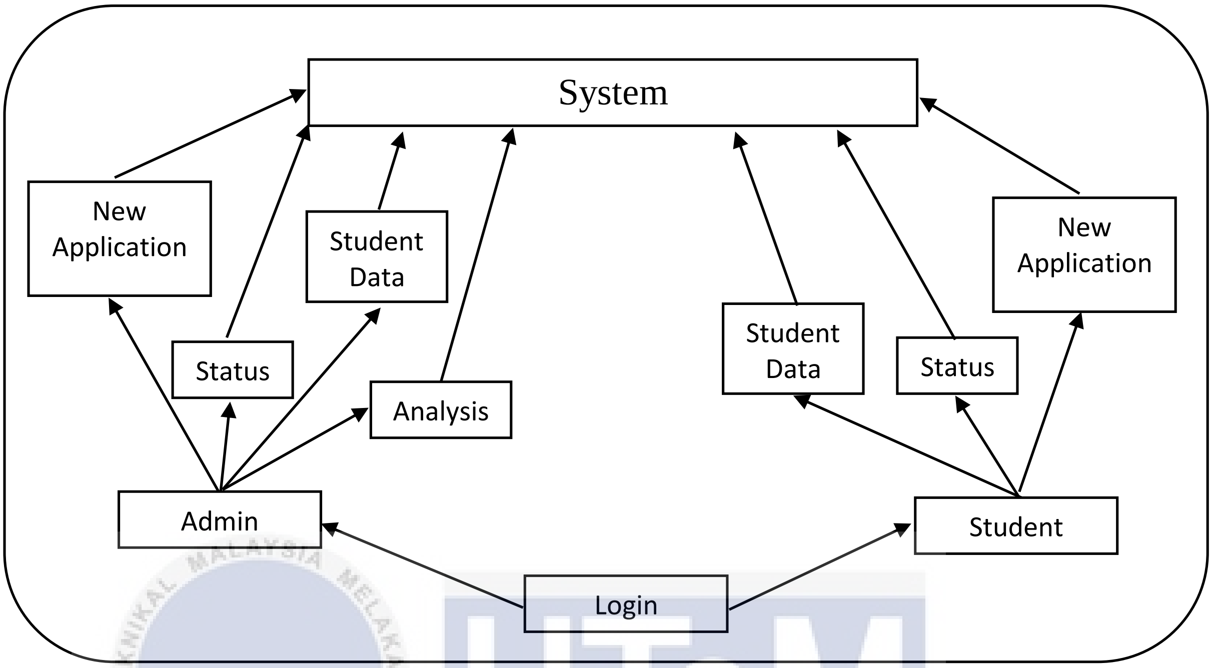


Figure 6.4 Bottom-up Testing Module for Leave Management System (LMS).

In White box testing, the test case is selected based on the software structure and implementation. It requires person who has knowledge on programming language to examine the system output and to ensure the performance of Leave Management System (LMS) works as expected. It also known as Unit Testing and Integration Testing.

The last strategy used is Black box testing. This test cases are focused on the functional specification of the system. It does not require any knowledge on the internal design and source code of the system. Each feature and functionality of GUI interfaces is tested without considering how the output is being generated.

Table 6.5 White-box Testing and Black-box Testing module of Leave Management System (LMS).

White-box Testing	Black-box Testing
• Unit Testing	• Functional Testing
• Integration Testing	• Performance Testing

- Security Testing	- System Testing
	- Usability Testing
	- User Acceptance Testing

6.4 Test Design

The test design is outline to ensure those important test is not overbooked. System developer may recreate the right sequence of the events that cased it by referring to the test design if there are any errors during the testing. Test Design for Leave Management System (LMS) testing will be defined in two main test that is test description and test data.

6.4.1 Test Description

The test case identification, test case and expected result for each module of Leave Management System (LMS) will be designed and documented in the test description. As for example, the admin login component is detailed in the table form as shown below while the rest of test description will be included in the Appendix C.

Table 6.6: Test Case for Admin Login Component.

Test Case ID	TEST_001-Admin Login		
Testing Phase	Unit Testing		
Description	To ensure the only authorized admin can access to the system.		
Test No	**Action**	**Expected Result**	**Actual Result**
Admin_Login_01	All fields leave blank.	System will display reminder to enter ID and Password.	OK
Admin_Login_02	User ID field leave blank.	System will display reminder to enter user ID.	OK
Admin_Login_03	Password field leave blank.	System will display reminder to enter password	OK
Admin_Login_04	Enter invalid user ID and password	System will display reminder to enter correct ID and Password.	OK

Admin_Login_05	Enter correct user ID and password	System will display reminder login success and direct to admin menu.	OK

6.4.2 Test Data

Test data is created based on the real data used in the testing of test description. It is important to covered all the test condition stated in test description. As for example, the test data for admin login component is detailed in table below. Meanwhile, the rest of the test data is included in the Appendix D.

Table 6.7: Test data for Admin Login Component.

Module	**Attribute**	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**
Admin Login	User ID	-	-	A001	A00	A001
	Password	-	123	-	234	123

6.5 Test Results and Analysis

The test results are the results collected from all the test that have been done on the Leave Management System (LMS). During the testing phase, any error or bugs detected is recorded so that it can be fixed later on. The test case will be conducted again to make sure that its meet it user requirement and bug free. As a results, the system is now ready to be delivered to the end-user. Table below shows each function that have been tested and the result of each test.

Table 6.8: Test Summary Result

Test Case ID	Module	Problem Submitted	Results (OK/Fail)
TEST_001	Admin Login Module	None	OK
TEST_002	Student Login	None	OK
TEST_003	Student Apply Leave	None	OK
TEST_004	Student Status	None	OK
TEST_005	Admin Process Leave	None	OK
TEST_006	Admin Status	None	OK
TEST_007	Admin Student Data	None	OK
TEST_008	Admin Analysis	None	OK
TEST_009	Integration Module	None	OK
TEST_010	System Module	None	OK

6.6 Conclusion

As conclusion, this chapter has discussed about the overall test process of Leave Management System (LMS). This process help developer to make sure that the system developed is complete and meet user requirement as well as it error and bug free. This process is critical as it can be the bench mark whether the developed system can be deliver or not.

In the next chapter, the strength and weakness of the developed system will be discussed. It also includes the suggestion to be made on the future to the system.

Chapter VII

CONCLUSION

7.1 Introduction

This chapter will discuss about the strength and weaknesses of the developed system. As a developer, he or she should know what the system can do and cannot do well. All of this will be list down and presented in this chapter. Besides, this chapter also will be discussing about the suggestion and recommendation that can be made to the system in the future. The recommendation will be able to cover the existing weaknesses of the system.

The contribution of the system will also be included in this chapter as it is the important to ensure that the developed system will bring good and benefits to user.

7.2 Observation on Strength and Weaknesses.

Leave Management System (LMS) have its own strength which can help user in many ways. But it also has weaknesses as the developer is a human which cannot run from doing mistake. The observation of strength and weaknesses will discuss below.

7.2.1 Strength

As mention earlier, Leave Management System (LMS) have several strengths that will be very helpful to the user. This system is developed after a detailed planning and observation on the current system have been made.

In order to make the system more reliable, the developer have list out the problem of current system and set some objectives that is going to be achieve by the system. Following point is the strength of Leave Management System (LMS) :

i. Enable student to apply for leave more easily.
ii. Save student time and reduce the work load for student to apply leave.
iii. Reduce the work load of admin in the administration office.
iv. Sent notification to student in form of email if their application is accepted or rejected.
v. Sent notification to lecturer who teach the student on the day student absent.
vi. Help admin to make analysis on each of the student based on the matric number.
vii. Help admin to make analysis based on type of leave student usually apply.
viii. Help admin make monthly and yearly analysis on student leave type.
ix. Reduce the processing time needed for admin to process leave application.

7.2.2 Weaknesses

Leave Management System (LMS) do have the weaknesses. These weaknesses should overcome in future in order make the system more complete and work efficiently. The weaknesses are :

i. There is no limitation on how many days' student can apply for leave.

ii. Although it will not be counted, student can apply leave on Saturday and Sunday.

7.3 Propositions for Improvement.

In order to make sure that the system more complete and more reliable, the weaknesses that have been identify should be overcome in the future. All the listed weaknesses should be the bench mark for the system improvement so that the system can meets its objective to help the user in their daily work. Following are the point of improvement that should be implemented on Leave Management System (LMS) in the future:

i. Limit the number of days' student can apply leave by setting constraint.
ii. Set some constraint for student so that they cannot apply leave with the start date on Saturday or Sunday.

7.4 System Contribution.

The contribution of Leave Management System (LMS) is for the student and also the admin. As for the student, they did not need to waste their time anymore to go to the administration office to take the application form, fill it and get permission from every lecturer and dean. Meanwhile, the admin did not need to take the application every day from the administration office lobby every day and check it manually one by one every single day. Admin also can make analysis more easily as the system have module that can automatically make the analysis and generate the bar graph and pie chart based on month, year and leave type.

In short, Leave Management System (LMS) will bring many benefit to user. It provides new approach for user rather than only rely on the existing system that is paper based system.

7.5 Conclusion

In conclusion, Leave Management System (LMS) has achieved the objectives that has been list out earlier in chapter I. Leave Management System (LMS) has improved the typical way to handle student leave application that before this using the paper based system which require a lot of workload as well as time consuming. Every module included in the system is well developed and can support and able to replace the existing system without any worries.

This system will bring out an effective enhancement of process and change in the existing system. Even though there are several improvements should be made, the system still more reliable and bring more good to user. By the development of Leave Management System (LMS), hopefully that the existing paper based system can be replace as the world is moving toward IT generation.

REFERENCES

George, Robbin (2013). What is the important of taking leave? [Online]

Retrieved on June,2016 from http://ewn.co.za/2013/08/22/The-importance-of-taking-leave

Technet.microsoft.com. (2014). Creating a Test Plan. [Online]

Retrieved on June, 2016 from http://teacnet.microsoft.com/en-us/library/cc781572(v=ws.10).aspx

Morgan, B (2014). An overview of common database architectures. [Online]

Retrieved on June, 2016 from

http://www.windowsecurity.com/whitepapers/software_engineering/An-Overview-of-Common-Database-Architectures.html

APPENDIX A

IMPLEMENTATION CODE

1. Trigger.

Leave Management System (LMS) are develop with the implementation of several types of trigger. The trigger are:

Before insert on table student.

```
create or replace trigger pk_student
before insert on student
for each row
declare s_id student.matric_no%TYPE;
begin
select student_seq.nextval into s_id from dual;
:new.matric_no := 'B0'||s_id;
end;
```

Before insert on table staff.

```
create or replace trigger pk_staff
before insert on staff
for each row
declare a_id staff.staff_id%TYPE;
begin
select staff_seq.nextval into a_id from dual;
:new.staff_id := 'S0'||a_id;
end;
```

Before insert on table admin.

```
create or replace trigger pk_admin
before insert on admin
for each row
declare v_id admin.admin_id%TYPE;
begin
select admin_seq.nextval into v_id from dual;
:new.admin_id := 'A00'||v_id;
end;
```

Before insert on table leave.

```
create or replace trigger pk_leave
before insert on leave
for each row
declare l_id leave.leave_id%TYPE;
begin
select leave_seq.nextval into l_id from dual;
:new.leave_id := 'L00'||l_id;
End;
```

Before insert on table leave.

```
create or replace trigger pk_leave_detail
before insert on leave_detail
for each row
declare ld_id leave_detail.l_id%TYPE;
begin
select leavedetail_seq.nextval into ld_id
from dual;
:new.l_id := 'LD0'||ld_id;
```

After insert on leave detail

```
create or replace trigger after_insert_leave
after insert on leave_detail
for each row
begin
update student set total_leave_taken =
total_leave_taken + :new.total_day
where matric_no= :new.matric_no;
end;
```

After delete on leave detail

```
create or replace trigger after_delete_leave
after delete on leave_detail
for each row
begin
insert into leave_detail_b(l_id, start_date,
end_date, status, total_day, leave_id, matric_no)
values (:old.l_id,:old.start_date,:old.end_date,
:old.status,:old.total_day,
:old.leave_id,:old.matric_no);
```

After update on leave detail

```
create or replace trigger after_update_status
after update on leave_detail
for each row
begin
if :new.status='Accepted' then
null;
else
update student set total_leave_taken =
total_leave_taken - :new.total_day
```

```
where matric_no= :new.matric_no;
end if;
end;
```

Before delete on leave detail

```
create or replace trigger before_delete_leave
before delete on leave_detail
for each row
begin
update student set total_leave_taken=
total_leave_taken - :new.total_day
where matric_no = :new.matric_no;
end;
```

2. Function.

Leave Management System (LMS) is develop with the implementation of oracle function. The function is :

Function to calculate the number of day.

```
create or replace function calculateday_try
(start_date date,end_date date)
return number
is
v_day varchar2(20);
t_day number:=0;
s_date date :=start_date;
e_date date :=end_date;
begin
while s_date <= e_date
loop
v_day:=to_char(s_date,'DAY');
if (v_day = ('SUNDAY   ') OR v_day= ('SATURDAY '))
 THEN
DBMS_OUTPUT.PUT_LINE (v_day||' falls on weekend');
NULL;
else
t_day := t_day +1;
end if;
s_date:=s_date +1;
end loop;
return t_day;
end;
```

3. Stored Procedure.

Leave Management System (LMS) are develop with the implementation of stored procedure. The stored procedure are as shown below :

Stored procedure for login of two users: Admin and Student.

```
create or replace procedure loginuser(
 p_id in   admin.admin_id%type,
 p_pass in  admin.password%type ,
who out varchar2)
 as

cc int;
cv int;

begin
select count(*) into cc from student
where matric_no = p_id and password = p_pass ;
select count(*) into cv from admin
where admin_id = p_id and password = p_pass ;
if cc = 1 then
      who := 'STUDENT';
      elsif cv = 1 then
      who := 'ADMIN';
      end if;
end;
```

Stored procedure to check date and get lecturer email address.

```
create or replace procedure check_date(
v_l_id leave_detail.l_id%TYPE,rc out sys_refcursor)
as
s_date date;
e_date date;
v_day varchar2(15);
v_matric varchar2(20);
begin
select s.matric_no,l.start_date,l.end_date
into v_matric,s_date,e_date
from leave_detail l,student s
where l.l_id=v_l_id and
l.matric_no = s.matric_no;
while s_date <= e_date
```

```
loop
v_day:=to_char(s_date,'day');
dbms_output.put_line('dayyy:'||v_day);
dbms_output.put_line('matricc:'||v_matric);

if v_day<>'saturday' or v_day<>'sunday' then

open rc for

select t.subject_code,a.staff_id,a.email
from leave_detail l,student s,timetable t,staff a
where l.l_id=v_l_id and
t.matric_no = v_matric and
l.matric_no = s.matric_no and
t.matric_no=s.matric_no and
t.staff_id=a.staff_id and
v_day = to_char(s_day,'day');

commit;

else
dbms_output.put_line('salah');
end if;
s_date := s_date + 1;
end loop;
end;
```

Stored procedure to delete student leave application

```
create or replace procedure deletedata_proc
(LD_id in leave_detail.l_id%TYPE)
as
begin
delete from leave_detail where l_id=LD_id;
commit;
end;
```

Stored procedure to calculate number of days and apply new leave.

```
create or replace procedure leave_proc_try(
start_date in out varchar2,
end_date in out varchar2,
leave_id in leave.leave_id%TYPE,
matric_no in student.matric_no%TYPE )
IS
total_day number;

begin


total_day := calculateday_try(to_date
(start_date,'YYYY-MM-DD'),to_date(end_date,
'YYYY-MM-DD'));
insert into leave_detail
(start_date,end_date,total_day,leave_id,matric_no)
values (to_date(start_date,'YYYY-MM-
DD'),to_date(end_date,'YYYY-MM-
DD'),total_day,leave_id,matric_no);

end;
```

Stored procedure to find student and determine number of leave taken based on leave type using student matric number.

```
create or replace procedure list_student_leave3(
m_no student.matric_no%TYPE, rc out sys_refcursor)
is
begin
OPEN rc FOR
SELECT COUNT(  l.leave_id) as
total,b.leave_type,s.student_name
from leave_detail l,leave b,student s
where l.matric_no=m_no
and l.matric_no=s.matric_no
and l.leave_id=b.leave_id
and l.matric_no LIKE '%' || m_no || '%'
group by s.student_name,b.leave_type;
end;
```

Stored procedure to lest all leave application that have been process.

```
create or replace procedure listdata_proc(
rc out sys_refcursor)
as
begin
open  rc for select
l.l_id,l.start_date,l.end_date,l.status,l.total_day,
b.leave_type,l.matric_no
```

```
from leave_detail l,leave b
where status !='In_Process' and
l.leave_id=b.leave_id
order by l_id desc;
end;
```

Stored procedure to lest all leave application that not yet processed.

```
create or replace procedure listleave_proc(
rc out sys_refcursor)
as
begin
open  rc for select
l.l_id,l.start_date,l.end_date,l.status,l.total_day,
b.leave_type,l.matric_no
from leave_detail l,leave b
where status='In_Process' and
l.leave_id=b.leave_id
order by l_id desc;
end;
```

Stored procedure to list all student leave application.

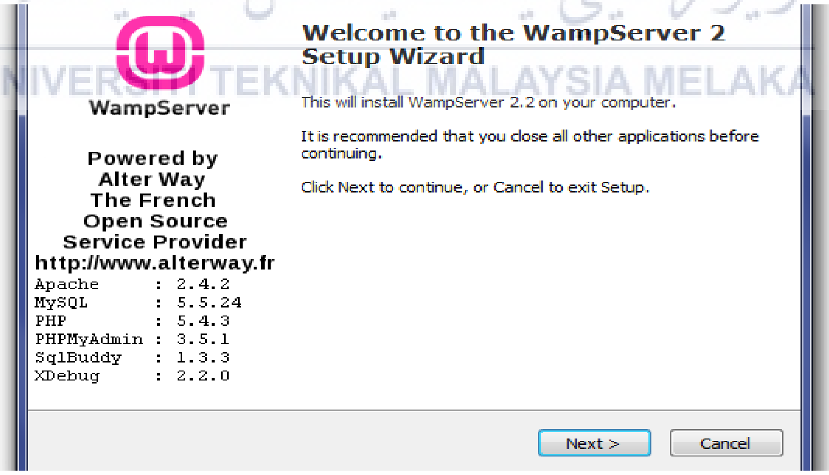
```
create or replace procedure liststudent1_proc(
m_no student.matric_no%TYPE,rc out sys_refcursor)
as
begin
open  rc for select
l.l_id,l.start_date,l.end_date,l.status,l.total_day,
l.matric_no,t.leave_type
from leave_detail l, leave t
where matric_no=m_no
and l.leave_id=t.leave_id
order by l_id desc;
end;
```

APPENDIX B

WAMP INSTALLATION

B.1 Installing WampServer

- To start the installation process, open the folder and double-click the installer file. A security warning window will open, ask confirmation to run the file.
- Click Run to start the installation process.
- The Welcome to the WampServer Setup Wizard screen was pop-pup. Click Next to continue the installation.



Welcome to the WampServer 2 Setup Wizard

This will install WampServer 2.2 on your computer.

It is recommended that you close all other applications before continuing.

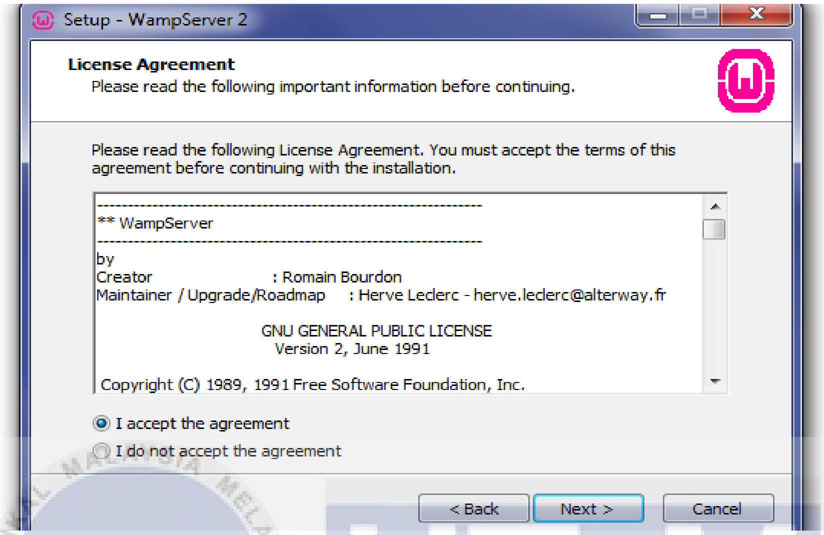
Click Next to continue, or Cancel to exit Setup.

WampServer

Powered by
Alter Way
The French
Open Source
Service Provider
http://www.alterway.fr

```
Apache     : 2.4.2
MySQL      : 5.5.24
PHP        : 5.4.3
PHPMyAdmin : 3.5.1
SqlBuddy   : 1.3.3
XDebug     : 2.2.0
```

- The next screen is the License Agreement. Check the radio button next to accept the agreement, then click Next to continue the installation.



License Agreement

Please read the following important information before continuing.

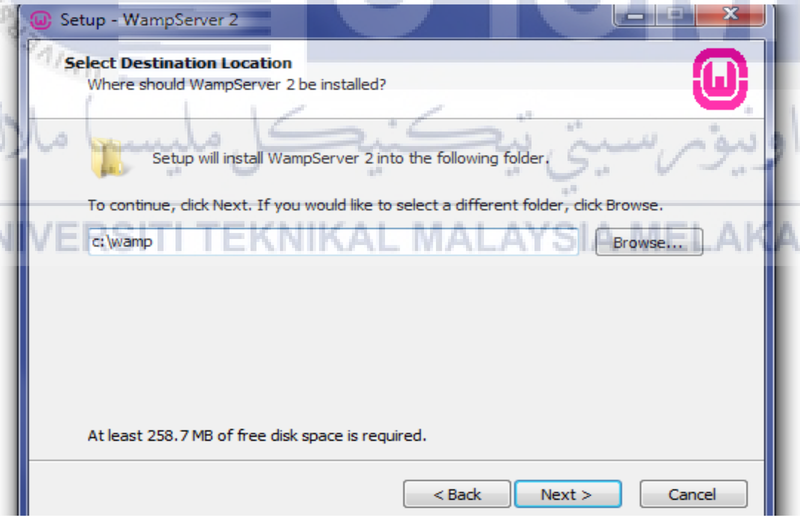
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

```
--------------------------------------------------------------
** WampServer
--------------------------------------------------------------
by
Creator                     : Romain Bourdon
Maintainer / Upgrade/Roadmap     : Herve Leclerc - herve.leclerc@alterway.fr

                    GNU GENERAL PUBLIC LICENSE
                      Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
```

● I accept the agreement

○ I do not accept the agreement

- Next Select Destination Location screen will be shown. Click Next to continue.



Select Destination Location

Where should WampServer 2 be installed?

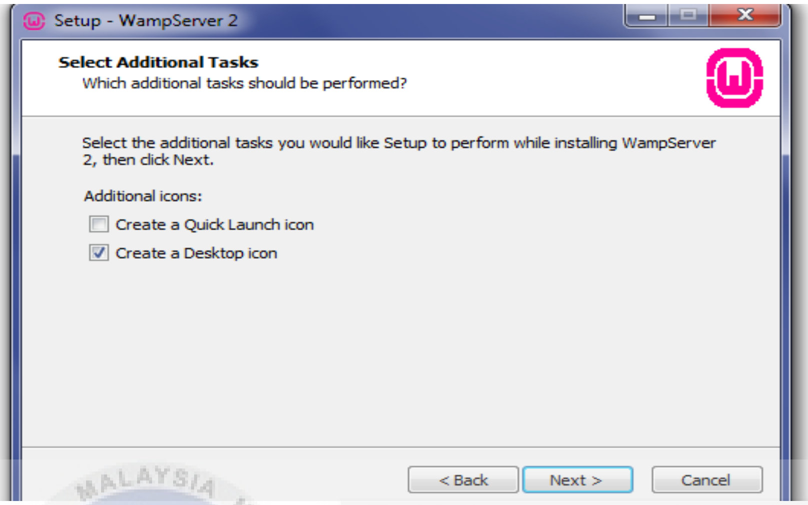
Setup will install WampServer 2 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

c:\wamp

At least 258.7 MB of free disk space is required.

- The next screen is the Select Additional Tasks screen. You will be able to select whether you would like a Quick Launch icon added to the taskbar or a Desktop icon created once installation is complete. Make a selection, then click Next to continue.


Setup - WampServer 2

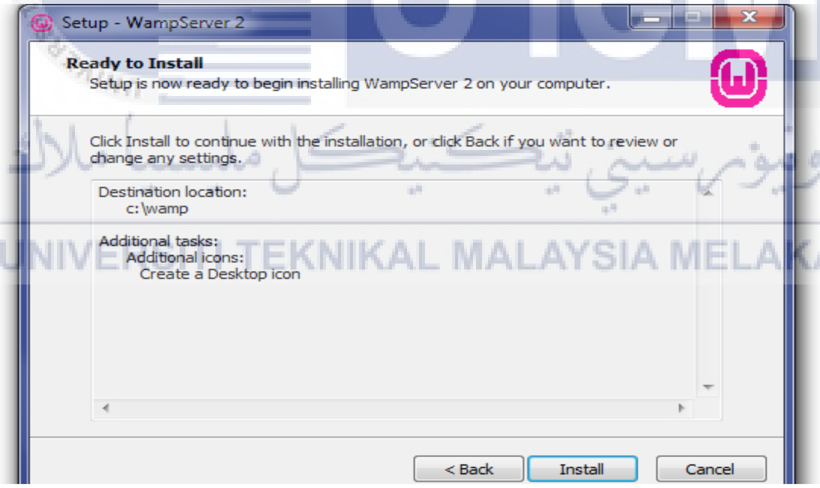
Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing WampServer 2, then click Next.

Additional icons:

☐ Create a Quick Launch icon
☑ Create a Desktop icon

- Next is the Ready To Install screen. Click Install to continue.

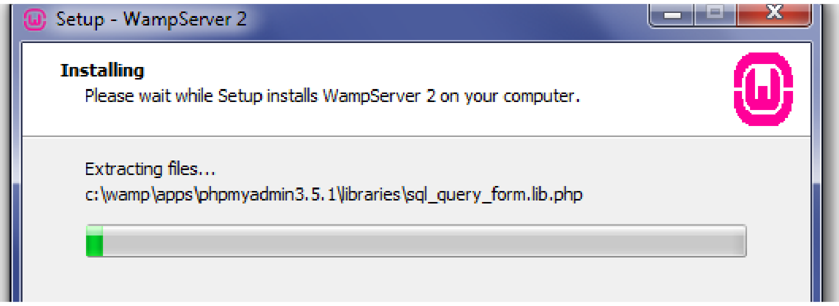

Setup - WampServer 2

Ready to Install
Setup is now ready to begin installing WampServer 2 on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    c:\wamp

Additional tasks:
    Additional icons:
       Create a Desktop icon
```

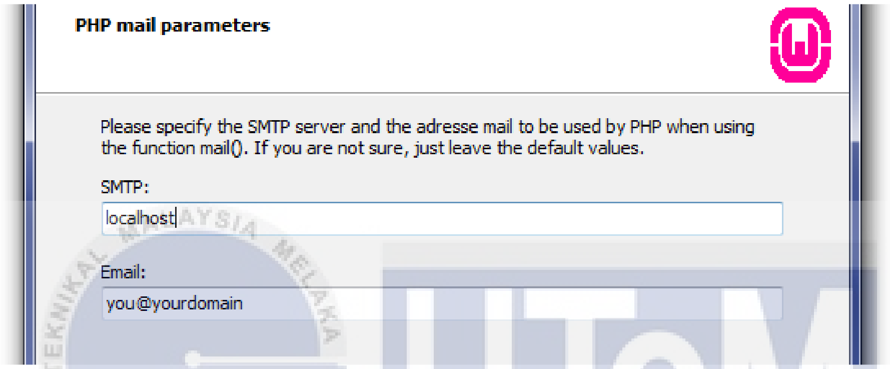
- WampServer will begin extracting files to the location that been selected.


Setup - WampServer 2

Installing
Please wait while Setup installs WampServer 2 on your computer.

Extracting files...
c:\wamp\apps\phpmyadmin3.5.1\libraries\sql_query_form.lib.php

- Choose the browser that will be set as a default browser and then click continue.
- Once the progress bar is completely green, the PHP Mail Parameters screen will appear. Leave the SMTP server as local host, and change the email address to one that had been choose. Click Next to continue.



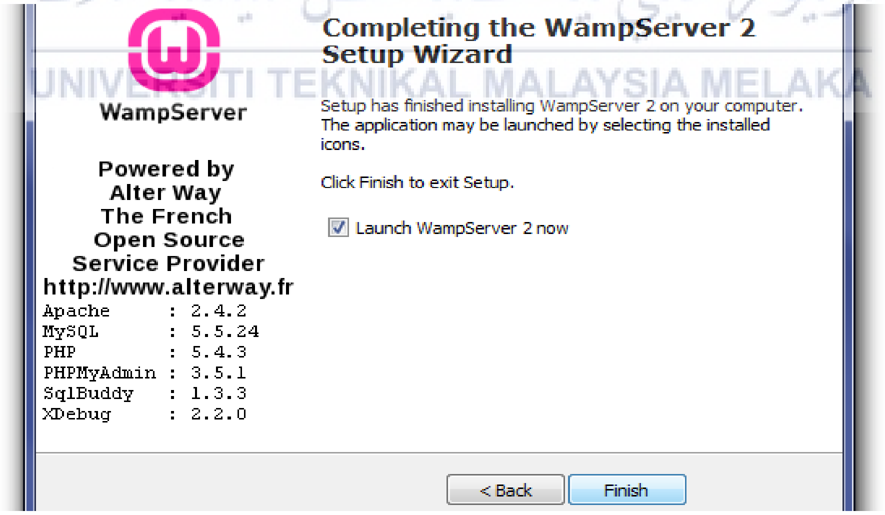
PHP mail parameters

Please specify the SMTP server and the adresse mail to be used by PHP when using the function mail(). If you are not sure, just leave the default values.

SMTP: localhost

Email: you@yourdomain

- The Installation Complete screen will now appear. Check the Launch WampServer Now box, then click Finish to complete the installation.



Powered by Alter Way The French Open Source Service Provider http://www.alterway.fr

```
Apache     : 2.4.2
MySQL      : 5.5.24
PHP        : 5.4.3
PHPMyAdmin : 3.5.1
SqlBuddy   : 1.3.3
XDebug     : 2.2.0
```

Completing the WampServer 2 Setup Wizard

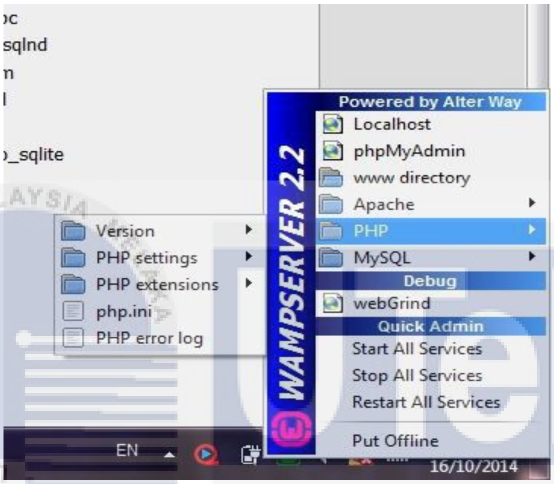
Setup has finished installing WampServer 2 on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

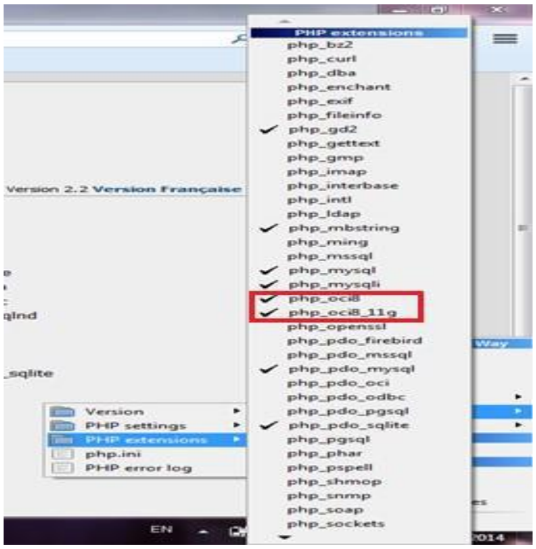
☑ Launch WampServer 2 now

B.2 Configuring WampServer

- Click on the WampServer icon, go to the php menu, and click on the php.ini option. This will open the php.ini file in your plain text editor. Adjust the following settings:



- Remove the ; symbol at the beginning of the list.
- To allow database oracle used php in WampServer click on php_oci8 and php_oci8_11g.



- Same like before, remove the ; symbol at the beginning of the list WampServer is ready.

APPENDIX C

TEST DESCRIPTION

Table C.1: Test Case for Student Login Component.

Test Case ID	TEST_002- Student Login		
Testing Phase	Unit Testing		
Description	To ensure the only authorized student can access to the system.		
Test No	**Action**	**Expected Result**	**Actual Result**
Student _Login_01	All fields leave blank.	System will display reminder to enter ID and Password.	OK
Student _Login_02	User ID field leave blank.	System will display reminder to enter user ID.	OK
Student _Login_03	Password field leave blank.	System will display reminder to enter password	OK
Student _Login_04	Enter invalid user ID and password	System will display reminder to enter correct ID and Password.	OK
Student _Login_05	Enter correct user ID and password	System will display reminder login success and direct to student menu.	OK

Table C.2: Test Case for Student Apply Leave Component.

Test Case ID	TEST_003- Student Apply Leave		
Testing Phase	Unit Testing		
Description	To ensure the only authorized student can apply leave.		
Test No	**Action**	**Expected Result**	**Actual Result**
Student_Apply_Leave_01	All fields leave blank.	System will display reminder to fill all field.	OK
Student_Apply_Leave_02	Start Date field leave blank.	System will display reminder to enter Start Date.	OK
Student_Apply_Leave_03	End Date field leave blank.	System will display reminder to enter End Date.	OK
Student_Apply_Leave_04	Reason field leave blank.	System will display reminder to enter Reason.	OK
Student_Apply_Leave_05	Enter start date later than end date	System will display reminder to enter end date later than start date.	OK
Student_Apply_Leave_06	All field enter correctly	System will display data have been recorded message and direct to new leave form.	OK

Table C.3: Test Case for Student Status Component.

Test Case ID	TEST_004- Student Status		
Testing Phase	Unit Testing		
Description	To ensure the student able to view their leave application status.		
Test No	**Action**	**Expected Result**	**Actual Result**
Student _Status_01	Select In process option from the status menu.	System will display all leave application that still in process.	OK
Student _Status_02	Select accepted option from the status menu.	System will display all leave application that have been accepted.	OK
Student _Status_03	Select rejected option from the status menu.	System will display all leave application that have been rejected.	OK

Table C.4: Test Case for Admin Process Leave Component.

Test Case ID	TEST_005- Admin Process Leave		
Testing Phase	Unit Testing		
Description	To ensure admin can update the leave status.		
Test No	**Action**	**Expected Result**	**Actual Result**
Admin_Process_Leave_01	Click the student detail button.	System will display student personal data.	OK
Admin_Process_Leave_02	Click the accept button.	System will update the status and sent notification to student and lecturer.	OK
Admin_Process_Leave_03	Click the reject button.	System will update the status and sent notification to student.	OK

Table C.5: Test Case for Admin Status Component.

Test Case ID	TEST_006- Admin Status		
Testing Phase	Unit Testing		
Description	To ensure the admin able to view all student leave application status.		
Test No	**Action**	**Expected Result**	**Actual Result**

Admin _Status_01	Select deleted option from the status menu.	System will display all leave application that have been deleted.	OK
Admin _Status_02	Select accepted option from the status menu.	System will display all leave application that have been accepted.	OK
Admin _Status_03	Select rejected option from the status menu.	System will display all leave application that have been rejected.	OK

Table C.6: Test Case for Admin Student Data Component.

Test Case ID	TEST_007- Admin Student Data		
Testing Phase	Unit Testing		
Description	To ensure the admin able to view particular student leave data.		
Test No	**Action**	**Expected Result**	**Actual Result**
Admin_Student_Data_01	Leave search field empty.	System will display error message to fill the search field.	OK
Admin_Student_Data_02	Enter search field with false matric number	System will display error message to fill the search field correctly.	OK
Admin_Student_Data_03	Enter search field with correct matric number	System will display student leave data based on the matric number.	OK

Table C.7: Test Case for Admin Analysis Component.

Test Case ID	TEST_008- Admin Analysis		
Testing Phase	Unit Testing		
Description	To ensure the admin make analysis based on student leave data.		
Test No	**Action**	**Expected Result**	**Actual Result**
Admin_Analysis_01	Leave all search field empty.	System will display error message to fill the search field.	OK
Admin_Analysis_02	Enter month field only.	System will display error message to fill the year field.	OK
Admin_Analysis_03	Enter year field only.	System will display error message to fill the month field.	OK
Admin_Analysis_04	Enter month and year field correctly.	System will display bar graph on number of student apply leave and number of accept and reject leave for the month and year.	OK

Table C.8: Test Case for Integration Module

Test Case ID	TEST_009- Integration Module		
Testing Phase	Integration Testing		
Description	To test all component can run smoothly after integration.		
Test No	**Action**	**Expected Result**	**Actual Result**
Integration_01	Test all module after integration.	Functionalities of each module can be performed as specified.	OK

Table C.9: Test Case for System Module

Test Case ID	TEST_010- System Module		
Testing Phase	System Testing		
Description	To ensure system meet its functional and non-functional requirement.		
Test No	**Action**	**Expected Result**	**Actual Result**
System_01	Test system whole module.	Functionalities of each module can be performed as specified and error free.	OK

APPENDIX D

TEST DATA

Table D.1: Test data for Student Login Component.

Module	Attribute	Test 1	Test 2	Test 3	Test 4	Test 5
Student Login	User ID	-	-	B0313010001	B0313010000	B0313010001
	Password	-	123	-	234	123

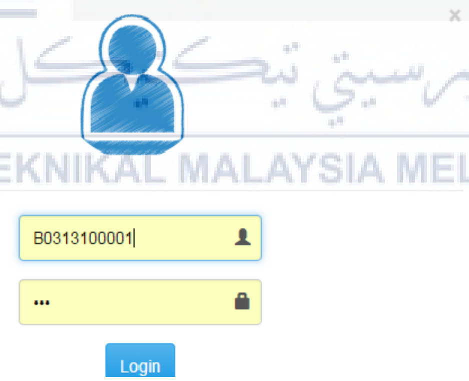
Table D.2: Test data for Student Apply Leave Component.

Module	Attribute	Test 1	Test 2	Test 3
Student Apply Leave	Start Date	-	12-05-2016	12-05-2016
	End Date	-	10-05-2016	13-05-2016
	Reason	-	Emergency	Emergency

APPENDIX E

USER MANUAL

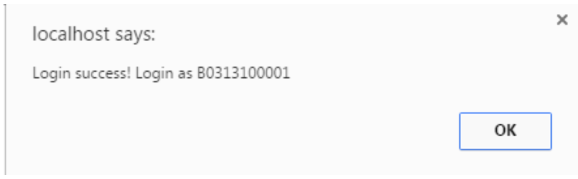
User Login



B0313100001

•••

1. Enter User ID and Password according to user privileges.
2. Click login button.
3. Pop up message will appear saying that login success



localhost says:

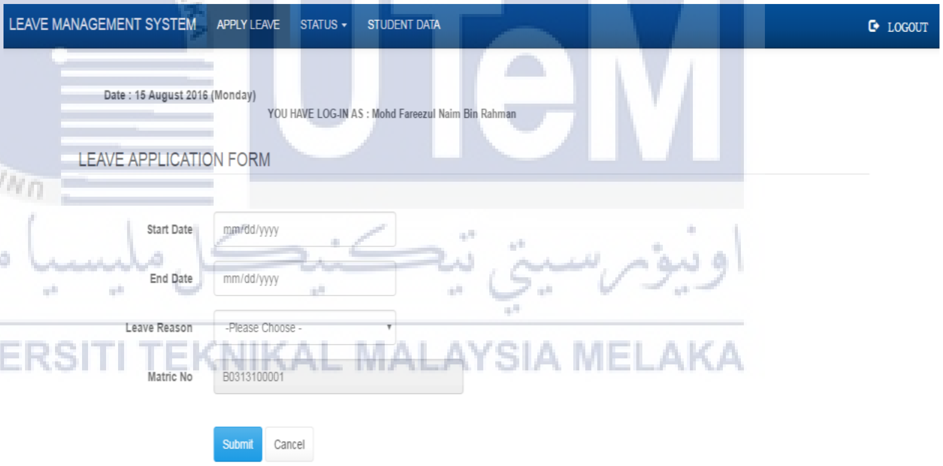
Login success! Login as B0313100001

4. After that user will be directed to menu based on user privileges.

Student

a. <u>Apply New Leave</u>

1. If the user is student, system will direct user to student menu. New application form will be display on screen.



Date : 15 August 2016 (Monday)

YOU HAVE LOG-IN AS : Mohd Fareezul Naim Bin Rahman

LEAVE APPLICATION FORM

Start Date: mm/dd/yyyy

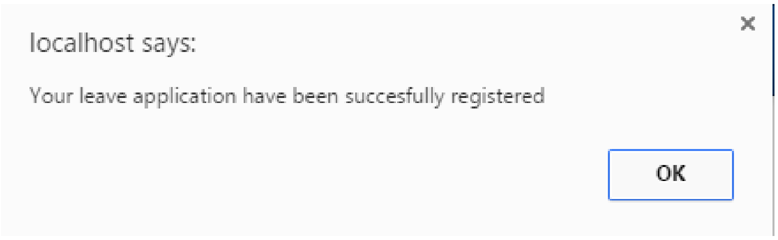
End Date: mm/dd/yyyy

Leave Reason: -Please Choose -

Matric No: B0313100001

2. Fill in all field and click submit button.

3. A pop up message will appear saying record have been save

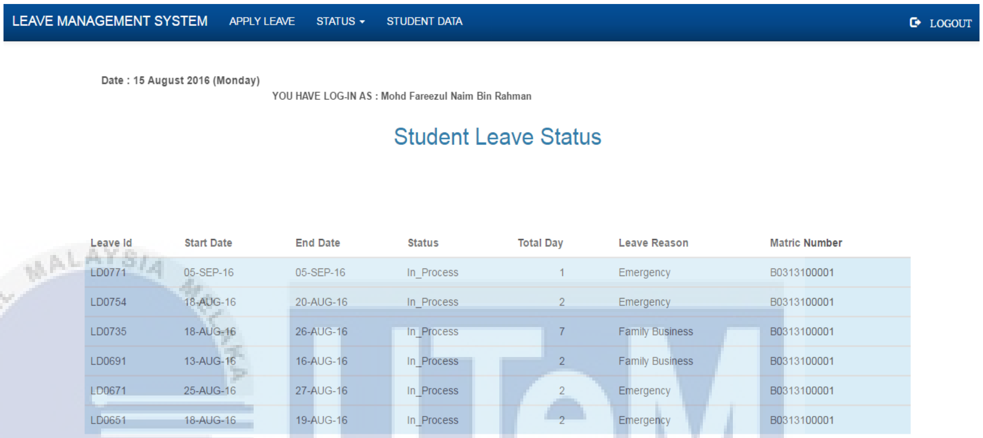


localhost says:

Your leave application have been succesfully registered

b. View Status

1. Click on status dropdown on the navigation bar and select In Process.
2. A table which contain data about leave that still in process will appear.



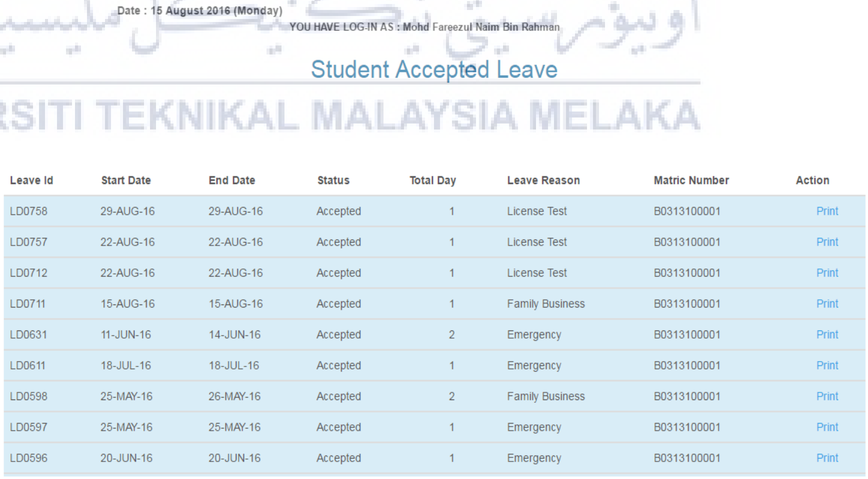
Date : 15 August 2016 (Monday)

YOU HAVE LOG-IN AS : Mohd Fareezul Naim Bin Rahman

Student Leave Status

Leave Id	Start Date	End Date	Status	Total Day	Leave Reason	Matric Number
LD0771	05-SEP-16	05-SEP-16	In_Process	1	Emergency	B0313100001
LD0754	18-AUG-16	20-AUG-16	In_Process	2	Emergency	B0313100001
LD0735	18-AUG-16	26-AUG-16	In_Process	7	Family Business	B0313100001
LD0691	13-AUG-16	16-AUG-16	In_Process	2	Family Business	B0313100001
LD0671	25-AUG-16	27-AUG-16	In_Process	2	Emergency	B0313100001
LD0651	18-AUG-16	19-AUG-16	In_Process	2	Emergency	B0313100001

3. Click on the status dropdown and select Accepted.
4. A table which contain data about leave that still in process will appear.



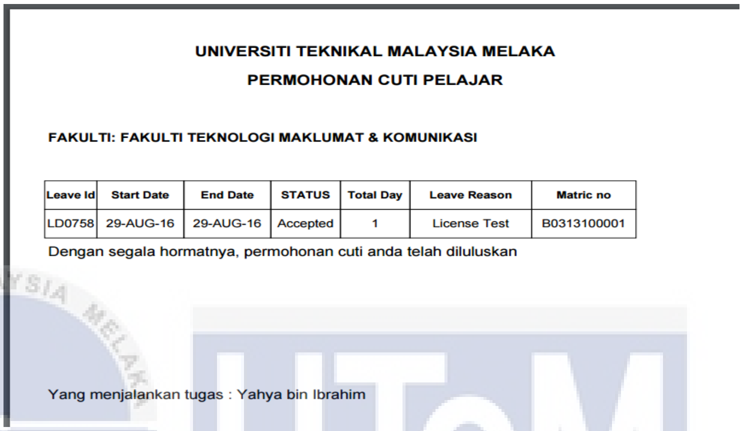
Date : 15 August 2016 (Monday)

YOU HAVE LOG-IN AS : Mohd Fareezul Naim Bin Rahman

Student Accepted Leave

Leave Id	Start Date	End Date	Status	Total Day	Leave Reason	Matric Number	Action
LD0758	29-AUG-16	29-AUG-16	Accepted	1	License Test	B0313100001	Print
LD0757	22-AUG-16	22-AUG-16	Accepted	1	License Test	B0313100001	Print
LD0712	22-AUG-16	22-AUG-16	Accepted	1	License Test	B0313100001	Print
LD0711	15-AUG-16	15-AUG-16	Accepted	1	Family Business	B0313100001	Print
LD0631	11-JUN-16	14-JUN-16	Accepted	2	Emergency	B0313100001	Print
LD0611	18-JUL-16	18-JUL-16	Accepted	1	Emergency	B0313100001	Print
LD0598	25-MAY-16	26-MAY-16	Accepted	2	Family Business	B0313100001	Print
LD0597	25-MAY-16	25-MAY-16	Accepted	1	Emergency	B0313100001	Print
LD0596	20-JUN-16	20-JUN-16	Accepted	1	Emergency	B0313100001	Print

5. If student click on print button, a pdf file will be auto-generated to print detail about their leave.



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

PERMOHONAN CUTI PELAJAR

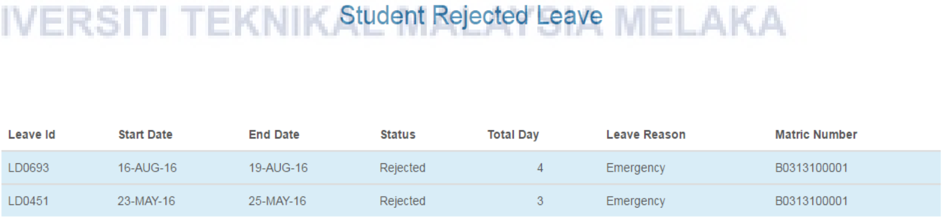
FAKULTI: FAKULTI TEKNOLOGI MAKLUMAT & KOMUNIKASI

Leave Id	Start Date	End Date	STATUS	Total Day	Leave Reason	Matric no
LD0758	29-AUG-16	29-AUG-16	Accepted	1	License Test	B0313100001

Dengan segala hormatnya, permohonan cuti anda telah diluluskan

Yang menjalankan tugas : Yahya bin Ibrahim

6. Click on the status dropdown again and select rejected.
7. All list of student rejected leave will be shown.



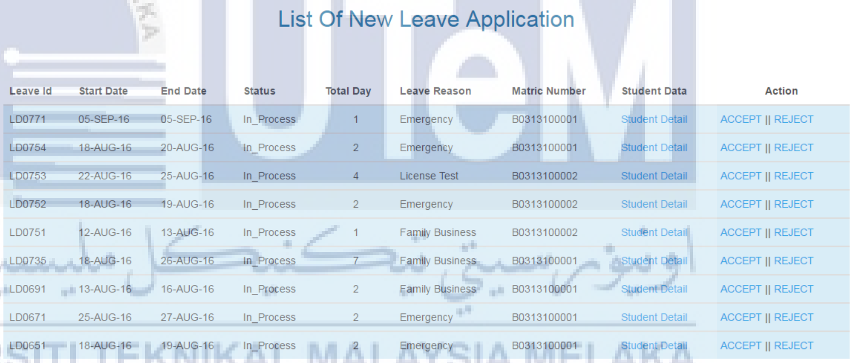
Student Rejected Leave

Leave Id	Start Date	End Date	Status	Total Day	Leave Reason	Matric Number
LD0693	16-AUG-16	19-AUG-16	Rejected	4	Emergency	B0313100001
LD0451	23-MAY-16	25-MAY-16	Rejected	3	Emergency	B0313100001

Admin

a. New Application.

1. If the user is admin, admin menu will be shown.
2. Admin can update student leave by clicking the button accept or reject.



List Of New Leave Application

Leave Id	Start Date	End Date	Status	Total Day	Leave Reason	Matric Number	Student Data	Action
LD0771	05-SEP-16	05-SEP-16	In_Process	1	Emergency	B0313100001	Student Detail	ACCEPT || REJECT
LD0754	18-AUG-16	20-AUG-16	In_Process	2	Emergency	B0313100001	Student Detail	ACCEPT || REJECT
LD0753	22-AUG-16	25-AUG-16	In_Process	4	License Test	B0313100002	Student Detail	ACCEPT || REJECT
LD0752	18-AUG-16	19-AUG-16	In_Process	2	Emergency	B0313100002	Student Detail	ACCEPT || REJECT
LD0751	12-AUG-16	13-AUG-16	In_Process	1	Family Business	B0313100002	Student Detail	ACCEPT || REJECT
LD0735	18-AUG-16	26-AUG-16	In_Process	7	Family Business	B0313100001	Student Detail	ACCEPT || REJECT
LD0691	13-AUG-16	16-AUG-16	In_Process	2	Family Business	B0313100001	Student Detail	ACCEPT || REJECT
LD0671	25-AUG-16	27-AUG-16	In_Process	2	Emergency	B0313100001	Student Detail	ACCEPT || REJECT
LD0651	18-AUG-16	19-AUG-16	In_Process	2	Emergency	B0313100001	Student Detail	ACCEPT || REJECT

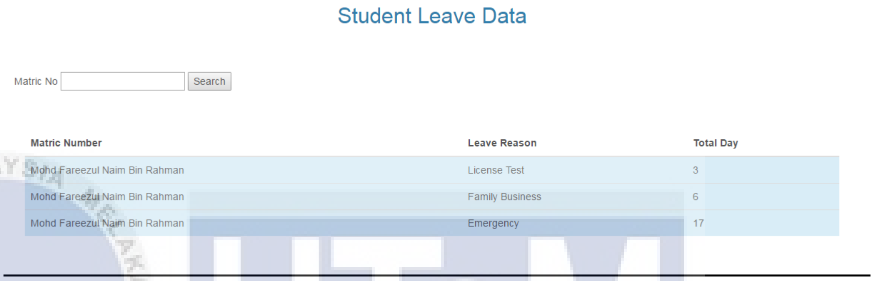
3. If admin click the accept button, the status will be change and email will be sent to student and lecturer.

b. Application Status.

1. This module is the same like student module. Just select which option admin want to view and system will display the data in table form.

c. Student Data.

1. When admin click on student data, admin will be directed to interface with a search box.
2. Enter the student matric number and all leave data about that student will be display.



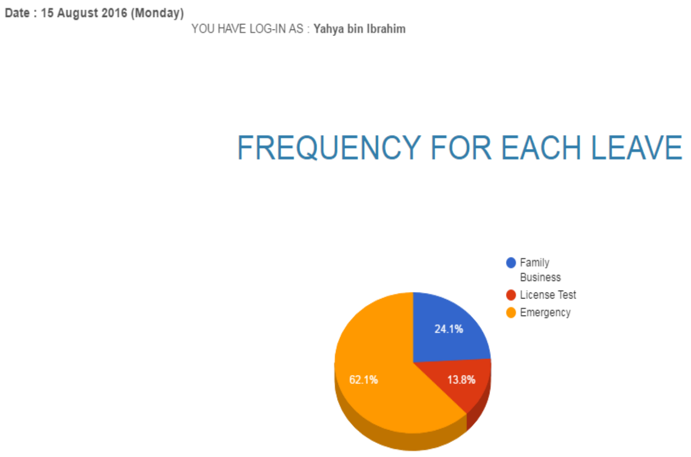
Student Leave Data

Matric No [   ] Search

Matric Number	Leave Reason	Total Day
Mohd Fareezul Naim Bin Rahman	License Test	3
Mohd Fareezul Naim Bin Rahman	Family Business	6
Mohd Fareezul Naim Bin Rahman	Emergency	17

d. Analysis.

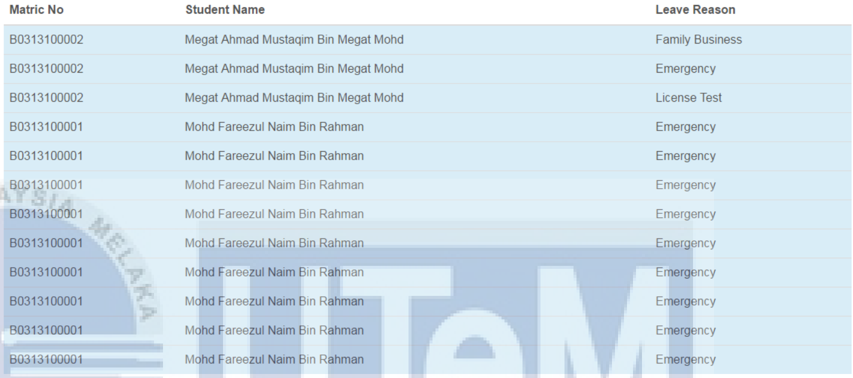
1. Click on analysis dropdown on the navigation bar and select total leave analysis.
2. Admin will be directed to a page where a pie chart and a table about the total of student applied based on leave type.



Date : 15 August 2016 (Monday)

YOU HAVE LOG-IN AS : Yahya bin Ibrahim

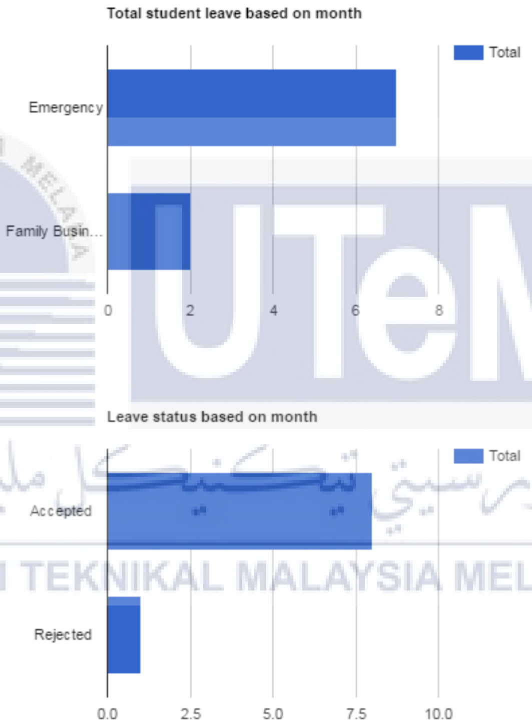
FREQUENCY FOR EACH LEAVE



Matric No	Student Name	Leave Reason
B0313100002	Megat Ahmad Mustaqim Bin Megat Mohd	Family Business
B0313100002	Megat Ahmad Mustaqim Bin Megat Mohd	Emergency
B0313100002	Megat Ahmad Mustaqim Bin Megat Mohd	License Test
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency
B0313100001	Mohd Fareezul Naim Bin Rahman	Emergency

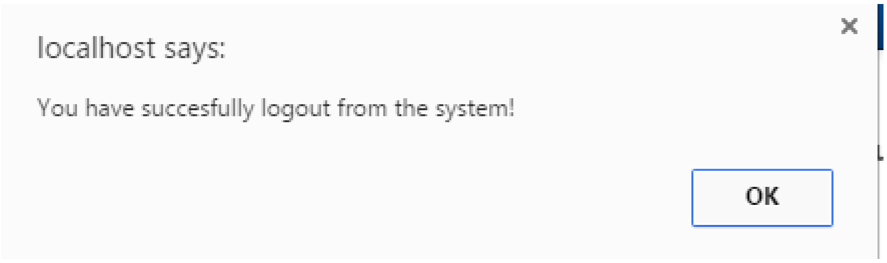
c. Monthly Analysis.

1. If admin select monthly analysis option, search box asking for month and year will be display.
2. Select desired month and year and click search button.
3. The bar graph will appear indicating number of student have applied for leave on that month and year will be display.
4. The number of rejected and accepted leave will also be presented in the form of bar graph.



User Logout

1. When user click the logout button on the navigation bar, user will direct out from the system and session will be destroy.
2. If user wish to use the system again, user will need to login again into the system.
3. A pop pup message will appear after user click the logout button.


localhost says:

You have succesfully logout from the system!