CAR PLATE NUMBER RECOGNITION (CPNR) DATABASE SYSTEM

NAQIBAH BINTI ZED

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: Car Plate Number Recognition (CPNR) Database System

SESI PENGAJIAN: 2015/2016

Saya NAQIBAH BINTI ZED

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

1. Tesis dan projek adalah hakmilik 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

_____ _____

Alamat tetap:
LOT 3399 KAMPUNG MELAYU
SUNGAI KROH AIR KUNING,
31920 KAMPAR PERAK
Tarikh:23 AUGUST 2016

Nama Penyelia:
DR NURUL AKMAR BT EMRAN

Tarikh: 23 AUGUST 2016

CAR PLATE NUMBER RECOGNITION (CPNR) DATABASE SYSTEM

NAQIBAH BINTI ZED

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2016

DECLARATION

I hereby declare that this project report entitled

CAR PLATE NUMBER RECOGNITION (CPNR) DATABASE SYSTEM

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

STUDENT : _____ Date : _____

(NAQIBAH BINTI ZED)

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

SUPERVISOR : _____ Date : _____

(DR NURUL AKMAR EMRAN)

DEDICATION

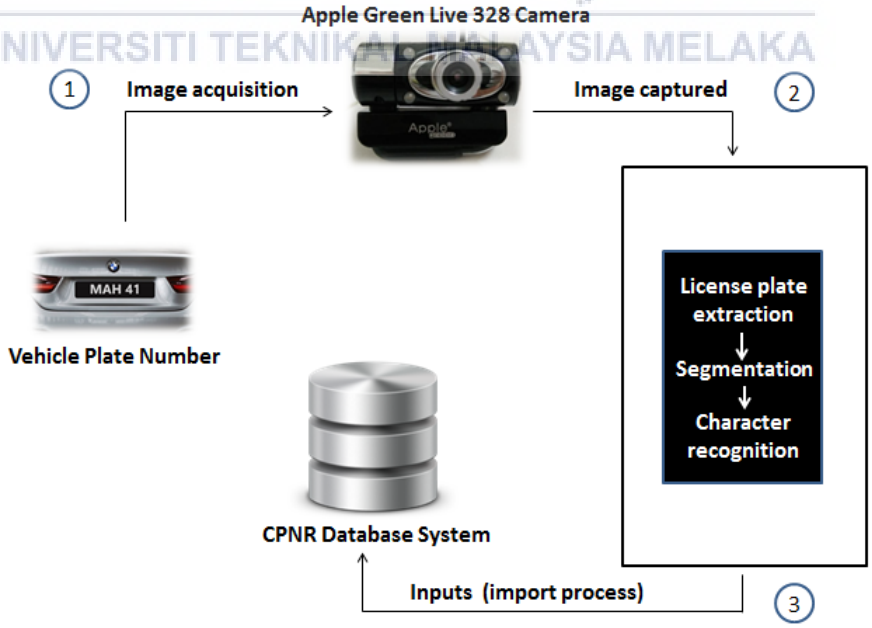
This dedication I dedicate to the people who helped in preparing this final year project. All guidances from them are given many benefits to me until the project is completed. Thanks to my mother Maimun binti Abd Talib as well as families and friends of the Faculty of Information Communication and Technology who involved directly or indirectly. Encouragement and passion that has given cause me to do this project well from beginning until the end. Next, thanks also for the recommendations which are very creative in producing a useful system. Besides that, do not forget to my supervisor, Dr Nurul Akmar binti Emran whose provides useful insights, advice and critism to build Car Plate Number Recognition (CPNR) database system that has many functions to the public. Finally, thanks to all attentions that has been given throughout under supervision Dr Nurul Akmar.

ACKNOWLEDGEMENTS

First of all, I wish to express my deep sense of gratitude and indebtedness to my respected supervisor, Dr Nurul Akmar binti Emran for the inspiration, guidance, well wishes and encouragement. She has been extremely helpful and give me inspiration of pursuing this project work and guide as an endeavor. She has been a constant source of motivation and encouragement for me. I have also thankful to all the faculty member of the department of Information and Communication Technology for bringing out best help during the period of my project work.

ABSTRACT

Car Plate Number Recognition (CPNR) database system is developed to help the security guard to monitor that every vehicle that arrived at UTeM is a registered vehicle. This system provides detailed information about the vehicle and its owner while the number after the plate number is detected. So, in order to enable this process to run smoothly, some information must be stored in the database. Among them are vehicle owner information, sticker registration and time in and time out. Thus, this system provides a number of forms to be filled online. Users also can make sticker registration through this system at any time. In addition, this system also has the functions to check summons, owner information management, sticker registration management, send an alert via email and generate reports. Furthermore, this system is used by two different users, which are staff and owner. Owners are student, staff or outsiders who work in UTeM for a certain time, while the staff is who managing the information. Overall, this system should be developed to ensure all the vehicles entered in UTeM are registered. Below is the framework of the integration between recognition camera and the database system.



ABSTRAK

Sistem Pengecaman Plat Kereta dibangunkan adalah bertujuan untuk membantu pihak keselamatan untuk memastikan setiap kenderaan yang masuk di UTeM adalah kenderaan berdaftar. Sistem ini akan memberikan maklumat terperinci mengenai kenderaan dan pemiliknya semasa nombor plat dikesan. Bagi membolehkan proses ini berjalan dengan lancar, beberapa maklumat perlu disimpan di dalam pangkalan data. Antaranya maklumat pemilik kenderaan, pendaftaran pelekat kenderaan dan waktu keluar dan masuk UTeM. Jadi, sistem ini menyediakan beberapa borang yang perlu diisi atas talian. Pengguna juga boleh membuat pendaftaran pelekat kenderaan melalui sistem ini pada bila-bila masa. Selain itu, sistem ini turut berfungsi untuk menyemak saman, pengurusan maklumat pengguna, pengurusan maklumat pelekat kenderaan, menghantar amaran melalui e-mel dan menghasilkan beberapa laporan. Disamping itu, sistem ini digunakan oleh dua pihak yang berbeza iaitu pengguna dan staf keselamatan. Pengguna adalah pelajar, staf atau orang luar yang bekerja di dalam UTeM manakalan staf adalah pihak keselamatan yang menguruskan maklumat-maklumat pengguna. Secara keseluruhannya, sistem ini wajar untuk dibangunkan bagi membantu pihak keselamatan memastikan kenderaan yang masuk di UTeM adalah berdaftar. Di bawah adalah rangka integrasi antara kamera dan sistem pangkalan data.

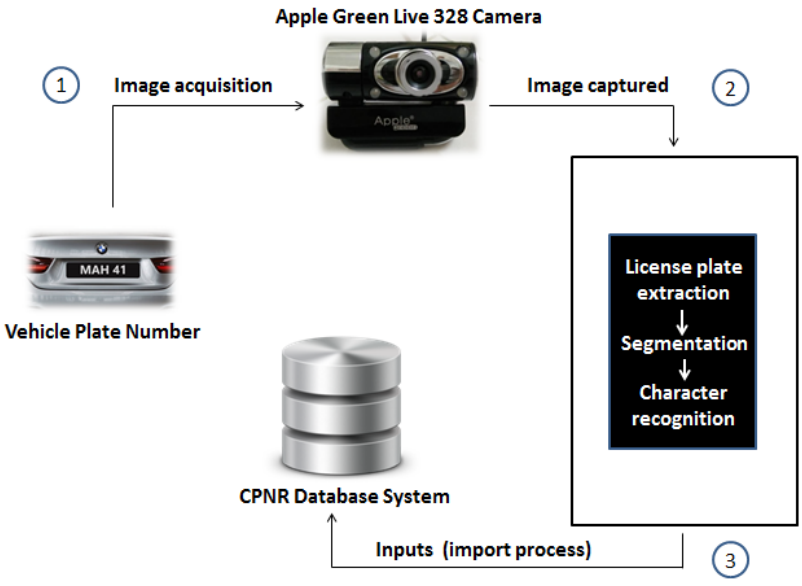


TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CPNR	-	Car Plate Number Recognition
ERD	-	Entity Relationship Diagram
DFD	-	Data Flow Diagram
CD	-	Context Diagram
DBLC	-	Database Life Cycle
SDLC	-	Software Development Life Cycle
GUI	-	Graphical User Interface
DDL	-	Data Definition Language
DML	-	Data Manipulation Language

CHAPTER I

INTRODUCTION

1.1 Project Background

The main purpose of this system is to develop a database in order to organize the information of vehicles and its owner. Nowadays, the access system by using license plate recognition has been done but the information only display in the monitor at the guard house. This information is limited because it can display the information at one place only and only security guard at there can know that information. According Tran *et al* (2005), any system that usually used in real-time systems, it requires not only accuracy but also fast processing. If this system is completed with online status monitoring application, it will be more efficient in term of checking the availability of

sticker by security guard. All of the security guard can monitor the movement in and out of the vehicles although they are not at the guard house.

The data is growing rapidly and require a management system for managing data properly. The challenging of management data is also one important aspect that needs to take into account. For example, the data comes from different data source of different types. So, with this database, it can be synchronized and put in one place that can later be retrieved by many people. Therefore, it is required to have one place to store all kinds of information which is called a database.

According to Robert (1994), a database is a persistent, logically coherent collection of inherently meaningful data, relevant to one aspects of the real world. Meanwhile, a database management system (DBMS) is a collection of programs that enables users to create and maintain a database.

In this project, a database for the information of vehicles and user will be built to make the data is available when needed. All the information of the vehicles will display via online. This will result in a better improvement in a current access system.

1.2 Problem Statement

There are many problems occur on the existing system, particularly in the access system management. Difficulty accessing data at a maximum level is often happens because of the data on paper and only be in one place only. Due to the inefficient access system management, there are some problems occur such as:

i. Having congestion at the main entrance.

The security guard still used the manual access system which is by checking the sticker on the vehicles to identify user identity by differentiate a student, staff or an outsider. As a result, this system will take a long time for the security to check the vehicles one by one. Besides that, it would cause difficulties to the various parties, especially to the security to know the user information based on the vehicle's number plate.

ii. No online database to keep all the information such as vehicle, user, summons and others

The efficiency to retrieve information is essential to ensure all the processes are running properly. So it is important if the database is realized in the form of online mode. The difficulty of obtaining data manually will also cause some problems such as the duplication of data, incorrect information, a slow process and will take a long time.

iii. Staff need to view and analyze the data on paper

The information also difficult to analyze because of the information is not in online mode. Staff needs to check the whole information on the paper of every sentence and make a report. This way will takes time and cause other errors that may occur from the dropout information.

1.3 Objectives

Based on the problem statement that has been stated above, the objectives of this project are:

i. To develop a database for storing the data in organized and be managed systematically.

The system need to store the information of vehicle and its user after the registration. User need to register the system to be authorized person. Then, they can insert their vehicle information as the way for vehicle registration in UTeM.

ii. To allow user to search for the availability of person in UTeM by searching using the plate number.

The purpose of the database is storing useful information to the end user. In this case, users easily make search in a situation where they want to know availability of person in this university.

iii. To allow online registration of owner's vehicle in anytime and at anywhere.

The system is developed to simplify many things. Among them is the registration sticker for those who bring vehicles into UTeM. Car Plate Number Recognition (CPNR) database system also provide registration stickers using this system in order to avoid congestion at the security office, making it easier for those who want to register at any time and improve the quality of the registration process.

iv. To produce a report by day and year

This system will help to produce a report from the various of information. Data collected will be presented in the graphic form and will help the user to better understand**.**

1.4 Scope

The scope for this system is divided into two categories which are system scope and user scope. The system scope explains about the functionality of what the system can do while the user scope is about the type of user and their tasks.

1.4.1 User Scope

The users are divided into three categories which are security guard, student and staff.

- Security guard

The security guard is responsible for detect sticker on vehicle by themselves. By using this system, it will simplify their task. For example, the camera will recognize the plate number and do some processing until it becomes readable text. Next, using the text it will search for user information and give a notification to the security guard. Then, Car Plate Number Recgnition (CPNR) database system provides access to the security guard in order to help them doing their work more effective and to avoid congestion at the main gate.

- Student and staff

The student and staff are allowed to use this system for several functions. There are sticker registration, search for user availability and view summons information. Car Plate Number Recognition (CPNR) database system provides a form for user to register the vehicle. Other than that, they are able to check the status availability of other person in UTeM. This function is provided especially for students to find the lecturer. They just need to enter some information needed by the system such as plate number to get the information required.

1.4.2 System Scope

The system scope is divided into seven modules which are login, user registration, plate number registration, vehicle information, staff and user information, summons and reporting.

1.4.2.1 Login

Login module is developing for all users of this system. They need to log into the system by key in their identity number and password. Login will authenticate the user either they are allowed to use the system or not. This module will be used by student and staff.

1.4.2.2 User registration

User registration module is developing for users who want to access the system need to be registered. For security guard or other staff they will be registered by the admin of this system for security reason. This module will be used by student and staff. User registration module also used stored procedure to join at least two tables to display information. Other than that, it used trigger to create unique id start with REG0.

1.4.2.3 Plate number registration

Plate number registration module allows every person who brings the vehicle into UTeM need to be registered in every year to allow only an authorized vehicle enters the UTeM. This system also provides the registration form in order to help them register in any time at anywhere. This module will be used by owner of the vehicle in order to register their vehicle.

1.4.2.4 Vehicle information

Vehicle information module allows user to view their vehicle information throughout this system. Besides that, they are able to search the status of vehicle whether it is in or outside UTeM. This module will be used by admin and user of the system. They can insert, update or delete the information. Vehicle information module also used stored procedure to join two tables, insert and delete.

1.4.2.5 Staff and user information

Staff and user information module shows a list of staff and user that register the system in order to get the right access. There are several information that be kept into database such as name, address, phone number, matric or staff number, email and image. This module will be used by user of the system. They are allowed to view and update the information. Staff and user information modules also used stored procedure to view, insert, update and delete. A trigger also executed for table staff to generate unique id that start with S00.

1.4.2.6 Summons

Summons module is developing to help the security to report on drivers who break the rules. Summons's form is provided in the system. Users also are allowed to view their summons information in details. This module will be used by security to insert summons and owner to view the summons. The summons module also used trigger to generate summons id and a trigger to generate unique id that start with SM0.

1.4.2.5 Reporting

Reporting module is develop for staff to track which user get into UTeM, how many times per day user entrance in UTeM, and the number of registration per year. These reporting will help staff to view the trend day to day and make analysis on that. The reporting also used stored procedure in order to generate the graph in term of data to be viewed.

1.5 Project Significant

Throughout some study for the development of Car Plate Number Recognition (CPNR) database system, it will benefit to everyone, especially the students and staff at UTeM. They are the main consumers of help in developing useful information.

In addition, this Car Plate Number Recognition (CPNR) database system enables them to register a vehicle in a systematic and process will be rapidly. This system provides a form to be filled on the vehicle information as well as the owners. The information has been filled will be saved into the database and so on admin will continue the registration process. Through this system, users can also find out the status of vehicle registration and stickers start and end dates. Besides that, the system will notify the user to renew their license if the expiration date has been reached.

Next, it is useful to staff who manage this system. Through this Car Plate Recognition Database System, all the information are easier to manage, especially taking care of a lot of information and may be the information is likely to overlap. The staff also can check whether the vehicle has been registered or not. Finally, it simplifies the search process.

1.6 Expected Output

Output 1: List of reports that can be generated from the input data.

Output 2: An automatic notification for the license expiry.

Output 3: Searching result based on one or more tables which is generated by join queries

1.7 Conclusion

At the end of the project, the developer needs to achieve the objectives that have been list out in the objectives section in this proposal. There are to develop a database to store the users and vehicles information in able to manage data systematically, to allow user to know the availability of person in UTeM by searching using the plate number and to help person in registration their vehicle in anytime and at anywhere. This system should be developed in more systematic way to make user easy to use so that any problems occur can be avoided. Throughout this database driven, the information can be retrieve by any authorized user. Therefore, database design as well as user interface need to perform in order to clarify the requirements of the system.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

This chapter covers the description on development methodology that is used in this project. Many methodologies outside there such as Agile, Waterfall, Rapid Application Development, System Development Life Cycle (SDLC) and Extreme Programming (EP) are useful in developing a system and it is depends on someone who want to work with. In this project, the methodology is based on the Database Life Cycle (DBLC). The method is use to achieve the objectives of the project that will accomplish a correct result. Generally, the steps are database initial study, database design, implementation and loading, testing and evaluation, operation and maintenance and evolution.

2.2 Project Methodology

This project used four major steps to implement project starting from database initial study until the evaluation. All the methods used for finding and analyzing data regarding the project related.

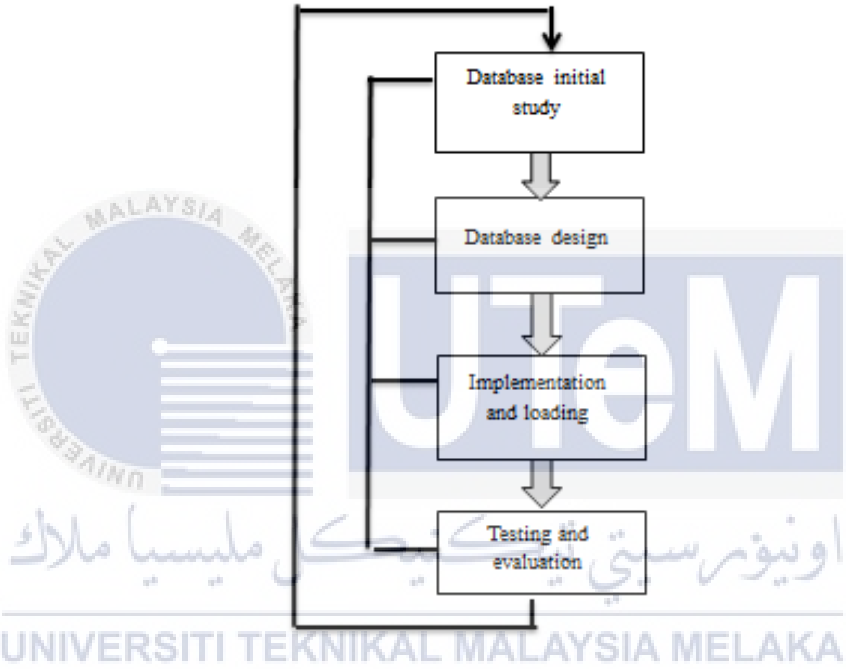


Figure 2.1 Database Life Cycle (DBLC)

2.2.1 Database Initial Study

In this step is to identify all the requirements and information such as hardware, software and planning must be done in the proper manner. This stage also requires analyzing the existing of car plate number system, defining what are the problems occur from the current system, state the objectives, scope and boundaries.

2.2.2 Database Design

Design is the process which is to visualizing the flow or data of the system by using Data Flow Diagram (DFD). Design is the alternate solution to give the explanation of the system workflow. In this case, to achieve the objectives, design can be part of the solution. It is easy to understand, manage and simple to work.

The database design phase is divided into three steps. There are conceptual database design, logical database design and physical database design. The purpose of conceptual database design is to build a conceptual model of Car Plate Number Recognition (CPNR) database system which the output of this step is Entity Relationship Digram (ERD) and normalized relation.

The second step is logical database design which is an agreement of definition and business logic. CPNR database system is logically explained through this stage by specifying data and queries based on information that user want. The output for this stage is relational data model. This step also includes the selection on DBMS that will be used.

The last step is physical database design that represents core business rules and data relationships at a detailed level. CPNR is presented in tables, columns, indexes, sequences and constraint after grouping attributes from the logical database. The output for this stage is technical specifications used during the implementation phase.

2.2.3 Implementation and Loading

In this phase, the DBMS software is selected earlier during logical design. The Entity Relationship Diagram (ERD) needs to be translated to the targeted DBMS that will be installed into the computer. The DBMS used is PostgreSQL while the language is Hypertext Preprocessor (PHP). The database produced and data is put into the

database tables. Then, user will add some future advantages such as set up the security standard, data integrity enforcement and place the backup and recovery procedures.

2.2.4 Testing and Evaluation

The database is tested for the performance to ensure all of the requirements achieved and to make sure the system is working well. It is tested during implementation and loading. However it is tested again and fine-tuned during testing. Lastly, the Car Plate Number Recognition (CPNR) database system will be evaluated its database and its application programs by the specific tester and be given a form to give the feedback.

2.3 Project Schedule and Milestones

Table 2.1 shown the project schedule and milestones for this project

Table 2.1 Project Schedule and Milestones

Milestones	Expected Documents	Dates
CPNR Project Planning	1. Identify all the requirements for CPNR system 2. State the problem, objective, scope and purpose	22 February 2016
CPNR Project Analysis	1. Analyst the project requirements in details 2. Prepare the project schedule and Gantt Chart 3. Analyst the current system in	2 March 2016

	market 4. Decide the hardware and software needed in the development of the project	
CPNR Project Design	1. Design the data flow of the project by design the Entity Relationship Diagram (ERD) and Data Flow Diagram (DFD) 2. Design the flowchart of the project	29 April 2016
CPNR Project Development and Implementation	Start with design interface and develops the project coding	19 May 2016
CPNR Project Testing	Test the system whether it is fulfill at the requirements of the project	1 June 2016
CPNR Project Completion	Final phase that have complete all functional, documentation and ongoing system through the life of the service	8 June 2016

2.3.2 Gantt chart

Figure 2.2 shows the Gantt chart for the project schedule of Car Plate Number Recognition (CPNR) database system

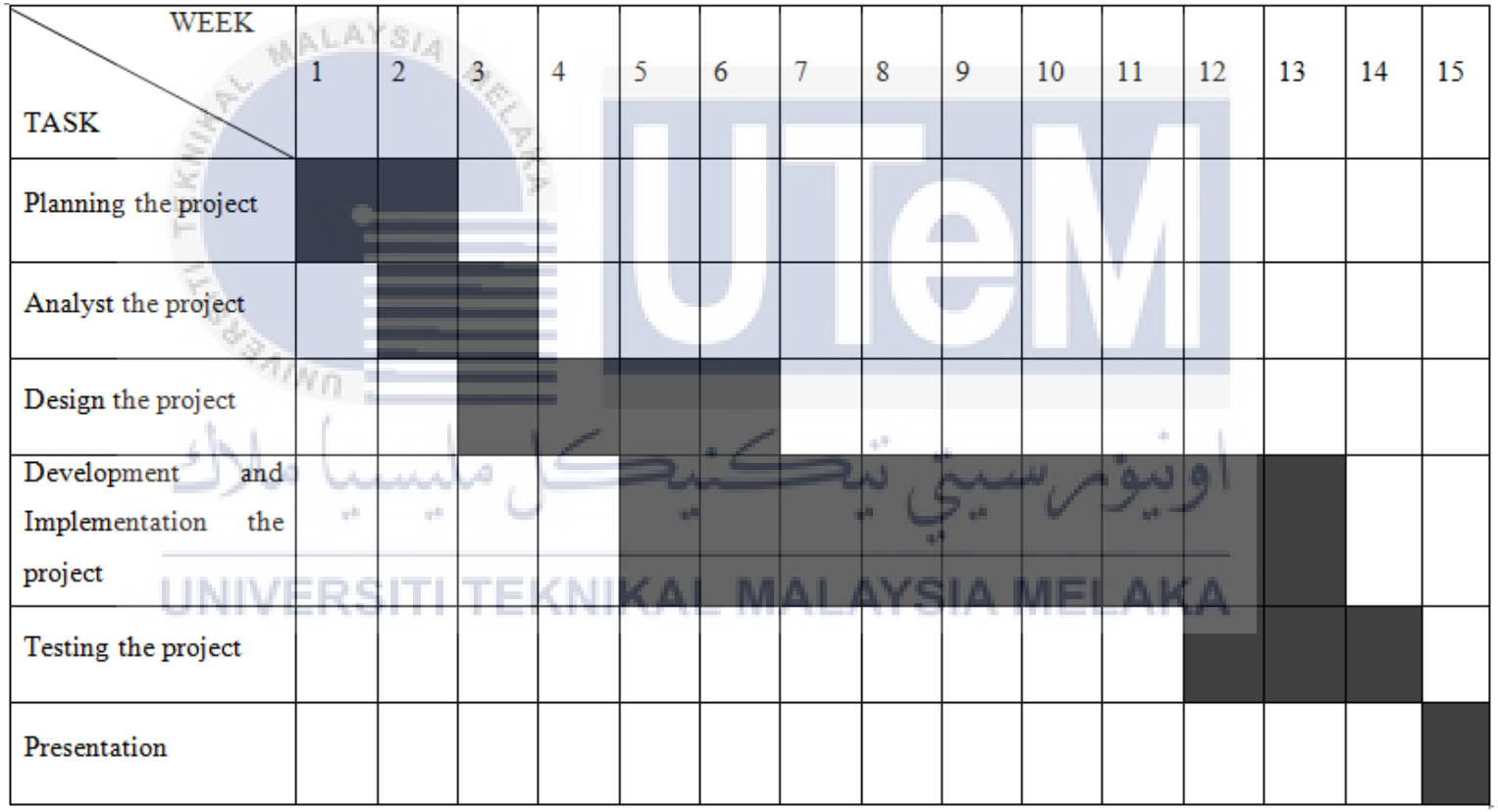


Figure 2.2 Gantt Chart

2.4 Conclusion

As a conclusion, every project has a different methodology that is being used to make the project success and working well. This project used Database Life Cycle (DBLC) as database system development methodology to achieve the objectives of the project that will accomplish a perfect result. This project methodology and planning explains in details all the DBLC stages that are used for this project.

CHAPTER III

ANALYSIS

3.1 Introduction

Database system in analysis phase is concerned with the identification and problems of the data element which is needed to support the data processing system of an organization, the placing of these elements into logical groups and the relationships between the resulting groups. System analysis often goes directly from fact finding to implementation dependent data analysis. Thus, Car Plate Number Recognition (CPNR) system is chosen to be the fact finding source. From the information provided by the respondent, the data is analyzed and build a system representation in the form of a conceptual data model which specifies the structure of the data and the processes which use the data.

3.2 Problem Analysis

The vehicle sticker is a good way to categorize a person registered or not registered. Each person must register vehicles to enable vehicle brought into the UTeM area. Security guard also needs to ensure that every vehicle that comes in is registered by checking the sticker affixed to the windscreen of the cars or motorcycle.

But, there are some problems occur from the old system where there is no data that can be retrieving at that time. The security guard manually checks the vehicle one by one and it will cause congestion if there are many vehicles at the same time. This system will provide an effective way for them to recognize the vehicle and avoid some problems there. Below is the flow how this old system works

3.2.1 Flowchart description

Based on the flow of the current system, there are two categories which are student site and staff site. Each of them has different flow of process and will be explained below.

Figure 3.1: Sticker registration by user

The process of car's owner registers a sticker. Each registration sticker, users must register at the security office. They have to fill in the form provided. In addition, copies of IC and licenses must be submitted with this form and they will be charged a fee of RM1 for each application. The form is sent to the officers. When all processes are complete then the sticker will be given to the user.

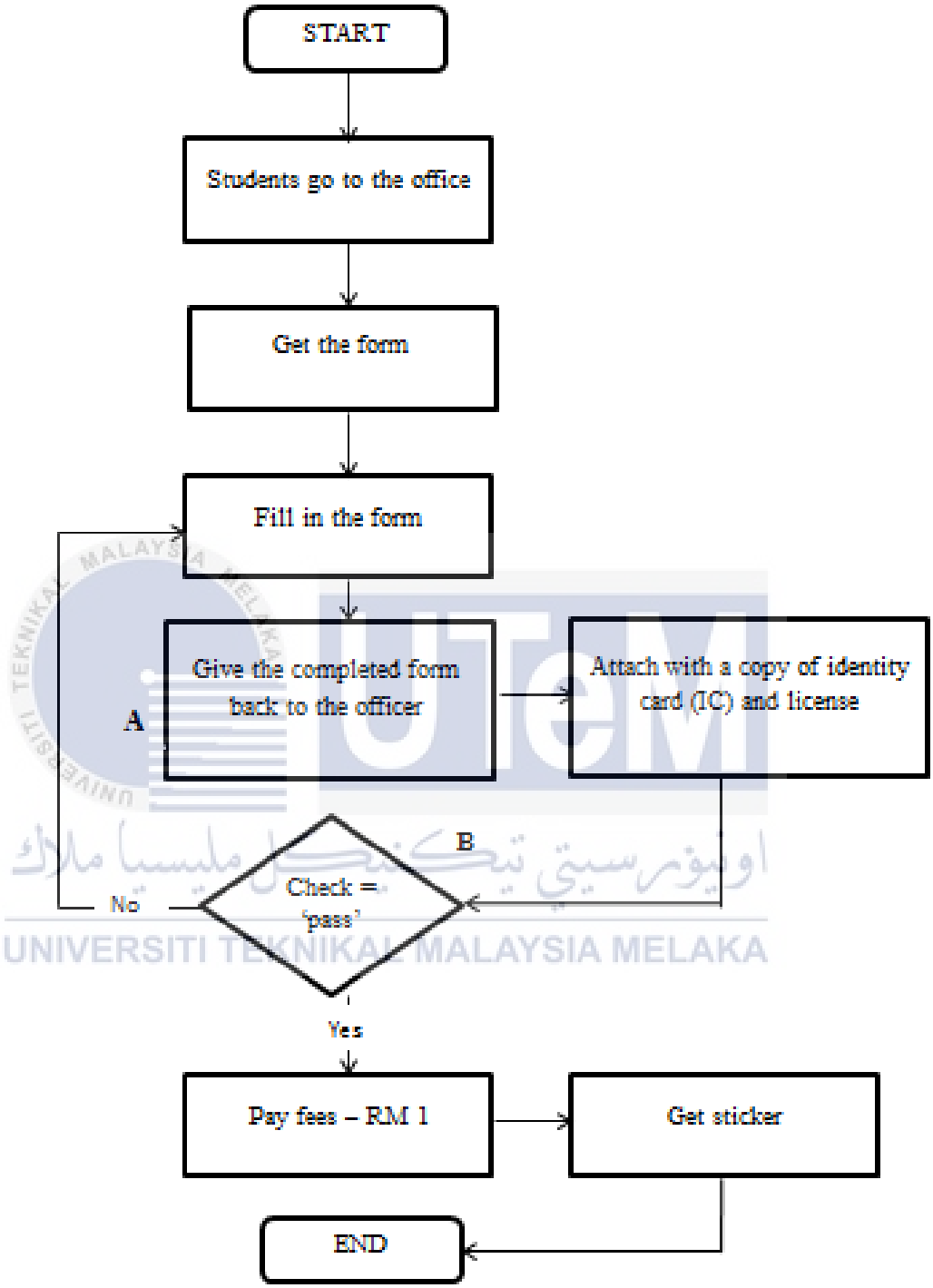


Figure 3.1 Flow Chart of Current System (Sticker Registration by User)

Figure 3.2: Sticker management by staff

The process to manage the sticker. First, employees must provide registration forms where on the form contains a space to be filled in as the user information, vehicle, license and other information. Forms that have been printed will be given out to users to complete. Then, once completed, the form will be reviewed and if all the information, students make a payment of RM1. The forms will be put in one place for use when necessary.

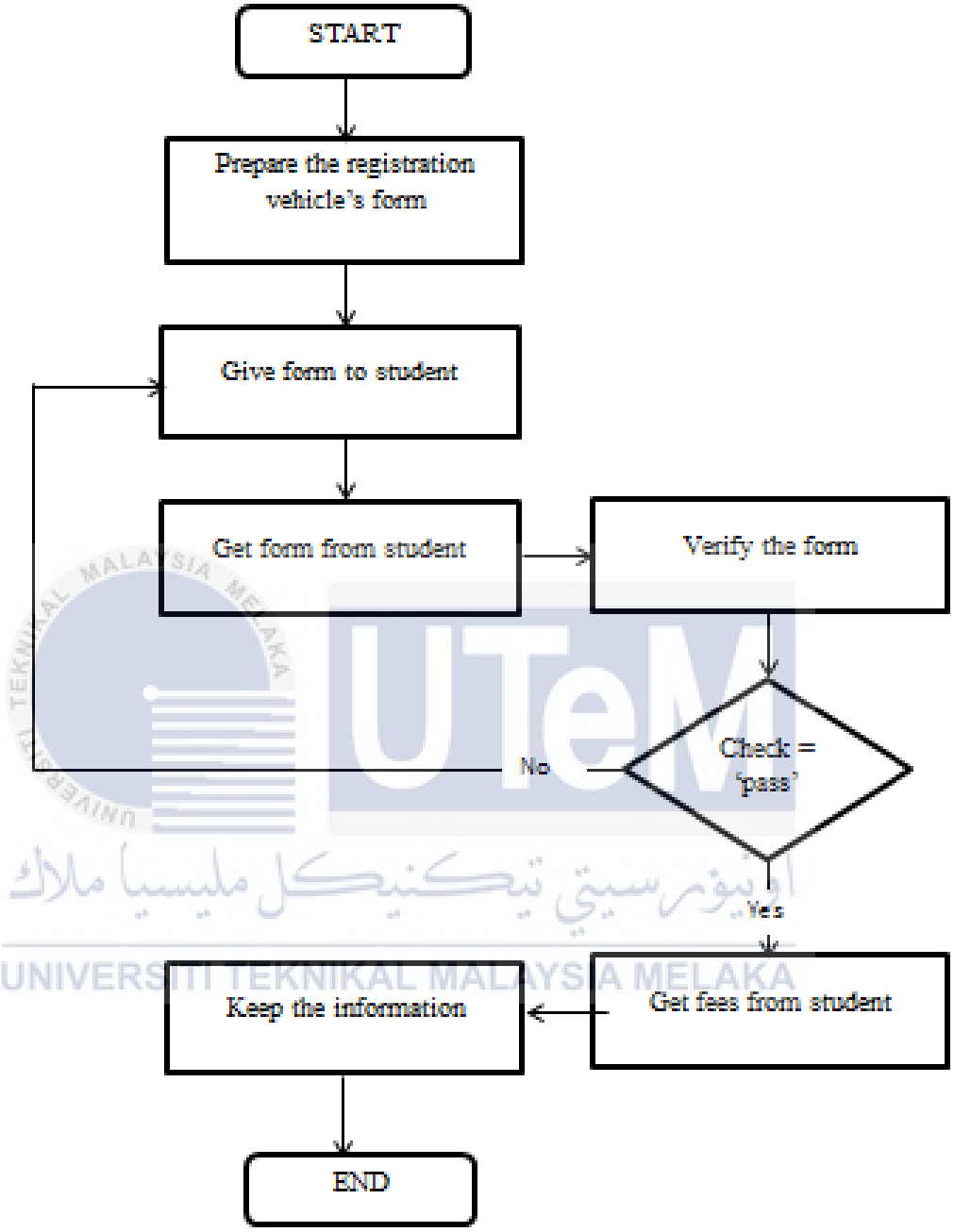


Figure 3.2 Sticker management by staff

Figure 3.3: Sticker Identification by Security Guard

This process is initiated by security guards who manage the online identification of sticker. According to current system, there is no online database system to store user information. So, it is hard for security guard to identify an authorized user. The current system shows the vehicles move into UTeM gate and security guard will recognize each of the vehicles by checking the availability of the sticker. Any user who has the sticker is permitted to enter UTeM area while on the other hand, they are asked to provide an explanation.

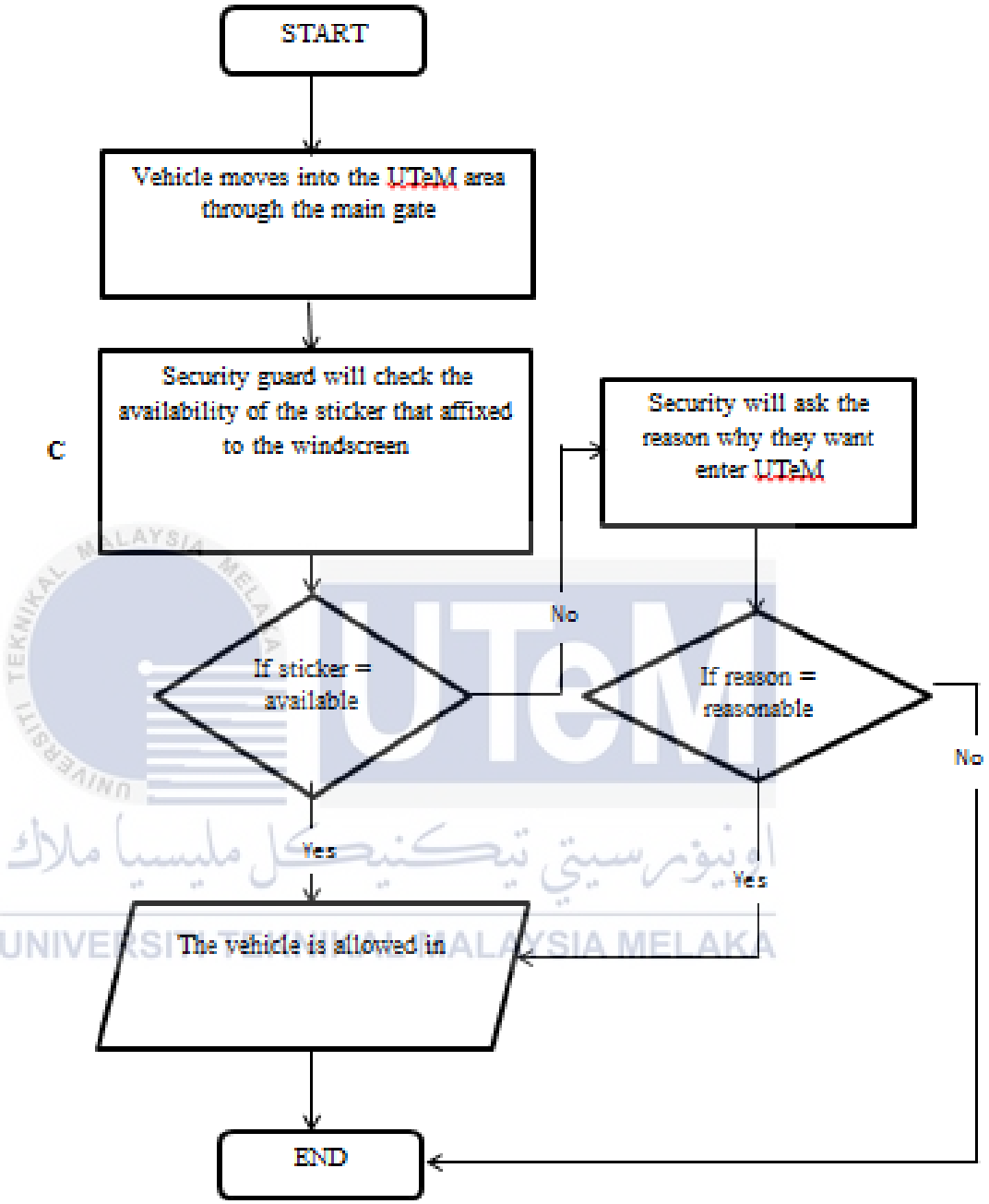


Figure 3.3 Flow Chart of Current System (Sticker Identification by Security Guard

3.3 The proposed improvements or solutions

According to the current manual system, a database system should exist to deal with the problems that occur. In addition, this CPNR Database System is able to store and manage data to support an orderly management. So a centralized system should be established to solve these problems. From the current system, there are some processes need to improve on its management. Below are the identify problem and suggestion for improvements:

A - Give the completed form back to the officer

Problem:

It is start from users need to go to security office. Users get the form and fill the requested information. After that, submit it to the officer and pay.

Suggestion of improvement:

To avoid wasting time and speed up the process, a system should be developed for systematic data management purposes. The system will facilitate access to information from the user and the information can be better managed.

B - Check = "pass"

Problem:

Officer themselves need to check one by one the registration form that given by users. It is takes time and possibility of mis-check.

Suggestion of improvement:

Officer does not need to check the information otherwise the system will process the information based on criteria that are required. The criteria will be added into the system so that each form is reviewed in accordance with the guidelines that have been set.

C - Security guard will check the availability of the sticker that affixed to the windscreen

Problem:

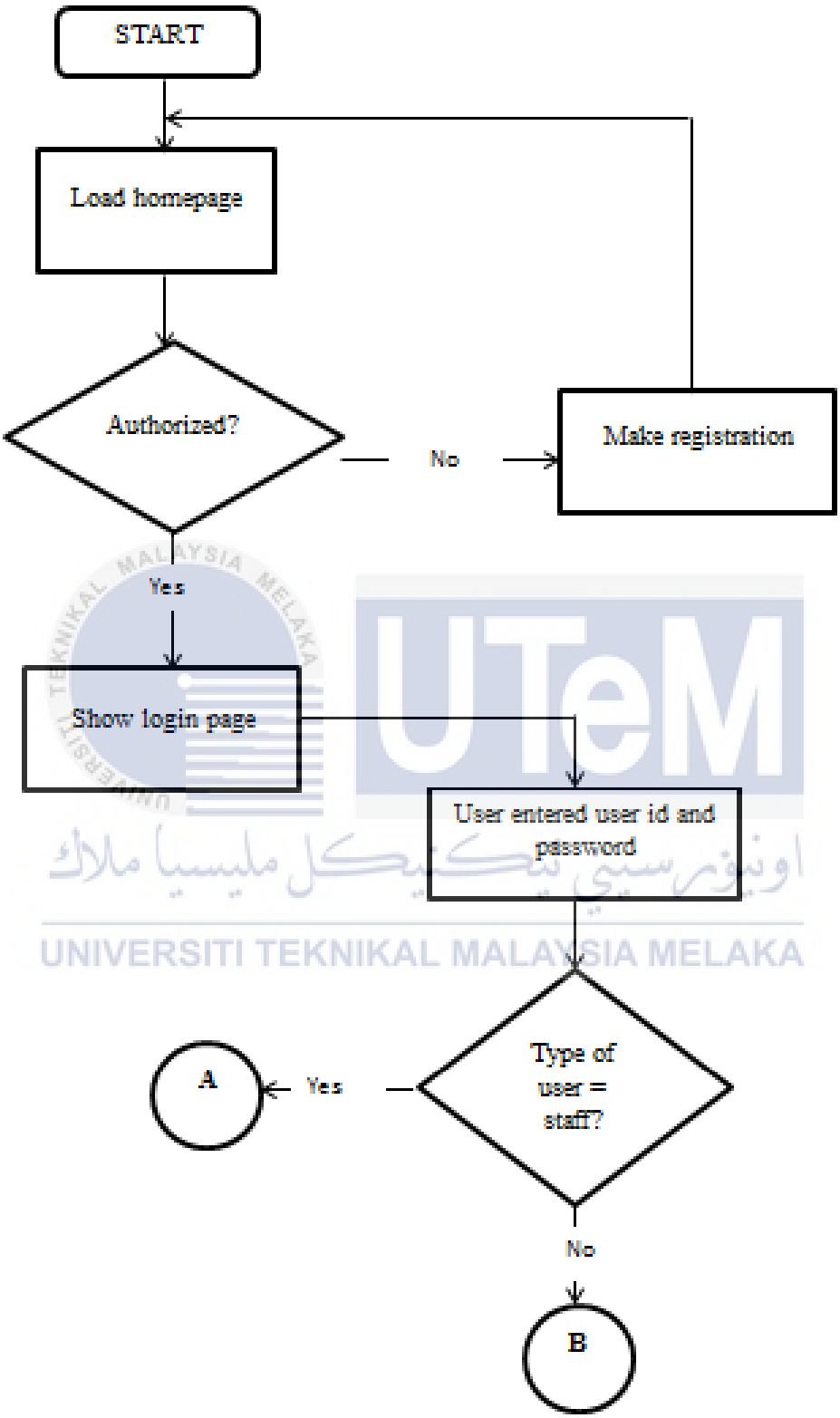
The problem occurs when there is no centralized data so that the security guard needs to check the vehicle on by one.

Suggestion of improvement:

Make a system in order to centralize the data. Security only needs to monitor the vehicles from inside through the system. The system also provides information on registered user, vehicle information and the record entry and exit.

Thus, the vehicle management system should be established which aims to manage vehicle data, owner, summons and able to record entry and exit of vehicles in UTeM. This system is mainly designed to simplify security guard to identify vehicles that have been registered or not. In addition, the system will be used by the student, lecturer or other type of staff and this system is managed by staff.

Figure 3.3.1 shows the activity takes in the CPNR Database System. Each module is shown in the diagram for more understanding the flow of the process.



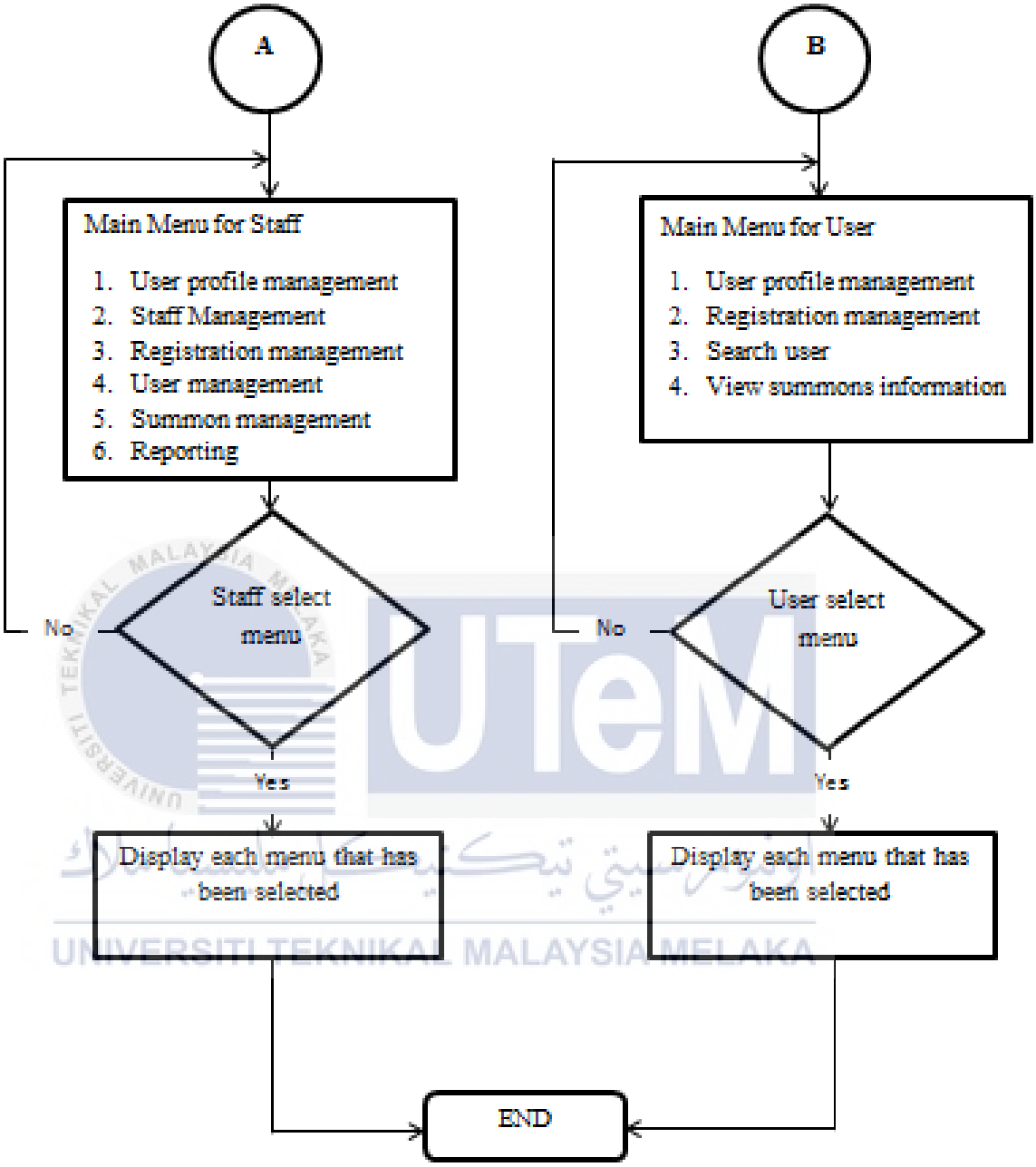


Figure 3.4 Flow Chart of Proposed System (CPNR Database System)

3.4. Requirement analysis of the to-be system

In this section, there are two types of requirement which is functional requirement and non-functional requirement.

3.4.1 Functional requirement (process model)

The functional requirement is a requirement that the main things that user expects from the software or system. It describes the behavior of the system rather than how the system operates.

3.4.1.1 Functions of the system

Table 3.1 Function of the System

FR_No	Requirement	Description
1	Registration	- The system provide registration for new users
2	Login	- The system provide login before user access the information
3	User Profile Management	- Admin can insert, update, delete and view user information - User only can register , update and view their profile
4	Vehicle's Registration Management	- The system allows user to make vehicle's registration - The system has the functionality of update, view and delete the registration information
5	Summons Management	- The system provide information of summons - The system allow user to add, update and delete summons's information
6	Car Daily Track Management	- The system allows user to insert, delete and view car daily track
7	Reporting	- The system will generate various type of reporting such as bar chart and table. - The report will shows many of data in simplified way to be more understandable.

3.5 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output.

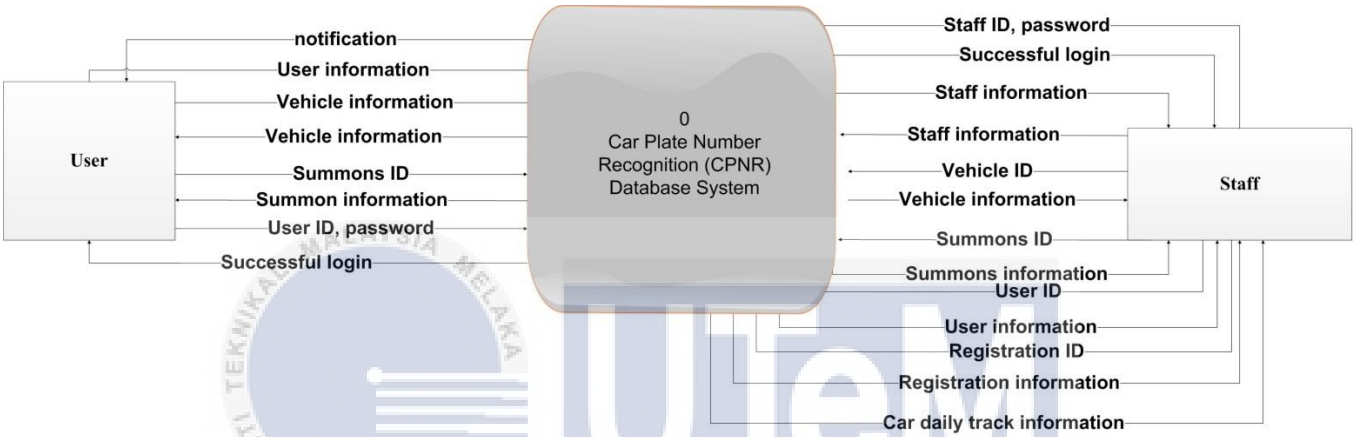


Figure 3.5 Context Diagram

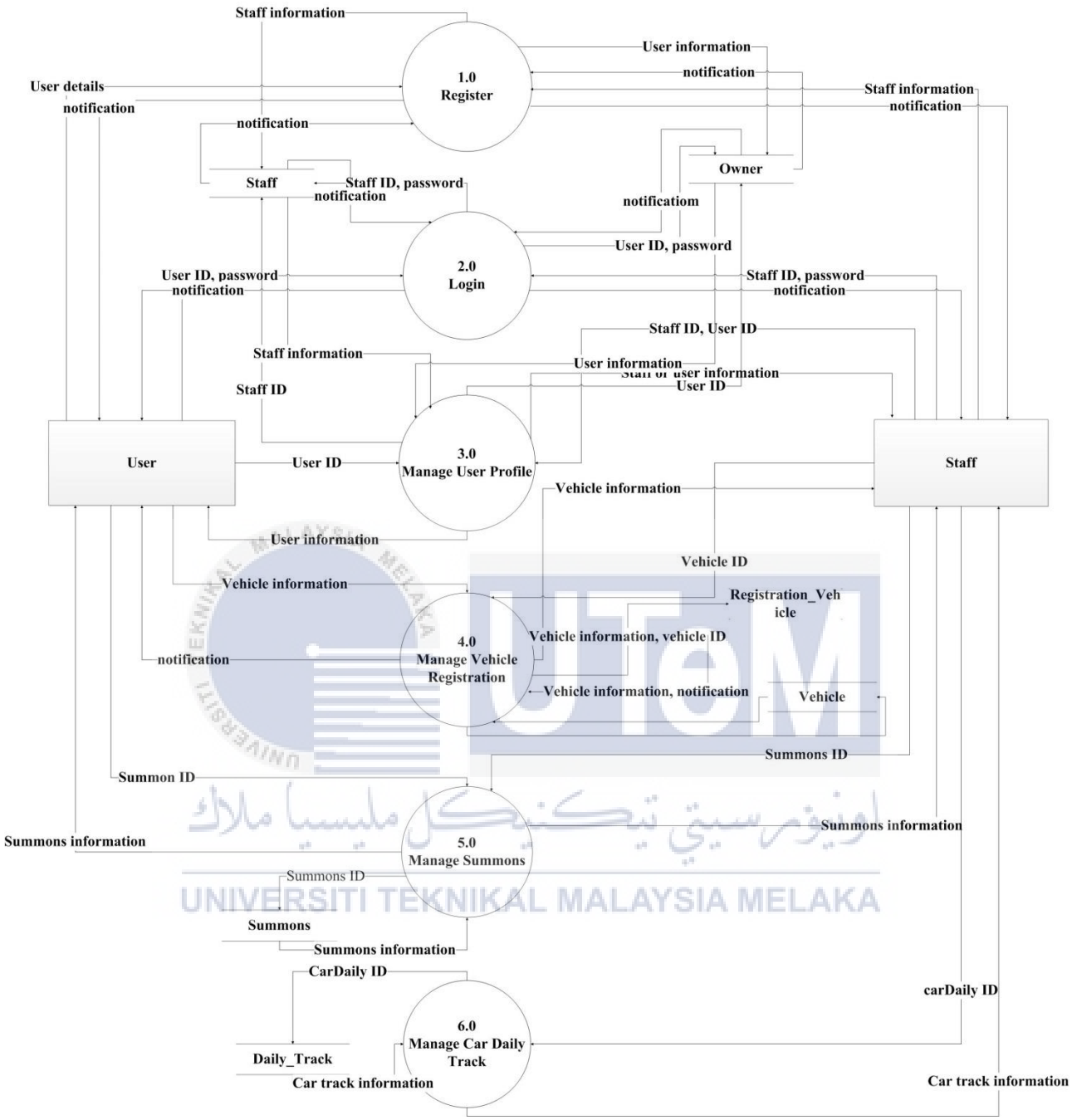


Figure 3.6 Data Flow Diagram (DFD) - Level 0

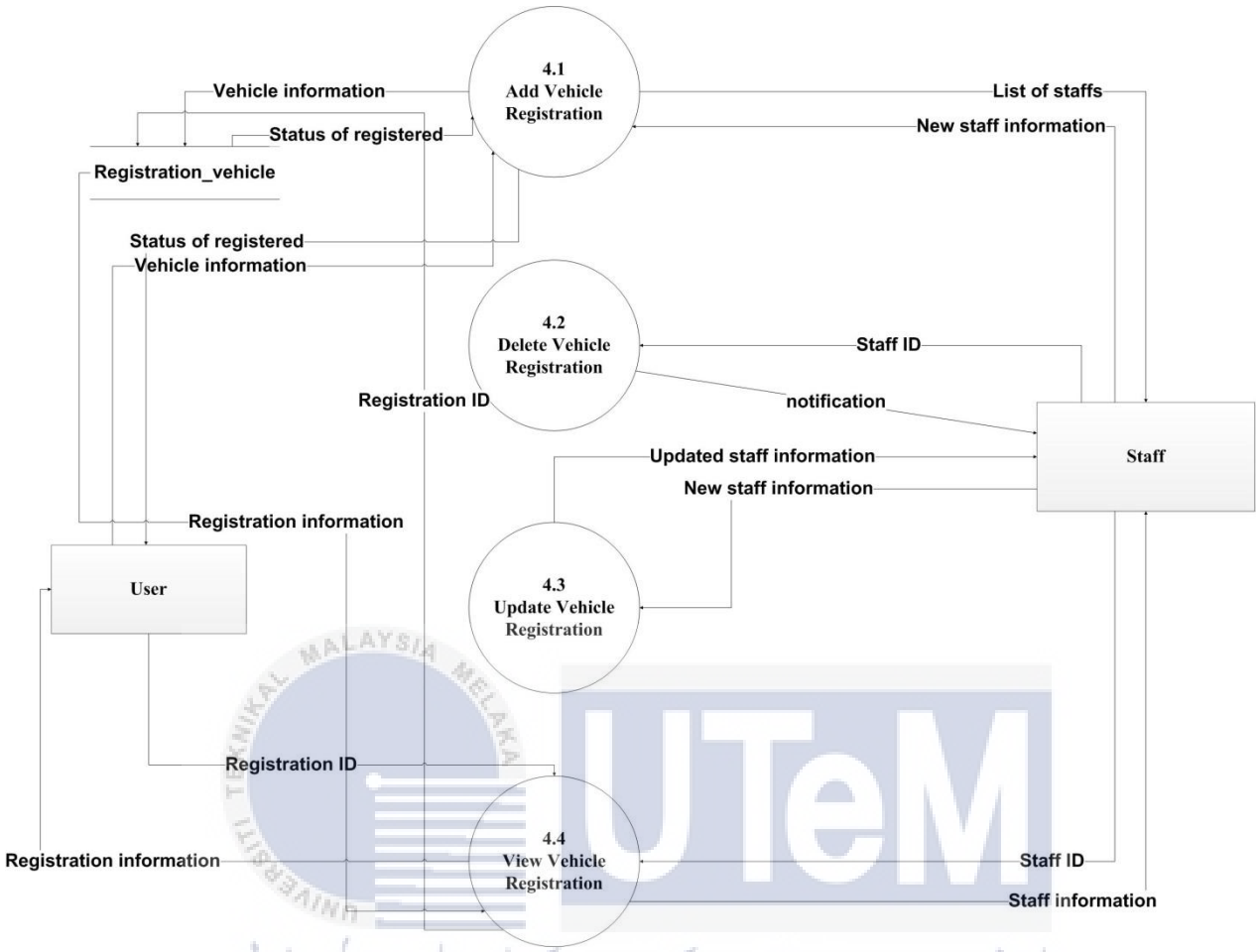


Figure 3.7 Data Flow Diagram (DFD) - Level 1 (Manage User Profile)

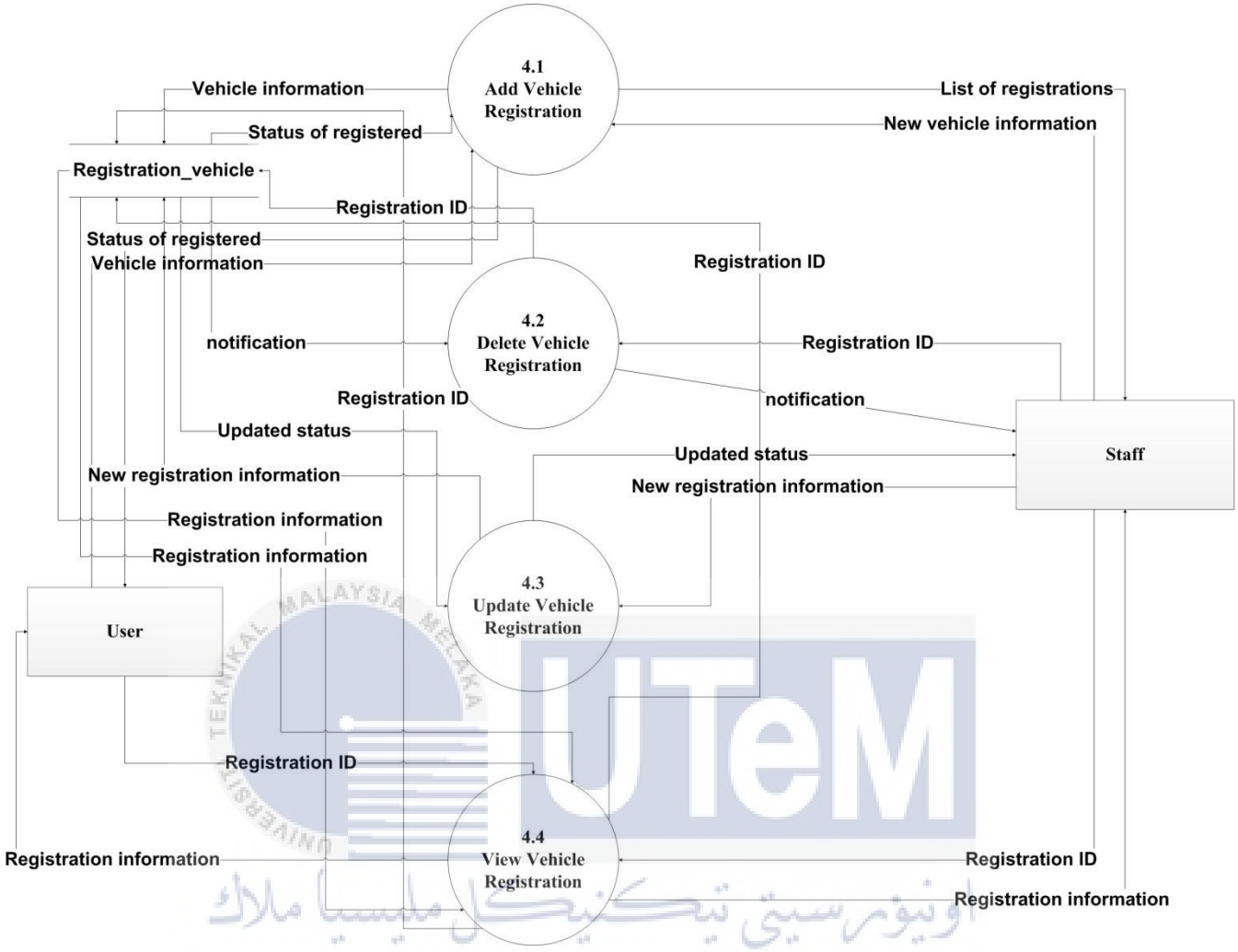


Figure 3.8 Data Flow Diagram (DFD) - Level 1 (Manage Vehicle Registration)

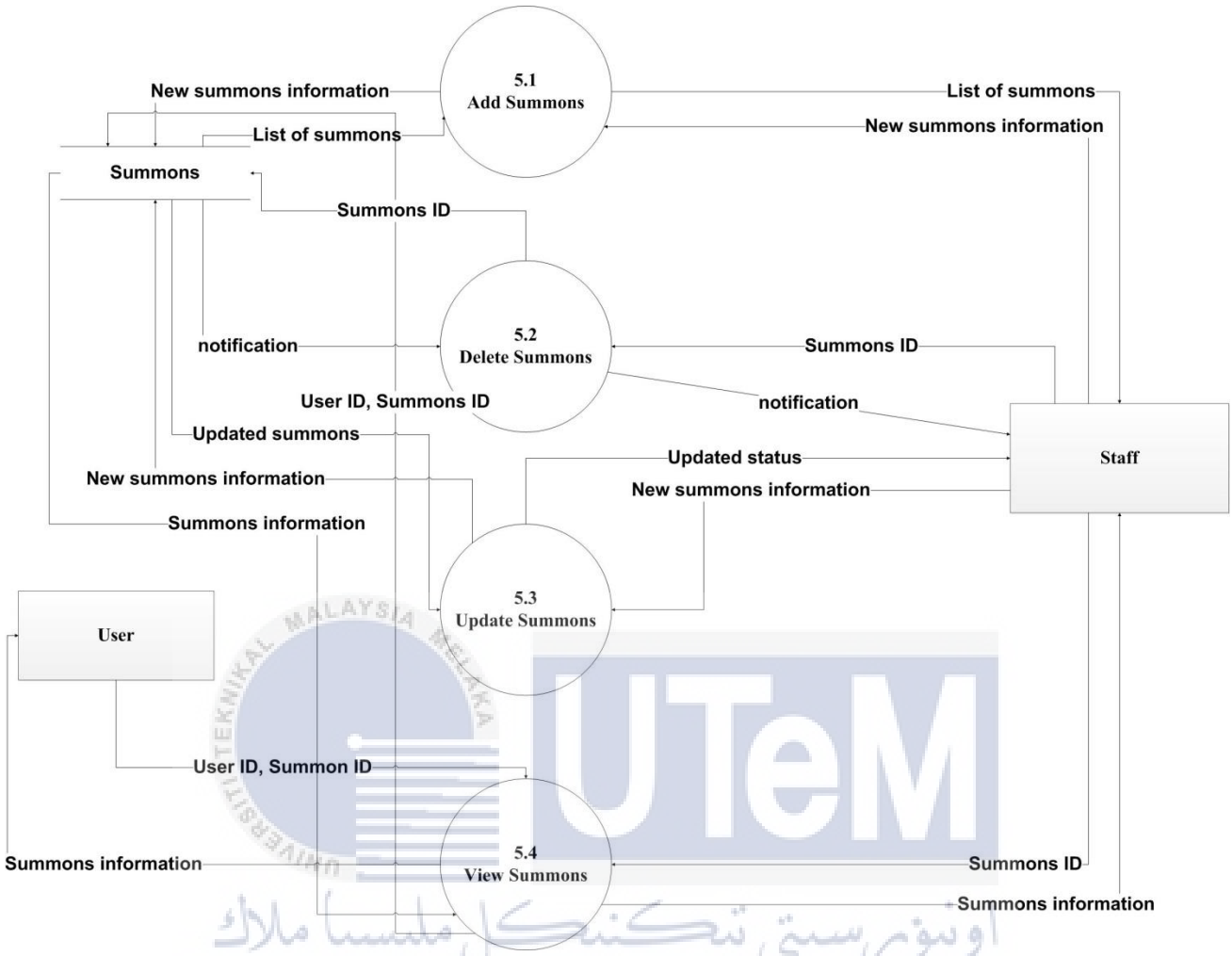


Figure 3.9 Data Flow Diagram (DFD) - Level 1 (Manage Summons)

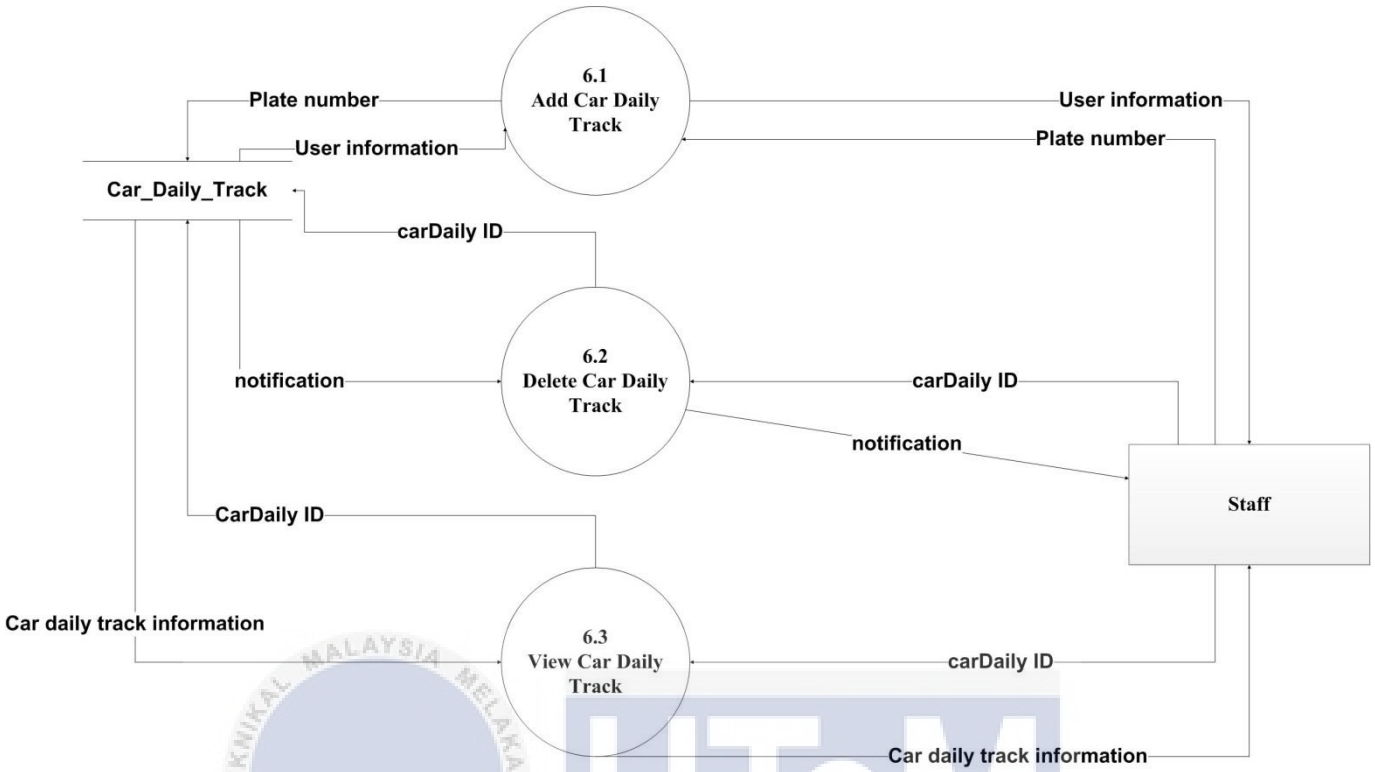


Figure 3.10 Data Flow Diagram (DFD) - Level 1 (Manage Car Daily Track)

3.6 Non-functional requirement

Table 3.2 List of Non-functional Requirement

NFR_No	Requirement	Description
1	Response Time	- Time taken to receive information
2	Usability	- System provide user friendly interface by simple flow of any process
3	Integrity	- Avoid any duplicate data by providing unique key
5	Correctness	- Correct in giving output for end-users

3.6.1 Others requirement

This section consist of software requirement, hardware requirement and other requirement

3.6.1.1 Software and Hardware requirement

Table 3.3 List of Software and Hardware Requirement

No	Software Requirement	Hardware Requirement
1	Microsoft Word 2010	Printer
2	Microsoft Visio 2010	Laptop
3	Microsoft Excel 2010	Mouse
4	Adobe Photoshop CS6	Hard Disk
5	Microsoft PowerPoint 2010	Thumb Drive
6	Notepad++	
7	Microsoft project 2010	

3.6.1.2 Other requirement

- Google Chrome

  One of the web browsers used to search any information related to this project.

- Internet access

  Allow computer to access data from outside and easily to find information.

- Prezi

  Slide presentation for final presentation.

3.7 Conclusion

In a conclusion, this chapter explains about the existing system and system to be developed. Some modules were added to make this system more function and interacting besides makes improvements from the existing system. The next chapter will explain about system design which is including the conceptual design, logical design and physical design.

CHAPTER IV

DESIGN

4.1 Introduction

This design phase will give the result of the analysis phase and next will explain more about the design of the system to-be. This chapter is related after completed the Context Diagram (CD) and Data Flow Diagram (DFD). Design is divided into two main categories which are input and output. Both of categories provide initial overview of Car Plate Number Recognition (CPNR) database system. Input is the process where users enter some information requested by the system. Meanwhile, the output is the interfaces that show to the users the information they need or requested. Besides that, design phase helps to solve problems during the process of implementation by giving the structure of

the system. The initial impression is important for individuals to add or subtract during the development of the real system.

4.2 Introductory preview to this chapter

Figure 4.2.1 shows the architecture view for Car Plate Number Recognition (CPNR) database system. The system architecture will cover all the processes. There are several processes in this system which is user registration, vehicle registration by user, summons management and car daily track management by staff. The process starts with user need to register into the system to be able become an authorized user. Then they are able to register their vehicles throughout this system by providing some information needed. The system will process the registration form and if it eligible applications then the user can print out the form. In addition, user will be provided summons information while staff can manage the summons.

The system architecture of this system is two-tier architecture. It refers to client/server in which the user interface runs on the client and the data structure gets stored on the server that comprises the two layers; the client application and data source. On client application side the code is written for saving the data in the PostgreSQL server database. The camera will detect plate number and sent the request to a server. Then it processes the request and sends back with data.

i. Client application: User interface

The interface is divided into two different categories which is administrator interface and user interface. The interfaces that administrator use are manage user profile, manage vehicle's registration, manage summons and manage car daily track which is insert, update and delete data. While the interfaces for user are managing profile, vehicle registration and view summons information. Both are using login and register interfaces.

ii. Data source: PostgreSQL DBMS

The information that is stored and retrieved from the database are owner, staff, v ehicle, daily_track, registration, vehicle_daily_track, daily_track and summons.

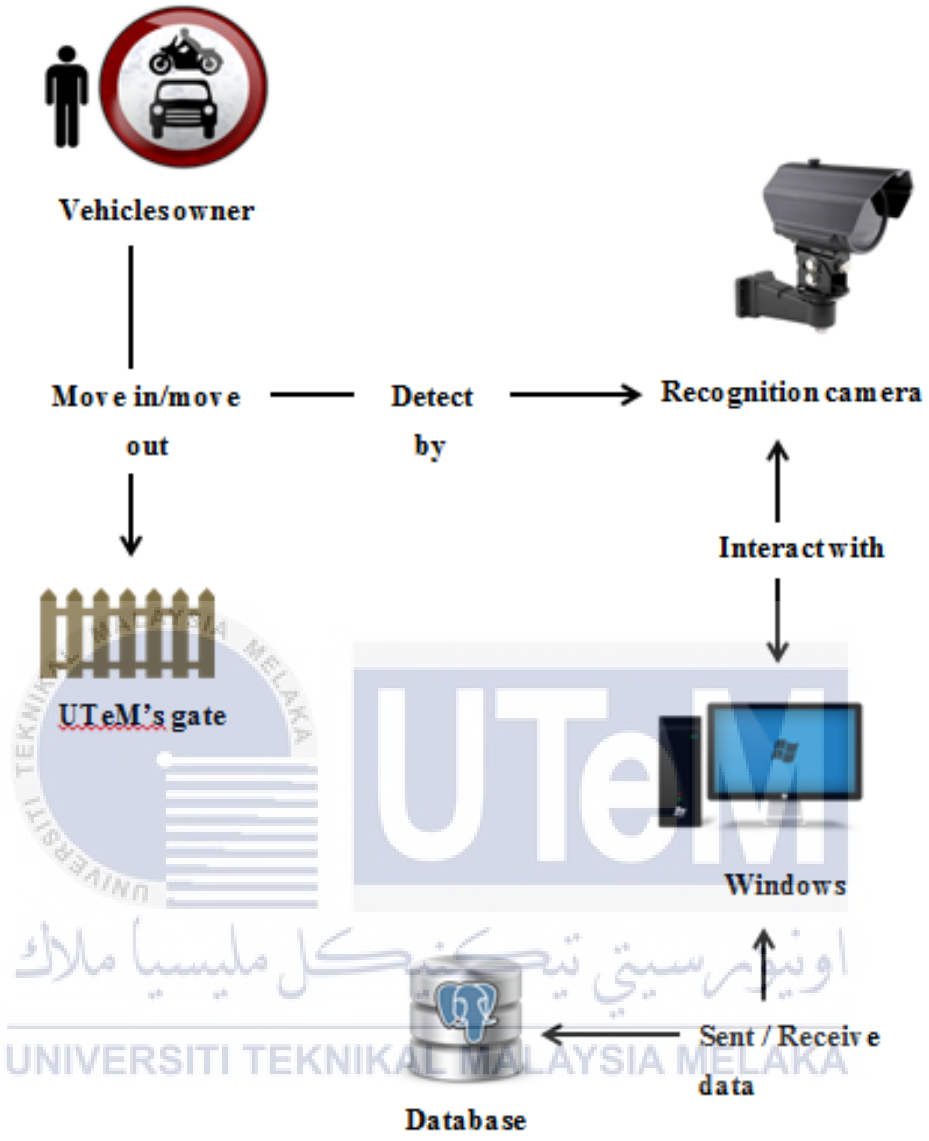


Figure 4.1 System Architecture Design for CPNR Database System

4.3 Conceptual Design

This section consists of two subsections which are Entity Relationship Diagram (ERD) and description by using business rules.

4.3.1 Entity Relationship Diagram

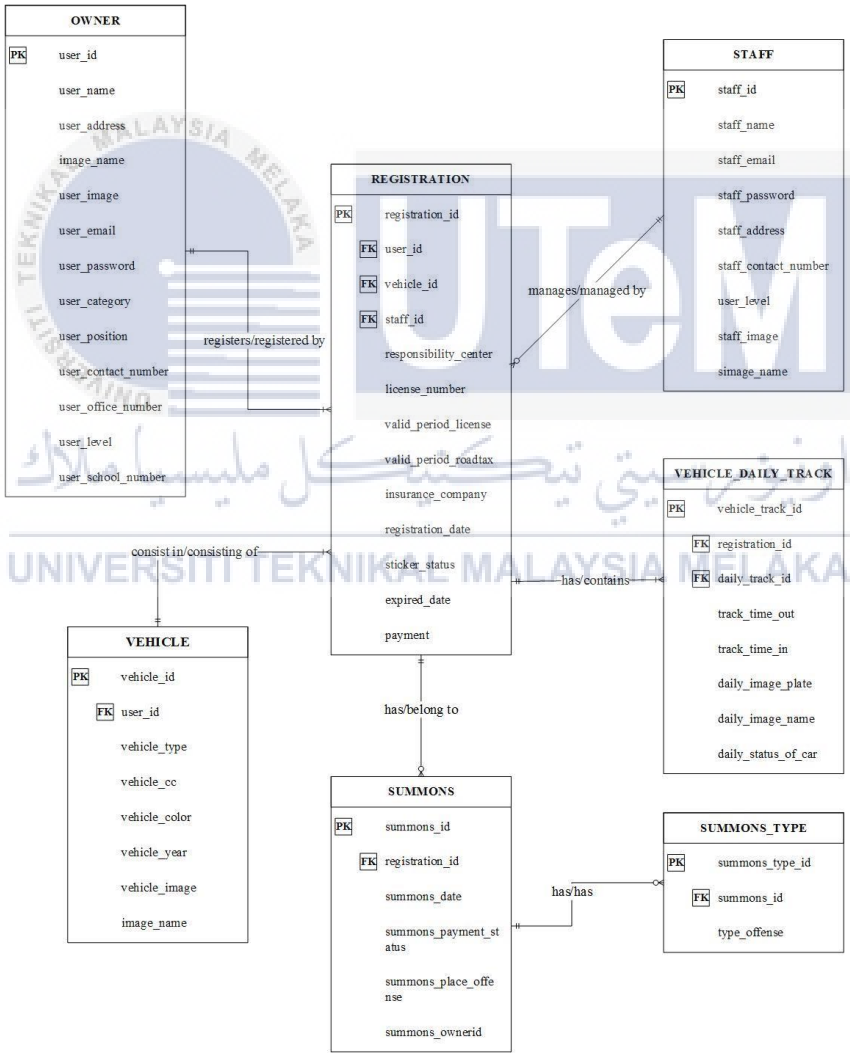


Figure 4.2 Entity Relationship Diagram (ERD)

4.3.2 Business Rules

1. One owner owns one or many vehicle.
   One vehicle is owned by one or many owner.
   Registration is the bridge between owner and vehicle.

2. One owner can register one or many registration.
   One registration is registered by only one owner.

3. One vehicle consists in one or many registration.
   One registration is consisting of only one vehicle.

4. One registration has zero or many summons.
   One summons belongs to only one registration.

5. One staff manages one or many vehicle
   One vehicle is managed by one or many staff
   Registration is the bridge between staff and vehicle

6. One registration has one or many daily_track.
   One daily_track contains only one registration.

7. One registration has one or many vehicle_daily_track.
   One vehicle_daily_track contains only one registration.

8. One summons belong to one registration.
   One registration has zero or many summons.

4.4 Logical Design

Logical design is dividing into three categories which are data dictionary, conceptual design using normalization and query design.

4.4.1 Data dictionary

Table 4.1 Data dictionary for vehicle

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
vehicle	vehicle_id	It refers to the plate number of vehicle	Character Varying (20)	AGM 6728	PK	Not null	-
	vehicle_type	Either car, motorcycle or others	Character Varying (20)	Car		Not null	-

	vehicle_cc	Number of cubic centimeters of the vehicle	Character Varying (20)	660		Null	-
	vehicle_color	Color of vehicle	Character Varying (20)	Silver		Null	-
	vehicle_year	Number of year of the car that has been used	Character Varying (20)	2008		Null	-

Table 4.2 Data dictionary for owner

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
owner	user_id	Matric number, staff number or worker pas number of the owner	Character Varying (20)	B031410056	PK	Not null	-
	user_name	Name of the owner	Character Varying (50)	Naqibah binti Zed		Not null	-
	user_address	Address of the owner	Character Varying (100)	No 75, Jalan DI 17, Taman Desa Idaman, 76100 Durian Tunggal		Not null	-
	user_ic	Identity card number of the owner	Character Varying (30)	931111-08-5222		Not null	-

	user_email	Email of the owner	Character Varying (50)	naqibahzed@yahoo.com		Not null	-
	user_password	Password to log into the system	Character Varying (50)	Naqibah123		Not null	-
	user_category	Category of owner	Character Varying (20)	Outsider student		Not null	-
	user_position	Position of owner	Character Varying (30)	Student		Not null	-
	user_contact_number	Contact number of the owner	Character Varying (20)	013-5454123		Not null	-
	user_office_number	Office number of the owner	Character Varying (20)	-		Null	-

Table 4.3 Data dictionary for staff

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
staff	staff_id	Identity number for staff	Character Varying (20)	ST01	PK	Not null	-
	staff_name	Name of the staff	Character Varying (50)	Muhammad Khairullah bin Hassan		Not null	-
	staff_email	Email address of the staff	Character Varying (50)	khairullah@utem.edu.my		Not null	-
	staff_password	Password to log into the system	Character Varying (30)	Khairullahhassanpass		Not null	-
	staff_address	Address of the staff	Character Varying (100)	No 9, Jalan TPJ 7, Taman Permata Jaya, 65454 Ayer Keroh Melaka		Not null	-
	staff_contact_number	Contact number of the staff	Character Varying (20)	019-7878123		Not null	-

Table 4.4 Data dictionary for daily_track

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
daily_track	daily_track_id	Identity number of daily track	Character Varying (20)	DT01	PK	Not null	-
	daily_image_plate	Image of plate number	Bytea	a.jpg		Null	-
	daily_status_of_car	Status of the car	Character Varying (20)	Available		Null	-

Table 4.5 Data dictionary for registration

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
registration	registration_id	Identity number of registration	Character Varying (20)	REG01	PK	Not null	-
	responsibility_center	Owner's responsibility center	Character Varying (30)	FTMK		Not null	-
	license_number	License number of the owner	Character Varying (50)	11547014		Not null	-
	valid_period_license	Period of validity of licenses	Character Varying (50)	02/05/2015		Not null	-
	valid_period_roadtax	Period of validity of road tax	Character Varying (50)	August 2016		Not null	-
	insurance_company	Insurance company name	Character Varying (50)	Alliance		Not null	-

	registration_date	Date of vehicle registration	Timestamp without time zone	24/04/2016		Not null	-
	sticker_status	Status of the sticker	Character Varying (50)	Valid		Null	-
	expired_date	Sticker valid date	Character Varying (50)	31 December 2016		Null	-
	payment	Payment information	Character Varying (20)	No		Null	-
	user_id	Identity number for owner	Character Varying (20)	B031410056	FK	Not null	Owner
	vehicle_id	Identity number for vehicle	Character Varying (20)	AGM 6728	FK	Not null	Vehicle
	staff_id	Identity number for staff	Character Varying (20)	ST01	FK	Not null	Staff

Table 4.6 Data dictionary for vehicle_daily_track

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
vehicle_daily_track	vehicle_track_id	Identity number for vehicle track	Character Varying (20)	VDT01	PK	Not null	-
	track_date	Date of track	Timestamp without time zone	13/05/2016		Null	-
	track_time_out	Vehicle's time out	Timestamp without time zone	09:15		Null	-
	track_time_in	Vehicle's time in	Timestamp without time zone	14:26		Null	-
	registration_id	Identity number of registration	Character Varying (20)	REG01		Not null	Registration
	daily_track_id	Identity number of track	Character Varying (20)	DT01		Not null	Daily_track

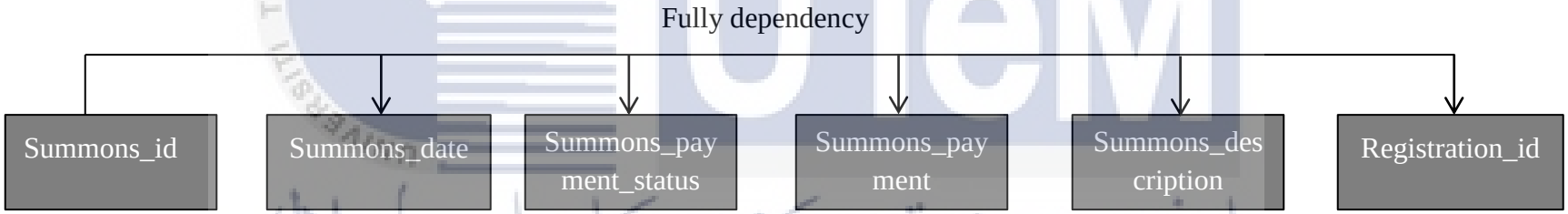
Table 4.7 Data dictionary for summons

Table Name	Attribute Name	Description	Type / Size	Format	PK/ FK	Null/Not Null	Related Table
summons	summons_id	Identity number of summons	Character Varying (20)	SM01	PK	Not null	-
	summons_date	Date of summons	Character Varying (30)	09/03/2016		Not null	-
	summons_payment_date	Payment date of summons	Character Varying (30)	12/03/2016		Null	-
	summons_payment	Total payment of summons	Character Varying (30)	30		Not null	-
	summons_description	Description of the summons	Character Varying (100)	Parking in a prohibited place		Not null	-
	registration_id	Identity number of registration	Character Varying (20)	REG01		Not null	Registration

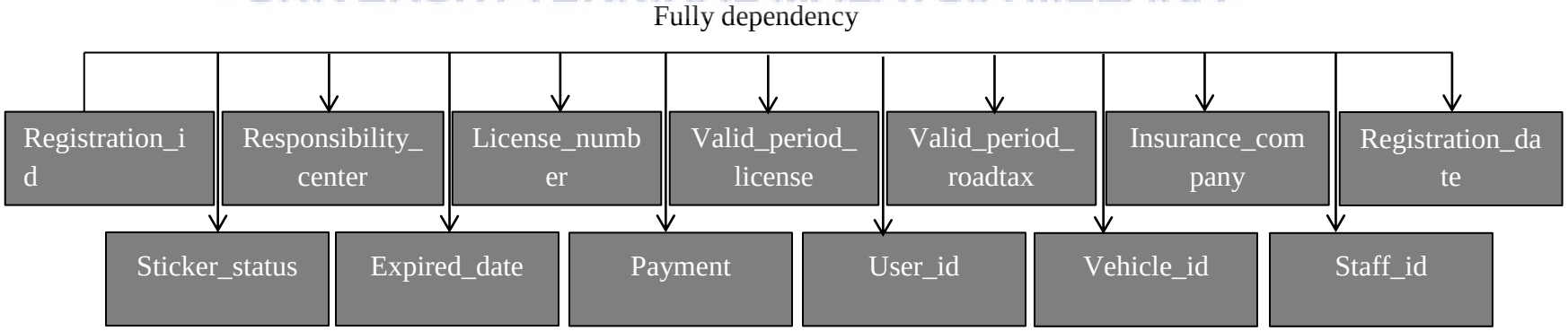
4.5 Conceptual Design using Normalization

The conceptual design using normalization is displayed using relational schema. Each table shows attributes, primary key and foreign key.

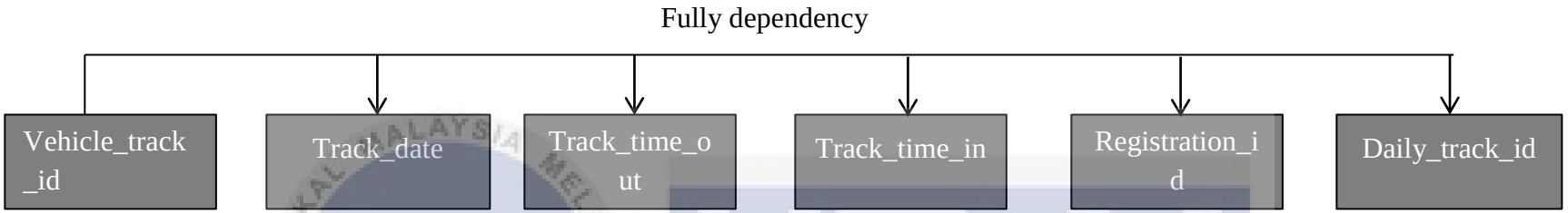
- Normalization of summons table : Summons table is already in 3 NF



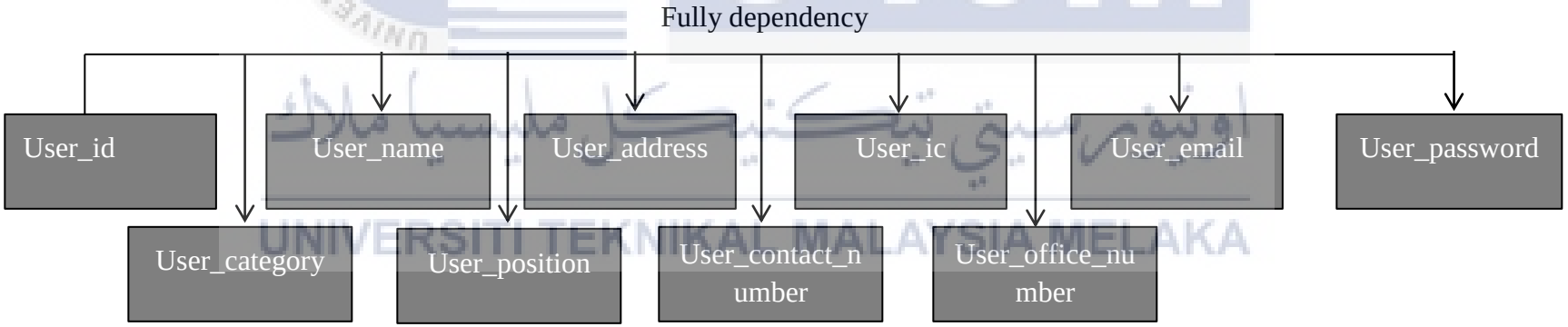
- Normalization of registration table : Registration table is already in 3 NF



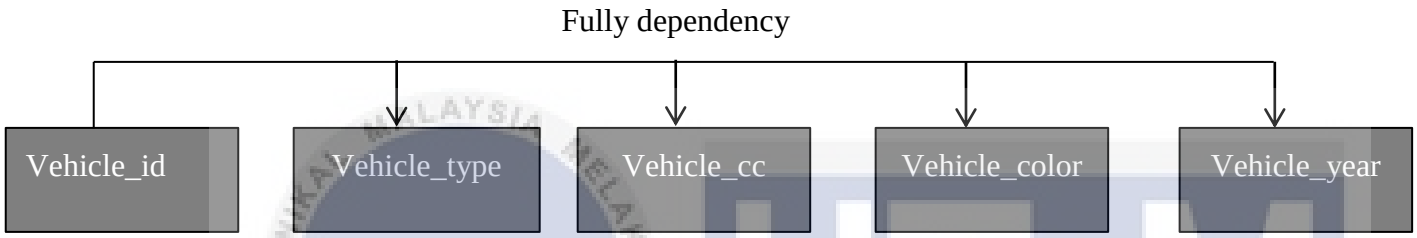
- Normalization of vehicle_daily_track table : Vehicle_daily_track table is already in 3 NF



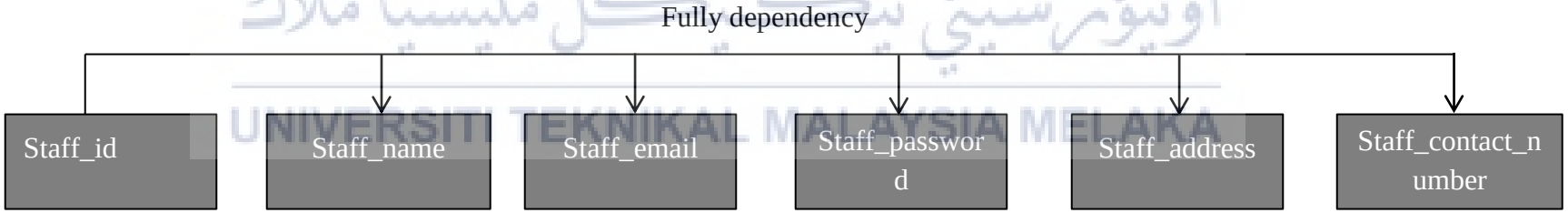
- Normalization of owner table : Owner table is already in 3 NF



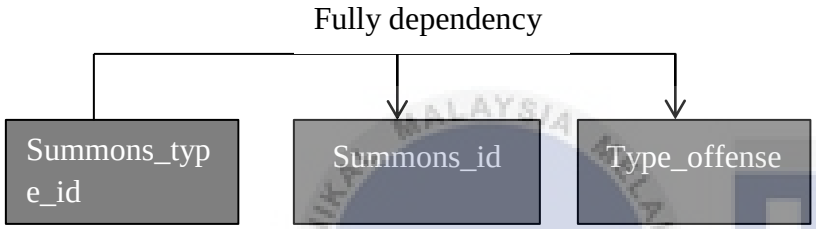
- Normalization of vehicle table : Vehicle table is already in 3 NF



- Normalization of staff table : Staff table is already in 3 NF



- Normalization of summos_type table : Summons_type table is already in 3 NF



4.5.1 Query Design

There are many query design that can be carried out to produce many different output. Each query is based on the requirement on user's need and it has its own reason and purpose. Table 8 shows the example of query design.

Table 4.8 Example of Query Design

Type of Query	Query	Description
Simple SQL query	Select * from summons;	View all data in summons table
Joint multiple table SQL query	`select r.registration_id, r.vehicle_id, r.user_id, r.valid_period_license, r.valid_period_roadtax from owner o, vehicle v, registration r where r.user_id= o.user_id AND r.vehicle_id = v.vehicle_id`	Join between owner, vehicle, registration to view the selected data that related each other
Subquery SQL	`select daily_track_id from daily_track where daily_track_id = (select max(daily_track_id) from daily_track)`	Select max daily track id first to select daily track id from daily track table if one user got more than one daily track id

Aggregate query	SELECT v.vehicle_id, count(his.registration_id) AS total, date_part('day', track_time_in) as day,date_part('month', track_time_in) as month, date_part('year', track_time_in) as year FROM history_vehicle_daily_track his, registration r, vehicle v WHERE his.registration_id = r.registration_id AND r.vehicle_id = v.vehicle_id AND operation ='UPDATE' GROUP BY v.vehicle_id, his.registration_id, date_part('day', track_time_in), date_part('month', track_time_in), date_part('year', track_time_in) ORDER BY date_part('day', track_time_in), date_part('month', track_time_in), date_part('year', track_time_in)	Using of count to calculate total of registration per person

4.6 Physical Design

Physical design consists of four categories which are selection of DBMS, trigger and stored procedure.

4.6.1 Selection of DBMS

Database is used primarily for data storage and retrieval. It handles a large amount of data and the data itself need to be retrieved and updated frequently. Database also allows orderly data storage, rapid data analysis and complex data analysis. Most databases like MySQL, SQL Server, PostgreSQL and Oracle 10g has been review one by one. Through the review process, some information about databases has gain like description of each of them, when it is useful and the benefits when use it. For this project, the most suitable is PostgreSQL based on a number of factors and specific reasons which mention in paragraph below.

PostgreSQL is open source software. The priority for this software is data integrity and reliability. As this project is using Windows, PostgreSQL also support this operating system. It supports highly performance and even on complex queries. In term of data storage, PostgreSQL provides many different data types for managing data beyond the basic Numeric, Text and Data types.

4.6.2 Trigger

There are some types of trigger that can be implemented in this system and it gives different meaning and process.

4.6.2.1 Trigger Before

Table 4.9 Example of Trigger Before

No	Function	Trigger	Description
1	CREATE OR REPLACE FUNCTION staff_id() RETURNS trigger AS $BODY$DECLARE ID int; BEGIN select NEXTVAL('staff_id_seq') into ID; NEW.staff_id := ('S00'|| ID); RETURN NEW; END; $BODY$ LANGUAGE plpgsql VOLATILE COST 100;	CREATE TRIGGER staff_id_trigger BEFORE INSERT ON staff FOR EACH ROW EXECUTE PROCEDURE staff_id()	To produce a primary key start with character for table staff

2	create function daily_track_id() RETURNS trigger AS $ daily_track _id_TRIGGER $DECLARE ID int; BEGIN select NEXTVAL(daily_track _seq') into ID; NEW. daily_track _id := ('DT0' || ID); RETURN NEW; END; $ daily_track _id_TRIGGER$ LANGUAGE plpgsql;	create trigger daily_track _id_TRIGGER BEFORE INSERT ON daily_track FOR EACH ROW EXECUTE PROCEDURE daily_track _id ();	To produce a primary key start with character for table daily_track
3	create function registration_id() RETURNS trigger AS $ registration _id_TRIGGER $DECLARE ID int; BEGIN select NEXTVAL(registration _seq') into ID; NEW. registration _id := ('REG0' || ID); RETURN NEW; END; $ registration _id_TRIGGER$ LANGUAGE plpgsql;	create trigger registration _id_TRIGGER BEFORE INSERT ON registration FOR EACH ROW EXECUTE PROCEDURE registration _id ();	To produce a primary key start with character for table registration

4	create function vehicle_daily_track_id() RETURNS trigger AS $ vehicle_daily_track _id_TRIGGER $DECLARE ID int; BEGIN select NEXTVAL(vehicle_daily_track _seq') into ID; NEW. vehicle_daily_track _id := ('VDT0' || ID); RETURN NEW; END; $ vehicle_daily_track _id_TRIGGER$ LANGUAGE plpgsql;	create trigger vehicle_daily_track _id_TRIGGER BEFORE INSERT ON vehicle_daily_track FOR EACH ROW EXECUTE PROCEDURE vehicle_daily_track _id ();	To produce a primary key start with character for table vehicle_daily_track
5	create function summons_type_id() RETURNS trigger AS $ summons_type_id _TRIGGER $DECLARE ID int; BEGIN select NEXTVAL(summons_type _seq') into ID; NEW. summons_type_id:= ('SUMTY0' || ID);	create trigger summons_type_id _TRIGGER BEFORE INSERT ON summons_type FOR EACH ROW EXECUTE PROCEDURE summons_type_id ();	To produce a primary key start with character for table summons_type

	RETURN NEW; END; $ summons _id_TRIGGER$ LANGUAGE plpgsql;		
6	create function summons_id() RETURNS trigger AS $ summons _id_TRIGGER $DECLARE ID int; BEGIN select NEXTVAL(summons _seq') into ID; NEW. summons _id := ('SM0' || ID); RETURN NEW; END; $ summons _id_TRIGGER$ LANGUAGE plpgsql;	create trigger summons _id_TRIGGER BEFORE INSERT ON summons FOR EACH ROW EXECUTE PROCEDURE summons _id ();	To produce a primary key start with character for table summons

4.6.2.2 Trigger after insert, delete and update

Table 4.10 Example of Trigger After Insert, Delete and Update

1	CREATE OR REPLACE FUNCTION history_vehicle_daily_track()
	RETURNS trigger AS
	$BODY$
	BEGIN
	IF tg_op = 'DELETE' THEN
	INSERT INTO history_vehicle_daily_track
	VALUES(old.vehicle_track_id, old.track_time_out,old.track_time_in,
	old.registration_id,old.daily_track_id, current_timestamp, tg_op);
	RETURN old;
	END IF;
	IF tg_op = 'INSERT' THEN
	INSERT INTO history_vehicle_daily_track
	VALUES(new.vehicle_track_id, new.track_time_out,new.track_time_in,
	new.registration_id,new.daily_track_id, current_timestamp, tg_op);
	RETURN new;
	END IF;
	IF tg_op = 'UPDATE' THEN
		A trigger to insert, delete and update any data into table history_vehicle_daily_track. Any new data, updated data or deleted data will be recorded into history_vehicle_daily_track table.

	INSERT INTO history_vehicle_daily_track VALUES(new.vehicle_track_id, new.track_time_out,new.track_time_in, new.registration_id,new.daily_track_id, current_timestamp, tg_op); RETURN new; END IF; END $BODY$ LANGUAGE plpgsql; CREATE TRIGGER trig_vehicle_daily_track AFTER INSERT OR DELETE OR UPDATE ON vehicle_daily_track FOR EACH ROW EXECUTE PROCEDURE history_vehicle_daily_track()	

4.6.3 Stored Procedure / Function

Table 4.11 Example of Stored Procedure / Function

1	CREATE OR REPLACE FUNCTION Insert_register_vehicle(user_id character varying, vehicle_id character varying(20), vehicle_type	Function to view list of vehicle

	character varying, vehicle_cc character varying(20), vehicle_color character varying, vehicle_year character varying(20))
	RETURNS void AS
	$BODY$
	BEGIN
	INSERT INTO vehicle(user_id, vehicle_id, vehicle_type, vehicle_cc, vehicle_color, vehicle_year)
	VALUES(user_id, vehicle_id, vehicle_type, vehicle_cc, vehicle_color, vehicle_year);
	END;
	$BODY$
	LANGUAGE 'plpgsql';	

2	CREATE FUNCTION viewregisteruser()
	RETURNS SETOF owner AS
	$$
	DECLARE
	rec record;
	BEGIN
	FOR rec IN (SELECT * FROM owner)
	LOOP
	RETURN NEXT rec;
	END LOOP;
	END;
	$$ LANGUAGE plpgsql;
		Function to view list of registered user

3	CREATE OR REPLACE FUNCTION deleteregistereduser(Id character varying(20))
	RETURNS character varying(20) AS
	$BODY$
	BEGIN
	delete from owner where user_id = Id;
	RETURN 1;
	END;
	$BODY$
	LANGUAGE plpgsql;
		Function to delete registered user

4	CREATE OR REPLACE FUNCTION deletesummons(Id character varying)
	RETURNS character varying AS
	$BODY$
	BEGIN
	delete from summons where summons_id = Id;
	RETURN 1;
	END;
	$BODY$
	LANGUAGE plpgsql;
		Function to delete summons

5	CREATE OR REPLACE FUNCTION deletestaff(Id character varying) RETURNS character varying AS	Function to delete staff

	$BODY$
	BEGIN
	delete from staff where staff_id = Id;
	RETURN 1;
	END;
	$BODY$
	LANGUAGE plpgsql;	

6	CREATE OR REPLACE FUNCTION deletevehicle(Id character varying)
	RETURNS character varying AS
	$BODY$
	BEGIN
	delete from registration where registration_id = Id;
	RETURN 1;
	END;
	$BODY$
	LANGUAGE plpgsql;
		Function to delete vehicle

7	CREATE OR REPLACE FUNCTION deletelogin(Id character varying)
	RETURNS character varying AS
	$BODY$
	BEGIN
	delete from verify_login where user_id = Id;
	RETURN 1;
		Function to delete user from login verify table

	END; $BODY$ LANGUAGE plpgsql;	

8	CREATE OR REPLACE FUNCTION updateregistereduser(id character varying(20), address character varying(100), jawatan character varying(30), contactno character varying(20), schoolno character varying(30))
	
	RETURNS void AS
	$BODY$
	BEGIN
	update owner
	set
	user_address = address,
	user_position = jawatan,
	user_contact_number = contactno,
	user_school_number = schoolno
	where user_id = id;
	END; $BODY$
	LANGUAGE plpgsql;
		Function to update registered user

9	CREATE OR REPLACE FUNCTION viewownervehicle() RETURNS TABLE (	Function to joint at least two tables

	registration_id character varying(100),
	vehicle_id character varying(20),
	user_id character varying(20),
	valid_period_license date,
	valid_period_roadtax character varying(50)) AS
	$$
	
	BEGIN
	FOR registration_id, vehicle_id, user_id, valid_period_license, valid_period_roadtax IN
	select r.registration_id, r.vehicle_id, r.user_id, r.valid_period_license, r.valid_period_roadtax
	from owner o, vehicle v, registration r where r.user_id= o.user_id AND r.vehicle_id = v.vehicle_id
	LOOP
	RETURN NEXT;
	END LOOP;
	END;
	$$ LANGUAGE plpgsql;
		To list useful information by combining registration, vehicle and owner table

10	CREATE OR REPLACE FUNCTION complex_view_report1()	Function complex

	RETURNS TABLE (vehicle_id character varying(100), total character varying(40), day character varying(100), month character varying(40), year character varying(40)) AS $$ BEGIN FOR vehicle_id, total, day, month, year IN SELECT v.vehicle_id, count(his.registration_id) AS total, date_part('day', track_time_in) as day,date_part('month', track_time_in) as month, date_part('year', track_time_in) as year FROM history_vehicle_daily_track his, registration r, vehicle v WHERE his.registration_id = r.registration_id AND r.vehicle_id = v.vehicle_id AND operation ='UPDATE' GROUP BY v.vehicle_id, his.registration_id, date_part('day', track_time_in),	To shows how many time every vehicle move into UTeM and its information

	date_part('month', track_time_in), date_part('year', track_time_in) LOOP RETURN NEXT; END LOOP; END; $$ LANGUAGE plpgsql;	

4.7 Graphical User Interface (GUI) Design

Graphical user interface is an interface that allow user to interact with it. User will key in information as an input into the system throughout the form on interface. The objectives of user interface are to get input from user and save it into database and to display output or any query based on user requirement.

The main interface for this system is account registration, login, user profile, vehicle registration, car daily track and summons. Both staff and owner can view all the interfaces but owner has some limitation on management of it.

Below are the examples of the interface that will implement.

- User main menu

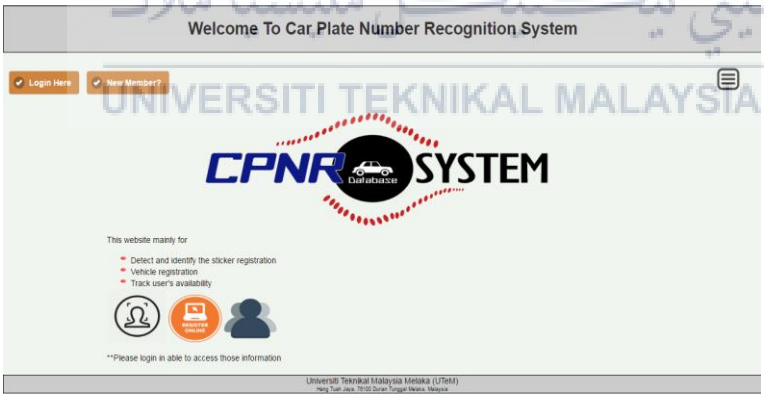


Figure 4.3 Web Interface for Main Menu

- User registration form

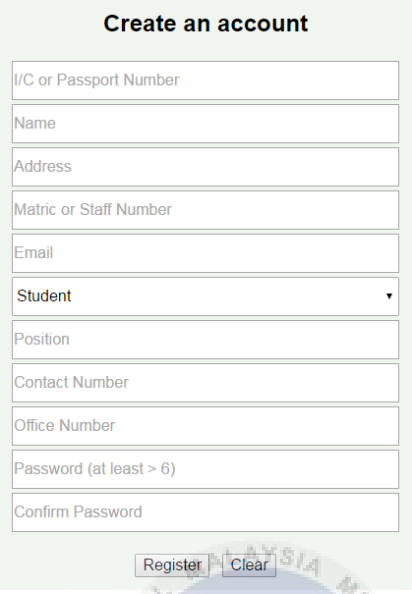


Figure 4.4 Web Interface for Registration

- User login form

Figure 4.5 Web Interface for Login

- Admin site: User Profile

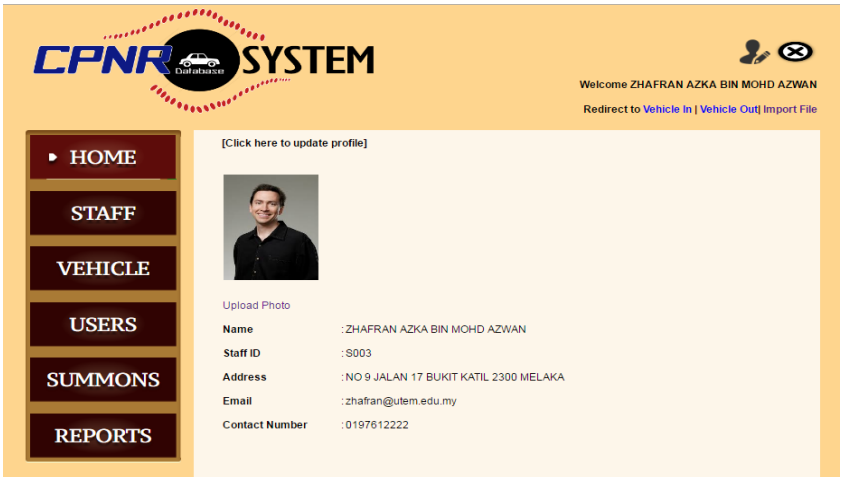


Figure 4.6 Web Interface for Admin - User Profile

- Admin site: Upload Photo

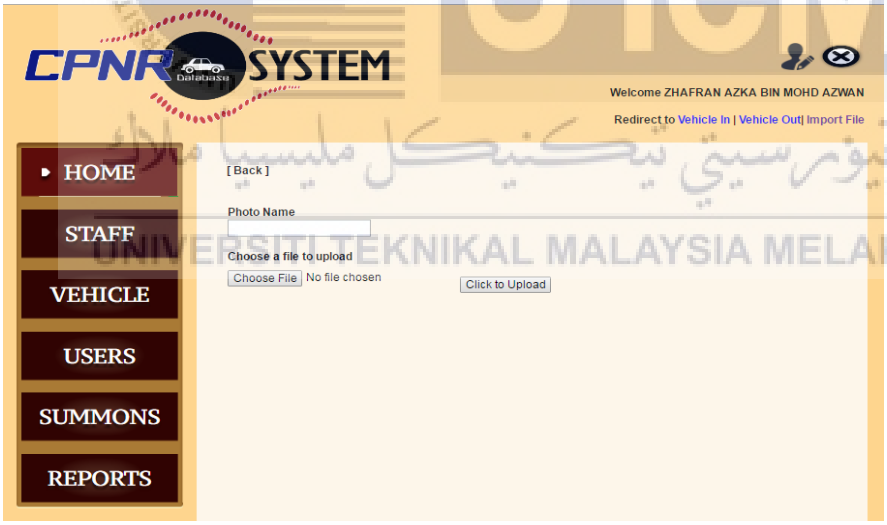


Figure 4.7 Web Interface for Admin - Upload Photo

- Admin site: Update Profile

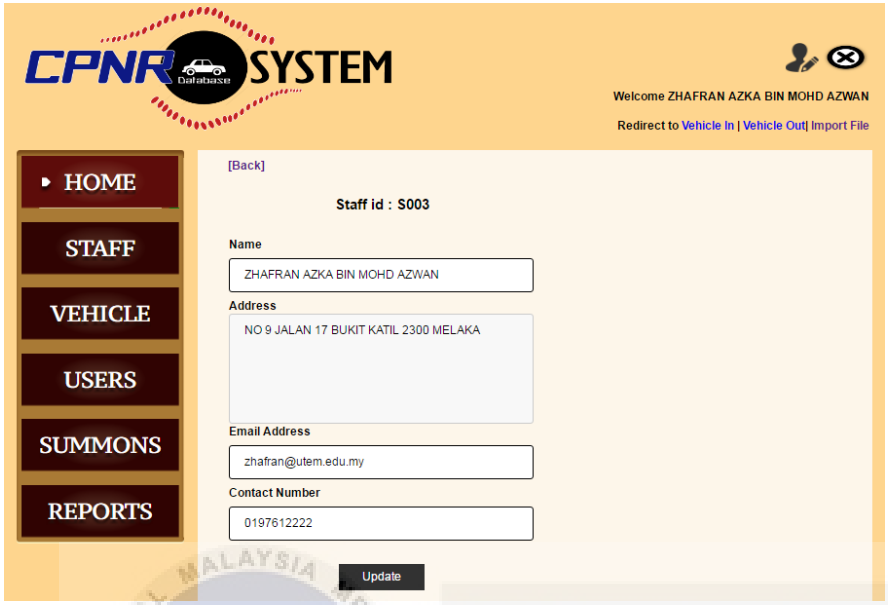


Figure 4.8 Web Interface for Admin - Update Profile

- Admin site: List of staff

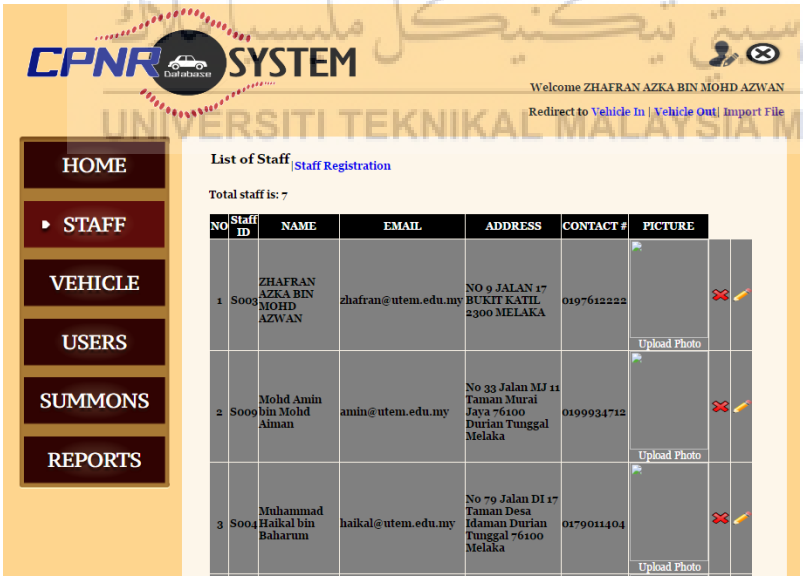


Figure 4.9 Web Interface for Admin - List of Staff

- Admin site: Registration form for staff

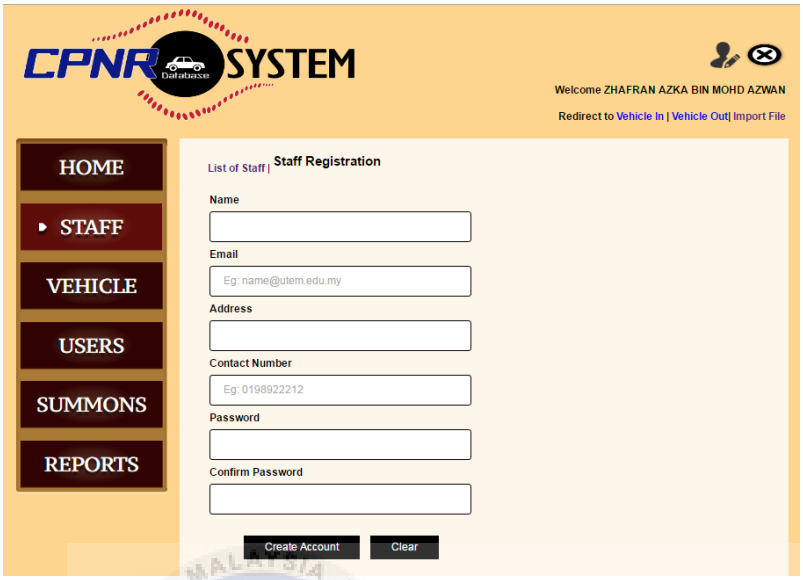


Figure 4.10 Web Interface for Admin - Staff Registration

- Admin site: List of registered vehicle

Figure 4.11 Web Interface for Admin - List of Registered Vehicle

- Admin site: More Information on Registered Vehicle

Figure 4.12 Web Interface for Admin - More Information on Registered User I

Figure 4.13 Web Interface for Admin - More Information on Registered User II

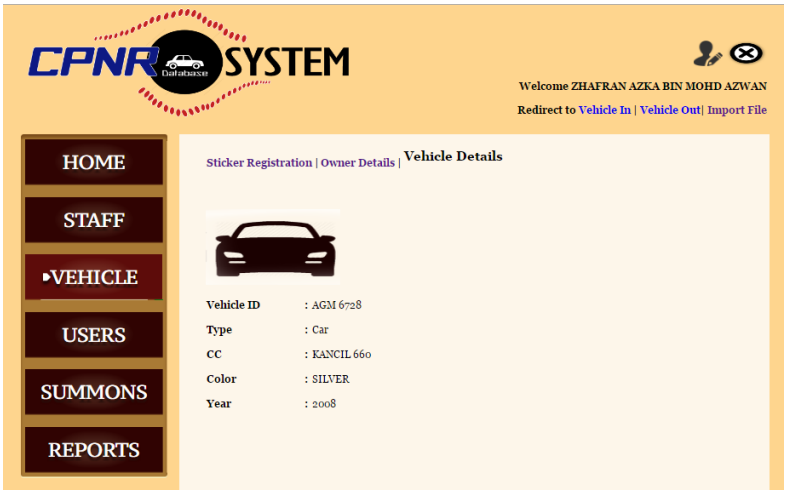


Figure 4.14 Web Interface for Admin - More Information on Registered User III

- Admin site: List of registered user

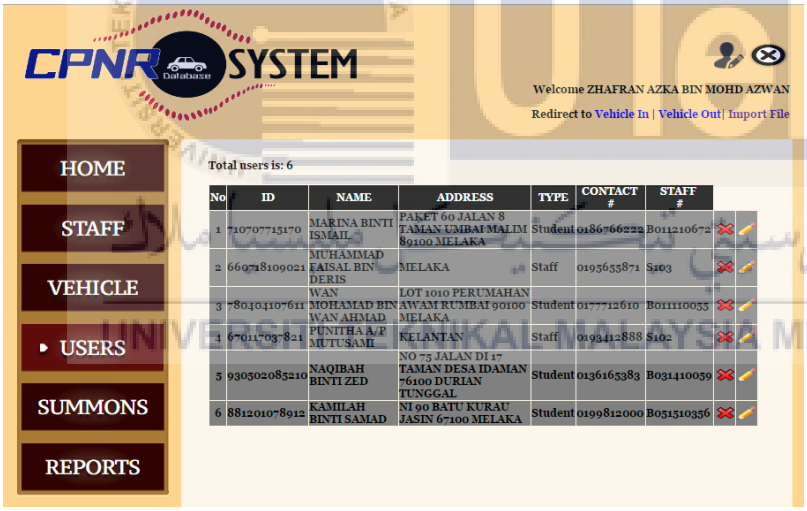


Figure 4.15 Web Interface for Admin - List of Registered User

- Admin site: List of summons history

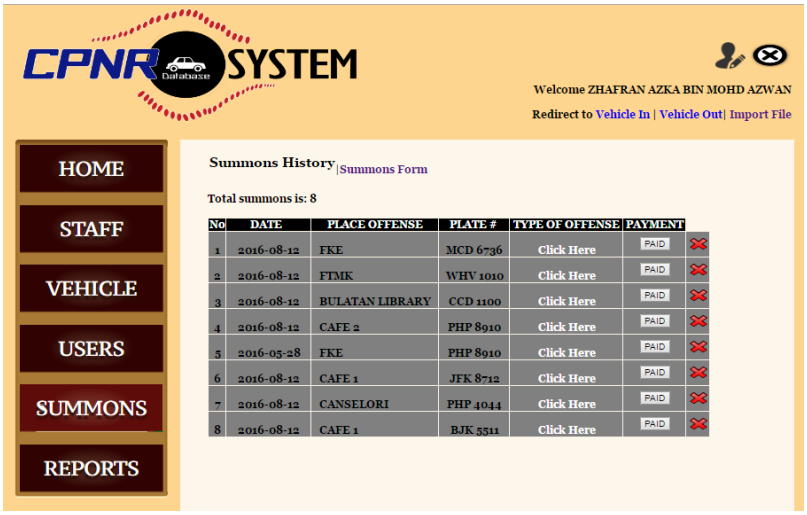


Figure 4.16 Web Interface for Admin - List of Summons History

- Admin site: Summons form

Figure 4.17 Web Interface for Admin - Summons Form

- Admin site: Report of user login

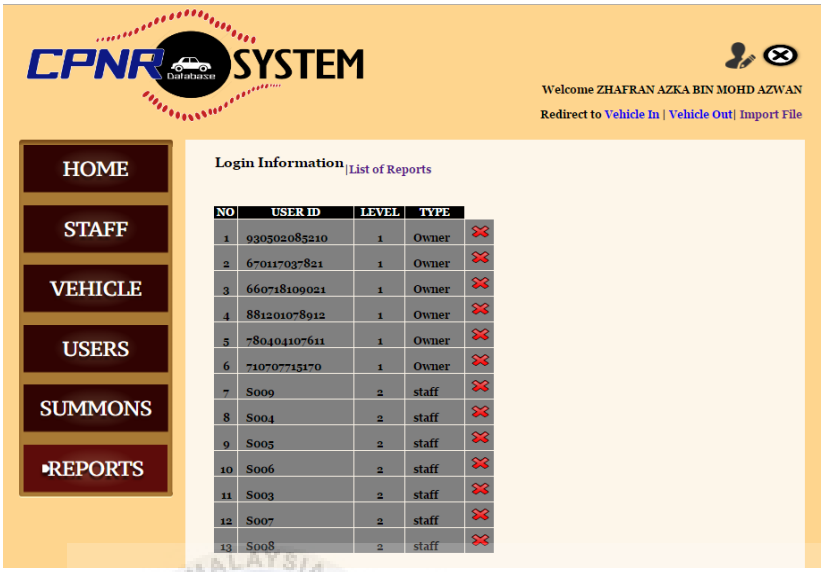


Figure 4.18 Web Interface for Admin - Report of User Login

- Admin site: Report of number of entrance per person in UTeM

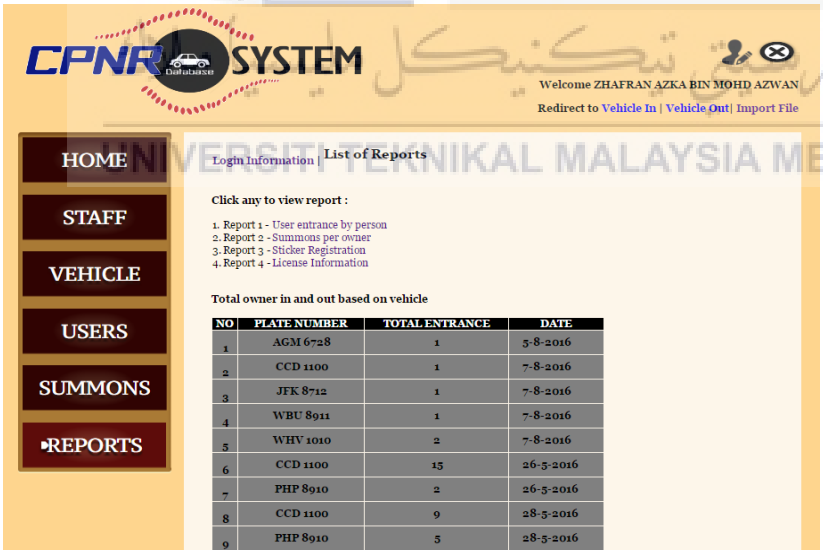


Figure 4.19 Web Interface for Admin - Report of Number User Entrance

- Admin site: Report of number of summons per person

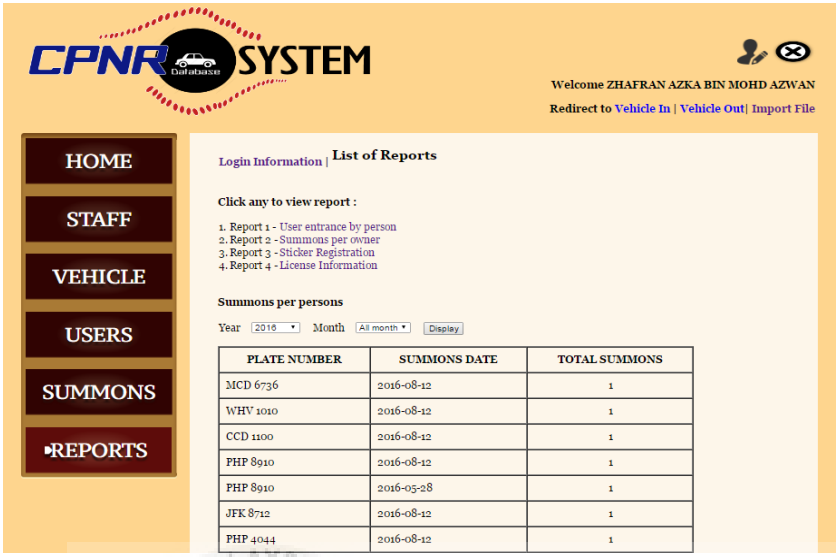


Figure 4.20 Web Interface for Admin - Report of Number of Summons

- Admin site: Report of owner's valid period license information

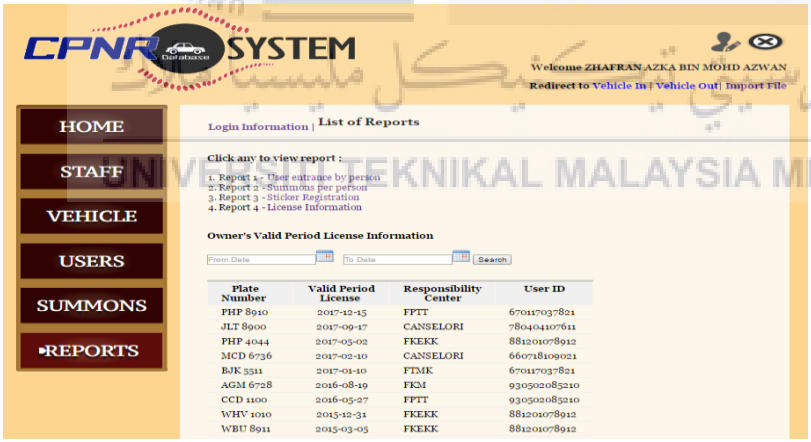


Figure 4.21 Web Interface for Admin - Report of Valid Period License

- Admin site: Report on Number of Sticker Registration

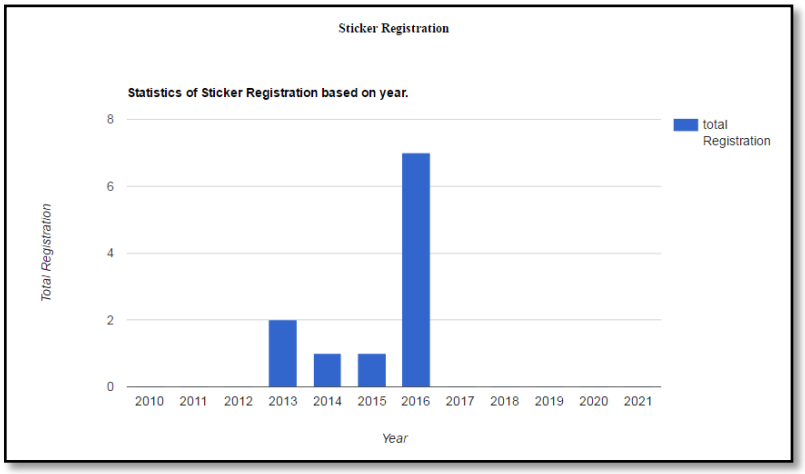


Figure 4.22 Web Interface for Admin - Report of Number of Sticker Registration

- Admin site: Track User In and Out

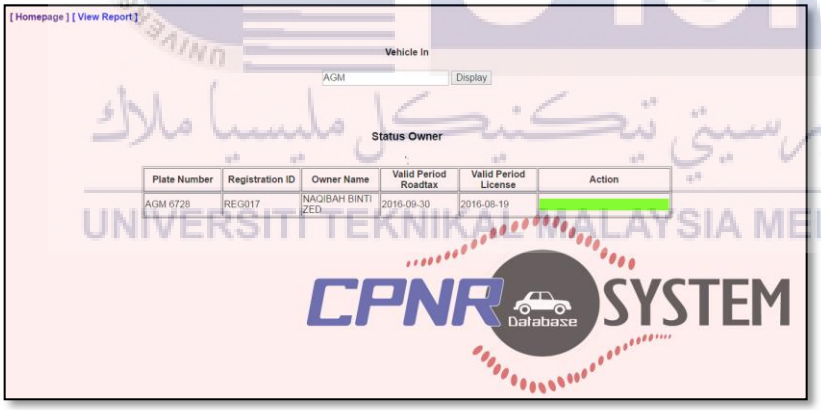


Figure 4.23 Web Interface for Admin - Track User Entrance I

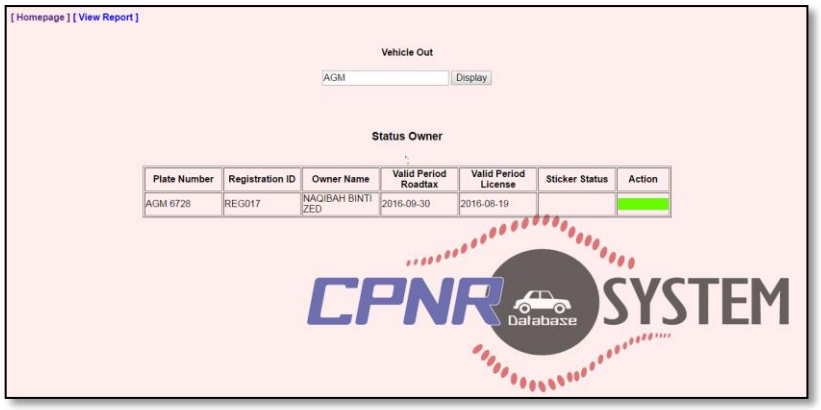


Figure 4.24 Web Interface for Admin - Track User Entrance II

- Admin site: Record on daily entrance

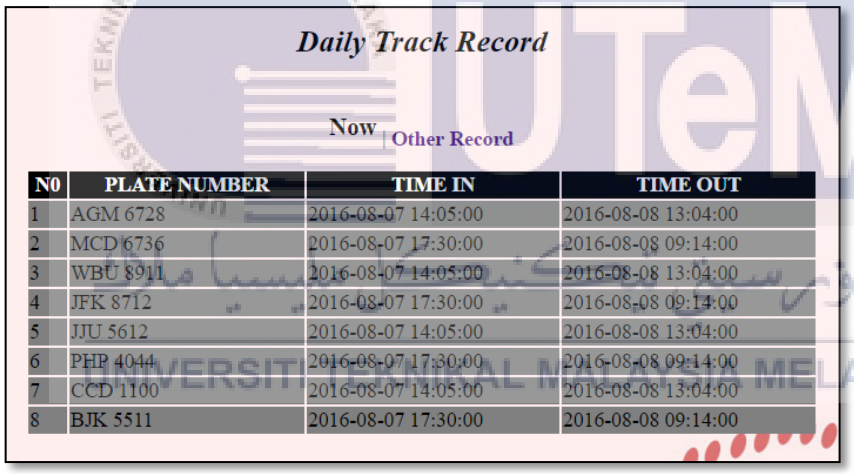
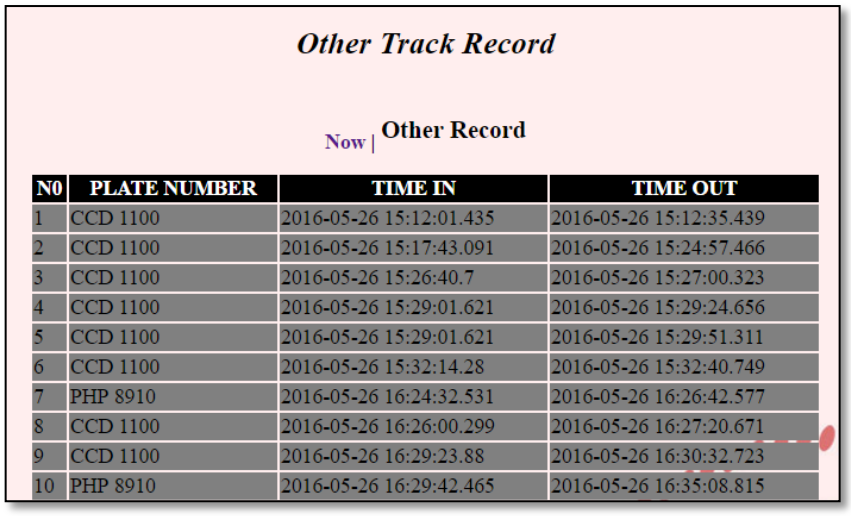


Figure 4.25 Web Interface for Admin - Daily Track Record I



Other Track Record

Now | Other Record

N0	PLATE NUMBER	TIME IN	TIME OUT
1	CCD 1100	2016-05-26 15:12:01.435	2016-05-26 15:12:35.439
2	CCD 1100	2016-05-26 15:17:43.091	2016-05-26 15:24:57.466
3	CCD 1100	2016-05-26 15:26:40.7	2016-05-26 15:27:00.323
4	CCD 1100	2016-05-26 15:29:01.621	2016-05-26 15:29:24.656
5	CCD 1100	2016-05-26 15:29:01.621	2016-05-26 15:29:51.311
6	CCD 1100	2016-05-26 15:32:14.28	2016-05-26 15:32:40.749
7	PHP 8910	2016-05-26 16:24:32.531	2016-05-26 16:26:42.577
8	CCD 1100	2016-05-26 16:26:00.299	2016-05-26 16:27:20.671
9	CCD 1100	2016-05-26 16:29:23.88	2016-05-26 16:30:32.723
10	PHP 8910	2016-05-26 16:29:42.465	2016-05-26 16:35:08.815

Figure 4.26 Web Interface for Admin - Daily Track Record II

- Admin site: Import csv file that contain user entrance

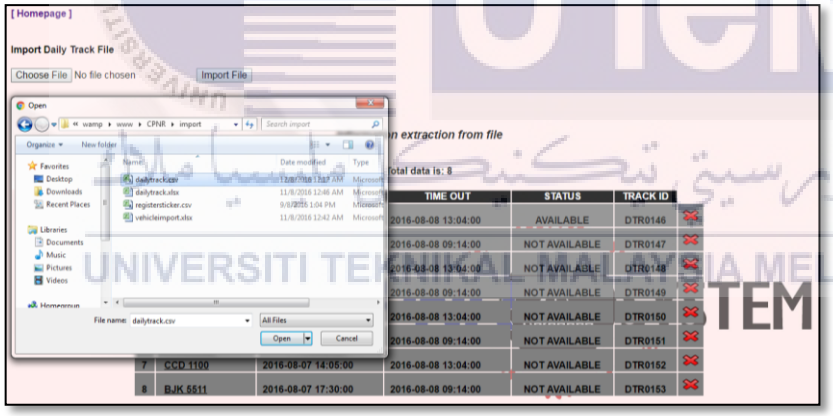


Figure 4.27 Web Interface for Admin - Import CSV File

- Owner site: Owner Profile Details

Figure 4.28 Web Interface for User - Profile Details

- Owner site: Update Profile

Figure 4.29 Web Interface for User - Update Profile

- Owner site: Upload Photo

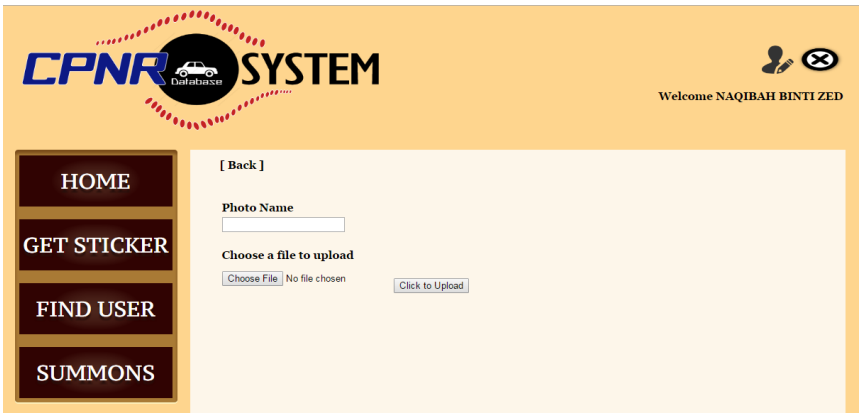


Figure 4.30 Web Interface for User - Upload Photo

- Owner site: List of Sticker Registration

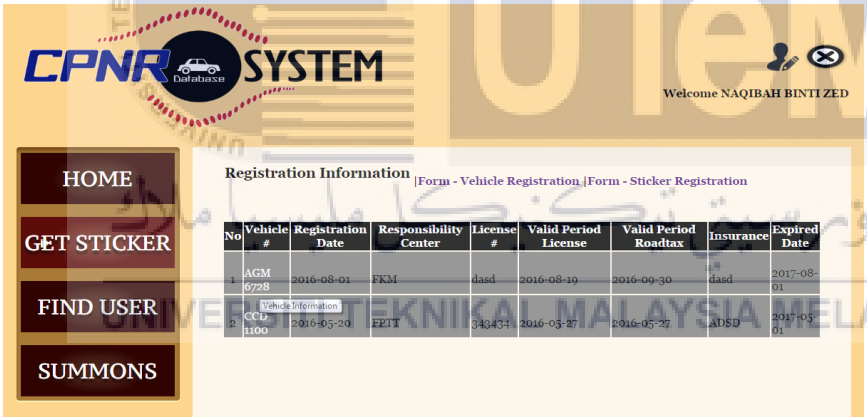


Figure 4.31 Web Interface for User - List of Sticker Registration

- Owner site: Vehicle registration form I

Figure 4.32 Web Interface for User - Vehicle Registration Form I

- Owner site: Vehicle registration form II

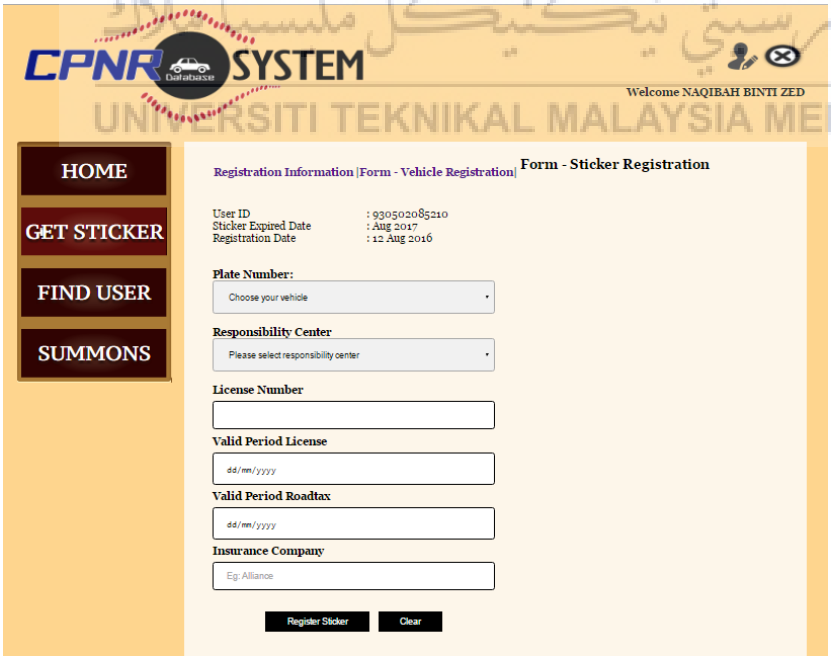


Figure 4.33 Web Interface for User - Vehicle Registration Form II

- Owner site: Find user page

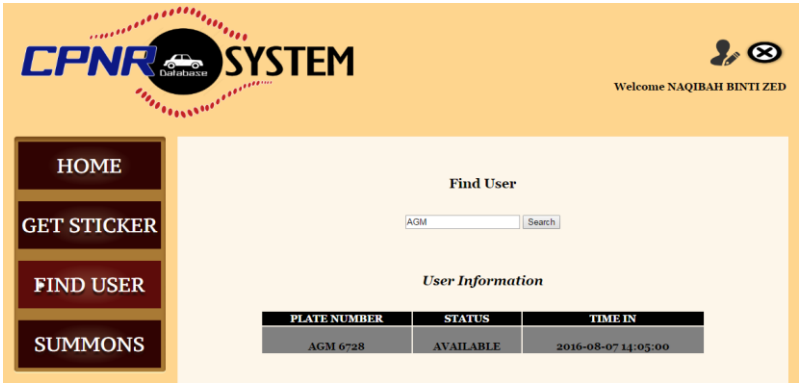


Figure 4.34 Web Interface for User - Find User

- Owner site: Owner summons information I

Figure 4.35 Web Interface for User - Summons Information I

- Owner site: Owner summons information II

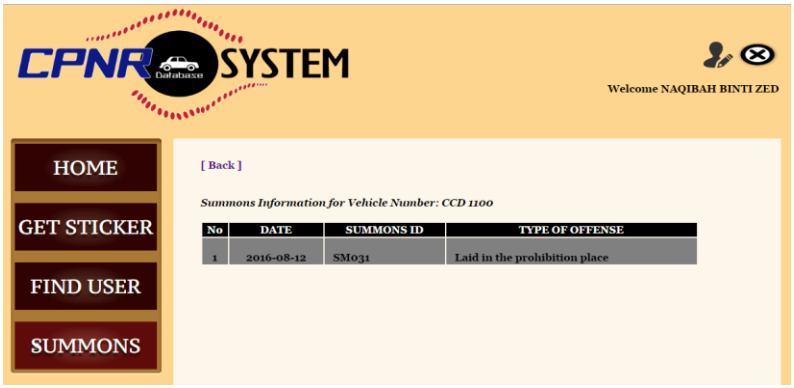


Figure 4.36 Web Interface for User - Summons Information II

4.8 Conclusion

The design phase is one of the key elements to the project. Without a detailed design, the system cannot be constructed, implemented or operated. Based on the physical and logical design that has been providing in this document, the structure of the database has been created within specific modules that will develop using PostgreSQL. The overall view of conceptual design shows the relationship for each entity that depending with each other. Thus, for the next chapter will discuss on implementations for the database and system that can be support the proposed objectives.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

The implementation phase concerns about describing how the information system will be deployed as an operational system. This phase includes efforts required to implement, identified and resolve system problems and plan for sustainment. The activities comprised in the framework implementation are system coding and debugging.

This chapter is an assurance that the system is developed to meet all the prerequisites that have been set out in the last chapter. The system will be produced stage by stage according to the requirements of module.

The main objective of this implementation phase is to define all planned activities in order to ensure successful implementation to production operations within of the requirements and duration of time.

5.2 Software Development Environment Setup

Software development environment should be set up before the system is developed. It enables to write programs for a particular language or platform. Management system for CPNR database system is a web based. The system and the database organization will be presented in the system which will be the essential access point.

The web server used is Wamp Server while PHP is the language to write software and Windows is a platform where the environment begins to set up. PostgreSQL connector integrates between information that is inserted through interface and the PostgreSQL database. Figure 5.2 shows the system framework for CPNR database system.

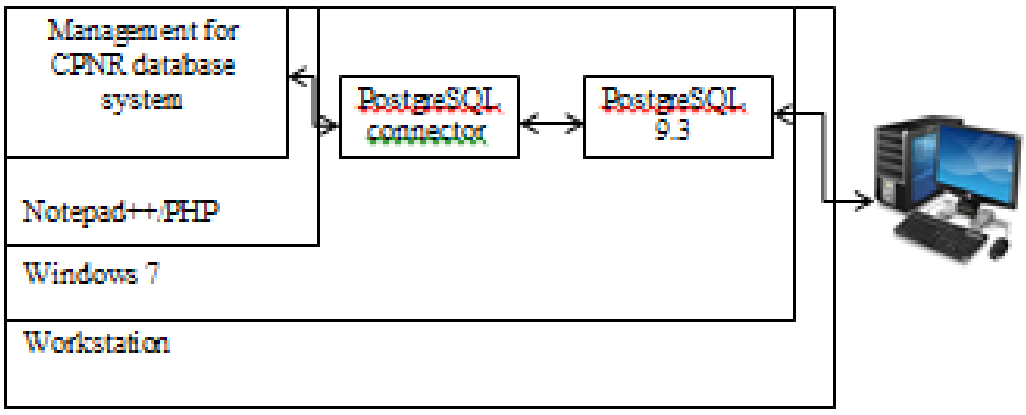


Figure 5.1 System Framework for CPNR Database System

5.2.1 Installation Setup

There are several steps in order to install PostgreSQL in Windows Platform successfully.

- Installation of Web Server

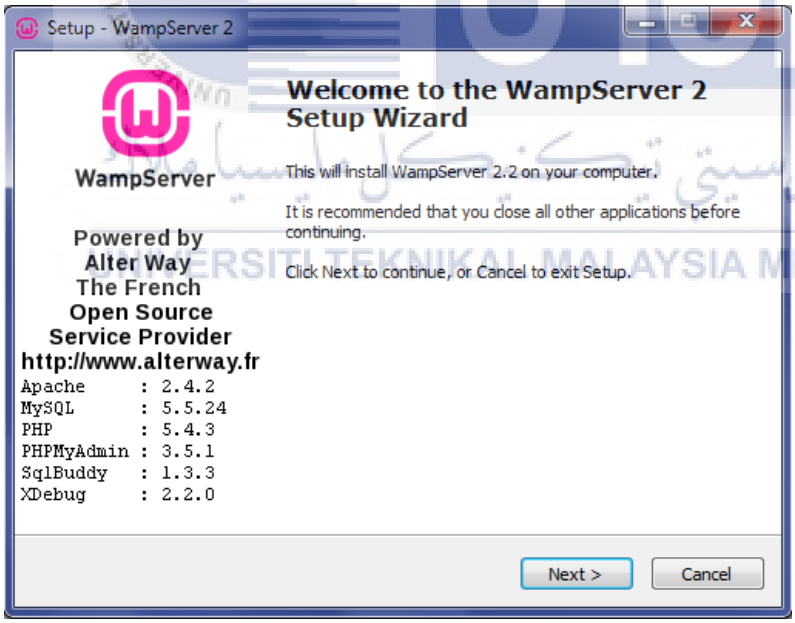
Step 1: Downloading WampServer from the available free software

Step 2: Installing WampServer in the computer

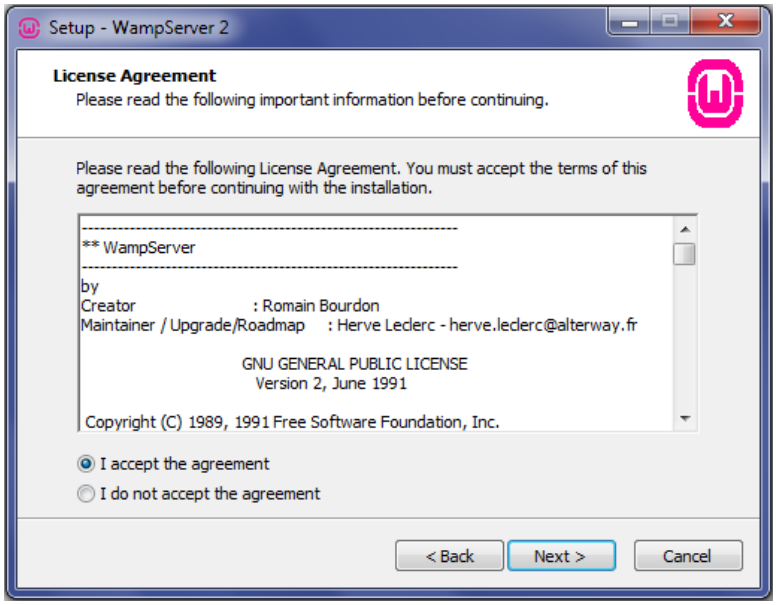
2.1 Open the WampServer folder and double click on the installer file

2.2 Click **run** to start the installation process

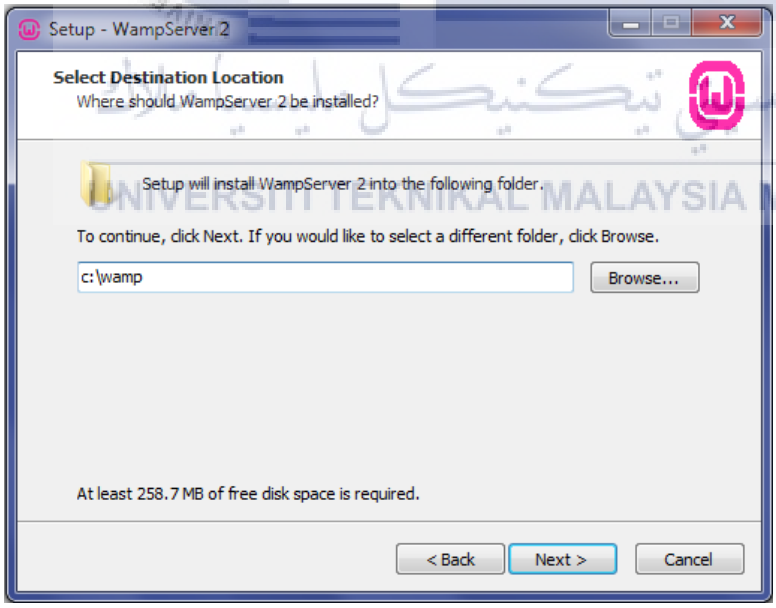
2.3 Setup – WampServer 2 screen will appear. Click **Next** button to continue



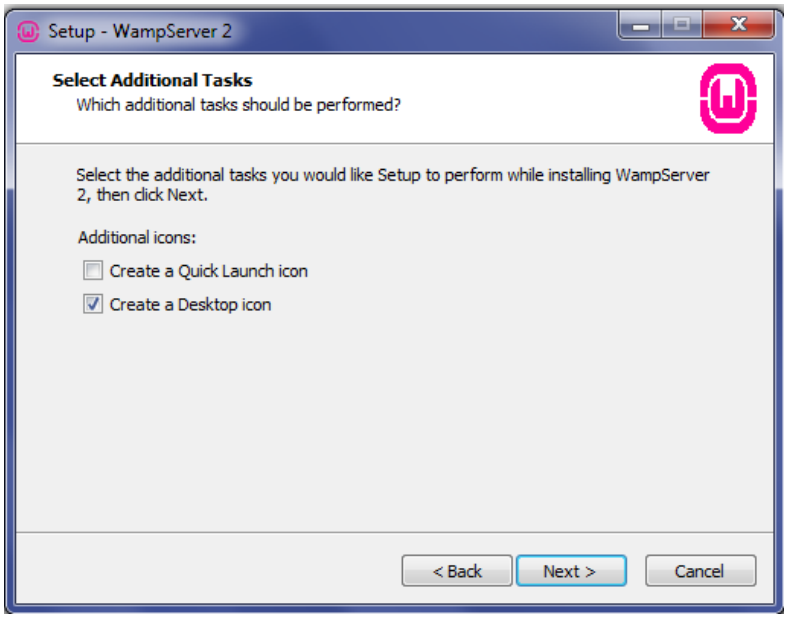
2.4 License Agreement screen will appear. Tick on the **I accept the agreement** and click **Next** button



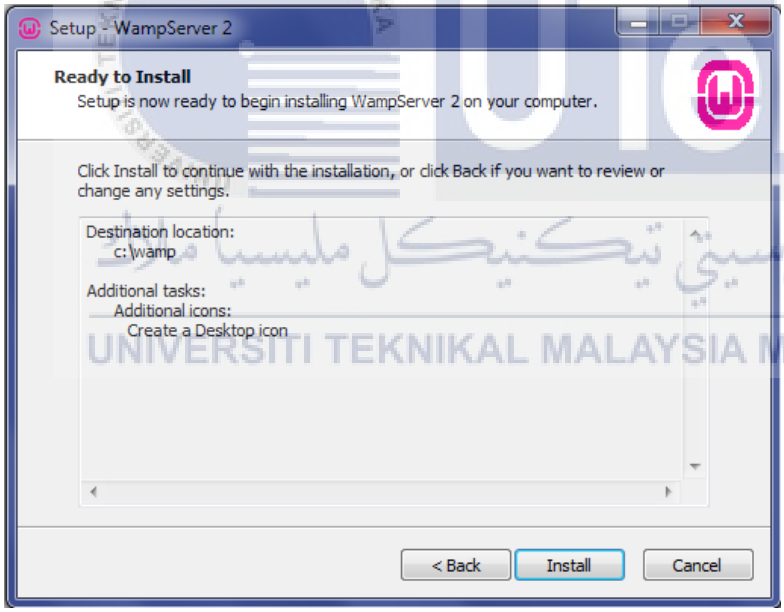
2.5 Select Destination Location screen will appear. Choose the location for WampServer and click **Next** button



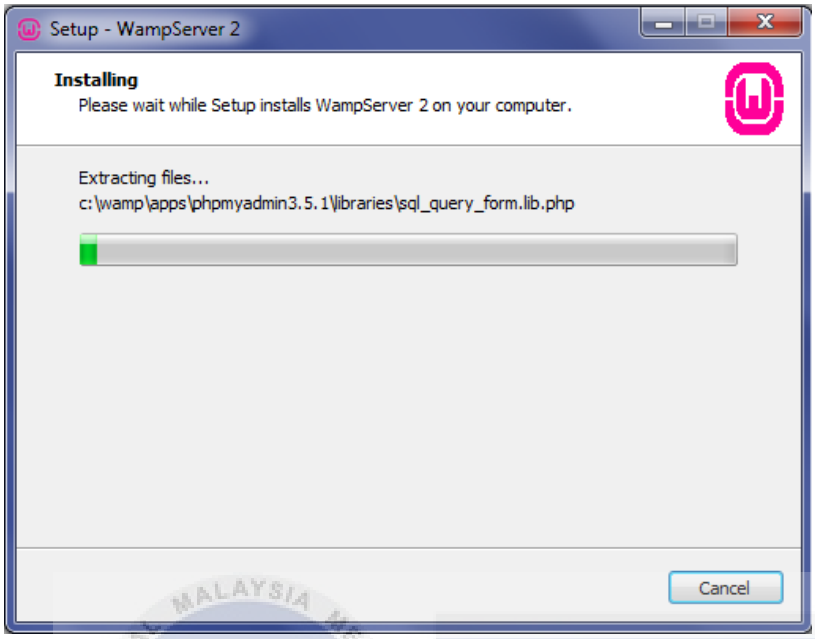
2.6 Select Additional Tasks screen will appear. Select either one of the option and click **Next** button



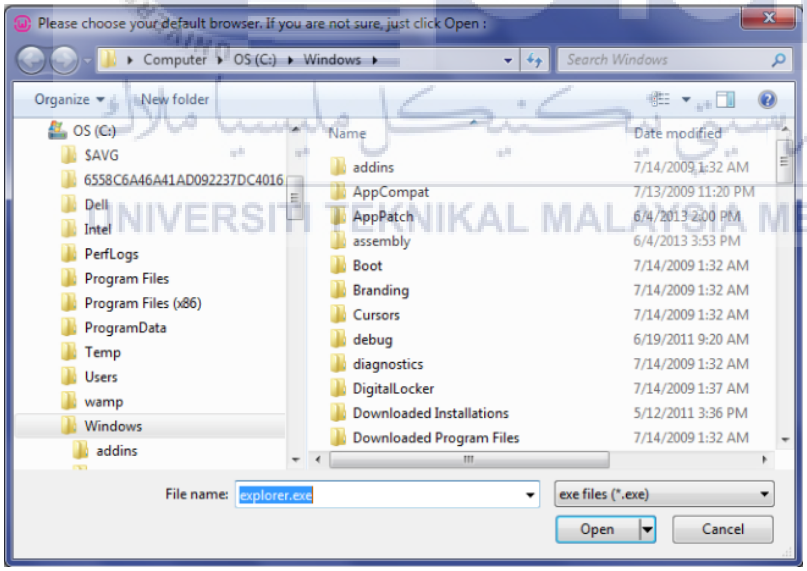
2.7 Ready to Install screen will appear. Click **Install** button to continue installation



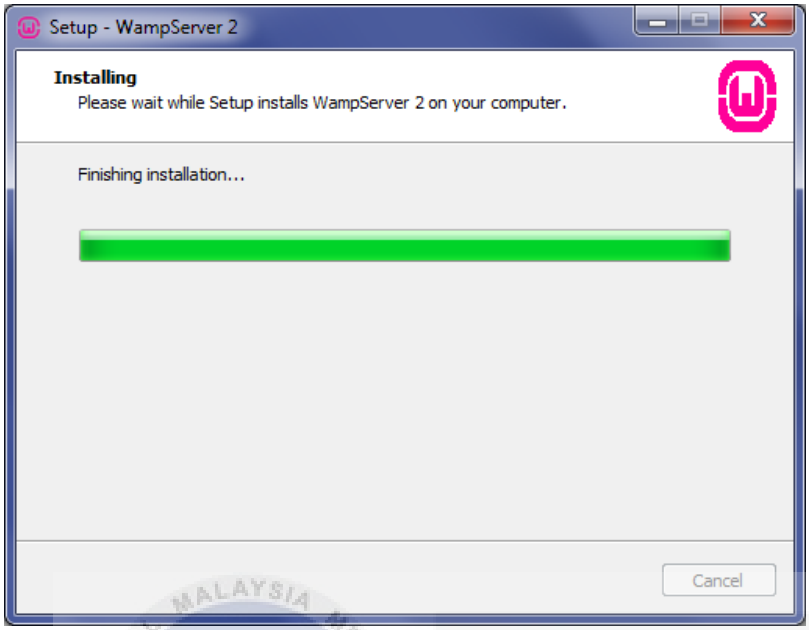
2.8 WampServer will begin to install



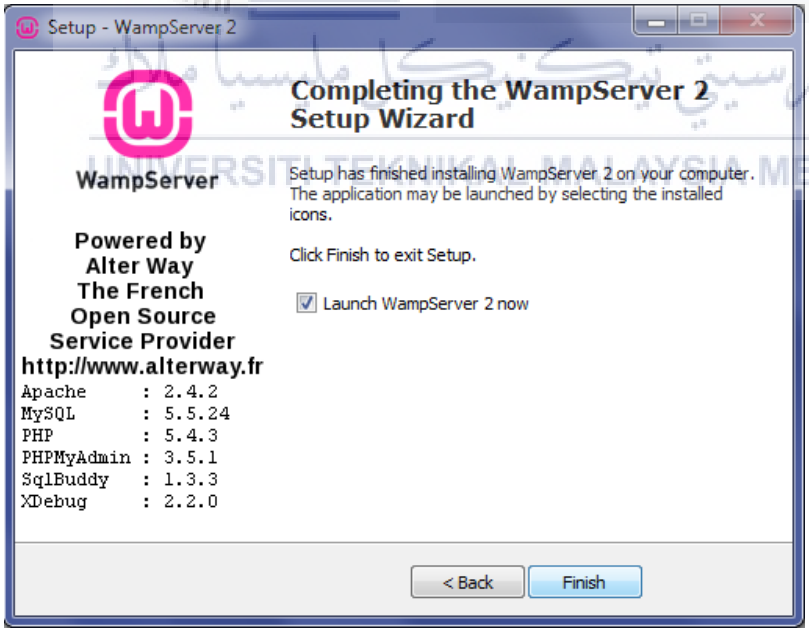
2.9 Select the default browser and click **Open** button



2.10 Then, setup screen will appear



2.11 Finally, the installation has complete

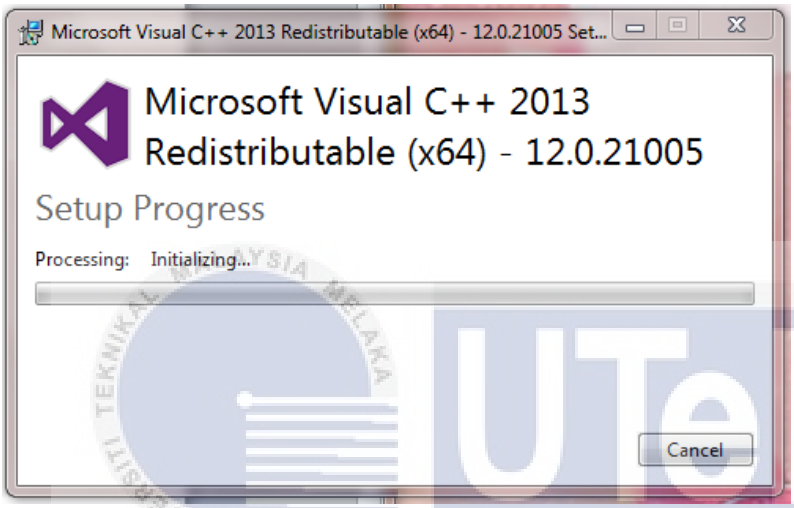


- Installation of PostgreSQL

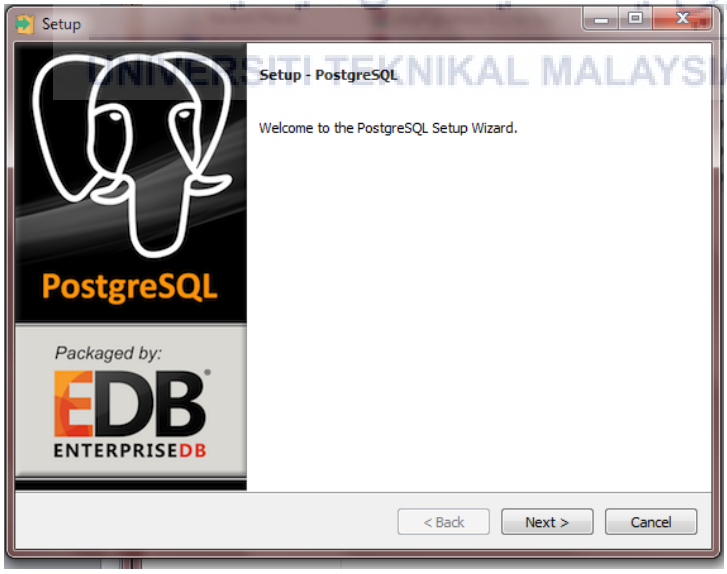
Step 1: Install PostgreSQL, GUI Administration and PHP Based Web Application Site

1.1 Double click on the installer icon

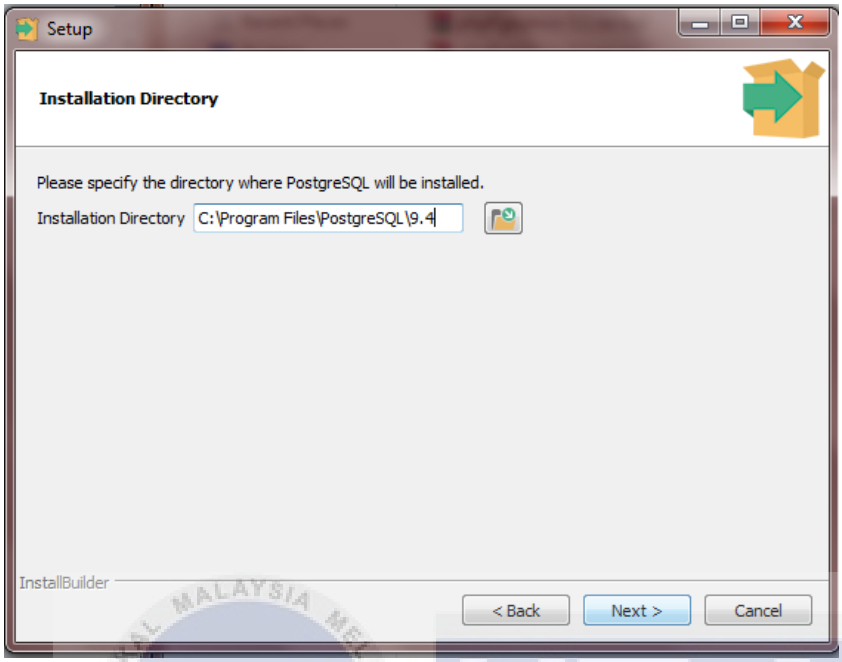
1.2 Setup process will be processing

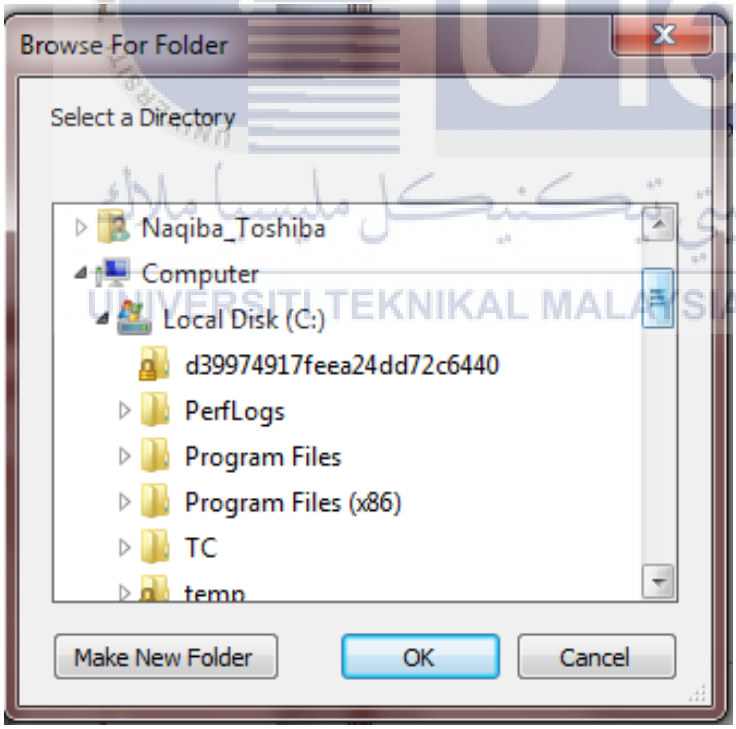


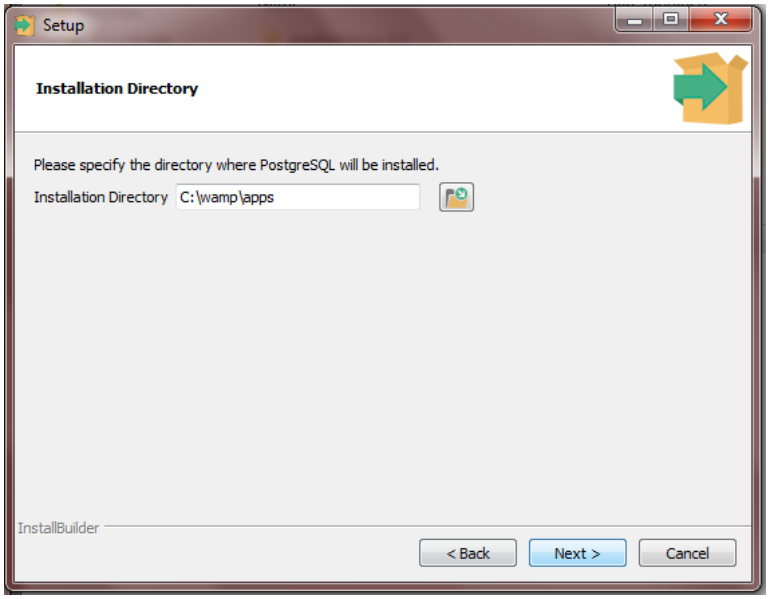
1.3 At the setup phase, click on Next button



1.4 Choose the installation directory



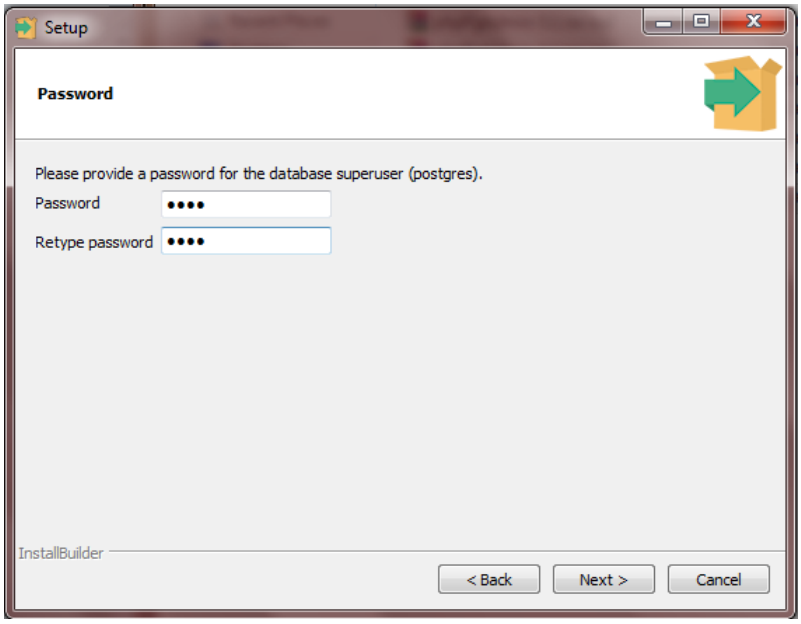




1.5 Continue installation by click on Next button

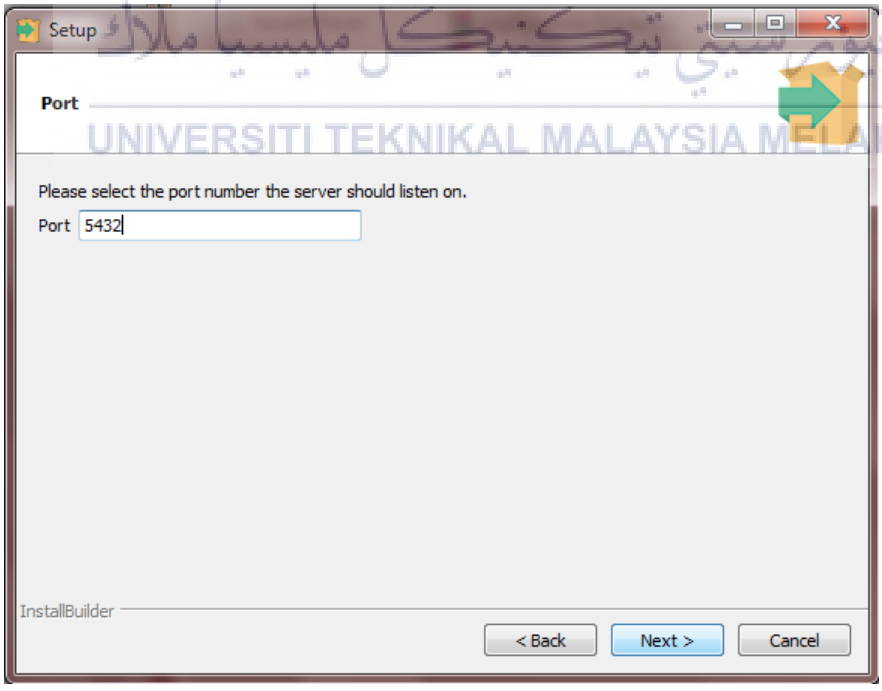
1.6 User need to set the login password for the root user postgres

1.7 Continue installation by click on Next button



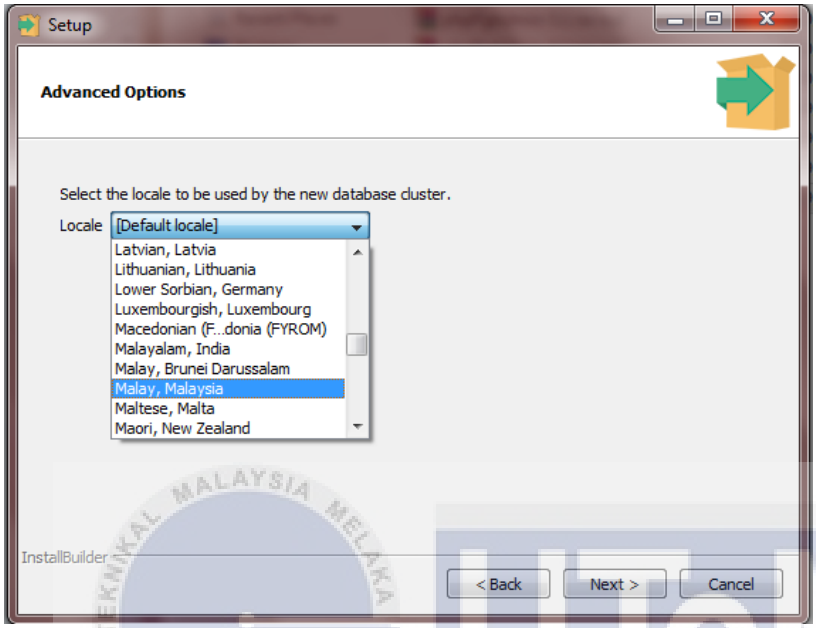
1.8 Then, port number screen will be displayed. Set the port number and make sure it does not conflict with the MySQL port 3306.

1.9 Continue installation by click on Next button

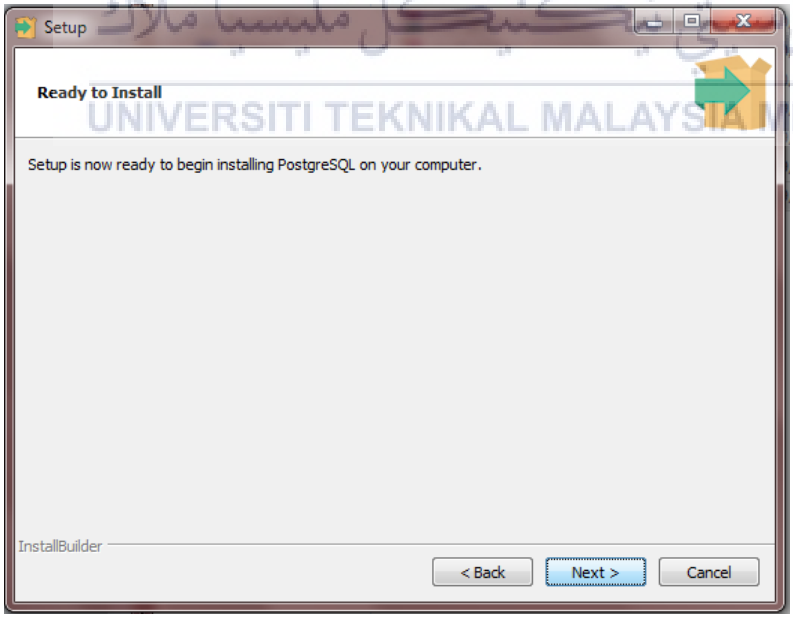


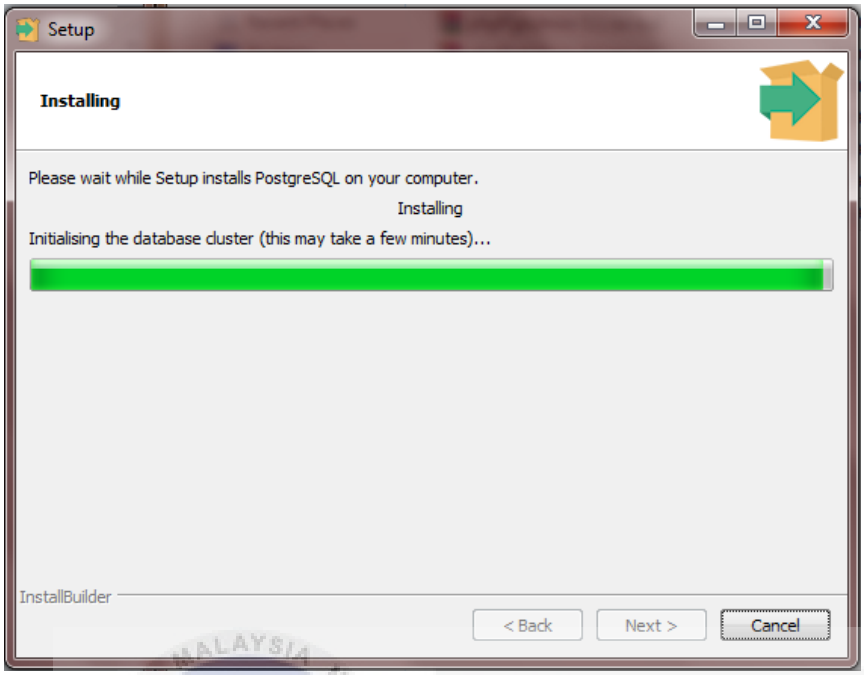
1.10 Set the Default Locale

1.11 Continue installation by click on Next button

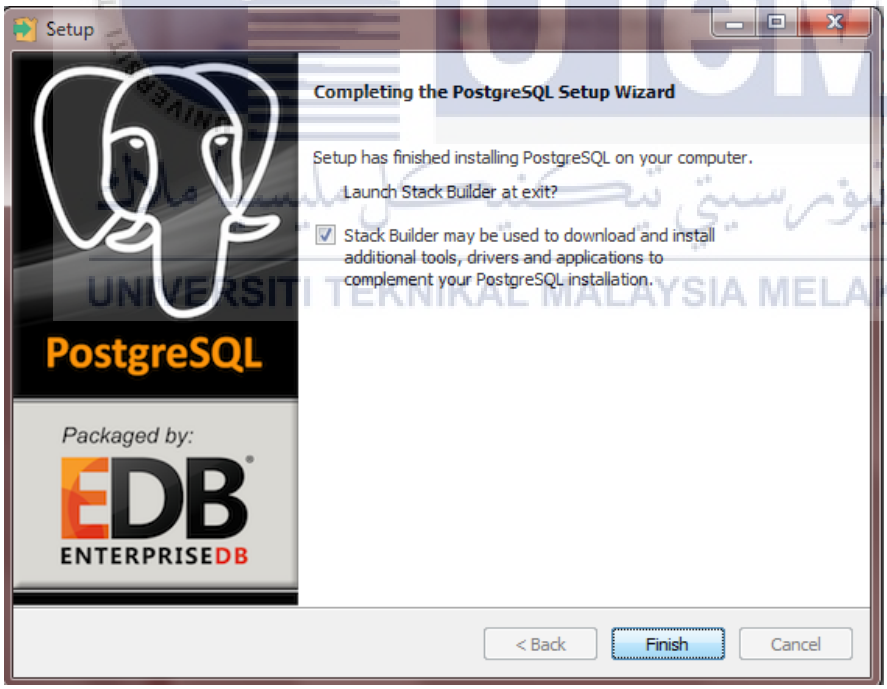


1.12 Now click Next to begin installation



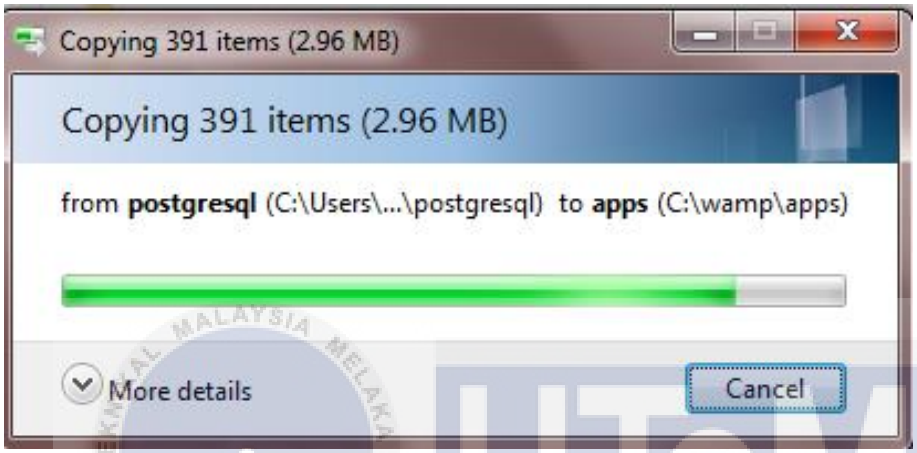


1.13 Then click Finish



Step 2: Copy phpPgAdmin-5.1 into computer

2.1 Copy phpPgAdmin 5.1 into Computer – Local Disk (C:) – wamp folder – apps

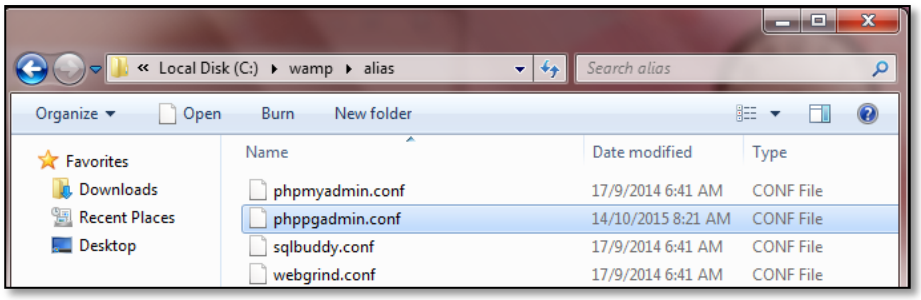


2.2 Go to Computer – Local Disk (C:) – wamp folder – alias

Make one file phppgadmin.conf into alias folder. This file contain the code as shown on the right

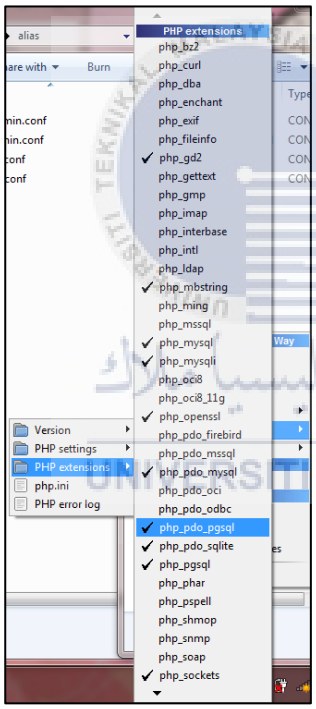
```
Alias /phppgadmin "C:/wamp/apps/phpPgAdmin-5.1/"

<Directory "C:/wamp/apps/phpPgAdmin-5.1/">
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
       Order Deny,Allow
         Allow from all
</Directory>
```



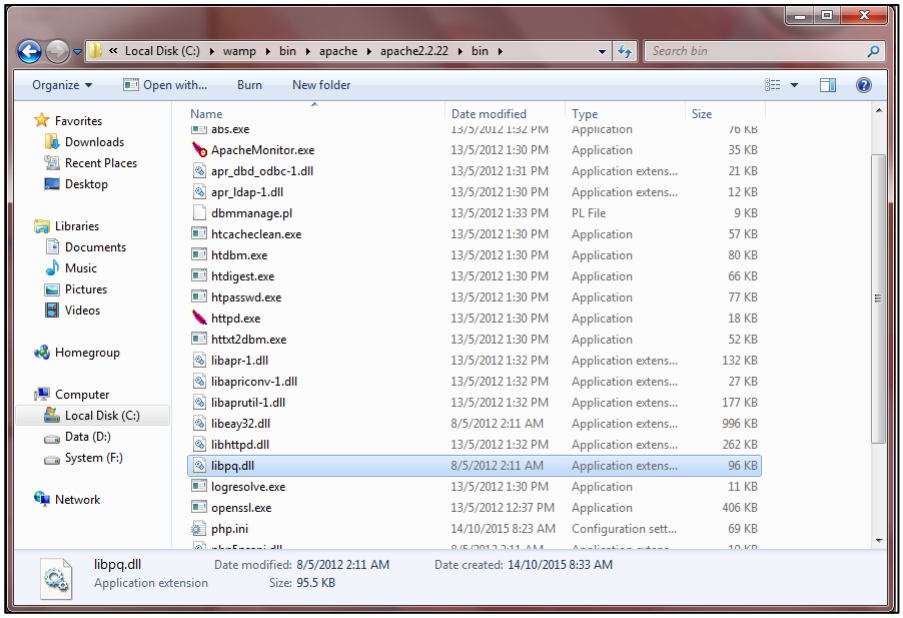
2.3 Then, go to wamp tray icon – click php – choose php extensions

Thick php_pgsql and php_pdo_pgsql

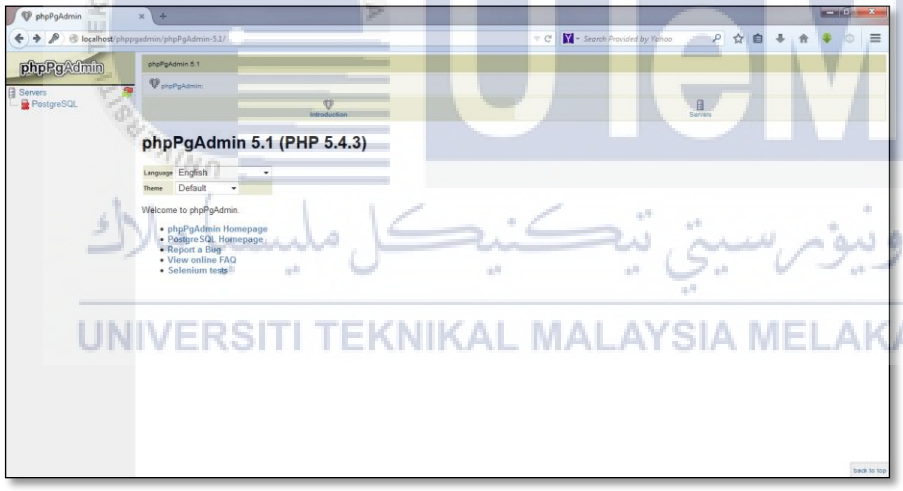


2.4 Then, go to Computer – Local Disk (C:) – wamp folder – bin – php – php5.4.3

Copy the libpq.dll and paste into Computer – Local Disk (C:) – wamp folder – bin – apache2.2.22 – bin



2.5 PhpPgAdmin is ready to use



5.2.2 Database Creation and Database Object Creation

Database creation

In order to create a database, the PostgreSQL server must be up and running. Database is created with the SQL command:-

```
CREATE DATABASE CPNR WITH OWNER = naqibah;
```

Database name must follow the usual rules for SQL identifiers. Owner is a super user or has the special CREATEDB privilege. Besides that, create database cannot be executed inside a transaction block.

Database Object Creation

Database objects in PostgreSQL consist of tables, views, functions, indexes and etc. There are eight tables for this CPNR database system which are owner, registration, staff, vehicle_daily_track, daily_track, summons, summons_type and vehicle. Other than that, there are several database objects that have been implemented in this system such as sequences, functions and triggers. The example will be shown below and in the next sub-title.

Example of create sequences

A sequence is used for generating a unique numeric identifier that is typically used to generate the primary key. Sequences are implemented in daily_track, registration, staff, summons, summons_type and vehicle_daily_track table to make auto generated numbers when data is inserted.

- Create sequence dailyTrackId_seq

```
CREATE SEQUENCE "daily_track_dailyTrackId_seq"
  INCREMENT BY 1
  START WITH 1;
```

- Create sequence registration_seq

```
CREATE SEQUENCE registration_seq
  INCREMENT BY 1
  START WITH 1;
```

- Create sequence staff_id_seq

```
CREATE SEQUENCE staff_id_seq
  INCREMENT BY 1
  START WITH 1;
```

- Create sequence summons_seq

```
CREATE SEQUENCE summons_seq
  INCREMENT BY 1
  START WITH 1;
```

- Create sequence summons_type_id_seq

```
CREATE SEQUENCE summons_type_id_seq
  INCREMENT BY 1
  START WITH 1;
```

- Create sequence vehicle_track_id_seq

```
CREATE SEQUENCE vehicle_track_id_seq
  INCREMENT BY 1
  START WITH 1;
```

5.3 Database Implementation

In the database implementation, the database will be tested using database queries, for example, restricting information, joins, aggregate functions and sub-query to verify the flow of information in the database is correct.

5.3.1 Data Definition Language

Data Definition Language (DDL) consist of script for create table and constraint that include in database implementation for CPNR database system.

5.3.1.1 Create Table

CPNR database system consists of eight tables which are Owner, Registration, Staff, Vehicle_Daily_Track, Daily_Track, Summons, Summons_Type and Vehicle table. These tables are used to stored related information in order to implement CPNR database system.

- Create table Owner

```
CREATE TABLE owner
(
  user_id character varying(20) NOT NULL,
  user_name character varying(50) NOT NULL,
  user_address character varying(100) NOT NULL,
  user_email character varying(50) NOT NULL,
  user_password character varying(256) NOT NULL,
  user_category character varying(20) NOT NULL,
  user_position character varying(30) NOT NULL,
  user_contact_number character varying(20) NOT NULL,
  user_office_number character varying(20),
  user_level character varying(10),
  user_school_number character varying(30),
  CONSTRAINT "OWNER_pkey" PRIMARY KEY (user_id)
);
```

- Create table Registration

```
CREATE TABLE registration
(
  registration_id character varying(100) NOT NULL,
  responsibility_center character varying(30) NOT NULL,
  license_number character varying(50) NOT NULL,
  valid_period_license date NOT NULL,
  valid_period_roadtax character varying(50) NOT NULL,
  insurance_company character varying(50) NOT NULL,
  registration_date date NOT NULL,
  sticker_status character varying(50),
  expired_date character varying,
  payment character varying(20),
  user_id character varying(20) NOT NULL,
  vehicle_id character varying(20) NOT NULL,
  staff_id character varying(20),
  CONSTRAINT registration_pkey PRIMARY KEY
(registration_id),
  CONSTRAINT registration_staff_id_fkey FOREIGN KEY
(staff_id)
  REFERENCES staff (staff_id) MATCH SIMPLE
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT registration_user_id_fkey FOREIGN KEY
(user_id)
  REFERENCES owner (user_id) MATCH SIMPLE
  ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT registration_vehice_id_fkey FOREIGN KEY
(vehicle_id)
  REFERENCES vehicle (vehicle_id) MATCH SIMPLE
  ON UPDATE CASCADE ON DELETE RESTRICT
);
```

- Create table Staff

```
CREATE TABLE staff
(
  staff_id character varying(20) NOT NULL,
  staff_name character varying(50) NOT NULL,
  staff_email character varying(50) NOT NULL,
  staff_password character varying(256) NOT NULL,
  staff_address character varying(100) NOT NULL,
  staff_contact_number character varying(20) NOT NULL,
  user_level character varying(10),
  staff_image oid,
  CONSTRAINT staff_pkey PRIMARY KEY (staff_id)
);
```

- Create table Vehicle_Daily_Track

```
CREATE TABLE vehicle_daily_track
(
  vehicle_track_id character varying NOT NULL,
  track_time_out timestamp without time zone,
  track_time_in timestamp without time zone,
  registration_id character varying NOT NULL,
  daily_track_id character varying NOT NULL,
  CONSTRAINT vehicle_daily_track_pkey PRIMARY KEY
(vehicle_track_id),
CONSTRAINT vehicle_daily_track_daily_track_id_fkey FOREIGN
KEY (daily_track_id)
REFERENCES daily_track (daily_track_id) MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE CASCADE,
CONSTRAINT vehicle_daily_track_registration_id_fkey FOREIGN
KEY (registration_id)
REFERENCES registration (registration_id) MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE CASCADE
);
```

- Create table Daily_Track

```
CREATE TABLE daily_track
(
  daily_track_id character varying NOT NULL,
  daily_status_of_car character varying(20),
  daily_image_name character varying(100),
  daily_image_plate character varying,
  CONSTRAINT daily_track_pkey PRIMARY KEY (daily_track_id)
);
```

- Create table Summons

```
CREATE TABLE summons
(
  summons_id character varying NOT NULL,
  summons_payment_status character varying(30),
  summons_payment character varying(30) NOT NULL,
  summons_place_offense character varying(100) NOT NULL,
  registration_id character varying NOT NULL,
  summons_ownerid character varying(100),
 summons_date date,
 CONSTRAINT summons_pkey PRIMARY KEY (summons_id),
 CONSTRAINT summons_registration_id_fkey FOREIGN KEY
(registration_id)
 REFERENCES registration (registration_id) MATCH SIMPLE
 ON UPDATE RESTRICT ON DELETE CASCADE
);
```

- Create table Summons_Type

```
CREATE TABLE summons_type
(
  summons_type_id character varying(100),
  summons_id character varying(100),
  type_offense character varying(200),
CONSTRAINT summons_type_summons_id_fkey FOREIGN KEY
(summons_id)
 REFERENCES summons (summons_id) MATCH SIMPLE
 ON UPDATE RESTRICT ON DELETE CASCADE
);
```

- Create table Vehicle

```
CREATE TABLE vehicle
(
  vehicle_id character varying(20) NOT NULL,
  vehicle_type character varying(20),
  vehicle_cc character varying(20),
  vehicle_color character varying(20),
  vehicle_year character varying(20),
  user_id character varying,
  CONSTRAINT "VEHICLE_pkey" PRIMARY KEY (vehicle_id),
  CONSTRAINT vehicle_user_id_fkey FOREIGN KEY (user_id)
  REFERENCES owner (user_id) MATCH SIMPLE
  ON UPDATE RESTRICT ON DELETE CASCADE
);
```

5.3.1.2 Create Constraint

Constraint is used to control over data in tables as any condition stated. Constraint can be not null, check, unique, primary key and foreign key that can be apply in the column in a table. Each of them has its own meaning and implementation.

- Not null constraint

```
CREATE TABLE summons
(
  summons_id character varying NOT NULL,
  summons_payment_status character varying(30),
  summons_payment character varying(30) NOT NULL,
  summons_place_offense character varying(100) NOT NULL,
  registration_id character varying NOT NULL,
  summons_ownerid character varying(100),
  summons_date date,
);
```


- Check constraint

```
ALTER TABLE vehicle_daily_track
ADD CONSTRAINT track_time_out CHECK
(track_time_out>track_time_in);
```

- Unique constraint

```
ALTER TABLE summons_type
ADD CONSTRAINT summons_type_id UNIQUE(summons_type_id);
```

- Primary key

```
CREATE TABLE daily_track
(
  daily_track_id character varying NOT NULL,
  daily_status_of_car character varying(20),
  daily_image_name character varying(100),
  daily_image_plate character varying,
  CONSTRAINT daily_track_pkey PRIMARY KEY (daily_track_id)
);
```

- Foreign key

```
CREATE TABLE summons_type
(
  summons_type_id character varying(100),
  summons_id character varying(100),
  type_offense character varying(200),
  CONSTRAINT summons_type_summons_id_fkey FOREIGN KEY
(summons_id)
  REFERENCES summons (summons_id) MATCH SIMPLE
  ON UPDATE RESTRICT ON DELETE CASCADE
);
```


5.3.2 Data Manipulation Language

Data Manipulation Language (DML) consist of script for insert, update and delete that include in database implementation for CPNR database system.

5.3.2.1 Insert Statement

- Insert into Owner

```
INSERT INTO owner VALUES ('930502085210', 'NAQIBAH BINTI ZED', 'NO 75 JALAN DI 17 TAMAN DESA IDAMAN 76100 DURIAN TUNGGAL', 'naqibahzed@yahoo.com', '03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4', 'Student', '-', '013-6165383', '000', '1', 'B031410059');
```

```
INSERT INTO owner VALUES ('950409078921', 'MOHD QAYYUM BIN ABD JALIL', 'NKJKA', 'BCSJH', '03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4', 'Student', 'C,KBSI', 'CBDK', 'BCSDK', '1', 'BCSK');
```

```
INSERT INTO owner VALUES ('670117037821', 'PUNITHA A/P MUTUSAMI', 'KELANTAN', 'punitha@yahoo.com', '03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4', 'Staff', 'LECTURER', '019-3412888', '2040', '1', 'S102');
```

- Insert into Registration

```
INSERT INTO registration VALUES ('REG01', 'FTMK', '989787', '2016-05-10', '2016-05-19', 'ALLIANCE', '2016-05-18', 'Not available', 'May 2017', 'not done', '930502085210', 'AGM 6728', 'null');
```

- INSERT INTO registration VALUES ('REG02', 'FPTT', '343434', '2016-05-27', '2016-05-27', 'ADSD', '2016-05-20', 'Not available', 'May 2017', 'not done', '930502085210', 'CCD 1100', 'null');

- INSERT INTO registration VALUES ('REG03', 'FPTT', 'W 327077', '2017-12-15', '2016-07-22', 'ALLIANCE', '2016-05-26', 'Not available', 'May 2017', 'not done', '670117037821', 'PHP 8910', 'null');

- Insert into Staff

- INSERT INTO staff VALUES ('S001', 'NUR MAISARAH ALIA BINTI KAMARUDDIN', 'maisalia@gmail.com', '03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4', 'NO 67 JALAN PANDAN INDAH 8 TAMAN KRUBUNG 67120 MELAKA ', '019-67125441', '2', NULL);

- INSERT INTO staff VALUES ('S003', 'ZHAFRAN AZKA BIN MOHD AZWAN', 'zhafran@yahoo.com', '8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92', 'NO 9 JALAN 17 BUKIT KATIL 2300 MELAKA', '019-7612222', '2', NULL);

- Insert into Vehicle_Daily_Track

- INSERT INTO vehicle_daily_track VALUES ('VDT082', NULL, '2016-05-29 11:32:13.248', 'REG02', 'DTR081');

- INSERT INTO vehicle_daily_track VALUES ('VDT084', NULL, '2016-05-29 11:32:45.296', 'REG04', 'DTR083');

```
INSERT INTO vehicle_daily_track VALUES ('VDT085', NULL,
'2016-05-29 11:33:10.473', 'REG01', 'DTR084');
```

- Insert into Daily_Track

```
INSERT INTO daily_track VALUES ('DTR081', 'AVAILABLE',
NULL, ' CCD 1100 ');
INSERT INTO daily_track VALUES ('DTR083', 'AVAILABLE',
NULL, ' JJC 9845 ');

INSERT INTO daily_track VALUES ('DTR084', 'AVAILABLE',
NULL, ' AGM 6728 ');
```

- Insert into Summons

```
INSERT INTO summons VALUES ('SM023', NULL, 'NOT PAID',
'FKE', 'REG03', '670117037821', '2016-05-28');

INSERT INTO summons VALUES ('SM024', NULL, 'NOT PAID',
'HEPA', 'REG04', '950409078921', '2017-01-17');

INSERT INTO summons VALUES ('SM022', NULL, 'PAID',
'FTMK', 'REG01', '930502085210', '2016-05-26');
```

- Insert into Summons_Type

```
INSERT INTO summons_type VALUES ('SUMTY020', 'SM022',
'Blocking the path');

INSERT INTO summons_type VALUES ('SUMTY026', 'SM024',
'Reckless driving');
```

```
➢ INSERT INTO summons_type VALUES ('SUMTY022', 'SM023',
  'No sticker');
```

- Insert into Vehicle

```
➢ INSERT INTO vehicle VALUES ('AGM 6728', 'Car', 'KANCIL
  660', 'SILVER', '2008', '930502085210');

➢ INSERT INTO vehicle VALUES ('CCD 1100', 'Motorcycle', '-
  ', 'BLACK', '2012', '930502085210');

➢ INSERT INTO vehicle VALUES ('PHP 8910', 'Motorcycle', '-
  ', 'BLACK', '2013', '670117037821');
```

5.3.2.2 Update Statement

- Update Vehicle_Daily_Track

```
UPDATE vehicle_daily_track
SET track_time_out = now()
WHERE daily_track_id = daily_track_id AND registration_id =
registration_id;
```

- Update Daily_Track

```
UPDATE daily_track
SET daily_status_of_car = "AVAILABLE"
WHERE daily_track_id = daily_track_id AND daily_image_plate
= daily_image_plate;
```

- Update Summons

```
UPDATE summons
SET summons_payment = "PAID"
WHERE summons_id = summons_id AND registration_id =
registration_id;
```

5.3.2.3 Delete Statement

- Delete Vehicle_Daily_Track

```
DELETE FROM vehicle_daily_track
WHERE daily_track_id = daily_track_id;
```

5.3.3 Stored Procedure / Function

Function in PostgreSQL also known as Stored Procedure. Function allows carrying out operations that take several queries and round trips in a single function. Besides that, database can reuse the function instead of a middle-tier or duplicating code. There are several functions that are created in this Car Plate Number Recognition (CPNR) database system including function that used complex query, join at least to table, view, update, delete and insert. Below shows examples of each function used.

- Function to display report using complex query

```
CREATE OR REPLACE FUNCTION complex_view_report1()
RETURNS TABLE (vehicle_id character varying(100), total
character varying(40),
day character varying(100), month character varying(40),
year character varying(40)) AS
$$
```

```
 BEGIN
FOR vehicle_id, total, day, month, year IN
SELECT v.vehicle_id, count(his.registration_id) AS total,
date_part('day', track_time_in) as day,date_part('month',
track_time_in) as month,date_part('year', track_time_in) as
year
FROM history_vehicle_daily_track his, registration r,
vehicle v
WHERE his.registration_id = r.registration_id AND
r.vehicle_id = v.vehicle_id AND operation ='UPDATE'
GROUP BY v.vehicle_id, his.registration_id,
date_part('day', track_time_in),
date_part('month', track_time_in), date_part('year',
track_time_in)
LOOP
RETURN NEXT;
END LOOP;
END;
$$ LANGUAGE plpgsql;
```


- Function to display owner vehicle using joint at least two tables

```
CREATE OR REPLACE FUNCTION viewownervehicle()
RETURNS TABLE (

registration_id character varying(100),
vehicle_id character varying(20),
user_id character varying(20),
valid_period_license date,
valid_period_roadtax character varying(50)) AS
$$
```

```
BEGIN
FOR registration_id, vehicle_id, user_id,
valid_period_license, valid_period_roadtax IN
select r.registration_id, r.vehicle_id, r.user_id,
r.valid_period_license, r.valid_period_roadtax
from owner o, vehicle v, registration r where r.user_id=
o.user_id AND r.vehicle_id = v.vehicle_id
LOOP
RETURN NEXT;
END LOOP;
END;
$$ LANGUAGE plpgsql;
```

- Function to display user information

```
CREATE FUNCTION viewregisteruser()
RETURNS SETOF owner AS
$$
DECLARE
rec record;
BEGIN
FOR rec IN (SELECT * FROM owner) LOOP
RETURN NEXT rec;
END LOOP;
END;
$$ LANGUAGE plpgsql;
```

- Function to insert vehicle information

```
CREATE OR REPLACE FUNCTION Insert_register_vehicle(user_id
character varying, vehicle_id
```

```
character varying(20), vehicle_type character varying,
vehicle_cc character varying(20), vehicle_color character
varying, vehicle_year character varying(20))
RETURNS void AS
$BODY$
BEGIN
INSERT INTO vehicle(user_id, vehicle_id, vehicle_type,
vehicle_cc, vehicle_color, vehicle_year)
VALUES(user_id, vehicle_id, vehicle_type, vehicle_cc,
vehicle_color, vehicle_year);
END;
$BODY$
LANGUAGE 'plpgsql' ;
```

- Function to delete user information

```
CREATE OR REPLACE FUNCTION deleteregistereduser(Id
character varying(20))
RETURNS character varying(20) AS
$BODY$
BEGIN
delete from owner where user_id = Id;
RETURN 1;
END;
$BODY$
LANGUAGE plpgsql;
```

- Function to update user information

```
CREATE OR REPLACE FUNCTION updateregistereduser(id
character varying(20), address character varying(100),
```

```
jawatan character varying(30), contactno character
varying(20), schoolno character varying(30))
RETURNS void AS
$BODY$
BEGIN
update owner
set
user_address = address,
user_position = jawatan,
user_contact_number = contactno,
user_school_number = schoolno
where user_id = id;
END;
$BODY$
 LANGUAGE plpgsql;
```

- Function to record old data, new data and update data in table history_vehicle_daily_track


```
CREATE OR REPLACE FUNCTION history_vehicle_daily_track()
RETURNS trigger AS
$BODY$
BEGIN
IF tg_op = 'DELETE' THEN
INSERT INTO history_vehicle_daily_track
VALUES (old.vehicle_track_id, old.track_time_out,
old.track_time_in, old.registration_id, old.daily_track_id,
current_timestamp, tg_op);
RETURN old;
END IF;
IF tg_op = 'INSERT' THEN
INSERT INTO history_vehicle_daily_track
```

```
VALUES (new.vehicle_track_id, new.track_time_out,
new.track_time_in, new.registration_id, new.daily_track_id,
current_timestamp, tg_op);
RETURN new;
END IF;
IF tg_op = 'UPDATE' THEN
INSERT INTO history_vehicle_daily_track
VALUES (new.vehicle_track_id, new.track_time_out,
new.track_time_in, new.registration_id, new.daily_track_id,
current_timestamp, tg_op);
RETURN new;
END IF;
END
$BODY$
LANGUAGE plpgsql;
```

- Function to add character for primary key

```
CREATE OR REPLACE FUNCTION staff_id()
RETURNS trigger AS
$BODY$DECLARE
ID int;
BEGIN select NEXTVAL('staff_id_seq') into ID;
NEW.staff_id := ('S00'|| ID);
RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

Table 5.1 Summary of Stored Procedure

Stored Procedure / Function

	Table	Description
Complex query	History_vehicle_daily_track, registration, vehicle	To display a report total in and out per owner in daily
Joint at least two tables	Owner, registration, vehicle	To display owner and its related vehicle and registration information
	Owner, registration, vehicle	To display list of user with expired license
DML insert	Vehicle	To insert vehicle information into vehicle table
DML view	Owner	To list all registered owner
DML delete	Owner	To delete data of owner
	Summons	To delete data of summons
	Staff	To delete data of staff
	Vehicle	To delete data of vehicle
DML update	Owner	To update data of owner

5.3.4 Triggers

Triggers in PostgreSQL are automatically invoked when a specified database event occur. PostgreSQL trigger can be stated before the operation is attempted, after operation has completed or instead of the operation. There are some triggers that applied for CPNR database system such as trigger after insert, update and delete and trigger before insert. Below shows examples of trigger used.

- Trigger after insert, update, delete on history_vehicle_daily_track table

```
CREATE TRIGGER trig_vehicle_daily_track
AFTER INSERT OR DELETE OR UPDATE ON
vehicle_daily_track
FOR EACH ROW
EXECUTE PROCEDURE history_vehicle_daily_track();
```

- Trigger before insert on table staff

```
CREATE TRIGGER staff_id_trigger
BEFORE INSERT ON staff
FOR EACH ROW
EXECUTE PROCEDURE staff_id();
```

Table 5.2 Summary of Trigger

Trigger

	Table	Description
After insert, update, delete	History_vehicle_ daily_track	The trigger will fire after user insert, update or delete data in vehicle_daily_track table. It will update any action into history_vehicle_daily_track table
Before insert	Registration	The trigger will fire before user inserts information into registration table. It will create unique and not null id starting with REG0
	Staff	The trigger will fire before user inserts information into staff table. It will create unique and not null id starting with S00
	Summons	The trigger will fire before user inserts information into summons table. It will create unique and not null id starting with SM0
	Summons_type	The trigger will fire before user inserts information into summons type table. It will create unique and not null id starting with SUMTY0
	Vehicle_daily_tra ck	The trigger will fire before user inserts information into vehicle_daily_track table. It will create unique and not null id starting with VDT0
	Daily_track	The trigger will fire before user inserts information into daily_track table. It will create unique and not null id starting with DTR0

5.3.5 Data Loading Process

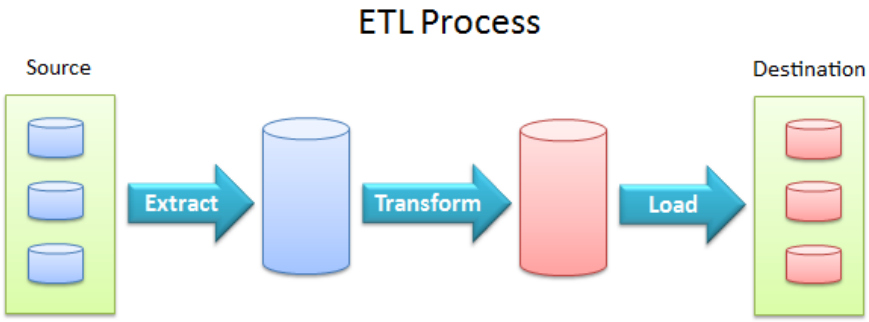


Figure 5.2 Diagram of Data Loading Process

CPNR database system is a system for obtaining and processing data in order to establish useful information to the user. Data is taken from a camera that captures images of license plates and converts it to text. After the process is carried out, this text will be transferred to the database in real time so the security guard will know whether the car has been registered or not.

Other than that, the data load from user input. The user will enter the required data to be processed by the system. Some of the data will be saved into tables in the database where a process of data transformation will carried out. Here, the data will transform into standardizing data entry by applying a set of rules or functions, making the data uniform.

5.4 Conclusion

As a conclusion, this part examined about the software development environment setup including installation and database creation and also database implementation that describes on DML, DDL, stored procedure, trigger and data loading process. It

incorporates the stage, software to be utilized to develop the application and other software used to help the running application.

The following chapter will discuss about the test plan, test strategy, test design, test result and analysis of the project. The testing phase will verify that the system meets the functional and non-functional requirements before it will be implemented to the end user and error can be avoided.

CHAPTER VI

TESTING

6.1 Introduction

The last activity to be carried out amid the advancement of this system is testing. The testing stage will be explained in detail in this chapter of Car Plate Number Recognition (CPNR) database system. The objective of testing is to point out the defects and errors that were made during development phase. Next, the defects and errors will be corrected for an effective performance of system application.

Next, testing is conducted to clarify the functional and non-functional requirement of the system. Testing is done by user that will use the system in a given time. It is required to get the feedback from the user to improve the system. Testing the

functional requirement is test on the ability the system to produce right output while testing the non-functional requirement is described how the system should do.

Besides that, the testing stage also will explain about test plan which comprises of test organization, test schedule and test environment. Meanwhile, test strategy covers the classes of tests. Test description and test data are included for testing. For more clarification will be explained in the sub-point of this part.

6.2 Test Plan

Test plan is the project plan for the testing work to be done. It is also helps to validate the acceptability of a software product and necessary to have written test plan in regulated environment. It includes test organization, test environment and test schedule. In general, test organization involved everyone including organization, person or company that tests the system to agreed requirements. Test environment comprise the area where software and hardware are setup to perform the testing while test schedule defines how many cycles and duration of test.

6.2.1 Test Organization

Test organization is a group of person who is responsible for testing the system through testing process. Testing should be performed by different individuals because the generated output also affects the quality system. The test for Car Plate Number Recognition (CPNR) database system will be carried out by system developer, project supervisor and student or staff.

System developer is a person who is responsible for developing Car Plate Number Recognition (CPNR) database system. Project supervisor is responsible for assisting and supervising the work of system developer while the system is being developed. Student or staff is act as a client to test this system. Table 6.1 shows the list of the tester of this system and their tasks.

Table 6.1 Test Organization Chart

Tester ID	Title/Position	Responsibilities
Tester 1	System developer	A person who is responsible for developing Car Plate Number Recognition (CPNR) database system throughout the customer requirements. As a tester, system developer test equipment and programming framework, using innovation and existing systems in addressing specific issues.

Tester 2	Project supervisor	Act as end client for admin and staff of the system. They will response based on system requirement and all the results will be recorded for enhancing the framework.
Tester 3	Student or Staff	Act as end client that register the vehicle for the sticker throughout the system. Every of the reaction taken will be recorded and take an action in order to improve the system.

6.2.2 Test Environment

A test of an environment is to test Car Plate Number Recognition (CPNR) database system that is going be performed by the testing team. It is a setup of software and hardware. This test will be built on both server and client. It also includes the location for the application software, the tools required to support testing and supporting data to conduct tests for

6.2.2.1 Environment Setup

In order to make a framework can run successfully for Car Plate Number Recognition (CPNR) database system, it need to be supervised and arranging the stage. Table 6.2 shows the application workspace specification.

Table 6.2 Application Workspace Specification

Environment Specification	Description
Operating system	Window 7
Processor	Intel Core i3
Random access memory (RAM)	4 GB
Database	PostgreSQL
Server	Wamp Server
Server scripting	PHP

6.2.2.2 Software Application

The software application includes every of the application that provides in the Car Plate Number Recognition (CPNR) database system. Below is the list of software application for Car Plate Number Recognition (CPNR) database system.

- Testing to authenticate user
- Testing on user registration
- Testing on update user profile
- Testing on vehicle registration
- Testing on sticker registration
- Testing on insert and view summons
- Testing on upload photo
- Testing on import and export file

6.2.2.3 System Software

System software is a program and tools that useful to run a computer's hardware and application programs. Below is the list of system software for Car Plate Number Recognition (CPNR) database system.

- Windows 7
- PostgreSQL 9.3
- WAMPP Server
- Notepad++
- Google Chrome (Web browser)
- Adobe Illustrator CS6

6.2.2.4 System Hardware

System hardware is the equipment used during the process of implementation of this system. Below is shown the list of system hardware for Car Plate Number Recognition (CPNR) database system.

- Laptop
- Mouse
- Printer
- Projector
- External hardisk
- Pendrive

6.2.3 Test Schedule

The test schedule defines when and by whom the test will be reviewed, tracked and approved. It is also collect all the information in the length of time during testing. A test schedule can be accurate with experience from previous testing effort along with a detailed understanding of the current testing goals. Table 6.3 shows the test schedule for developing of Car Plate Number Recognition (CPNR) database system.

Table 6.3 Test Schedule for CPNR Database System

Module	Test Activity	Duration (Days)	Cycle (Times)
Register	Test unit integration, testing and user acceptance	1	5

Login	Test unit integration, testing and user acceptance	1	10
User profile	Test unit integration, testing and user acceptance	1	6
Vehicle registration	Test unit integration, testing and user acceptance	5	10
Sticker registration	Test unit integration, testing and user acceptance	5	10
Summons management	Test unit integration, testing and user acceptance	4	7
Car daily track management	Test unit integration, testing and user acceptance	3	6
Login reporting	Test unit integration, testing and user acceptance	6	15
Summons reporting	Test unit integration, testing and user acceptance	6	12
Valid period license reporting	Test unit integration, testing and user acceptance	6	12
Error handling	Test unit integration, testing and user acceptance	3	15

6.3 Test Strategy

Test strategy is a guideline for a long-term plan of action that explains test design. It will describe how the test has been carried out, the technique to be used and which modules to test. All the partners including developer, designer and customer will together accomplish to achieve the objective through test strategy.

Black-box testing is in which test will carry out without reference to the internal structure and basically test on functional requirement. This testing can be applied at every level of software testing to identify what the normal output will come after the input of data.

While white-box testing is an opposite of black box testing which tests internal structure of an application. The tester determines the output by test through the program code. Today, it can be applied at the unit, integration and system level. The good at this testing is that it empowers us to see what is occurring inside the application.

The strategy to be selected is black-box testing for the Car Plate Number Recognition (CPNR) database system. Table 6.4 shows the difference between white box testing and black box testing.

Table 6.4 Black-box vs White-box

Black-box testing	White-box testing
Testing without reference to the internal structure	Testing with reference to the internal structure
Also known as closed box testing	Also known as clear box testing
Performed by end user, tester and developer	Performed by tester and developer

6.3.1 Classes of Tests

There are three classes of tests which are security test, error handling test and user acceptance test. There are two test classes that be implemented in the Car Plate Number Recognition (CPNR) database system; security test and error handling test.

i. Security test

The security test is a process to verify that the information system protects the data. It is intended to reveal any flaws that may lead to the security violation. The security test is implemented to check the level of user before they can log into the system, forgot password and encrypted the password in the first user register the system.

ii. Error handling test

Error handling test was utilized to ensure the system to get right data from the customer. The system will give an error message if any incorrect data entered by the customer. This is essential to make sure data entered is correct and filled by them. Error handling test is implemented for all form in the system to check user input before keep it into the database. An error or popup will be given to notify user.

iii. User acceptance test

User acceptance test is tested by the actual user of the system to make sure the system can handle its required tasks in real world scenario. The graphical user interface (GUI) must be comprehension and clear among of variety of IT knowledge level.

6.4 Test Implementation

There are two types of test implementation which are test description and test data. Test description is an activity that needs to be done to identify the best process information. At that point, it will describe the test case with an expected result and test data about user acceptance.

6.4.1 Test Description

Test description explains the test case identification; test cases and expected result for each module are designed and documented. A test case is a set of conditions under which a tester in order to verify compliance with a specific requirement. Certain information is needed as an input to test the system.

6.4.1.1 User Authentication Management

Table 6.5 Test Description for User Authentication Management

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC1_UA_1	Invalid IC number or password	Unit testing / integration	Login failed. Wrong IC number or password popup will appear.	Black box/White box	Security test/ Error handling test

TC1_UA_2	Valid IC number and password	Unit testing / integration	The system can be accessed.	Black box/White box	Security test/ Error handling test
TC1_UA_3	IC number or password is blank	Unit testing / integration	Login failed. Wrong IC number or password popup will appear.	Black box/White box	Security test/ Error handling test

6.4.1.2 User Registration Management

Table 6.6 Test Description for User Registration Management

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC2_UR_1	No input for all fields	Unit testing / integration	Notification popup will be appeared starting from the first blank field.	Black box/White box	Error handling test
TC2_UR_2	Valid input for all fields	Unit testing / integration	The user successfully registers the information.	Black box/White box	Error handling test

6.4.1.3 User Profile Management

Table 6.7 Test Description for User Profile Management

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC3_UP_1	Click on *[click here to update profile]* link	Unit testing / integration	A form for update user information will appear	Black box/White box	Error handling test
TC3_UP_2	New input for existing data is inserted	Unit testing / integration	Information is successfully updated	Black box/White box	Error handling test

6.4.1.4 Vehicle Registration Management

Table 6.8 Test Description for Vehicle Registration Management

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC4_VR_1	Click on *register vehicle* link	Unit testing / integration	A form for register the vehicle will appear.	Black box/White box	Error handling test
TC4_VR_2	No input for all fields	Unit testing / integration	Notification popup will be appeared starting from the first blank	Black box/White box	Error handling test

			field.		
TC4_VR_3	Valid input for all fields	Unit testing / integration	The user successfully registers the vehicle.	Black box/White box	Error handling test

6.4.1.5 Sticker Registration Management

Table 6.9 Test Description for Sticker Registration Management

Test Case ID	**Description**	**Testing Type**	**Expected Result**	**Test Strategy**	**Test Class**
TC5_SR_1	Click on *register sticker* link	Unit testing / integration	A form for register the sticker will appear.	Black box/White box	Error handling test
TC5_SR_2	No input for all fields	Unit testing / integration	Notification popup will be appeared starting from the first blank field.	Black box/White box	Error handling test
TC5_SR_3	Valid input for all fields	Unit testing / integration	The user successfully registers the sticker.	Black box/White box	Error handling test

6.4.1.6 Summons Management

Table 6.10 Test Description for Summons Management

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC6_SM_1	Click on *add summons* link	Unit testing / integration	A form for summons will appear.	Black box/White box	Error handling test
TC6_SM_2	No input for all fields	Unit testing / integration	Notification popup will be appeared starting from the first blank field.	Black box/White box	Error handling test
TC6_SM_3	Valid input for all fields	Unit testing / integration	The summons information successfully recorded.	Black box/White box	Error handling test

6.4.1.7 Car Daily Track Management

Table 6.11 Test Description for Car Daily Track Management

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC7_CD_1	Click on *vehicle in* link	Unit testing / integration	A form for search and record vehicle	Black box/White box	Error handling test

			in will appear.		
TC7_CD_2	Click on *vehicle out* link	Unit testing / integration	A form for search and record vehicle out will appear.	Black box/White box	Error handling test
TC7_CD_3	No input for all fields	Unit testing / integration	*Please insert at least 3 characters* notification popup will appear.	Black box/White box	Error handling test
TC7_CD_4	Valid input for all fields	Unit testing / integration	The information successfully records in the system.	Black box/White box	Error handling test

6.4.1.8 Login Reporting

Table 6.12 Test Description for Login Reporting

Test Case ID	**Description**	**Testing Type**	**Expected Result**	**Test Strategy**	**Test Class**
TC8_LR_1	Click on *login info* link	Unit testing / integration	Login information page will appear.	Black box/White box	Error handling test
TC8_LR_2	Click delete icon	Unit testing / integration	The information selected is	Black box/White box	Error handling test

			deleted.		

6.4.1.9 Summons Reporting

Table 6.13 Test Description for Summons Reporting

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC9_SR_1	Click on *summons per date* link	Unit testing / integration	Summons reporting page will appear.	Black box/White box	Error handling test
TC9_SR_2	Valid input for all fields	Unit testing / integration	A table of summons information will be shown.	Black box/White box	Error handling test

6.4.1.10 Valid Period License Reporting

Table 6.14 Test Description for Valid Period License Reporting

Test Case ID	Description	Testing Type	Expected Result	Test Strategy	Test Class
TC10_VPR_1	Click on *license information* link	Unit testing / integration	Valid period license reporting page will appear.	Black box/White box	Error handling test
TC10_VPR_2	Valid input for all fields	Unit testing / integration	A list of a valid period license will	Black box/White box	Error handling test

			appear based on range provided.		
TC10_VPR_3	Click button search	Unit testing / integration	A list of a valid period license will appear based on range provided.	Black box/White box	Error handling test

6.4.2 Test Data

Test data is used to affirm the expected result, for example, confirming system behavior towards invalid input data.

6.4.2.1 User Authentication Management

Table 6.15 Test Data for User Authentication Management

Column Name	TD1_UA_1	TD1_UA_2	TD1_UA_3
Test Case ID	TC1_UA_1	TC1_UA_2	TC1_UA_3
	No data is inserted by the user **IC number:** **Password:**	Correct data is inserted by the user **IC number:** 930502085210 **Password:**1234	Wrong data is inserted by the user **IC number:** 901209071223 **Password:**12891

Result Test Data	Login failed. Wrong IC number or password popup will appear.	The system can be accessed.	Login failed. Wrong IC number or password popup will appear.

6.4.2.2 User Registration Management

Table 6.16 Test Data for User Registration Management

Column Name	TD2_UR_1	TD2_UR_2
Test Case ID	TC2_UR_1	TC2_UR_2
	No data is inserted by the user **IC or passport number:** **Name:** **Address:** **Matric or staff number:** **Email:** **Level:** **Position:** **Contact number:** **Office number:** **Password:** **Confirm password:**	Correct data is inserted by the user **IC or passport number:** 950110076514 **Name:** Dhabitah binti Zed **Address:** Kampar, Perak **Matric or staff number:** D031610089 **Email:** dhabitzed@yahoo.com **Level:** student **Position:** student **Contact number:** 019-5516341 **Office number:** - **Password:** 8978912 **Confirm password:** 8978912
Result Test Data	Notification popup will be appeared starting from the	The user successfully registers the information.

	first blank field.	

6.4.2.3 User Profile Management

Table 6.17 Test Data for User Profile Management

Column Name	TD3_UP_1	TD3_UP_2
Test Case ID	TC3_UP_1	TC3_UP_2
	Users click on *click here to update profile* link	Correct data is inserted by the user **Name:** Nur Maisarah Alia binti Kamaruddin **Staff ID:** S001 **Address:** No 67 Jalan Pandan Indah 8 Taman Krubung 67120 Melaka **Email:** maisalia@gmail.com **Contact number:** 019-67125441
Result Test Data	A form for update user information will appear	Information is successfully updated

6.4.2.4 Vehicle Registration Management

Table 6.18 Test Data for Vehicle Registration Management

Column Name	TD4_VR_1	TD4_VR_2	TD4_VR_3
Test Case ID	TC4_VR_1	TC4_VR_2	TC4_VR_3
	Users click on *register vehicle* link	Wrong data is inserted by the user	Correct data is inserted by the user

		Plate number: **Type of vehicle:** **Vehicle CC:** **Vehicle color:** **Vehicle year:**	**Plate number:** AGM 7812 **Type of vehicle:** Car **Vehicle CC:** Kancil 850 **Vehicle color:** Silver **Vehicle year:** 2010
Result Test Data	A form for register the vehicle will appear.	Notification popup will be appeared starting from the first blank field.	The user successfully registers the vehicle.

6.4.2.5 Sticker Registration Management

Table 6.19 Test Data for Sticker Registration Management

Column Name	TD5_SR_1	TD5_SR_2	TD5_SR_3
Test Case ID	TC5_SR_1	TC5_SR_2	TC5_SR_3
	User click on *register vehicle* link	No data is inserted by the user **Plate number:** **Responsibility center:** **License number:** **Valid period**	Correct data is inserted by the user **Plate number:** AGM 7812 **Responsibility center:** FTMK **License number:**

		license: **Valid period roadtax:** **Insurance company:**	891291 **Valid period license:** 2017-09-01 **Valid period roadtax:** 2016-12-01 **Insurance company:** ALLIANCE
Result Test Data	A form for register the sticker will appear.	Notification popup will be appeared starting from the first blank field.	The user successfully registers the sticker.

6.4.2.6 Summons Management

Table 6.20 Test Data for Summons Management

Column Name	TD6_SM_1	TD6_SM_2	TD6_SM_3
Test Case ID	TC6_SM_1	TC6_SM_2	TC6_SM_3
	User click on *add summons* link	No data is inserted by the user **Date:** **Plate number:** **Place of offense:** **Type of offense:**	Correct data is inserted by the user **Date:** 24 Jul 2016 **Plate number:** AGM 7812 **Place of offense:** FKE **Type of offense:** Blocking the path

Result Test Data	A form for summons will appear.	Notification popup will be appeared starting from the first blank field.	The summons information successfully recorded.

6.4.2.7 Car Daily Track Management

Table 6.21 Test Data for Car Daily Track Management

Column Name	TD7_CD_1	TD7_CD_2	TD7_CD_3	TD7_CD_4
Test Case ID	TC7_CD_1	TC7_CD_2	TC7_CD_3	TC7_CD_4
	User click on *vehicle in* link	User click on *vehicle out* link	No data is inserted by the user **Plate number:**	Correct data is inserted by the user **Plate number:** AGM 7812
Result Test Data	A form for search and record vehicle in will appear.	A form for search and record vehicle out will appear.	*Please insert at least 3 characters* notification popup will appear.	The information successfully records in the system.

6.4.2.8 Login Reporting

Table 6.22 Test Data for Login Reporting

Column Name	TD8_LR_1	TD8_LR_2	TD8_LR_3
Test Case ID	TC8_LR_1	TC8_LR_2	TC8_LR_3
	User click on *login info* link	User click on delete icon from the selected information	
Result Test Data	Login information page will appear.	The information selected is deleted.	

6.4.2.9 Summons Reporting

Table 6.23 Test Data for Summons Reporting

Column Name	TD9_SR_1	TD9_SR_2
Test Case ID	TC9_SR_1	TC9_SR_2
	User click on *summons per date* link	Correct data is inserted by the user **Year:** 2016 **Month:** 5
Result Test Data	Summons reporting page will appear.	A table of summons information will be shown.

6.4.2.10 Valid Period License Reporting

Table 6.24 Test Data for Valid Period License Reporting

Column Name	TD10_VPR_1	TD10_ VPR _2	TD10_ VPR _3
Test Case ID	TC10_ VPR _1	TC10_ VPR _2	TC10_ VPR _3
	User click on *license information* link	Correct data is inserted by the user **Start date:** 11-04-2016 **Last date:** 14-07-2016	Click on search button
Result Test Data	Valid period license reporting page will appear.	A list of a valid period license will appear based on range provided.	A list of a valid period license will appear based on range provided.

6.5 Test Result and Analysis

Test result and analysis was included to record all the research on the Car Plate Number Recognition (CPNR) database system whether the test will generate successful or fail condition. All the prediction of this system will determine the system can operate more effectively.

6.5.1 User Authentication Management

Table 6.25 Test Result for User Authentication Management

Module / Component		**Result**		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC1_UA_1	TD1_UA_1	Invalid IC number or password	√ (14/8/16)	
TC1_UA_2	TD1_UA_2	Valid IC number or password	√ (14/8/16)	√ (29/7/16)
TC1_UA_3	TD1_UA_3	IC number or password is blank	√ (14/8/16)	

6.5.2 User Registration Management

Table 6.26 Test Result for User Registration Management

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC2_UR_1	TD2_UR_1	No input for all fields	√ (14/8/16)	
TC2_UR_2	TD2_UR_2	Valid input for all fields	√ (14/8/16)	

6.5.3 User Profile Management

Table 6.27 Test Result for User Profile Management

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC3_UP_1	TD3_UP_1	Click on *[click here to update profile]* link	√ (14/8/16)	
TC3_UP_2	TD3_UP_2	New input for existing data is inserted	√ (14/8/16)	

6.5.4 Vehicle Registration Management

Table 6.28 Test Result for Vehicle Registration Management

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC4_VR_1	TD4_VR_1	Click on *register vehicle* link	√ (14/8/16)	
TC4_VR_2	TD4_VR_2	No input for all fields	√ (14/8/16)	
TC4_VR_3	TD4_VR_3	Valid input for all fields	√ (14/8/16)	

6.5.5 Sticker Registration Management

Table 6.29 Test Result for Sticker Registration Management

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC5_SR_1	TD5_SR_1	Click on *register sticker* link	√ (14/8/16)	
TC5_SR_2	TD5_SR_2	No input for all fields	√ (14/8/16)	
TC5_SR_3	TD5_SR_3	Valid input for all fields	√ (14/8/16)	

6.5.6 Summons Management

Table 6.30 Test Result for Summons Management

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC6_SM_1	TD6_SM_1	Click on *add summons* link	√ (14/8/16)	
TC6_SM_2	TD6_SM_2	No input for all fields	√ (14/8/16)	
TC6_SM_3	TD6_SM_3	Valid input for all fields	√ (14/8/16)	

6.5.7 Car Daily Track Management

Table 6.31 Test Result for Car Daily Track Management

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC7_CD_1	TD7_CD_1	Click on *vehicle in* link	√ (14/8/16)	
TC7_CD_2	TD7_CD_2	Click on *vehicle out* link	√ (14/8/16)	
TC7_CD_3	TD7_CD_3	No input for all fields	√ (14/8/16)	
TC7_CD_4	TD7_CD_4	Valid input for all fields	√ (14/8/16)	

6.5.8 Login Reporting

Table 6.32 Test Result for Login Reporting

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC8_LR_1	TD8_LR_1	Click on *login info* link	√ (14/8/16)	
TC8_LR_2	TD8_LR_2	Click delete icon	√ (14/8/16)	

6.5.9 Summons Reporting

Table 6.33 Test Result for Summons Reporting

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC9_SR_1	TD9_SR_1	Click on *summons per date* link	√ (14/8/16)	
TC9_SR_2	TD9_SR_2	Valid input for all fields	√ (14/8/16)	

6.5.10 Valid Period License Reporting

Table 6.34 Test Result for Valid Period License Reporting

Module / Component		Result		
Test Case ID	**Test Data ID**	**Description**	**Success/date**	**Fail/date**
TC10_VPR_1	TD10_VPR_1	Click on *license information* link	√ (14/8/16)	
TC10_VPR_2	TD10_VPR_2	Correct input for all fields	√ (14/8/16)	
TC10_VPR_3	TD10_VPR_3	Click on search button	√ (14/8/16)	√ (29/7/16)

6.6 Test analysis

Based on the test result above, there are some errors in the early stages of system development. Among them are all users are able to access the system even uses the ID and password that is not valid. This problem occurs because there is no control of the user in the coding and data entry does not conform to the data stored in the database,

Then there is an error in which data remains on display to the user when the search button is pressed. Once analyzed, this error occurs as a result of which no validation implement in the coding that will notify the user when entering the wrong data.

6.6 Conclusion

As the conclusion, testing is the essential part of every system development. The system developer needs to test the system with the client to verify the requirements in order to guarantee the system work effectively and proficiently. Throughout the testing result, the developer can analyze the system performance and fixed them if any error occurred. Other than that, to achieve customer satisfaction, the developer needs to make sure all the requirements are works and the system is useful for them.

It is very important to have user-friendly system between the customer and the system. The better the communication between the system and database will facilitate all the requests from customer to the system as well as to retrieve and manipulate data which based on authorization setting. Each stage has been explained in detail in this chapter.

CHAPTER VII

CONCLUSION

7.1 Introduction

The last chapter will conclude of this project. Car Plate Number Recognition (CPNR) database system was successfully completed within the time given. This chapter will divide into several sub-topics. Firstly, it will explain on the weaknesses and strength of this system. Based on the weaknesses, the developer will take an action to enhance the system to ensure it is more beneficial for everyone. Secondly, recommendation for improvement will be described in the following sub-topic and lastly will state the project contribution for this system.

7.2 Observation on Weakness and Strength

Each system will have a value of its own advantages and disadvantages. The advantage is strength to the system while the disadvantage is a system weakness.

7.2.1 Strength

Below are some strength that have been identified in this system. There are:

i. As a data repository for various information especially vehicle and owner information. This system will help security guard to recognize vehicle owner that move in and out from UTeM.

ii. Facilitate the searching for user data through license plate number. A variety of data about the user are recorded in the system and the data are classified by its type. As the data is stored in a centralized database, it allows for data storage and retrieval.

iii. This system will simplify the process of the sticker registration. Online registration is available all the time for those who want to bring the vehicle into UTeM especially to students and UTeM's staff. It is also very useful to anyone who wants to register at their appropriate time and place.

iv. Provide useful information to staff, especially with the diversity report. This report helps the management to analyze peak hour and also find out which vehicle often in and out from UTeM.

v. This system is classified into two types of user. One is the management and another is the owner of the car. They have been categorized into two user levels and have different access to the system accordingly to its user level.

7.2.2 Weaknesses

Below are some weaknesses that have been identified in this system. There are:

i. The system does not have a backup and recovery in the event of any problems. Data is not stored other in the database as well and it makes data easily lost and so causes other problems such as the detection of users who have registered in UTeM and identify incoming vehicles whether they are registered or not.

ii. The system is not connected with incoming information in a real time. Thus, it makes difficult for the system to generate advances business intelligent to the user of the system.

iii. The system cannot communicate with the device that captures plate number and process it until get useful data in order to identify owner information and validate the registration of the car. When there is no communication between them, data entry must be typed own.

7.3 Proposition for Improvement

There are few suggestions for improvement of this system so that the system run more smoothly and will give beneficial to users. Some of the proposed improvements to be implemented in the future are as below

i. To solve the problem of data loss and data asynchronous, the system must implement a backup and recovery. It is important to ensure that data is always available at any time. In addition, if data is deleted, modified or stored, it should be recorded in the database.

ii. In a case of users forgot their passwords, administrators must reset the password and enforce them to change after they managed to get into the system. This improves system security from other users intrudes into the system.

iii. Implement system prediction for future. Using existing information, it allows the system to make the prediction and thus produce very useful data. It also helps the users to develop a strategy for the future based on that prediction.

iv. Send notification via text messaging and through email in case of a license has expired. This improvement will alert the user at that time and help them to renew their licenses before incurring to any legal action by the authorities.

v. Link together with the image capture device and can be also implemented in a mobile phone. The security guard can control the system by using the mobile phone and it can be also carry everywhere.

7.4 Project Contribution

There are many project contributions that have been identified by this system. This system mainly contributes to the university as a system to store data of a vehicle registration in UTeM. One of them is, the system helps security guard to identify the authorized vehicles. The system uses the license plate number as input to find user information to replace the manual method in which a security guard must observe the sticker affixed to each vehicle entering.

The second contribution, the system provides registration sticker to those who want to register vehicles without having to go to the security office. After registering online, users need to go to the security office to claim sticker. It will also speed up the process as well as allow users to perform the registration without having to wait their turn.

Finally, this system will generate several of reports. Among them are user entrance by a person, summons per owner, sticker registration and valid license information report. These reports will help management to examine users trends which recorded the number they in and out from UTeM in a day, the number of summons has been charged to the user, place where accidents often occur and many others.

7.5 Conclusion

Overall, this system has achieved all the objectives that have been stated. The objectives consist of to develop a database for storing data in organized and be managed systematically, to allow find out other users availability by searching using plate number and to generate reports that needed by the officer.

After this system has completed, as a conclusion, this system has successfully met all the requirements and objectives that stated in the earlier development of the system. However, there are still some weaknesses in some parts of the system that need to be improved in the future. The improvements are essential to make the system more meaningful to the users. This is based on the factor of time constraints during development of the system.

Nevertheless, as long as this system meets the objectives stated, it indicates the system has reached its goal on why it is produced and be useful to users.

References

Robert Robbins. (1994,1995). *Database Fundamental.* Retrieved from http://www.esp.org/db-fund.pdf

TMap. (n.d.). Retrieved from Test Organization: http://www.tmap.net/building-blocks/test-organization

Bibliography

Bruce Momjian . (2005, 4 21). Retrieved 2016, from Open Source Software: https://momjian.us/main/writings/pgsql/aw_pgsql_book/node13.html

FYICenter.com. (2008). Retrieved from What is the purpose of test plan: http://sqa.fyicenter.com/FAQ/Testing-Techniques/What_is_the_purpose_of_a_test_plan_.html

Ilia Alshanetsky. (n.d.). Retrieved 2016, from Introduction to PostgeSQL: https://ilia.ws/files/confoo_pgsql.pdf

ISTQBExamCertification.com. (n.d.). Retrieved from Purpose of Test Plan: http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/

Jaideep Khanduja. (2008, 9 12). Retrieved from Quality Assurance and Project Management: http://itknowledgeexchange.techtarget.com/quality-assurance/what-is-a-testing-environment-for-software-testing/