NISSA’S ONLINE ORDERING COOKIES AND MARKETING SYSTEM

ZULIYANA BINTI ZULBAHARI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS THESIS

JUDUL : NISSA'S ONLINE ORDERING COOKIES AND MARKETING SYSTEM

SESI PENGAJIAN : SEMESTER 2015/2016

Saya ZULIYANA BINTI ZULBAHARI
(HURUF BESAR)

mengaku membenarkan tesis (PSM/ ~~Sarjana/~~ ~~Doktor Falsafah~~) ini disimpan di perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat- syarat kegunaan seperti berikut :

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

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

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

___√___ TIDAK TERHAD

Zuliyana	
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat tetap : No 50, Jalan Dato Abdul Hamid 65, Taman Sentosa, 41200 Klang, Selangor.	NURUL AKMAR EMRAN Nama Penyelia
Tarikh: 24/8/2016	Tarikh: 24/8/2016

NISSA’S ONLINE ORDERING COOKIES AND MARKETING SYSTEM

ZULIYANA BINTI ZULBAHARI

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

NISSA'S ONLINE ORDERING COOKIES AND MARKETING SYSTEM

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

STUDENT : Zuliyana Date : 24/8/2016

(ZULIYANA BINTI ZULBAHARI)

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

SUPERVISOR : Date : 24/8/2016

(DR NURUL AKMAR BINTI EMRAN)

DEDICATION

I would like to express my special dedication to
my utmost understanding and beloved parents who inspires
me to always be outstanding and never give up throughout my
education journey. They are my backbone.

ACKNOWLEDGMENTS

I would like to thank Universiti Teknikal Malaysia Melaka (UTeM), for giving us; students from Faculty of Information and Communication Technology (FICTS) the chance to complete our final year project within two semesters or 28 weeks. Without the duration given, our systems might not be completed in time.

Next appreciation goes to my supervisor; Dr Nurul Akmar binti Emran for all the supports, ideas, comments and feedbacks in order to see valuable results came from her PSM students. Without her help and guidance, I would not be able to be where I am standing now.

Not to forget, thank you to my parents; Zulbahari bin Yaacob and Zaini binti Ibrahim for all the words of inspiration given just to see their daughter excels in this subject. Last but not least, to all colleagues, thank you for the help in completing this project as a whole. Thank you, Allah S.W.T.

ABSTRACT

Nissa’s Online Ordering Cookies and Marketing System (NOOCMS) is a web-database system proposed to support customer order management for small online business ventured by work-from-home people. Most of business operators in this domain chose file-based systems (such as Excel Spreadsheets) to record and manage their business data for its easy to-set-up and learn characteristics. Unfortunately, as the size of customer order data grows, file based system is no longer sufficient to handle large number of queries and complex reporting functions in timely manner. In addition, data quality problems such as accuracy, data redundancy and incompleteness are the common problems faced in file-based ordering system . Lack of backup facility in the system has also the business from data loss and eventually profit loss. Even though ordering systems are common nowadays, ordering systems that supports online businesses to migrate and integrate their customer orders data from file-based systems to a systematic database system is limited. Hence, this system allows customers to view various types of cookies, placing orders, update or delete orders made, and upload payment receipts into the system. The uploaded picture will then be approved by NOOCMS’s staff before their orders are being processed. NOOCMS’s staff can import and export order menu from an Excel Spread Sheet into the database and vice versa. Last but not least, this system will generate monthly revenue reports to be viewed by system admin.

ABSTRAK

Nissa’s Online Ordering Cookies and Marketing System (NOOCMS) adalah sebuah sistem berlandaskan platform web-database dan dicadangkan untuk menyokong order-order dari pelanggan bagi bisnes kecil-kecilan yang dibuat dari rumah. Kebanyakkan operator-operator bisnes pada zaman sekarang lebih memilih untuk menyimpan rekod ke dalam sistem manual seperti menggunakan file Excel kerana bagi mereka, cara ini adalah lebih mudah untuk dilakukan. Akan tetapi, semakin hari saiz tempahan dari pelanggan akan semakin bertambah dan sistem manual ini tidak akan dapat menampung jumlah data yang terlampau banyak. Ini akan menjurus kepada punca berlakunya masalah kualiti data seperti ketepatan, pertindihan data serta ketidaksempurnaan data. Selain itu, kekangan dalam data sandaran juga boleh menyebabkan kehilangan data dan juga keuntungan. Walaupun sistem yang membenarkan pelanggan untuk membuat tempahan online ini telah banyak dipasarkan, akan tetapi system yang membenarkan bisnes untuk memindahkan serta mengintegrasikan data dari system manual (Excel Spread Sheet) terus ke dalam database adalah sangat terhad. Oleh tu, NOOCMS adalah sebuah system yang menerima tempahan dari pelanggan. Di samping itu, pelanggan juga boleh melihat pelbagai jenis biskut yang dipaparkan, kemaskini mahupun memadam makumat tempahan yang telah dibuat, dan memuat naik gambar resit tempahan ke dalam sistem ini. Resit tempahan yang telah dimuat naik ke dalam sistem akan diterima oleh staff system sebelum tempahan diproses. Staff NOOCMS juga boleh import serta export tempahan-tempahan pelanggan dari Excel Spread Sheet ke dalam database dan juga sebaliknya. Akhir sekali, system ini akan menghasilkan laporan pendapatan bulanan untuk dilihat oleh sistem admin.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NOOCMS	-	Nissa’s Online Ordering Cookies and Marketing System
RAD	-	Rapid Agile Development
SDLC	-	System Development Life Cycle
DBLC	-	Database Development Life Cycle
WhatsApp	-	What’s Application
PHP	-	Hypertext Pre-Processor

CHAPTER I

INTRODUCTION

1.1 Introduction

Nissa’s Online Ordering Cookies and Marketing System (NOOCMS) is a system that is going to be developed to receive order from customers. Customers can view various types of cookies, place their orders and upload payment receipts into the system. The uploaded picture will then be approved by NOOCMS’s staff before their orders are being processed.

There are some problems in the current situation of Nissa’s cookies business that motivated me to build this system. One of them is the difficulty to manage orders from customers because they are taking orders manually. Every orders made by customers were recorded and saved by using file-based system. This is one of the reasons why the business and its production slowed down. When we save important things on papers, books or laptop, there might be problems occurred when it went missing or damage. This is what we would want to prevent from happening.

Besides developing a system that interacts with customers and staff, I plan to apply some marketing strategies to gain and attract more customers. This marketing strategy will be applied by using social media accounts such as Facebook and Instagram. Studies have shown that every time a prospective client clicks on the business advertisement we advertised, their interaction will be shared with his or her friends (via Facebook and Instagram). Their association with the product advertised is then spread similarly to word of mouth. So if we net one targeted client with an advertisement, we will also netting several more. Therefore, the fan base has the potential to grow exponentially.

The methods that I use in this project are research and interview. These methods are used to gain more information about marketing and to get to know more on the current cookies business. Some of the questions that will be asked during interview:

1) How long have you been running this business and through which medium your business had been advertised so far?
2) How do customers place their orders?
3) How do you keep customer's information and orders?
4) How do you receive and approve customer's payment?

Endeavour (n.d.) once said that manual system will give pressure for people to correct all details of their work most of the time. The problem is that, humans are not perfect. With manual system, the service is dependent on individuals and this requires the management to train their staff continuously in order to keep them motivated and to ensure they are following correct procedures. It can be all too easy to accidentally switch details and end up with inconsistency of data entry or in hand written orders. This has the effect of not only causing problems with customer service but also making information unable to be used for reporting or in data searching. Reporting and checking that data is robust can be time consumption and expensive.

Besides that, it takes more and physical space and energy to keep track of paper documents, to find information and to keep details secure. When mistakes are made or changes and corrections are needed, usually a manual transaction must be redone rather than just updated. This is because, manual or partially automated systems information has to be written down or entered more than once. Systemisation can reduce the amount of data entry duplication.

In a conclusion, online ordering system will accommodate users more on the cost, time, energy, and prevent data redundancies compared to the manual system.

1.2 Problem Statements

Generally, there is no any online application have been implemented in this Nissa's Cookies Business (NCB). All orders and customer's information are kept in the file-based system which is not really safe. Once the file went missing, there would be a big problem because most of the time, there are no data backup and when catastrophe occurs, all files will be lost. On the other hand, problems will occur when NCB wants to keep track of the archives; extra time will be needed to search through the history of orders, sales or customers information.

Besides that, all orders are being made by customers manually where they will have to make calls, send messages or meet the owner of NCB by themselves. This will definitely waste customer's time and NCB's as well. Same goes when customers are making payments which they have to send the pictures of payment receipt through messages. This will burden NCB's owner to keep track all payment records. By developing a system that will allow customer to upload their payment receipt onto the database, this can really solve the problem.

Due to the manual data retrieving by admin when scanning monthly reports, data redundancy will definitely appear because each customer might make more than one order every month. With this, NOOCM system is going to be equipped to retrieve accurate monthly reports and ease every operation that the admin wish to make.

Therefore, NOOCMS will help to solve these problems by advertising NCB's product on an online website so that the information can be widely spread to the community and customer can just click on the pictures, fill up a form, upload the payment receipt, and the ordering is done. Customer will always prefer things that will ease them and save their precious time.

1.3 Objectives

1. To allow NOOCMS's customers to place their orders by using a web based environment.
2. To accommodate NOOCMS's admin to save and keep track all customer's information and orders into a proper system with a database.
3. To allow NOOCMS's admin to view accurate sales report without data redundancy by enforcing key constraints.

1.4 Scope

Table 1.1: List of each scope in this Nissa's Online Ordering and Marketing System

SYSTEM SCOPE	EXPLAINATION
Customer	1) Login 2) View Cookies 3) Make Order and Update Order 4) Upload Payment Receipt 5) Search cookies by entering some keyword
Staff	1) Login 2) View customers information 3) Update Customers information 4) View, add, update, or delete choices of cookies

	available in the system 5) View and approve orders made by customers 6) Approve orders paid by customers by checking the uploaded payment receipt image 7) Import customers order data from Excel Spread Sheet into database 8) Export customers order data from database into Excel Spread Sheet
Admin	1) Add, view, update, and delete staff's information and details 2) View and delete customer's information and details 3) View monthly profit report 4) View total menu sales report
System	1) Allow customers to upload pictures 2) Generate monthly profit report 3) Generate total menu sales report
Marketing	Marketing strategy will be implied by creating accounts in: 1) Facebook 2) Instagram

Table 1.2: The list of software and hardware that will be used in this project

SOFTWARE	PURPOSE
Microsoft Office Word 2010	To write documents
Microsoft Office Project 2010	To produce Work Breakdown Structure, Gantt Chart and Milestones
Microsoft Visio 2010	To create Entity Relationship Diagram (ERD), Flow Charts, and Data Flow Diagram
Adobe Photoshop	To make NCOOMS logo
Notepad	To backup all of the coding

HARDWARE	PURPOSE
Notebook/Laptop	To work on the project
Printer	To print documents

The programming language that will be used in this project is PHP while the software used to support this language is Adobe Dreamweaver. This software is used to implement coding in producing the system interfaces with its functional requirements.

Meanwhile, the Operating System that will be used is Windows 8.1 and the Database Management System (DBMS) is Oracle 11g.

1.5 Project Significance

This database project is to accommodate customers in placing orders through an online system. Customers do not need to waste their precious time driving to the bakery nor calling the bakery staff to make their orders. By using this system, customers can easily order their favourite cookies at home or anywhere they like and the orders will be delivered straight to their doorsteps (by posting the products through a third medium; example: PosLaju). NOOCMS's staff will make sure that all of the orders will reach to the customers in a short period of time. This system will definitely help NOOCMS's customers and staffs in saving their time, energy and cost.

1.6 Expected Output

Output 1: Multi-dimensional searching results screens which are generated based on join queries and sub-queries results.

Output 2: Sales monitoring screen for staff to keep track daily orders before they can approve it.

Output 3: Accurate monthly sales report which can be generated through the system.

1.7 Conclusion

As a conclusion, this Nissa's Online Ordering Cookies and Marketing System will be developed in order to provide services such as placing orders, uploading pictures, approving payments, accepting orders, importing and exporting orders, and producing monthly report. The importance of this project has been explained to convince targeting users that there are many benefits that they could gain from this system.

For the next chapter (Chapter 2), the project methodology and planning will be elaborated into details. The Database Life Cycle (DBLC), project schedule and milestones will also be included.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. Examples of system development methodology are Waterfall, Prototyping, Spiral, Incremental, Agile, and Rapid Application Development (RAD).

2.2 Project Methodology

This project is using Agile Development methodology as the System Development Life Cycle (SDLC) approach.

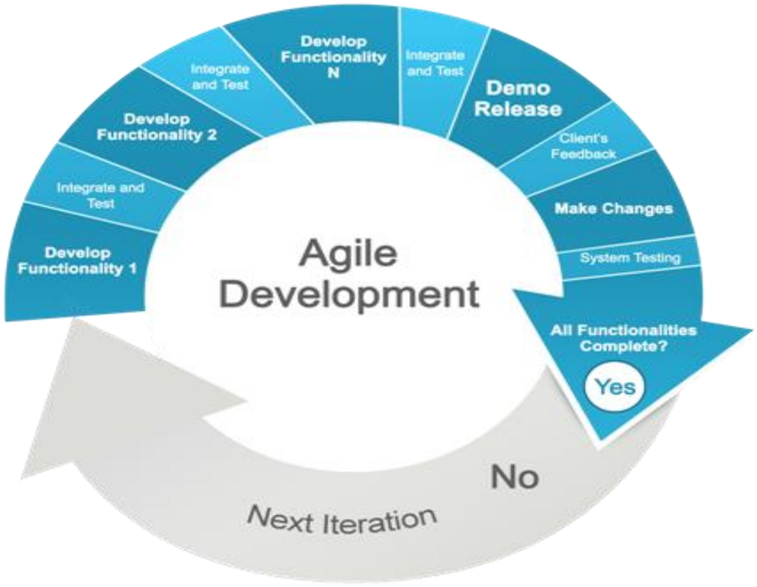


Figure 2.1: Agile Development diagram for SDLC approach (Sybite Technologies, n.d.)

There are going to be many testing conducted as this project is being develop to detect the defects of system earlier. This is to ensure that the system meets all the requirements and to avoid wasting of time and cost in a long term.

As for the database methodology on Database Development Life Cycle (DBLC), it is going to be developed as a top down approach and implemented separately with SDLC.

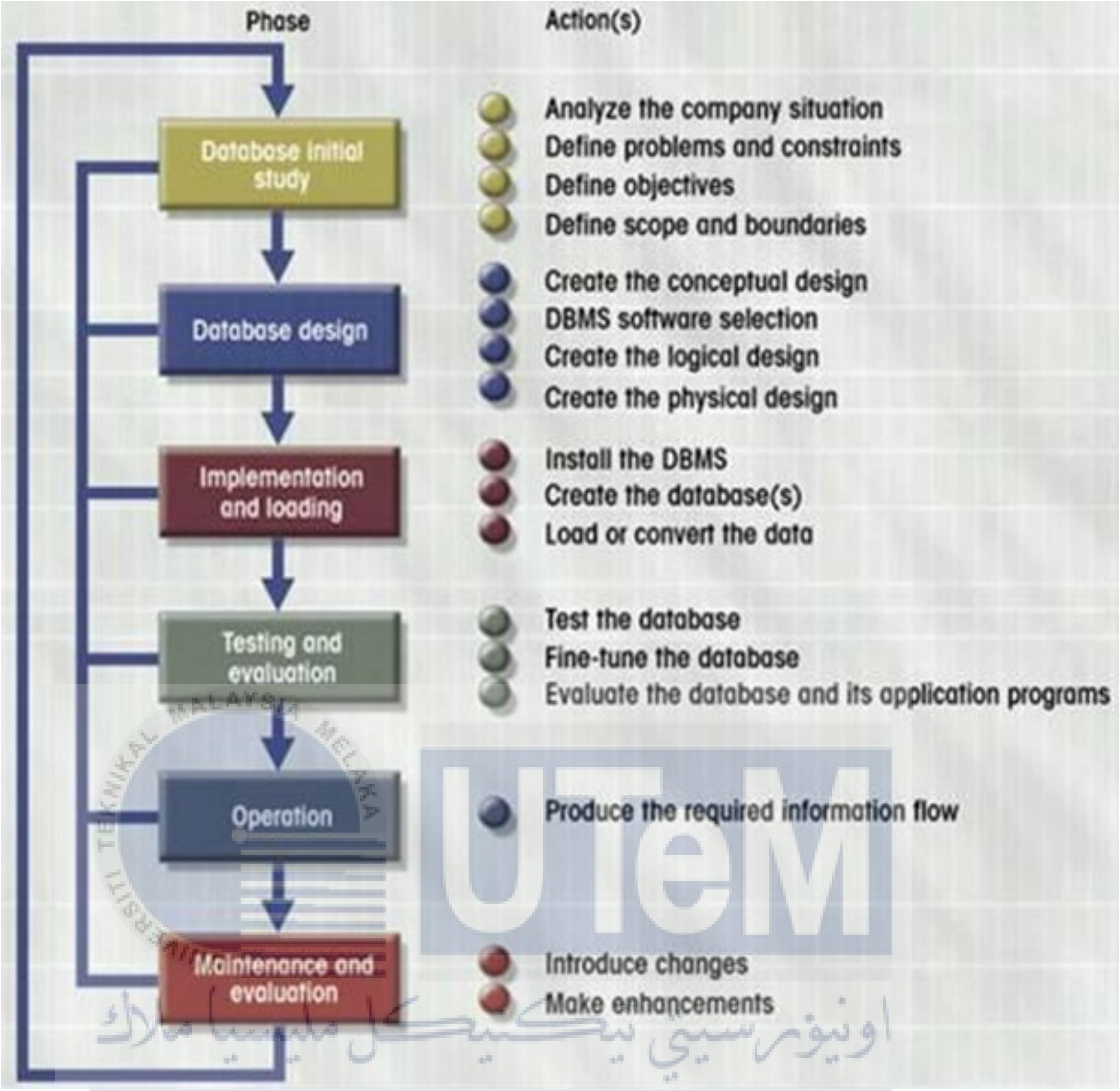

Phase
Action(s)
Database initial study
Analyze the company situation
Define problems and constraints
Define objectives
Define scope and boundaries
Database design
Create the conceptual design
DBMS software selection
Create the logical design
Create the physical design
Implementation and loading
Install the DBMS
Create the database(s)
Load or convert the data
Testing and evaluation
Test the database
Fine-tune the database
Evaluate the database and its application programs
Operation
Produce the required information flow
Maintenance and evaluation
Introduce changes
Make enhancements

Figure 2.2: Top down approach diagram for database development methodology by using DBLC (Peter Estonian, 2014)

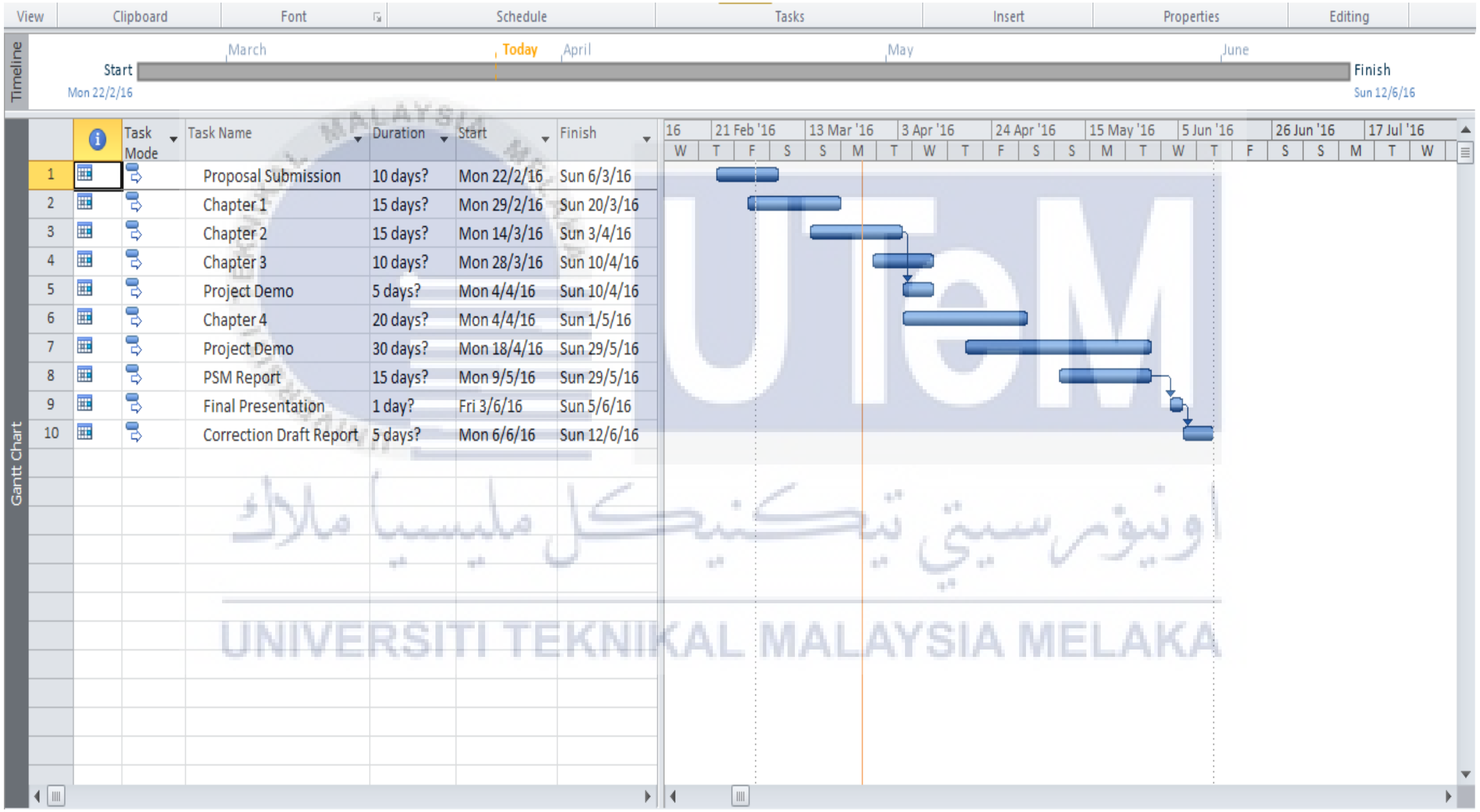
2.3 Project Schedule and Milestones

Table 2.1: NOOCMS's Schedule for PSM 1

Milestones	**Expected Documents**	**Dates (Weeks)**
Proposal correction and submission	Proposal	22/2/2016 - 6/3/2016 (Week 1 & 2)
Chapter 1	Objectives, Scope, Project Significance	29/2/2016 – 20/3/2016 (Week 2 – Week 4)
Chapter 2	Project Methodology, Project Schedule and Milestones	14/3/2016 – 3/4/2016 (Week 4 – Week 6)
Chapter 3	Functional Requirements, Non-Functional Requirements	28/3/2016 – 10/4/2016 (Week 6 – Week 7)
Project Demo	-	4/4/2016 – 10/4/2016 (Week 7)
Chapter 4	ERD, Data Dictionary, Normalization, Queries, Graphical User Interface (GUI)	4/4/2016 – 1/5/2016 (Week 7 – Week 10)
Project Demo	-	18/4/2016 – 29/5/2016 (Week 9 – Week 14)
PSM Report	Chapter 1 – Chapter 4	9/5/2016 – 29/5/2016 (Week 12 – Week 14)
Final Presentation	-	30/5/2016 – 5/6/2016 (Week 15)
Correction Draft Report	Chapter 1 – Chapter 4	6/6/2016 – 12/6/2016 (Week 16)

Table 2.2: NOOCMS's Schedule for PSM 2

Milestones	Expected Documents	Dates (Weeks)
PSM Report	Chapter 5	21/6/2016 (Week 1)
Project Demo	-	20/7/2016 (Week 5)
PSM Report	Chapter 6	25/7/2016 (Week 6)
Project Demo	-	3/8/2016 (Week 7)
PSM Report	Chapter 7	8/8/2016 (Week 7)
Final Presentation	Final Report (Chapter 1 – Chapter 7)	22/8/2016 (Week 9)
Correction Report	Chapter 5 – Chapter 7	23/8/2016 – 26/8/2016 (Week 10)



	Task Mode	Task Name	Duration	Start	Finish
1		Proposal Submission	10 days?	Mon 22/2/16	Sun 6/3/16
2		Chapter 1	15 days?	Mon 29/2/16	Sun 20/3/16
3		Chapter 2	15 days?	Mon 14/3/16	Sun 3/4/16
4		Chapter 3	10 days?	Mon 28/3/16	Sun 10/4/16
5		Project Demo	5 days?	Mon 4/4/16	Sun 10/4/16
6		Chapter 4	20 days?	Mon 4/4/16	Sun 1/5/16
7		Project Demo	30 days?	Mon 18/4/16	Sun 29/5/16
8		PSM Report	15 days?	Mon 9/5/16	Sun 29/5/16
9		Final Presentation	1 day?	Fri 3/6/16	Sun 5/6/16
10		Correction Draft Report	5 days?	Mon 6/6/16	Sun 12/6/16

Figure 2.3: NOOCMS's Gantt Chart

2.4 Conclusion

In this chapter, it is mainly discussed on the project methodology and planning that was used to define during the completion of this project. It includes the introduction of project methodology, database development methodology, and project schedule with milestones. Agile Development Methodology is used for System Development Life Cycle (SDLC) approach while a top down approach is implemented separately for the Database Development Life Cycle (DBLC).

For the next chapter (Chapter 3), the project analysis will be discussed into details. The problem analysis, proposed improvements/solutions, non-functional requirements, functional requirements, and other requirements will be included.

CHAPTER III

ANALYSIS

3.1 Introduction

Allison and Kendra (2001) stated that, analysis is the study we do in order to figure out what to do. It is a review process to compare between the current or existing system with the system which is to be built. The analysis phase is sometimes referred to a front-end analysis where you normally perform some type of analysis throughout the entire process. This 'front-end' process is where the identification of main problem is performed such as problem identifying, job analysing and selecting the tasks to be trained (U.S. Army Field Artillery School, 1984).

This analysis report is to explain the system to-be and compared it with the current system analysis. This analysis part will examine each and every aspects related to the existing system and the system that is going to be developed. However, for this project, the current system to be analysed will be the manual type of system where every task and activity will be done by human. Analysis will be carried out by describing the system using Flow Charts, Data Flow Diagram (DFD), and Context Diagram.

3.2 Problem Analysis

This part is to investigate and describe the current (as-is) system scenario or situation aided by flow chart diagrams.

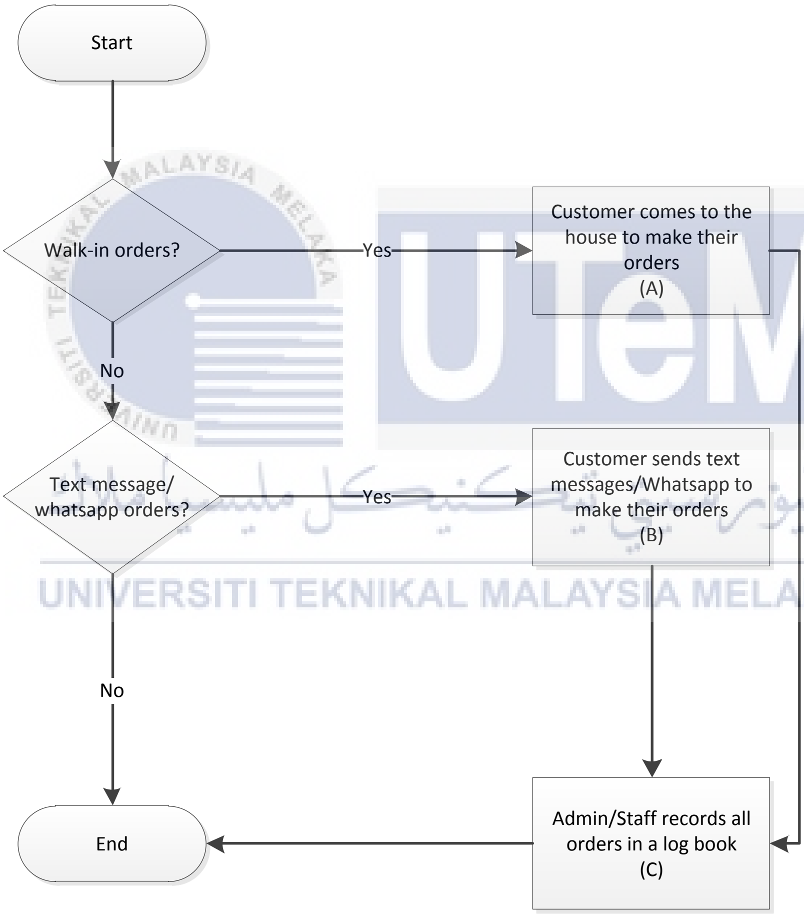

Start
Walk-in orders?
Yes
Customer comes to the house to make their orders (A)
No
Text message/ whatsapp orders?
Yes
Customer sends text messages/Whatsapp to make their orders (B)
No
End
Admin/Staff records all orders in a log book (C)

Figure 3.1: Flow chart of current (as-is) system which shows how customer and staff making and accepting orders manually

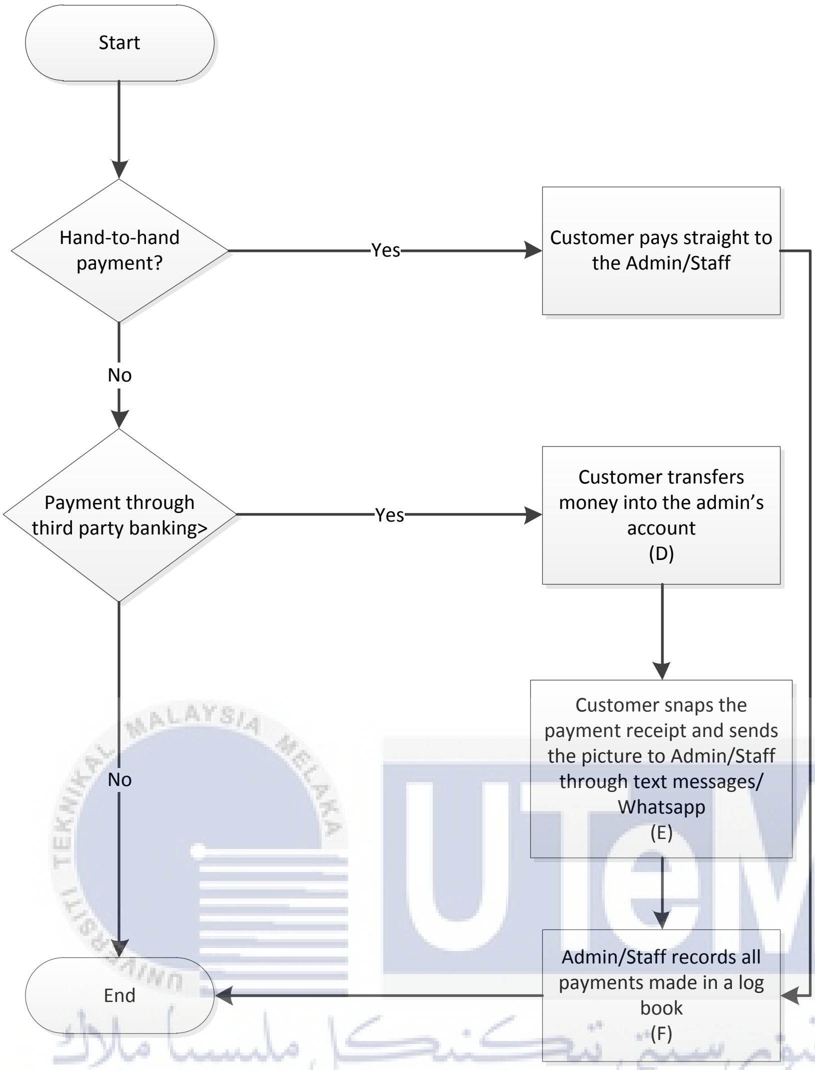

Start
Hand-to-hand payment?
Yes
Customer pays straight to the Admin/Staff
No
Payment through third party banking>
Yes
Customer transfers money into the admin's account (D)
Customer snaps the payment receipt and sends the picture to Admin/Staff through text messages/ Whatsapp (E)
No
Admin/Staff records all payments made in a log book (F)
End

Figure 3.2: Flow chart of current (as-is) system which shows how customer and staff making and accepting payments manually

Table 3.1: Problem description for labelled part in flow charts shown

Label (Figure)	Problem Description
A (Figure 1)	It will be very time consuming for customer to go to the baker's house to make their orders.
B (Figure 1)	It will be very time consuming for customer place their orders through text messages or WhatsApp.
C (Figure 1)	It is dangerous for staff to keep all orders from customers in a log book because if the book went missing or damaged, all data will be lost. Besides that, data redundancy will occur if customer repeats order and this will make it hard to keep track or refer to their last purchases.
D (Figure 2)	It will be very time consuming for customer to transfers money into admin's account to make their payment.
E (Figure 2)	It will be very time consuming for customer to sends their payment receipts through text messages or WhatsApp.
F (Figure 2)	It is dangerous for staff to keep all payment records made from customers in a log book because if the book went missing or damaged, all data will be lost. Besides that, data redundancy will occur if customer has made their payment several times and this will make it hard to keep track or refer to their last purchases.

3.3 The Proposed Improvements/Solution

This part is to describe and illustrate the proposed improvements (to-be) system in form of flow chart diagrams.

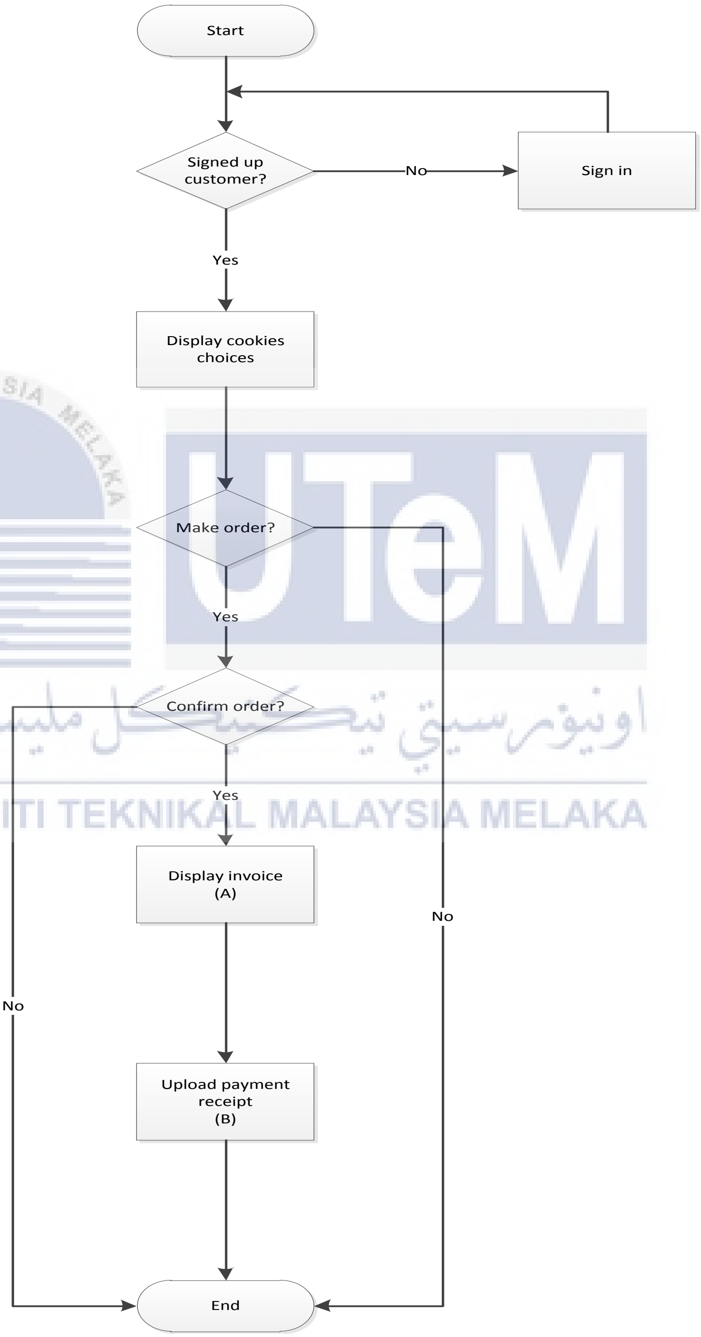


Figure 3.3: Flow chart of (to-be) system which shows how customer making their orders through a website

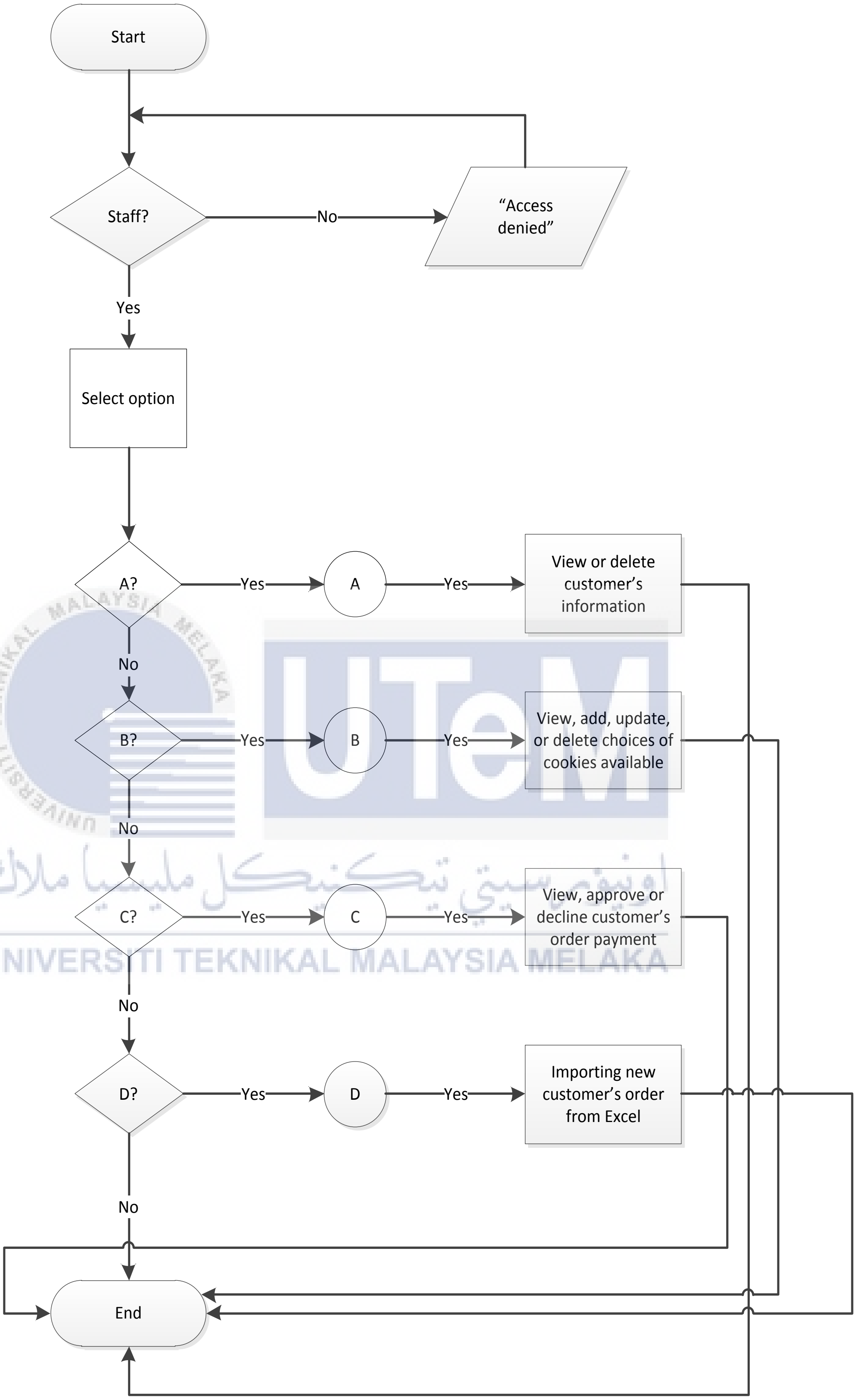

Start
Staff?
No
"Access denied"
Yes
Select option
A?
Yes
A
Yes
View or delete customer's information
No
B?
Yes
B
Yes
View, add, update, or delete choices of cookies available
No
C?
Yes
C
Yes
View, approve or decline customer's order payment
No
D?
Yes
D
Yes
Importing new customer's order from Excel
No
End

Figure 3.4: Flow chart of (to-be) system which shows staff's activity scope

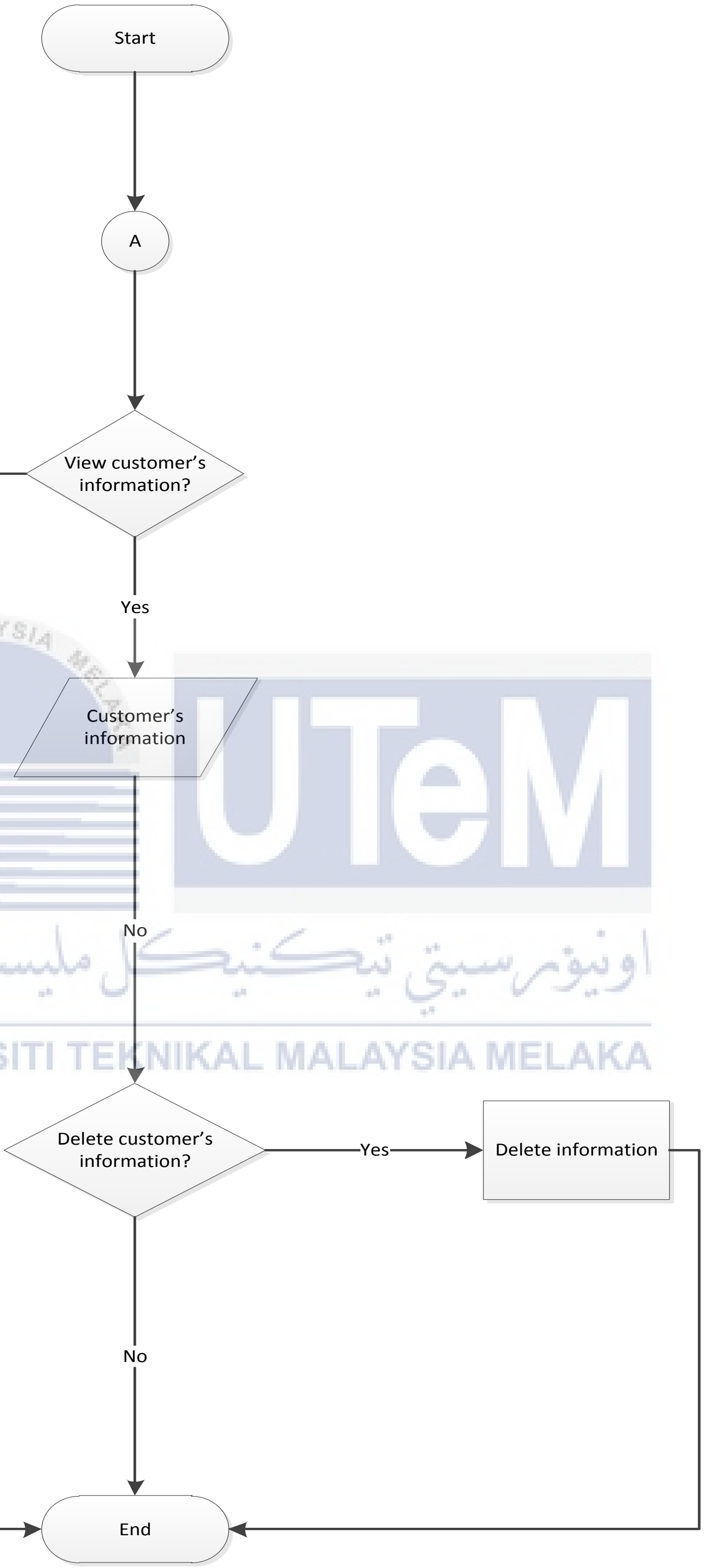


Figure 3.5: Flow chart of (to-be) system which shows staff viewing or deleting customer's information

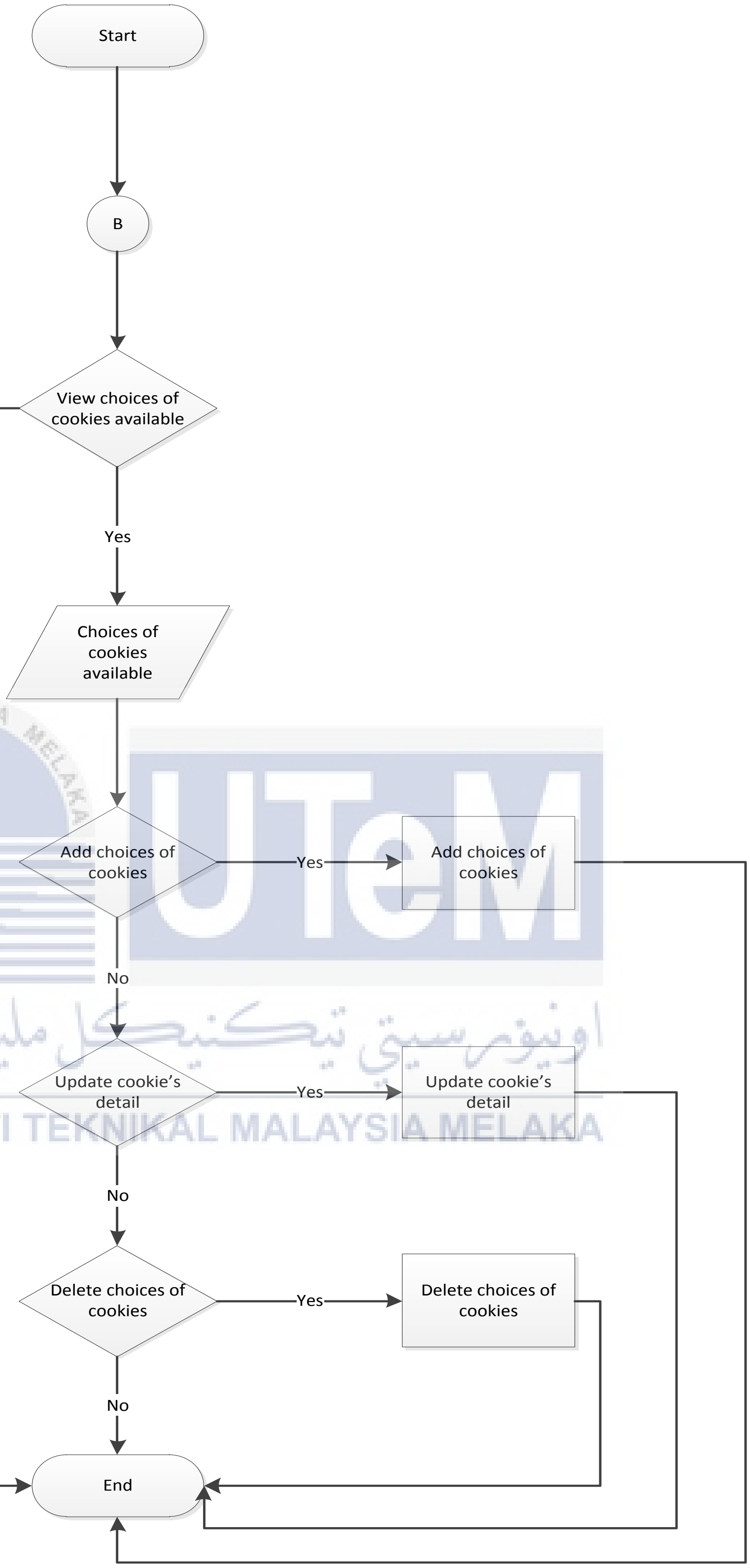


Figure 3.6: Flow chart of (to-be) system which shows staff viewing, updating or deleting cookie's choices

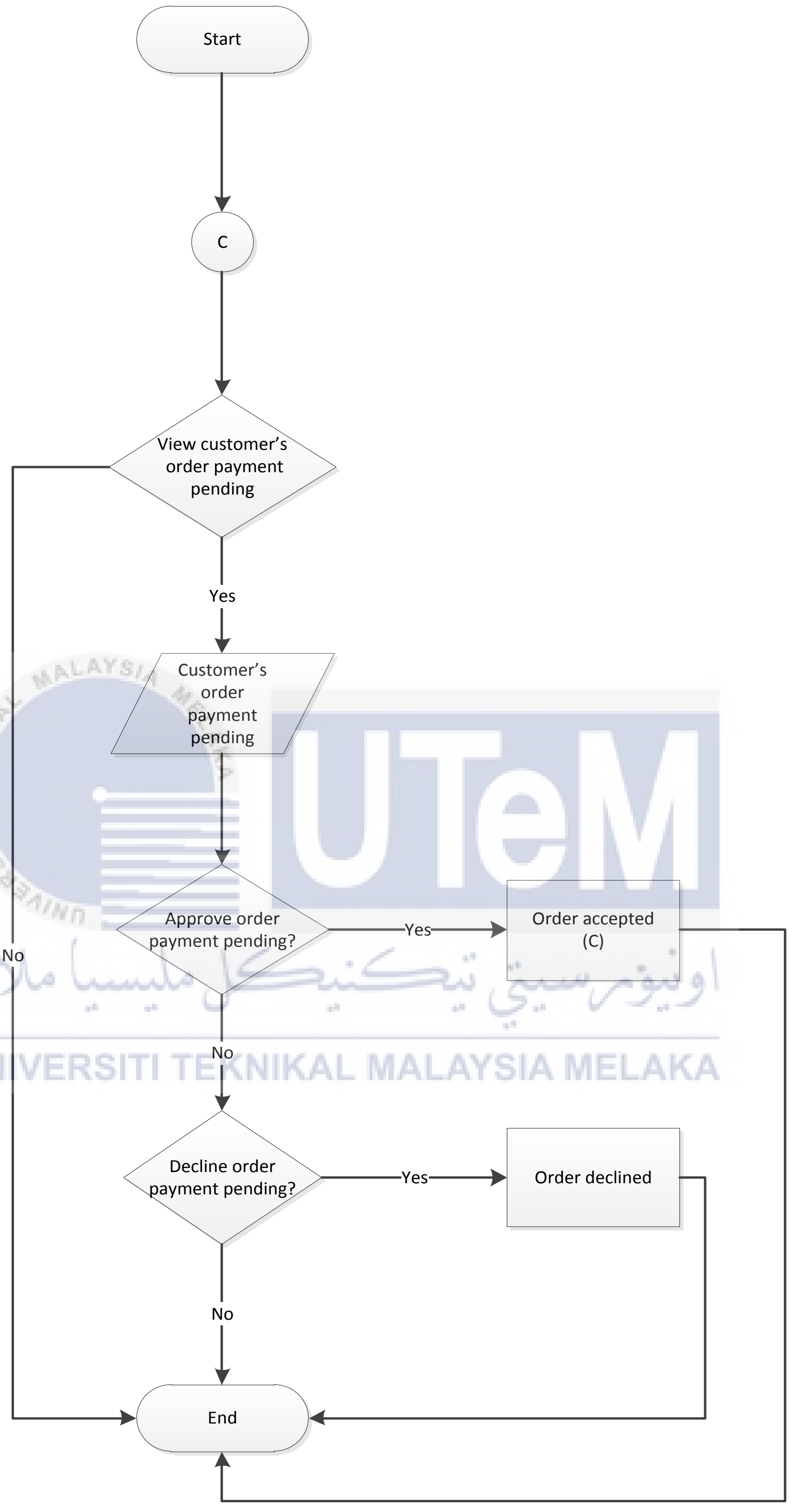

Start
C
View customer's order payment pending
Yes
No
Customer's order payment pending
Approve order payment pending?
Yes
Order accepted (C)
No
Decline order payment pending?
Yes
Order declined
No
End

Figure 3.7: Flow chart of (to-be) system which shows staff viewing, approving or declining customer's order payment

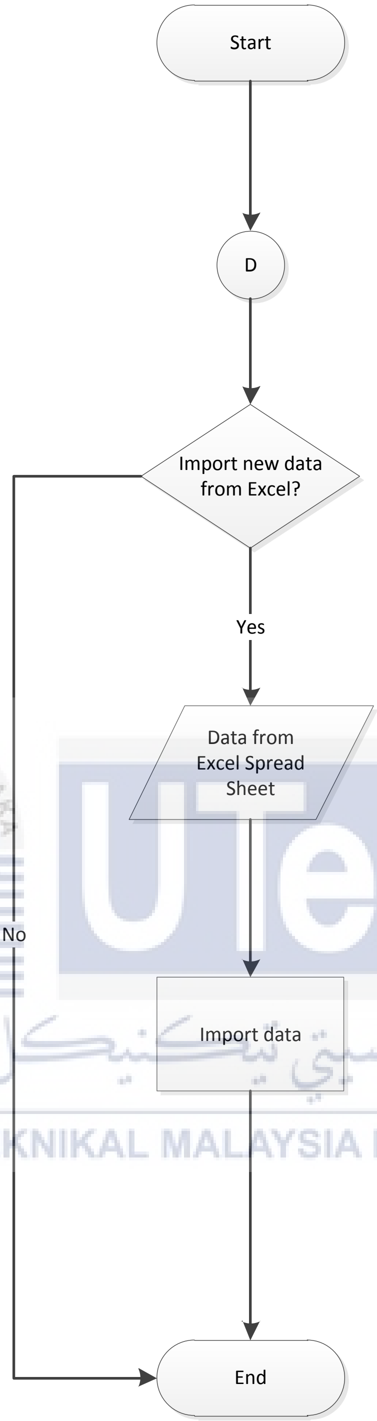


Figure 3.8: Flow chart of (to-be) system which shows staff importing customer's orders from Excel Spread Sheet into Database

Table 3.2: Labelled part in flow charts in 3.3 is the solution to all problems stated in the flow charts shown

Label (Figure)	Problem Description
A (Figure 3)	Customers do not have to go to the baker’s house or sending text messages/WhatsApp to make orders anymore. This system will accommodate customers by saving their precious time and energy in making their orders through an online website.
B (Figure 7)	Customers do not have to send their payment receipts through text messages/WhatsApp anymore. This system will accommodate customers by saving their precious time and energy by allowing them to upload their payment receipts straight into the system.
C (Figure 7)	This part will reduce the risks of losing orders and payments made from customers because every orders and transactions will be recorded and saved. Besides that, data redundancy will not occur because all customers’ archives will be kept using referential keys and this will ease NOOCMS’s staff to make their search in the future.

3.3.1 Online Ordering System Review

Below are the examples of online ordering websites that were taken from the Internet.

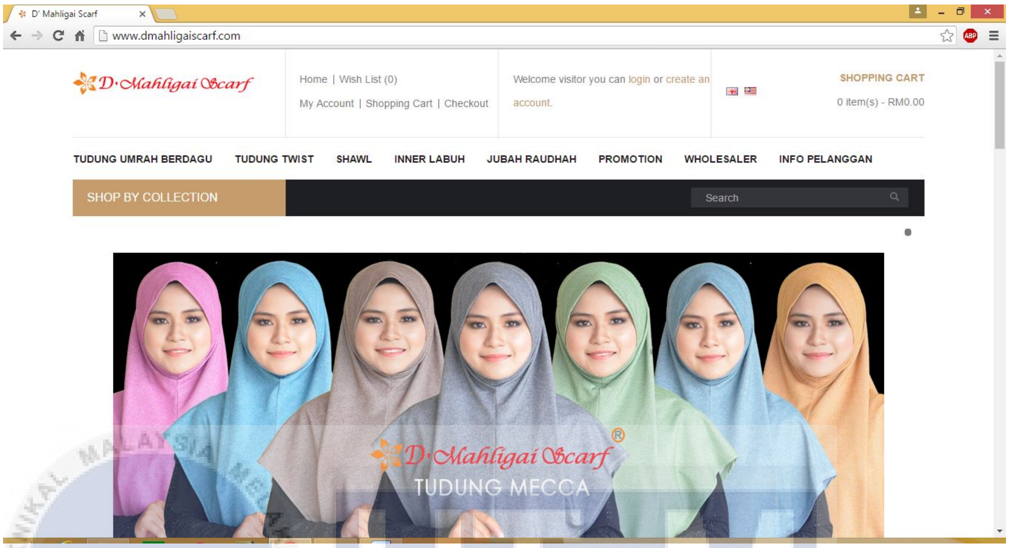


Figure 3.9: D'Mahligai official website

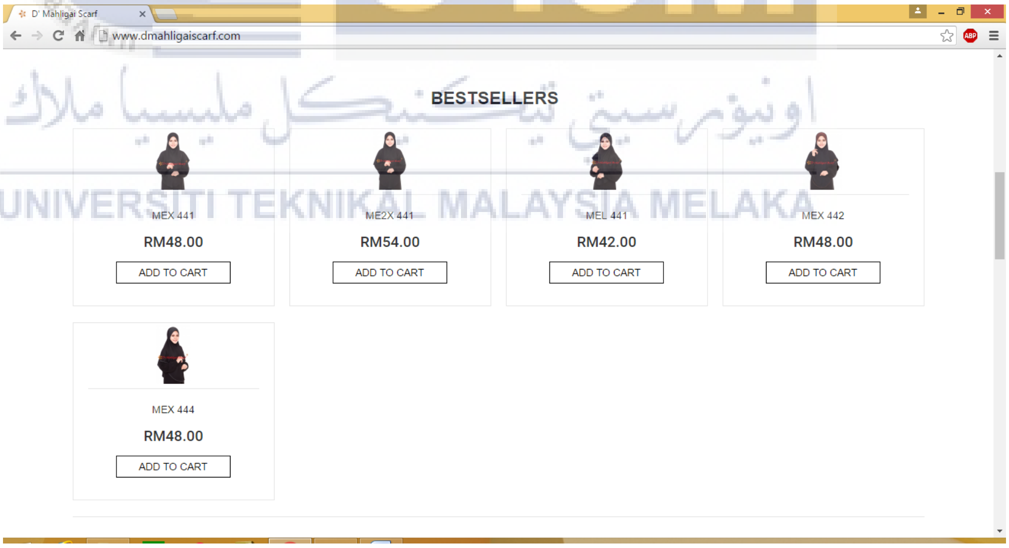
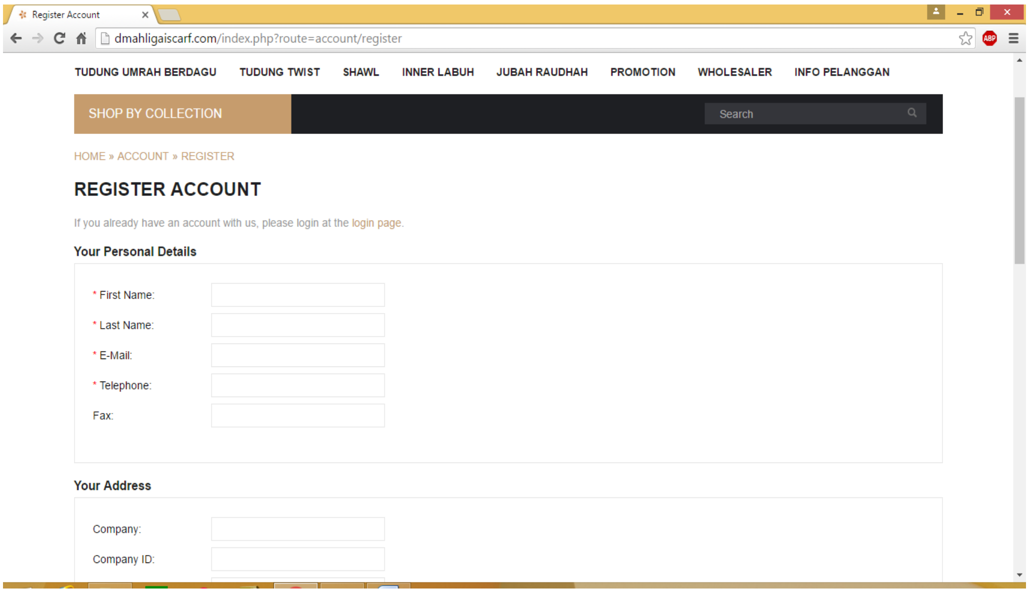


Figure 3.10: D'Mahligai official website (Add to cart function)


Register Account

dmahligaiscarf.com/index.php?route=account/register

TUDUNG UMRAH BERDAGU TUDUNG TWIST SHAWL INNER LABUH JUBAH RAUDHAH PROMOTION WHOLESALER INFO PELANGGAN

SHOP BY COLLECTION

Search

HOME » ACCOUNT » REGISTER

REGISTER ACCOUNT

If you already have an account with us, please login at the login page.

Your Personal Details

* First Name:

* Last Name:

* E-Mail:

* Telephone:

Fax:

Your Address

Company:

Company ID:

Figure 3.11: D'Mahligai official website (Registration form for new customer)

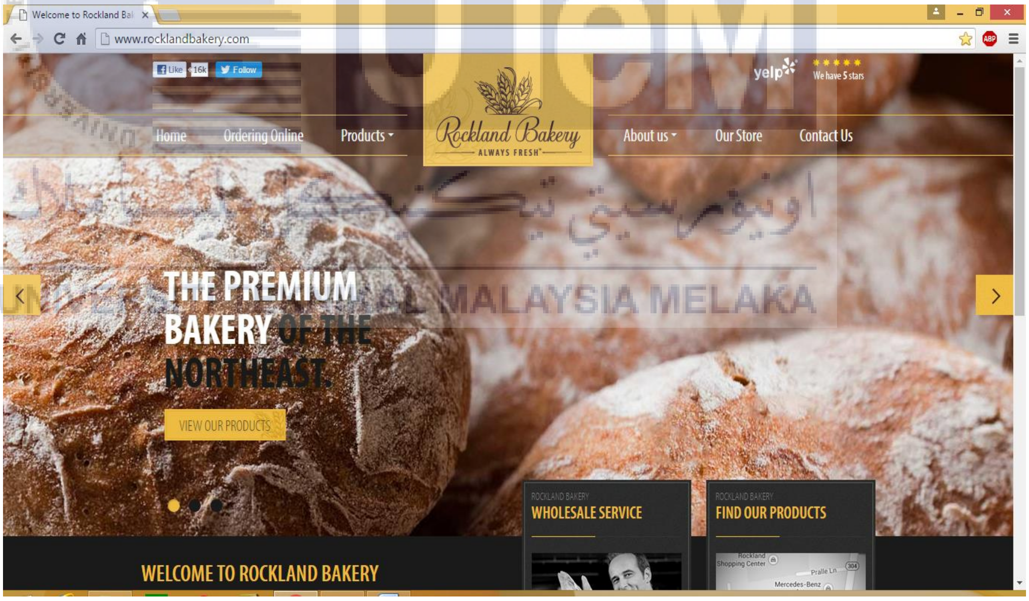


Figure 3.12: Rockland Bakery official website

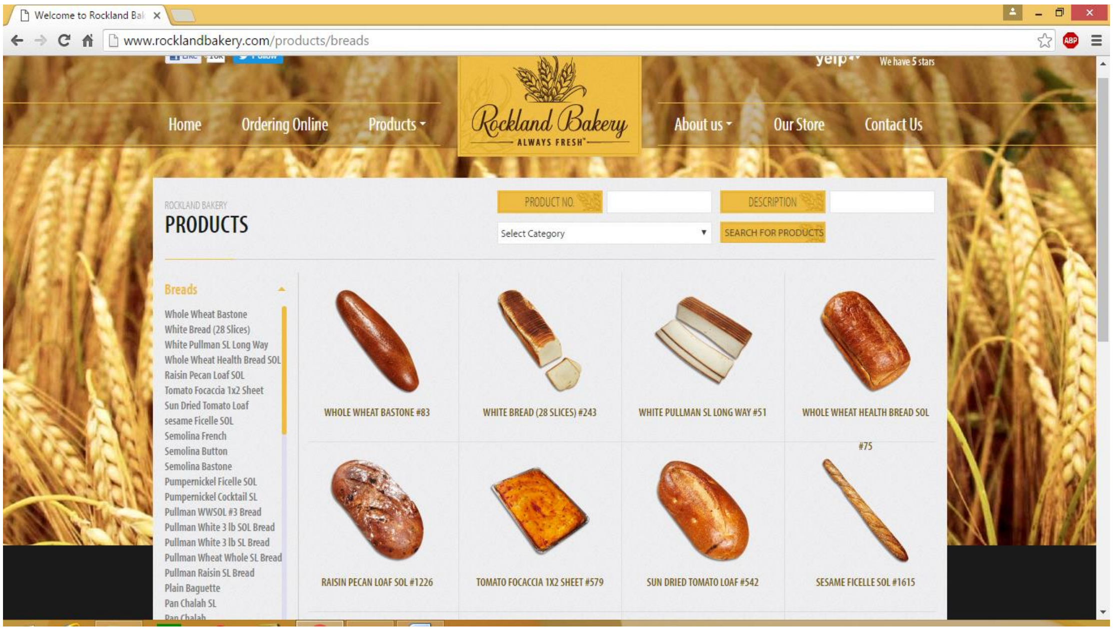
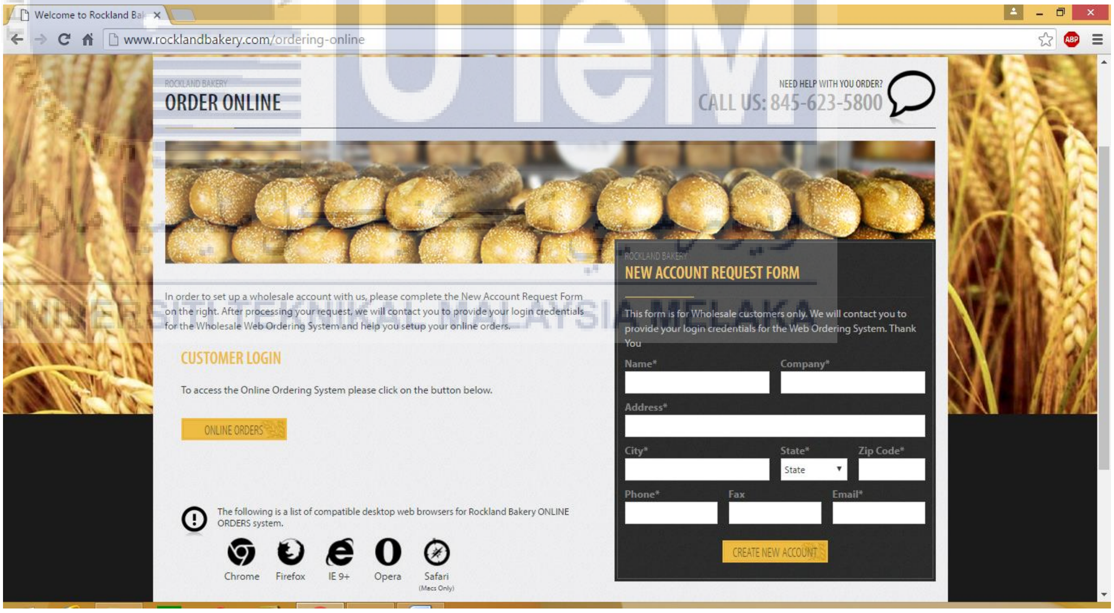


Figure 3.13: Rockland Bakery official website (Product's detail and prices)



In order to set up a wholesale account with us, please complete the New Account Request Form on the right. After processing your request, we will contact you to provide your login credentials for the Wholesale Web Ordering System and help you setup your online orders.

To access the Online Ordering System please click on the button below.

The following is a list of compatible desktop web browsers for Rockland Bakery ONLINE ORDERS system.

This form is for Wholesale customers only. We will contact you to provide your login credentials for the Web Ordering System. Thank You

Figure 3.14: Rockland Bakery official website (Registration form for new customer)

Based on the observation and reviews that have been made to both online ordering systems; D'Mahligai Scarf and Rockland Bakery, there are several things that every online ordering system supposed to have which are:

1) Product details and information with pictures to attract customer's attention
2) Add to cart function to ease customers in picking up and calculate total payment of their chosen items
3) Customer's registration forms if it is their first time making orders through the website. This is to capture customer's important information for future use.

For this NOOCM system, a function which will allow customers to upload their payment receipt will be implemented. This will save customer's time rather than they will have to send the receipt pictures to NOOCM's staff every time they have made their payment. With this, the risks of losing payment's details made from customers can be prevented because archives will be saved and can easily be retrieved when needed.

3.4 Requirement Analysis Of The To-Be System

3.4.1 Functional Requirements (Process Model)

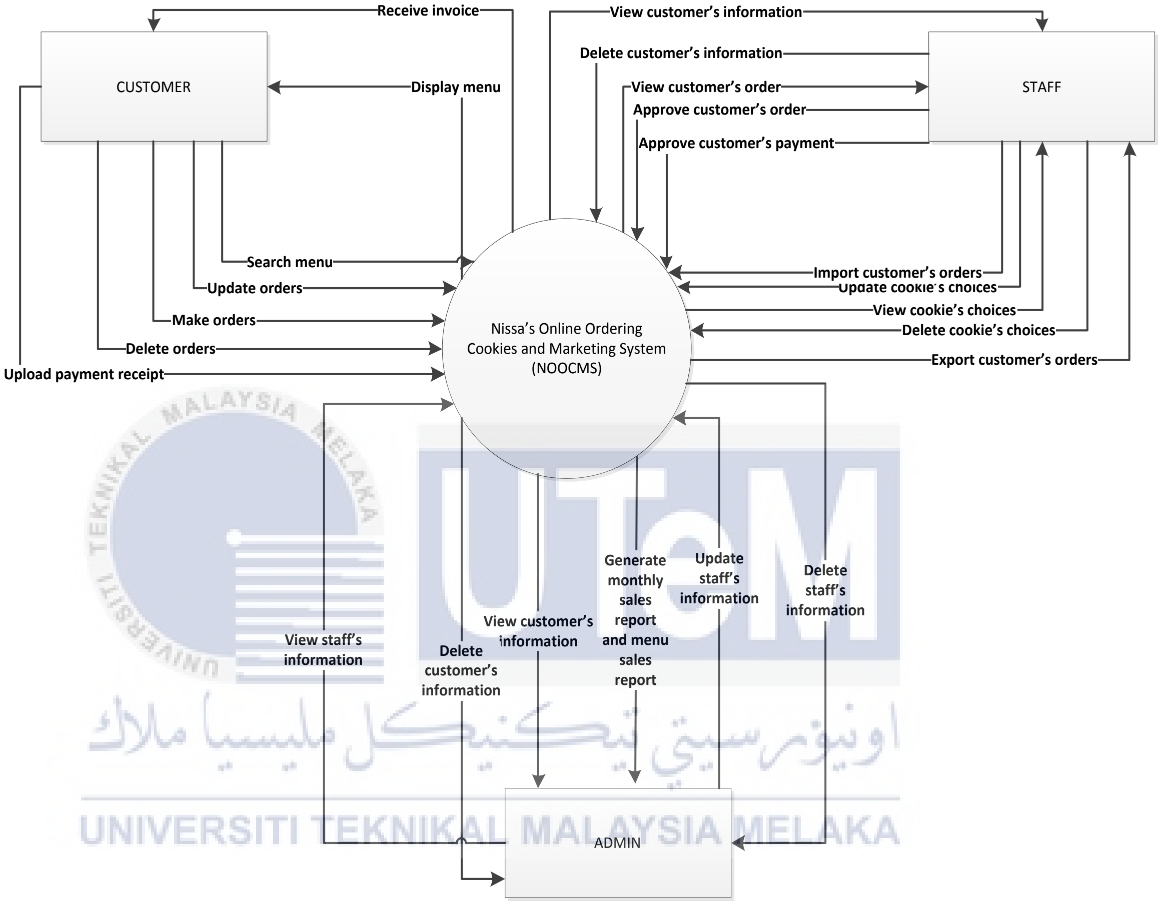

CUSTOMER
STAFF
ADMIN
Nissa's Online Ordering Cookies and Marketing System (NOOCMS)
Receive invoice
Display menu
Search menu
Update orders
Make orders
Delete orders
Upload payment receipt
View customer's information
Delete customer's information
View customer's order
Approve customer's order
Approve customer's payment
Import customer's orders
Update cookie's choices
View cookie's choices
Delete cookie's choices
Export customer's orders
View staff's information
Delete customer's information
View customer's information
Generate monthly sales report and menu sales report
Update staff's information
Delete staff's information

Figure 3.15: Context Diagram for NOOCMS

Yourdon context diagram is used in developing this system. There are three entities (actors) that involve in this system which are Admin, Staff, and Customer. Each of these entities carries out several relationships within this NOOCM system. It is either the relationship flows into the system or out from the system. If the relationship flows into the system, it means that the data is provide or entered by the actors and if the relationship flows out from the system, the data is retrieved or gained from the system.

Data Flow Diagram of to-be system

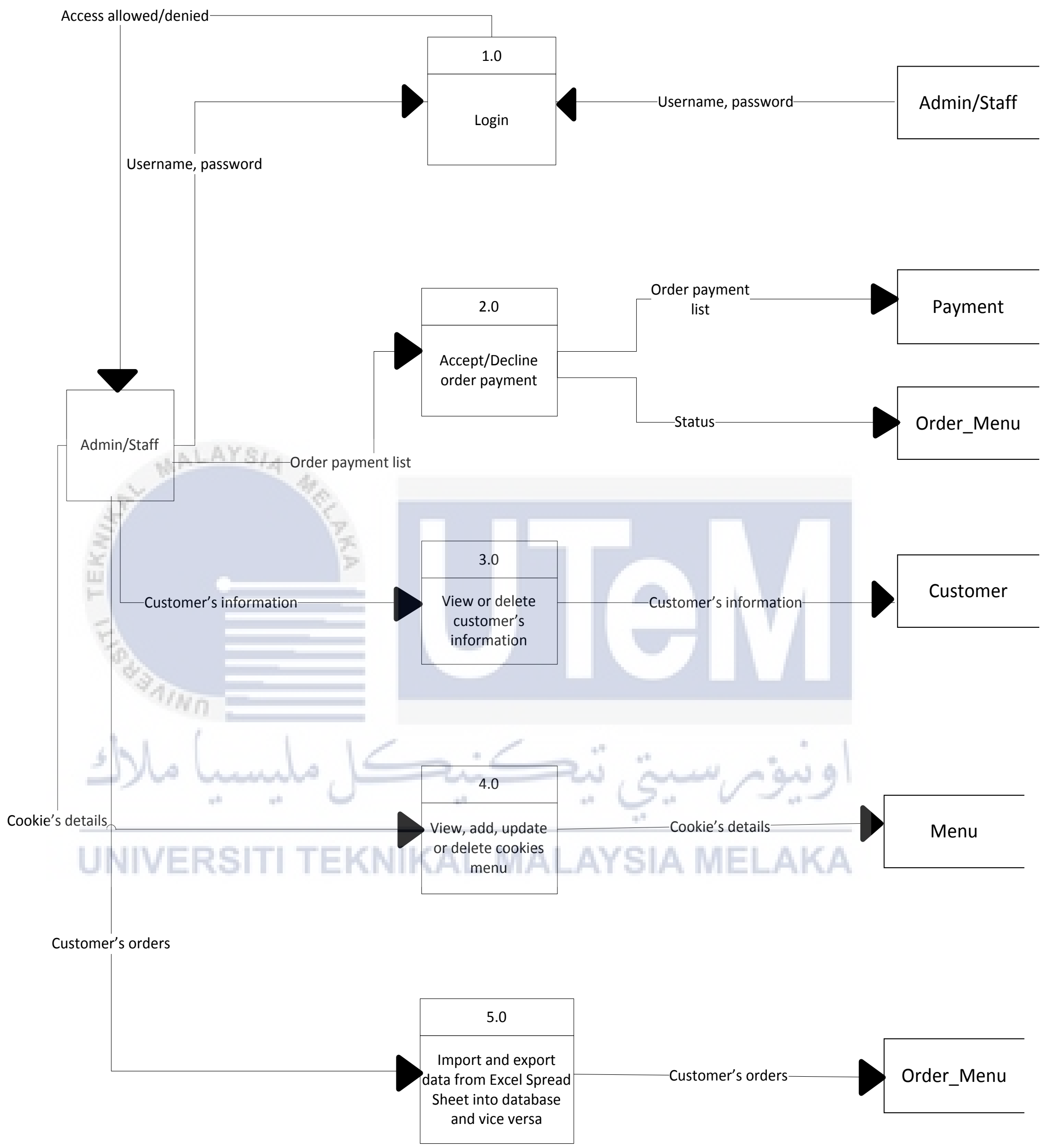


Figure 3.16: Data Flow Diagram for NOOCMS's staff

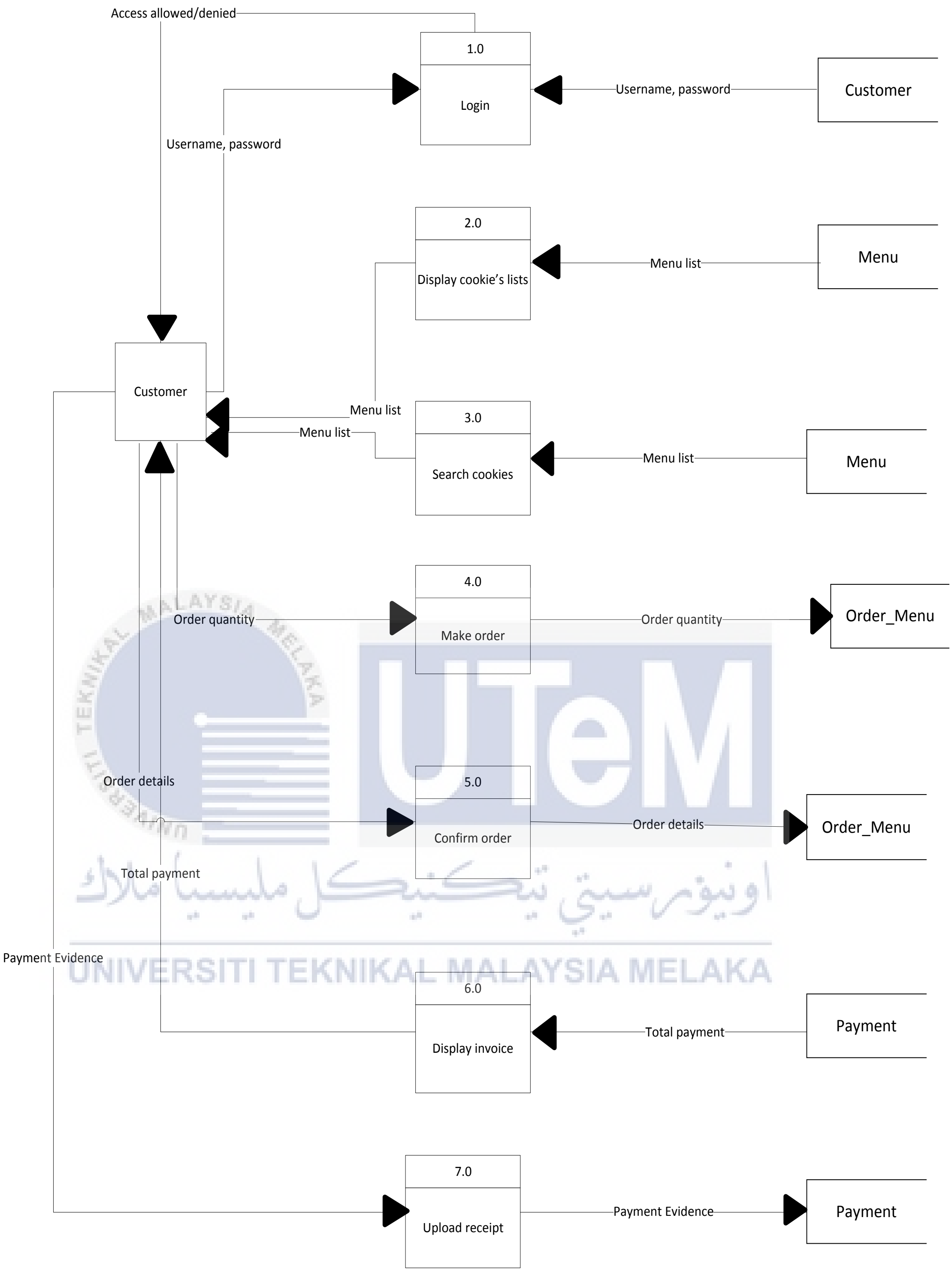


Figure 3.17: Data Flow Diagram for NOOCMS's customer

Gane Sarson data flow diagram (Level 0) is used in developing this system. It is being switched from Yourdon’s diagram to this Gane Sarson because there were some constraint occurred during producing these diagrams.

There are two data flow diagrams which hold two external entities (actors) which are Staff and Customer. Each entity carries out several processes within this NOOCM system. The processes carried out were retrieved from the tables inside the database. There are some data which were being retrieved from the database while some data were being inserted by the customer or staff. For example, during login, customer inserted their password and username and the process were to be verified by the system by checking the similarities between username and password inserted with the existing username and password inside the database table.

3.4.2 Non-Functional Requirement

i. Performance requirement: How fast is the query result to be displayed?

ii. Integrity requirement: Data integrity – Referential integrity between database tables and interfaces.

A performance testing will be conducted to understand the behaviour of the system which can determine how fast the system can work to display results. It will be tested by recording the response time taken for queries to be executed.

An integrity testing will be conducted to understand the referential integrities connected between tables in the database and interfaces. It will be tested by checking the accuracy of referential integrities in related tables.

3.4.3 Other Requirements

Table 3.3: List of software that will be used in this project

SOFTWARE	PURPOSE
Microsoft Office Word 2010	To write documents
Microsoft Office Project 2010	To produce Work Breakdown Structure, Gantt Chart and Milestones
Microsoft Visio 2010	To create Entity Relationship Diagram (ERD), Flow Charts, and Data Flow Diagram
Adobe Photoshop	To make NCOOMS logo
Notepad	To backup all of the coding

Table 3.4: List of hardware that will be used in this project

HARDWARE	PURPOSE
Notebook/Laptop	To work on the project
Printer	To print documents

The programming language that will be used in developing this project is Hypertext Pre-processor (PHP) and the software is Adobe Dreamweaver CS3. As for the Operating System (OS) and Database Management System (DBMS), Windows 8.1 and Oracle 11g were chosen respectively.

3.5 Conclusion

Flow Charts of current (as-is) and to-be system had been shown to explain the concept of how those system functions. Context Diagram is to explain on the whole system to be developed in general, while the Data Flow Diagram has scrutinized more on the input and output details needed for each step carried out by the system.

The development technique of system analysis has helped to gain better understanding of overall structure in the system organizations. System analysis techniques can be considered as the first step in deciding the complexities of the real world into a model that can be implemented on a computer and could be accessed by many users.

From the analysis carried out, a conclusion that can be made is that this system to-be of Nissa's Online Ordering Cookies and Marketing (NOOCM) is a system that can achieve high efficiency if it is developed properly and complete with all the functionalities discussed.

In the next chapter (Chapter 4), the design of system will be explained into details by elaborating and describing the Entity Relationship Diagram (ERD) together with its business rules, data dictionary, normalization of data, and Graphical User Interface (GUI).

CHAPTER IV

DESIGN

4.1 Introduction

Designing is the most crucial phase in the development of a system. The logical system design arrived as a result of system analysis and is converted into physical system design. Physical design is a detailed description of what is needed to solve the original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up into details. Data structure, control process, interface, documentation, and procedures are decided at this stage.

There are several techniques that can be used to describe the design of the system. For example, it can be shown through Entity Relationship Diagram (ERD), business rules, data dictionary, data normalization, the selection of Database Management System (DBMS), and the Graphical User Interface (GUI).

4.2 System Architecture Design

Architecture views are representations of the overall architecture that are meaningful to one or more stakeholders in the system. The architect chooses and develops a set of views that will enable the architecture to be communicated to, and understood by all the stakeholders, and enable them to verify that the system will address their concerns.

In this system, the architecture view used is a web-based application system. Web-based application system refers to any program that is accessed over a network connection using HTTP, rather than existing within a device's memory. Web-based application often runs inside a web browser.

Web-based system is chose as the architecture view in this project is because it suits the needs of customers who will place their orders online. Customers can place their orders by surfing the website easily. A web-based application is accessible anywhere and through any range of devices. Besides that, it is easier to install or to be maintained and it will grow as we grow.

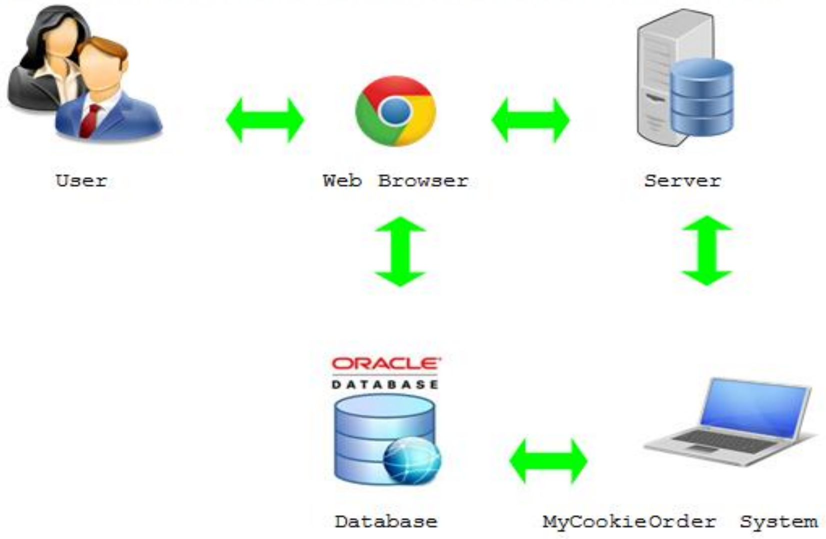


Figure 4.1: A web based system application that could be accessed by customers anywhere and anytime

4.2.1 Conceptual Design

Below are the Entity Relationship Diagram (ERD) of Nissa's Online Ordering Cookies and Marketing System (NOOCMS). NOOCMS's ERD consists of six entities; Customer, Payment, Staff, Order, Order_Menu, and Menu. Each entity has several attributes for data to be captured and saved and in order for it to be retrieved again.

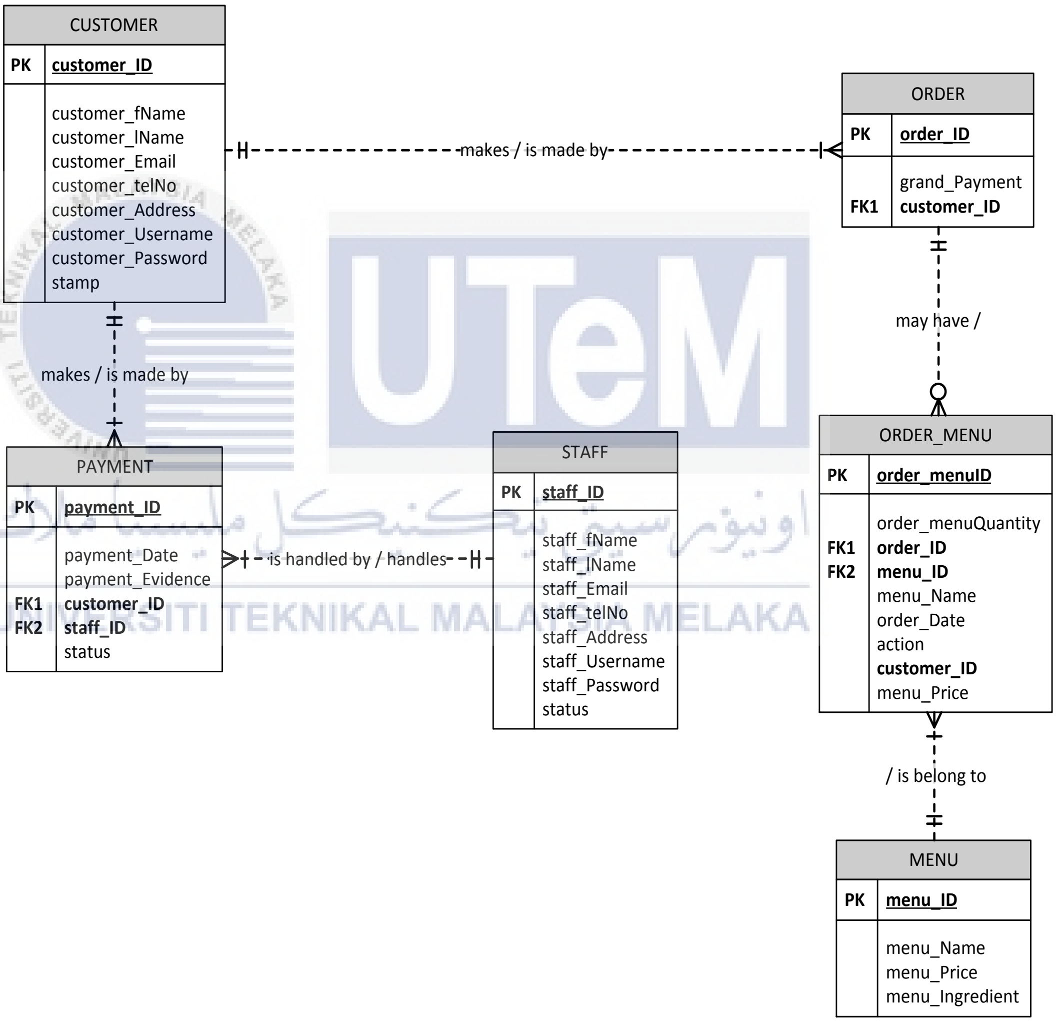


Figure 4.2: NOOCM's ERD

Below are NOOCMS business rules that explains the relationship among each entity based on the ERD shown above.

Table 4.1: NOOCMS's Business Rules

Business Rule	Description
1	Each customer must make at least one order
	Each order is made by only one customer
2	Each customer must make at least one payment
	Each payment is made by only one customer
3	Each staff must handle at least one payment
	Each payment is handled by only one staff
4	Each order may have many menu
	Each menu is belong to many orders

4.2.2 Logical Design

Below is the data dictionary of NOOCMS that contains attributes of each table and some explanation on the data type, data constraint, reference table, the attribute's description.

Table 4.2: NOOCMS's Data Dictionary

Table	Column	Data type	Constraint	Reference Table
Customer	customer_ID	VARCHAR2(20)	Primary Key	-
	customer_fName	VARCHAR2(20)	Not null	-
	customer_lName	VARCHAR2(20)	Not null	-
	customer_Email	VARCHAR2(50)	Unique	-
	customer_telNo	NUMBER	Unique	-
	customer_Address	VARCHAR2(100)	Not null	-
	customer_Username	VARCHAR2(20)	Unique	-
	customer_Password	VARCHAR2(20)	Unique	-
	stamp	NUMBER	Not null	-
Menu	menu_ID	VARCHAR2(20)	Primary key	-
	menu_Name	VARCHAR2(50)	Not null	-
	menu_Price	NUMBER(7,2)	Not null	-
	menu_Ingredient	VARCHAR2(100)	Not null	-
Order	order_ID	VARCHAR2(20)	Primary key	-
	grand_Payment	NUMBER(7,2)	Not null	-
	customer_ID	VARCHAR2(20)	Foreign key	Customer

Order_Menu	order_menuID	VARCHAR2(20)	Primary key	-
	order_menuQuantity	NUMBER	Not null	-
	order_ID	VARCHAR2(20)	Foreign key	Order
	menu_ID	VARCHAR2(20)	Foreign key	Menu
	menu_Name	VARCHAR2(50)	Not null	-
	order_Date	DATE	Not null	-
	action	VARCHAR2(20)	Not null	-
	customer_ID	VARCHAR2(20)	Foreign key	Customer
	menu_Price	NUMBER(7,2)	Not null	-
Payment	payment_ID	VARCHAR2(20)	Primary key	-
	payment_Date	DATE	Not null	-
	payment_Evidence	BLOB	-	-
	customer_ID	VARCHAR2(20)	Foreign key	Customer
	staff_ID	VARCHAR2(20)	Foreign key	Staff
	status	VARCHAR2(20)	Not null	-
Staff	staff_ID	VARCHAR2(20)	Primary key	-
	staff_fName	VARCHAR2(20)	Not null	-
	staff_lName	VARCHAR2(20)	Not null	-
	staff_Email	VARCHAR2(50)	Unique	-
	staff_telNo	NUMBER	Unique	-
	staff_Address	VARCHAR2(100)	Not null	-
	staff_Username	VARCHAR2(20)	Unique	-
	staff_Password	VARCHAR2(20)	Unique	-
	status	NUMBER	Not null	-

Below are the NOOCMS's normalization's of data among six tables that consists in the ERD.

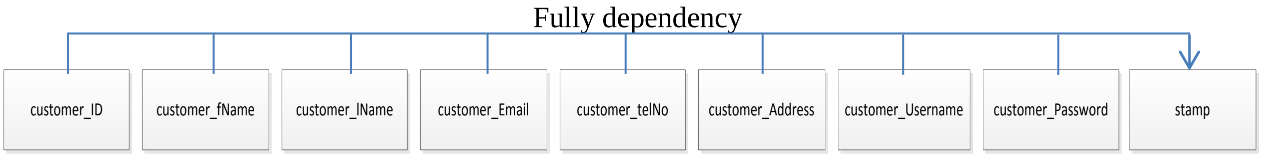


Figure 4.3: Normalization of Customer table

Customer table is already in 3NF because all of its attributes are fully dependency towards each other.

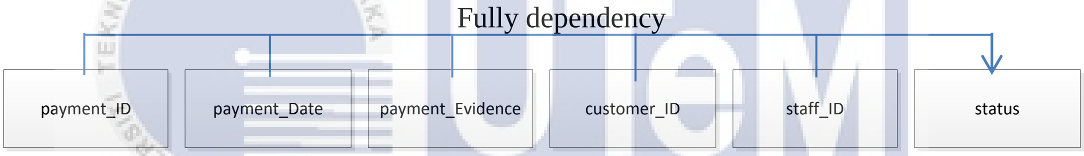


Figure 4.4: Normalization of Payment table

Payment table is already in 3NF because all of its attributes are fully dependency towards each other.

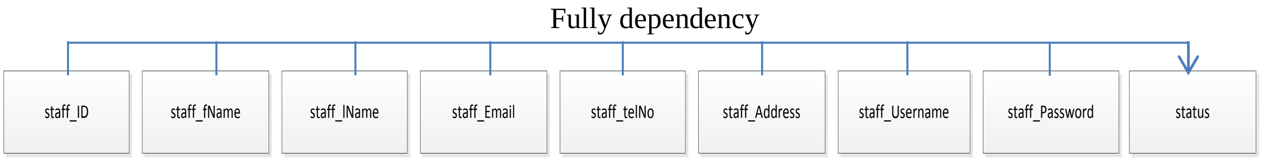


Figure 4.5: Normalization of Staff table

Staff table is already in 3NF because all of its attributes are fully dependency towards each other.

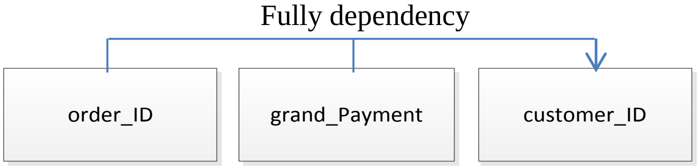


Figure 4.6: Normalization of Order table

Order table is already in 3NF because all of its attributes are fully dependency towards each other.

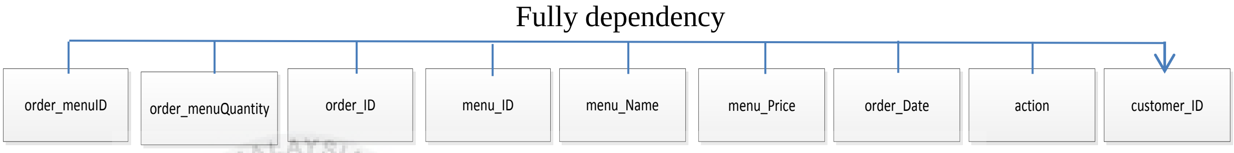


Figure 4.7: Normalization of Order_Menu table

Order_Menu table is already in 3NF because all of its attributes are fully dependency towards each other.

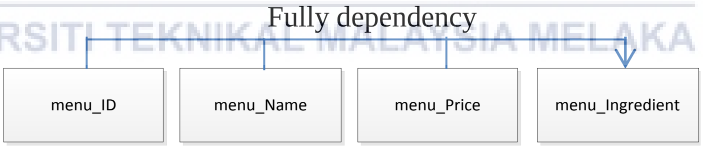


Figure 4.8: Normalization of Menu table

Menu table is already in 3NF because all of its attributes are fully dependency towards each other.

Based on the normalization done towards all six tables, it is proved that every table has been normalized up until the 3NF. Hence, the ERD is done completely.

4.2.3 Physical Design

-Adopted from www.bingapis.com-

The Database Management System (DBMS) selected for this project is Oracle 11g. This is because, Oracle has many advantages and features that make it known by all and therefore makes it as the world's largest enterprise software company. Oracle comes with new versions with new features implemented in the new version while the features of earlier versions are still being maintained. One important aspect is that Oracle databases tend to be backwards compatible. When Oracle releases a new version, their documentation contains a list of all the features new to that version thus makes it user friendly for one to learn the new features.

Oracle is a database that responds very well with excellent performance in demanding environments. Oracle is a major database where the added features pass the ACID test, which is important in ensuring the integrity of data. Data is the heart of any system in an organization. A reliable and adequate database system must have the following properties:

Atomicity

- The results of a transaction's execution are either all committed or all rolled back.

Consistency

- The database is transformed from one valid state to another valid state. Illegal transactions are not allowed and if an integrity constraint cannot be satisfied, then the transaction is rolled back.

Isolation

- The results of a transaction are invisible to other transactions until the transaction is complete thus increasing the security of data.

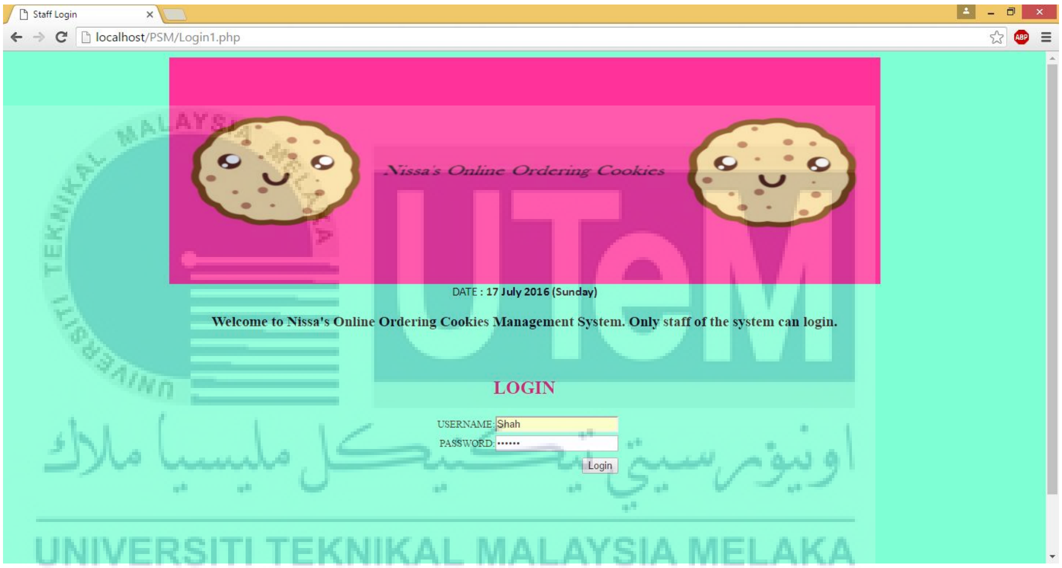
Durability

- Once committed (completed), the results of a transaction are permanent and survive future system and media failures, thus ensuring maintenance and protection of data.

All the above criteria explained are well maintained by Oracle database. Therefore, by using Oracle as this project's DBMS, it will bring a lot of benefits to the system.

4.3 Graphical User Interface (GUI) Design

Below are the some of the NOOCMS's interfaces that have been implemented some triggers and stored procedures through the database. The triggers used in this system are before insert triggers, which is to generate primary key automatically for all tables while the stored procedures used are for inserting, updating, deleting, and viewing data from tables. All figures shown below are specifically for inserting, updating, and deleting data on table Menu.



DATE : 17 July 2016 (Sunday)

Welcome to Nissa's Online Ordering Cookies Management System. Only staff of the system can login.

LOGIN

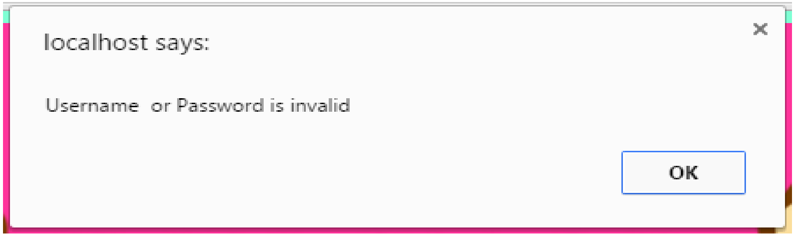
USERNAME: Shah

PASSWORD: ••••••

Login

Figure 4.9: Login interface for NOOCMS's

This required data during this stage are staff's Username and Password. If the Username and Password entered are correct, then the user will be linked straight to the main page of NOOCMS (Figure 11). Otherwise, there will be a pop-up (as in Figure 10).

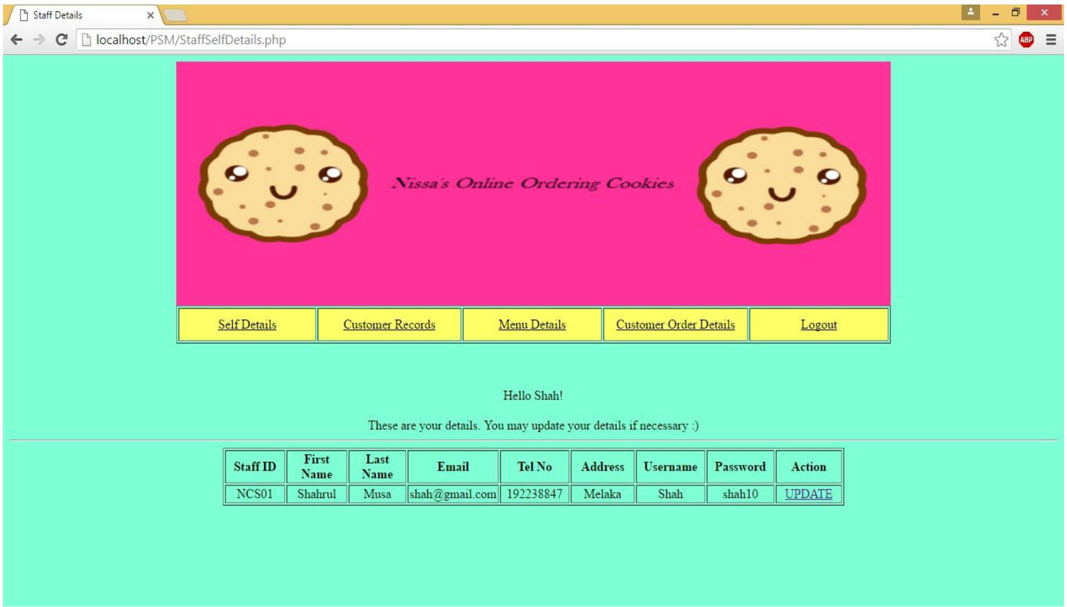


localhost says:

Username or Password is invalid

OK

Figure 4.10: Error handling on wrong input for Username and Password

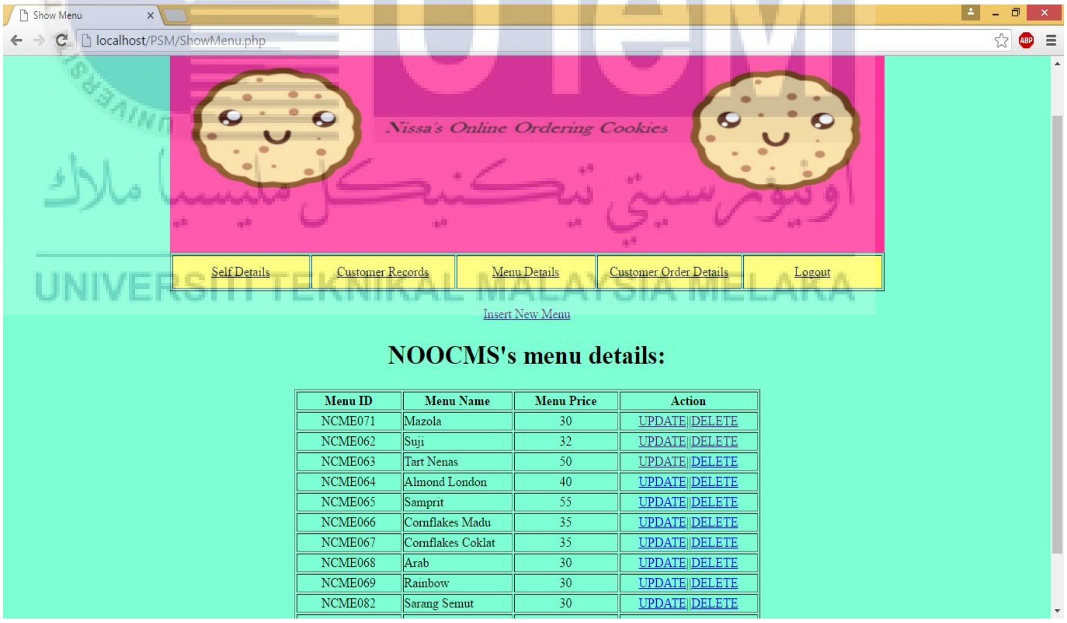


Hello Shah!

These are your details. You may update your details if necessary :)

Staff ID	First Name	Last Name	Email	Tel No	Address	Username	Password	Action
NCS01	Shahrul	Musa	shah@gmail.com	192238847	Melaka	Shah	shah10	UPDATE

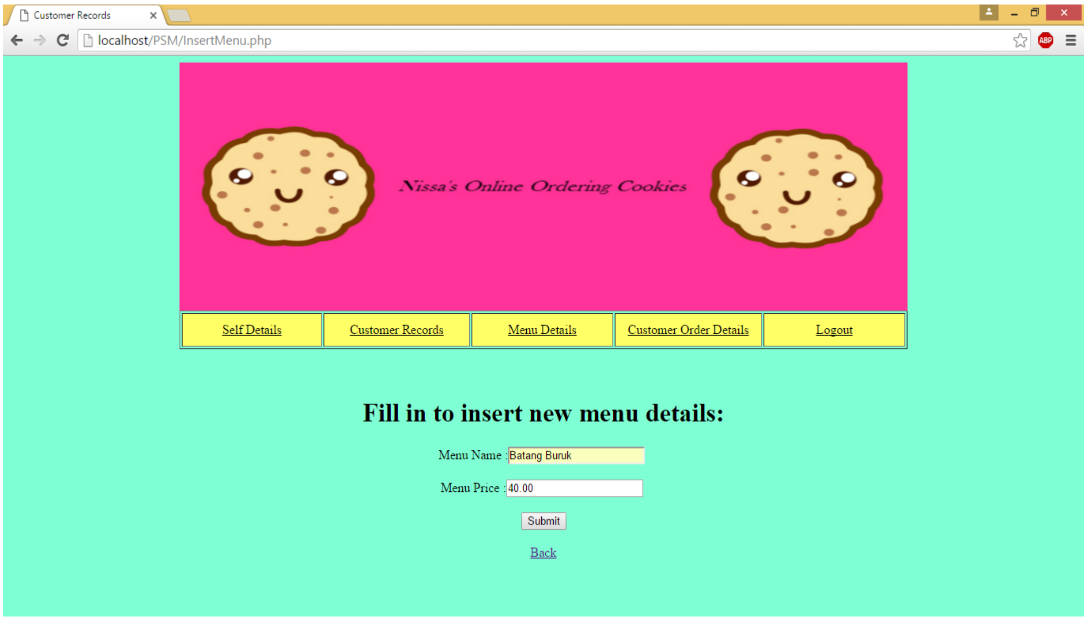
Figure 4.11: Main page of NOOCMS



NOOCMS's menu details:

Menu ID	Menu Name	Menu Price	Action
NCME071	Mazola	30	UPDATE|DELETE
NCME062	Suji	32	UPDATE|DELETE
NCME063	Tart Nenas	50	UPDATE|DELETE
NCME064	Almond London	40	UPDATE|DELETE
NCME065	Samprit	55	UPDATE|DELETE
NCME066	Cornflakes Madu	35	UPDATE|DELETE
NCME067	Cornflakes Coklat	35	UPDATE|DELETE
NCME068	Arab	30	UPDATE|DELETE
NCME069	Rainbow	30	UPDATE|DELETE
NCME082	Sarang Semut	30	UPDATE|DELETE

Figure 4.12: The existing data from table Menu

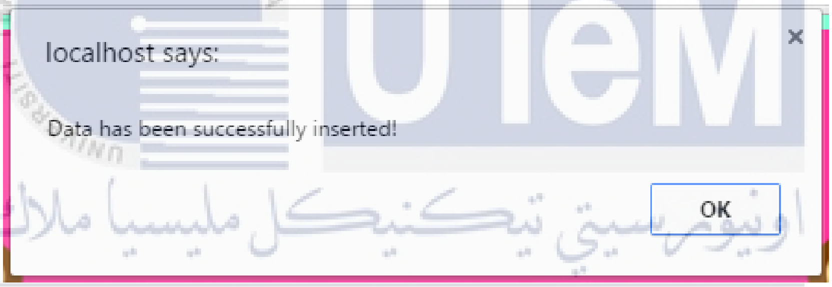


Fill in to insert new menu details:

Menu Name : Batang Buruk

Menu Price : 40.00

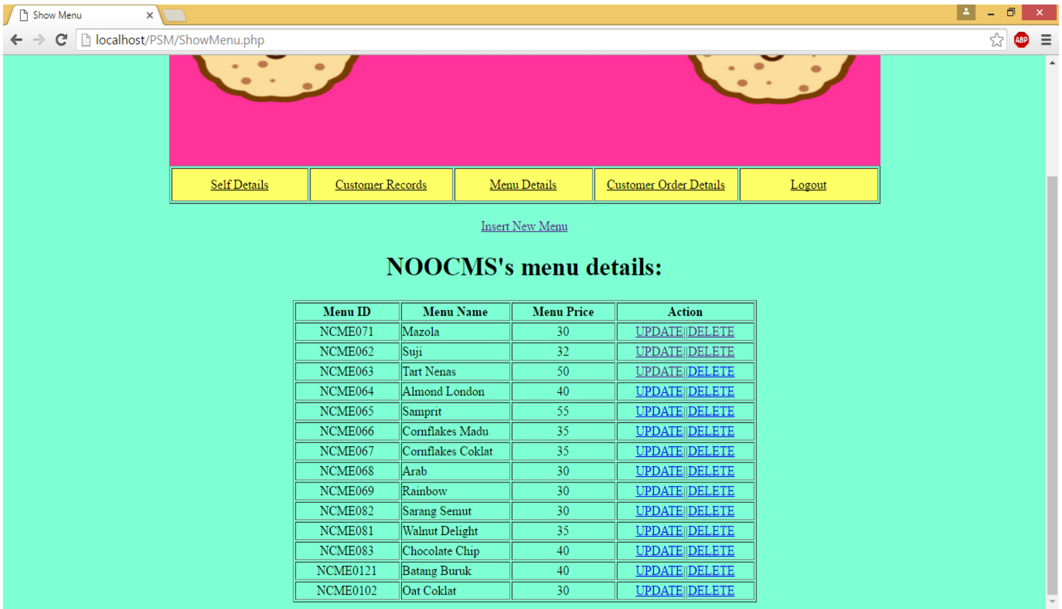
Figure 4.13: Inserting data into table Menu through web server using stored procedure



localhost says:

Data has been successfully inserted!

Figure 4.14: Pop-up is shown to inform user that data has been successfully inserted

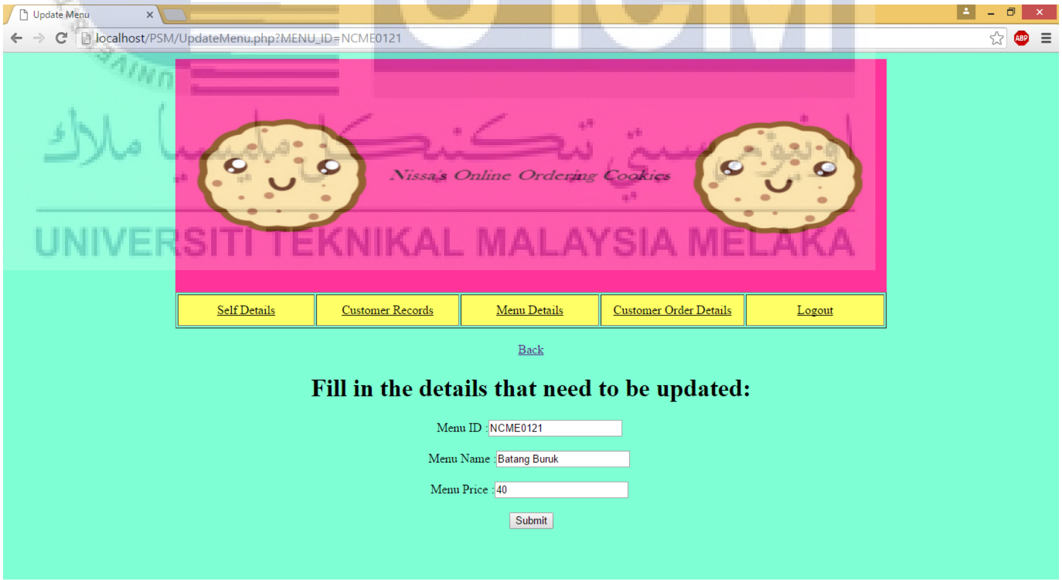


Insert New Menu

NOOCMS's menu details:

Menu ID	Menu Name	Menu Price	Action
NCME071	Mazola	30	UPDATE‖DELETE
NCME062	Suji	32	UPDATE‖DELETE
NCME063	Tart Nenas	50	UPDATE‖DELETE
NCME064	Almond London	40	UPDATE‖DELETE
NCME065	Samprit	55	UPDATE‖DELETE
NCME066	Cornflakes Madu	35	UPDATE‖DELETE
NCME067	Cornflakes Coklat	35	UPDATE‖DELETE
NCME068	Arab	30	UPDATE‖DELETE
NCME069	Rainbow	30	UPDATE‖DELETE
NCME082	Sarang Semut	30	UPDATE‖DELETE
NCME081	Walnut Delight	35	UPDATE‖DELETE
NCME083	Chocolate Chip	40	UPDATE‖DELETE
NCME0121	Batang Buruk	40	UPDATE‖DELETE
NCME0102	Oat Coklat	30	UPDATE‖DELETE

Figure 4.15: The new data is successfully inserted into the Menu table



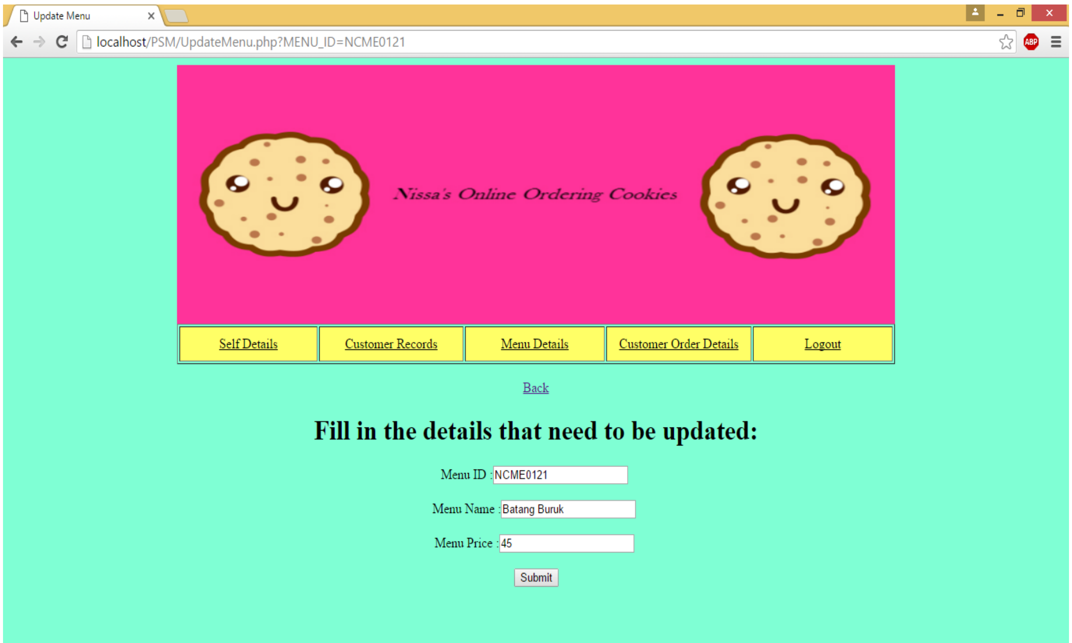
Fill in the details that need to be updated:

Menu ID : NCME0121

Menu Name : Batang Buruk

Menu Price : 40

Figure 4.16: Updating data from the Menu table through web server using stored procedure



Fill in the details that need to be updated:

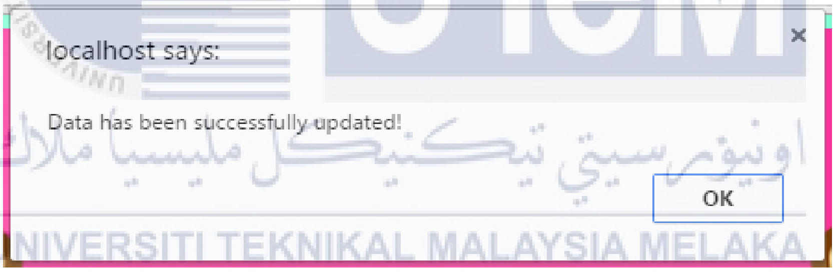
Menu ID : NCME0121

Menu Name : Batang Buruk

Menu Price : 45

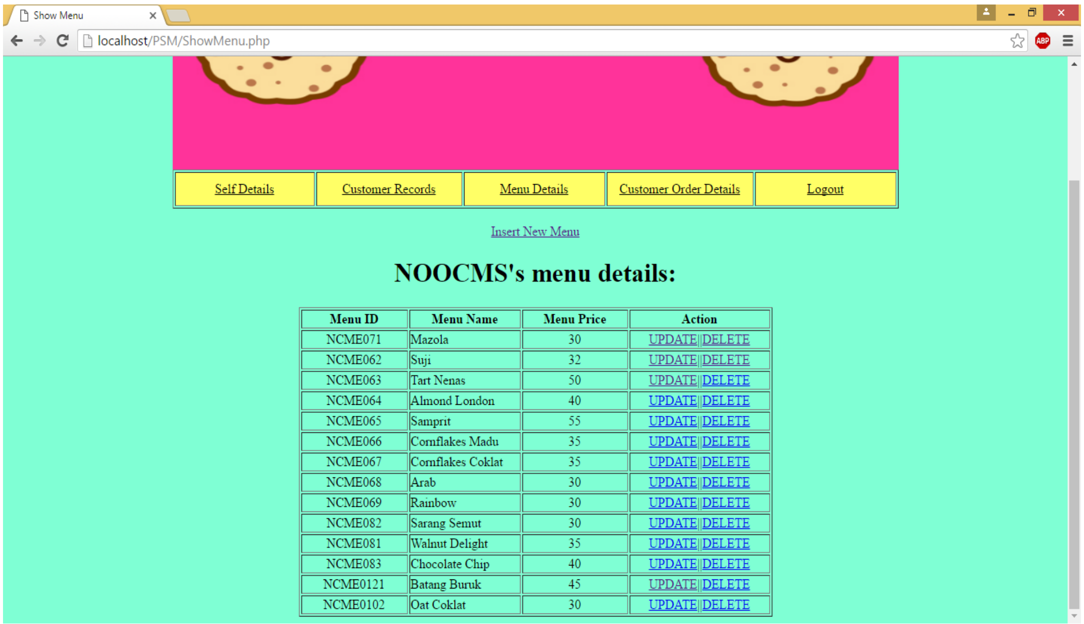
Submit

Figure 4.17: Updating data from the Menu table through web server using stored procedure


localhost says:

Data has been successfully updated!

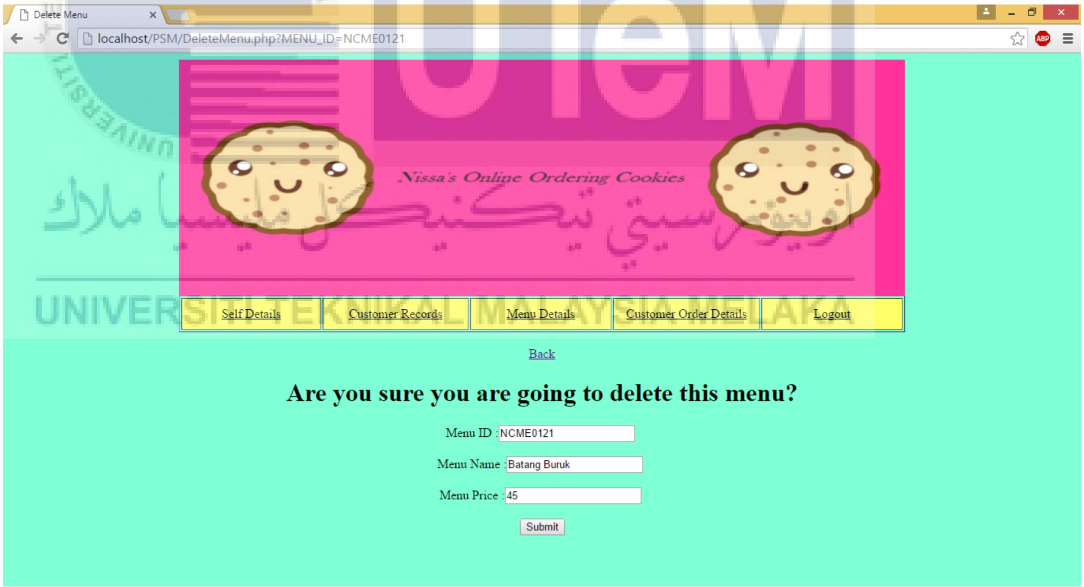
Figure 4.18: Pop-up is shown to inform user that data has been successfully updated



NOOCMS's menu details:

Menu ID	Menu Name	Menu Price	Action
NCME071	Mazola	30	UPDATE|DELETE
NCME062	Suji	32	UPDATE|DELETE
NCME063	Tart Nenas	50	UPDATE|DELETE
NCME064	Almond London	40	UPDATE|DELETE
NCME065	Samprit	55	UPDATE|DELETE
NCME066	Cornflakes Madu	35	UPDATE|DELETE
NCME067	Cornflakes Coklat	35	UPDATE|DELETE
NCME068	Arab	30	UPDATE|DELETE
NCME069	Rainbow	30	UPDATE|DELETE
NCME082	Sarang Semut	30	UPDATE|DELETE
NCME081	Walnut Delight	35	UPDATE|DELETE
NCME083	Chocolate Chip	40	UPDATE|DELETE
NCME0121	Batang Buruk	45	UPDATE|DELETE
NCME0102	Oat Coklat	30	UPDATE|DELETE

Figure 4.19: The new data has been updated inside the Menu table



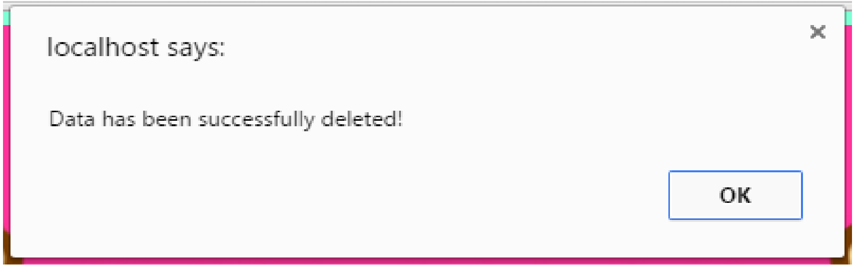
Are you sure you are going to delete this menu?

Menu ID : NCME0121

Menu Name : Batang Buruk

Menu Price : 45

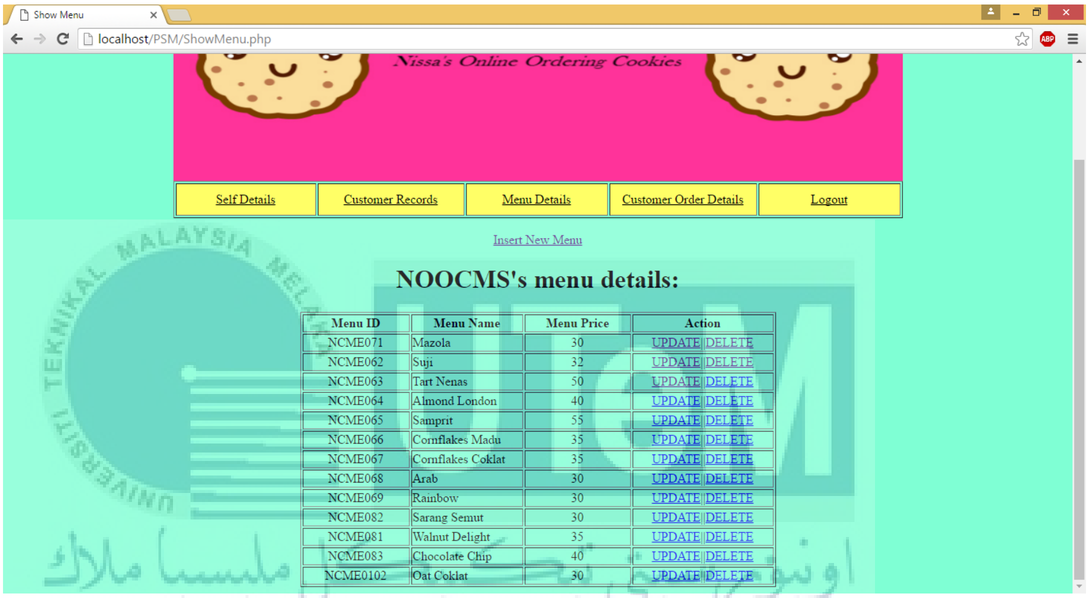
Figure 4.20: Deleting data from table Menu through web server using stored procedure



localhost says:

Data has been successfully deleted!

Figure 4.21: Pop-up is shown to inform user that data has been successfully deleted



NOOCMS's menu details:

Menu ID	Menu Name	Menu Price	Action
NCME071	Mazola	30	UPDATE||DELETE
NCME062	Suji	32	UPDATE||DELETE
NCME063	Tart Nenas	50	UPDATE||DELETE
NCME064	Almond London	40	UPDATE||DELETE
NCME065	Samprit	55	UPDATE||DELETE
NCME066	Cornflakes Madu	35	UPDATE||DELETE
NCME067	Cornflakes Coklat	35	UPDATE||DELETE
NCME068	Arab	30	UPDATE||DELETE
NCME069	Rainbow	30	UPDATE||DELETE
NCME082	Sarang Semut	30	UPDATE||DELETE
NCME081	Walnut Delight	35	UPDATE||DELETE
NCME083	Chocolate Chip	40	UPDATE||DELETE
NCME0102	Oat Coklat	30	UPDATE||DELETE

Figure 4.22: The new data has been deleted from the Menu table

4.4 Conclusion

This chapter has discussed on the project design which includes Entity Relationship Diagram and Business Rules for the Conceptual Design, Data Dictionary and Normalization for the Logical Design, Selection of DBMS for the Physical Design, and the Graphical User Interface (GUI) Design. By discussing these different designs into details, the knowledge and understanding of this system will be gained and this will be very useful during the implementation part.

In the next chapter which is Chapter 5, the implementation of this system while it is being developed will be discussed into details by explaining the system database environment setup (installation steps), description of database with the database object creation, and database implementation which consists of Data Definition Language, Data Control Language and Data Manipulation Language.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

As stated by Rouse (n.d), in an information technology (IT) context, software or hardware implementation encompasses all the post-sale processes involved in something operating properly in its environment, including analysing requirements, installation, configuration, customization, running, testing, systems integrations, user training, delivery and making necessary changes. The word ‘deployment’ is sometimes used to mean the same thing.

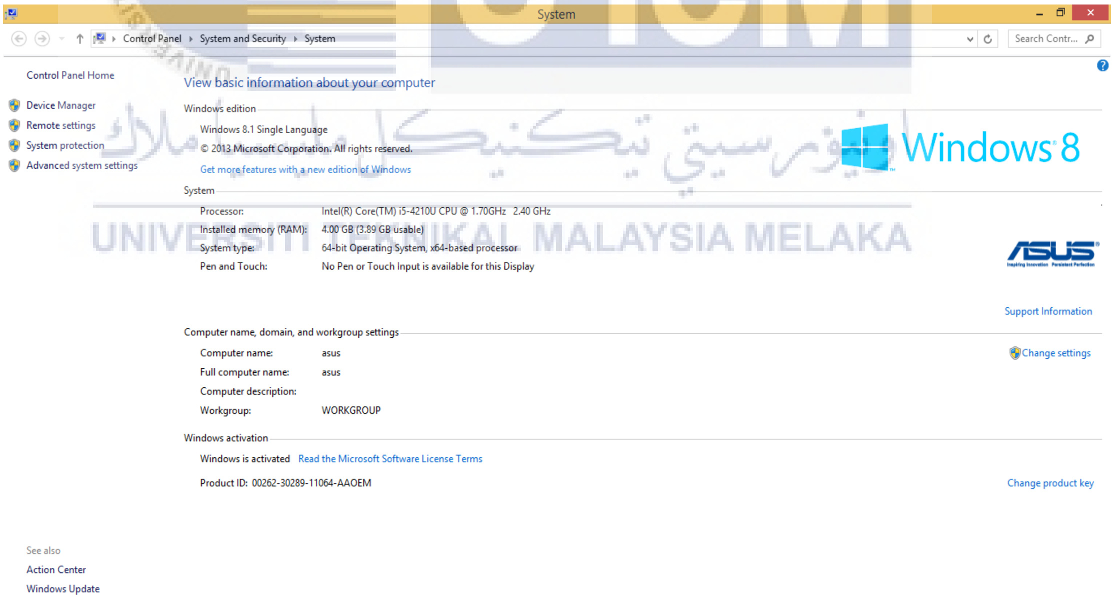
Software or hardware implementations should always be designed with the end user in mind and the implementation process usually benefits from user involvement and their support. If users participate in the design and implementation of the system, ideally it will serve the business objectives more accurately and reflect to its priorities and the ways in which users prefer to work. Their involvement in the process also makes them more receptive to changes that need to be implemented because they have undergone first-hand experience of what the system comprises.

In this project, the implementation phase activity will involve the software development environment setup and the database implementation. This includes the installation steps for the database chosen and description of all database creation, database object, stored procedures, and triggers. The expected output after this completion of implementation phase is that it will produce a useful system that will benefit end users well.

5.2 Software Development Environment Setup

Installation of the Operating System and Database Software:

i. Installation of database software ORACLE 11g in Windows 8.1 is successful.



View basic information about your computer

Windows edition

Windows 8.1 Single Language
© 2013 Microsoft Corporation. All rights reserved.
Get more features with a new edition of Windows

System

Processor: Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz
Installed memory (RAM): 4.00 GB (3.89 GB usable)
System type: 64-bit Operating System, x64-based processor
Pen and Touch: No Pen or Touch Input is available for this Display

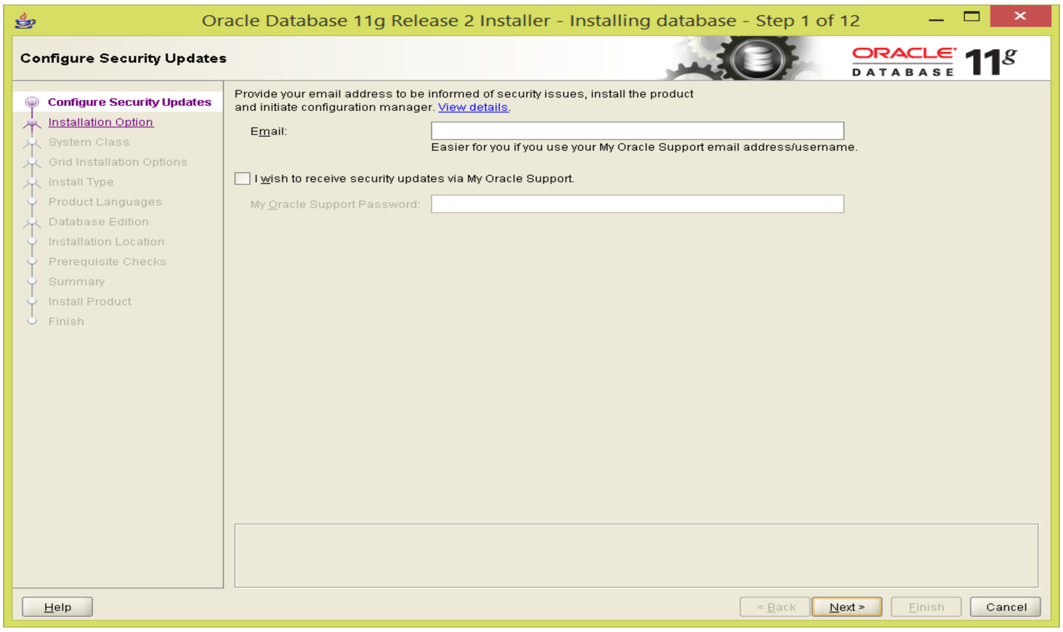
Computer name, domain, and workgroup settings

Computer name: asus
Full computer name: asus
Computer description:
Workgroup: WORKGROUP

Windows activation

Windows is activated Read the Microsoft Software License Terms
Product ID: 00262-30289-11064-AAOEM

Figure 5.1: Windows 8.1 Operating System



Oracle Database 11g Release 2 Installer - Installing database - Step 1 of 12

Configure Security Updates

Provide your email address to be informed of security issues, install the product and initiate configuration manager. View details.

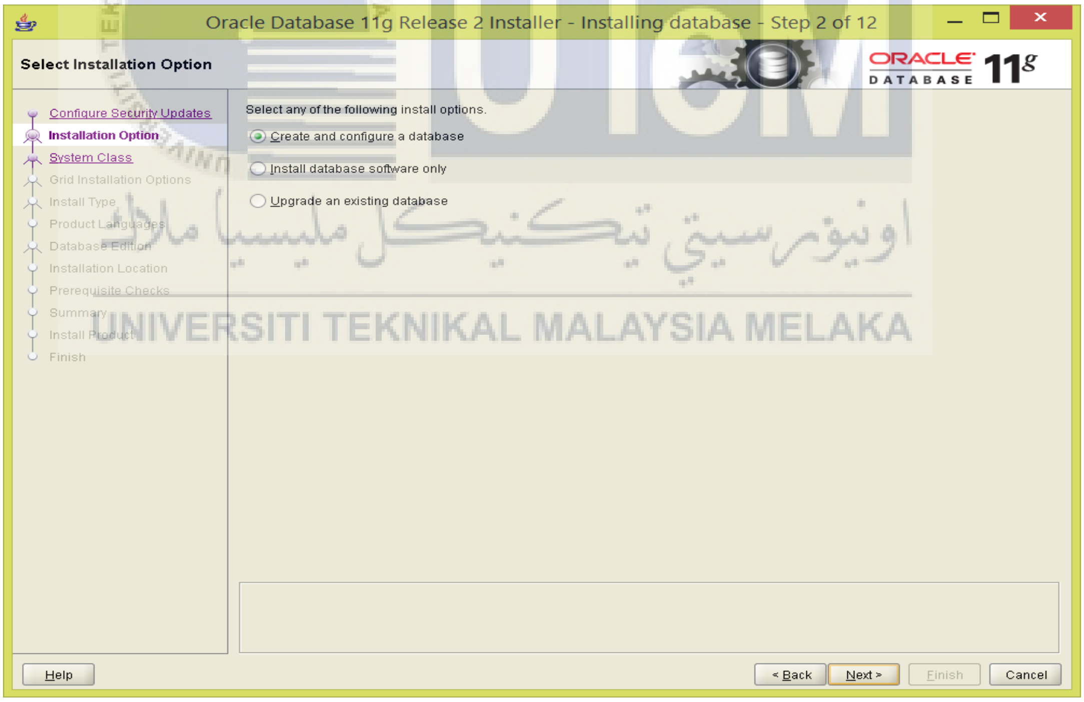
Email:

Easier for you if you use your My Oracle Support email address/username.

□ I wish to receive security updates via My Oracle Support.

My Oracle Support Password:

Figure 5.2: Installation of Oracle 11g database



Oracle Database 11g Release 2 Installer - Installing database - Step 2 of 12

Select Installation Option

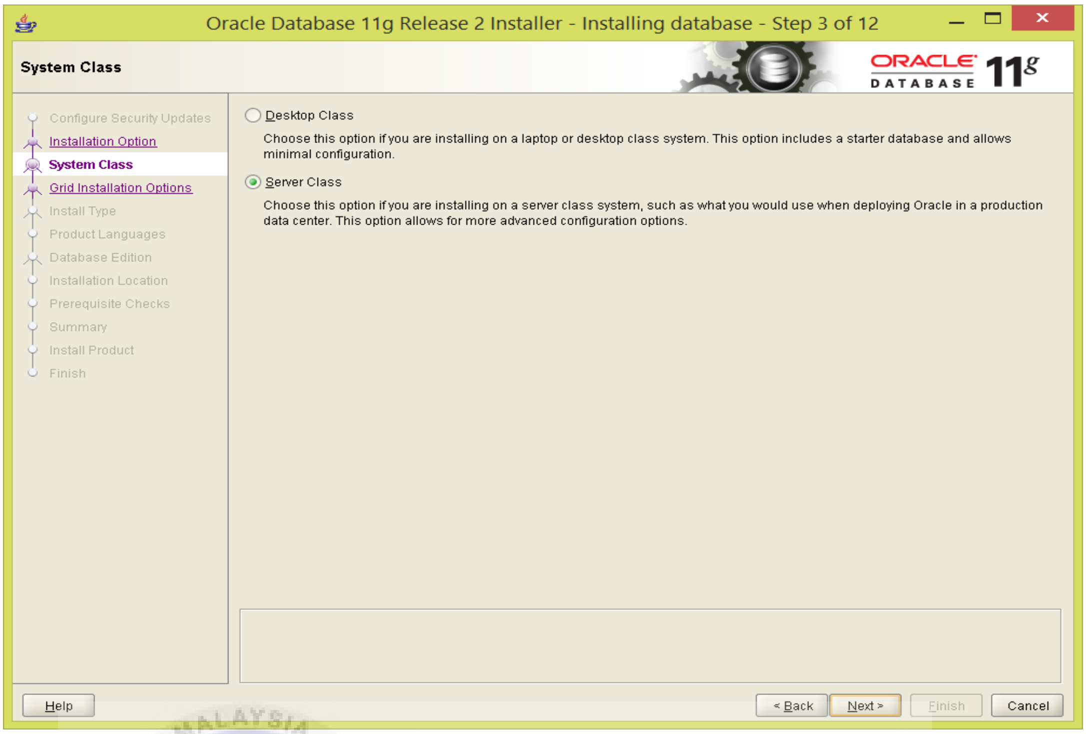
Select any of the following install options.

● Create and configure a database

○ Install database software only

○ Upgrade an existing database

Figure 5.3: Installation of Oracle 11g database



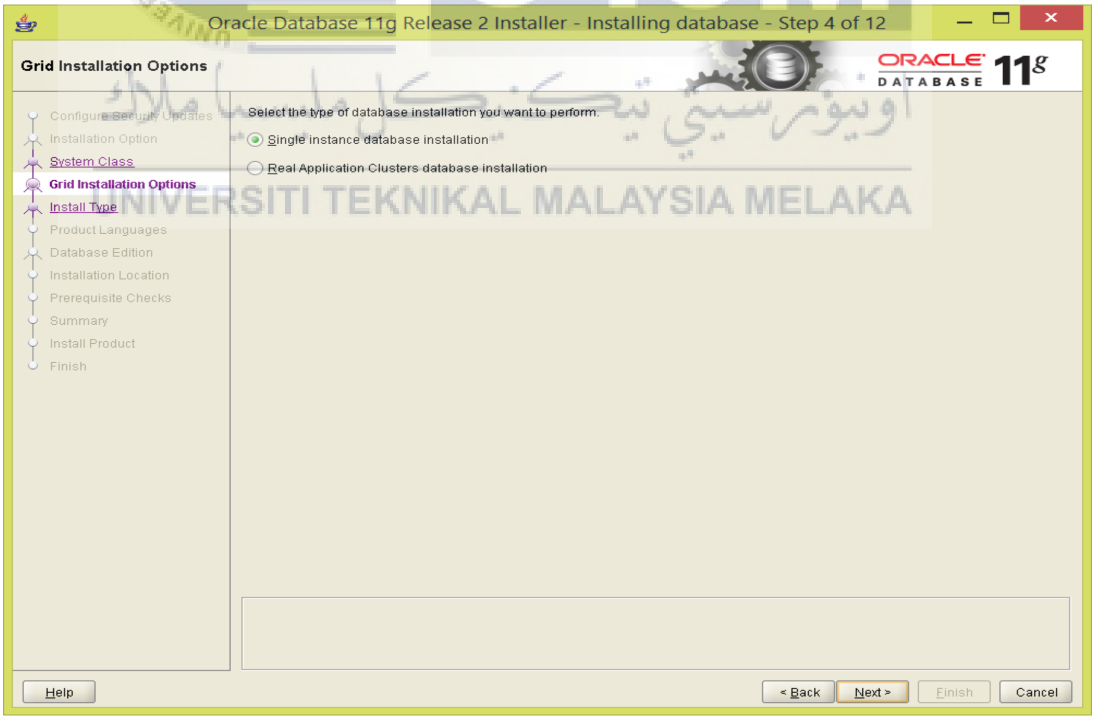
○ Desktop Class

Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and allows minimal configuration.

● Server Class

Choose this option if you are installing on a server class system, such as what you would use when deploying Oracle in a production data center. This option allows for more advanced configuration options.

Figure 5.4: Installation of Oracle 11g database

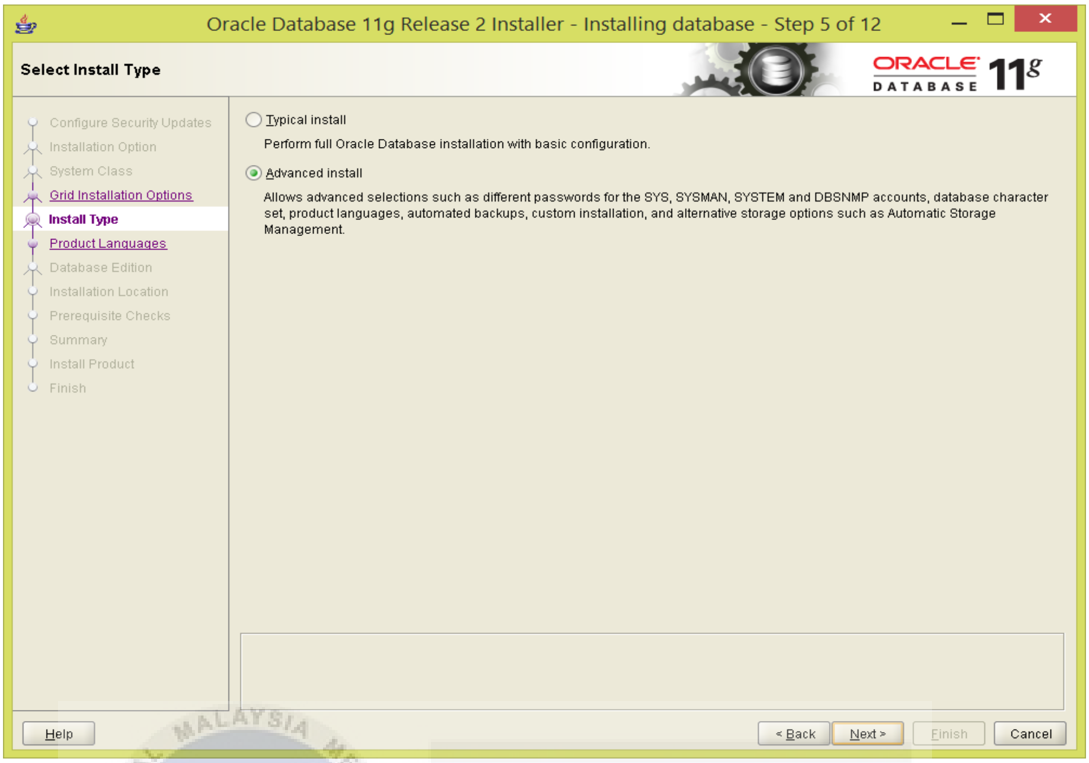


Select the type of database installation you want to perform.

● Single instance database installation

○ Real Application Clusters database installation

Figure 5.5: Installation of Oracle 11g database



Select Install Type

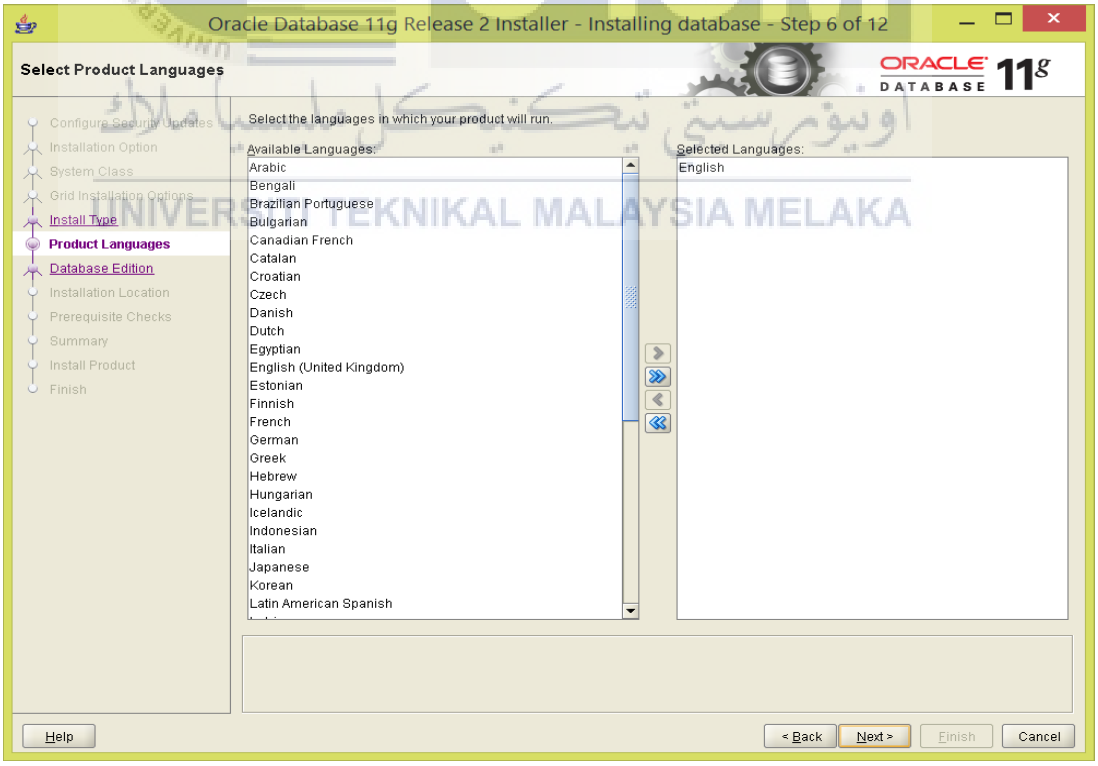
○ Typical install

Perform full Oracle Database installation with basic configuration.

● Advanced install

Allows advanced selections such as different passwords for the SYS, SYSMAN, SYSTEM and DBSNMP accounts, database character set, product languages, automated backups, custom installation, and alternative storage options such as Automatic Storage Management.

Figure 5.6: Installation of Oracle 11g database



Select Product Languages

Select the languages in which your product will run.

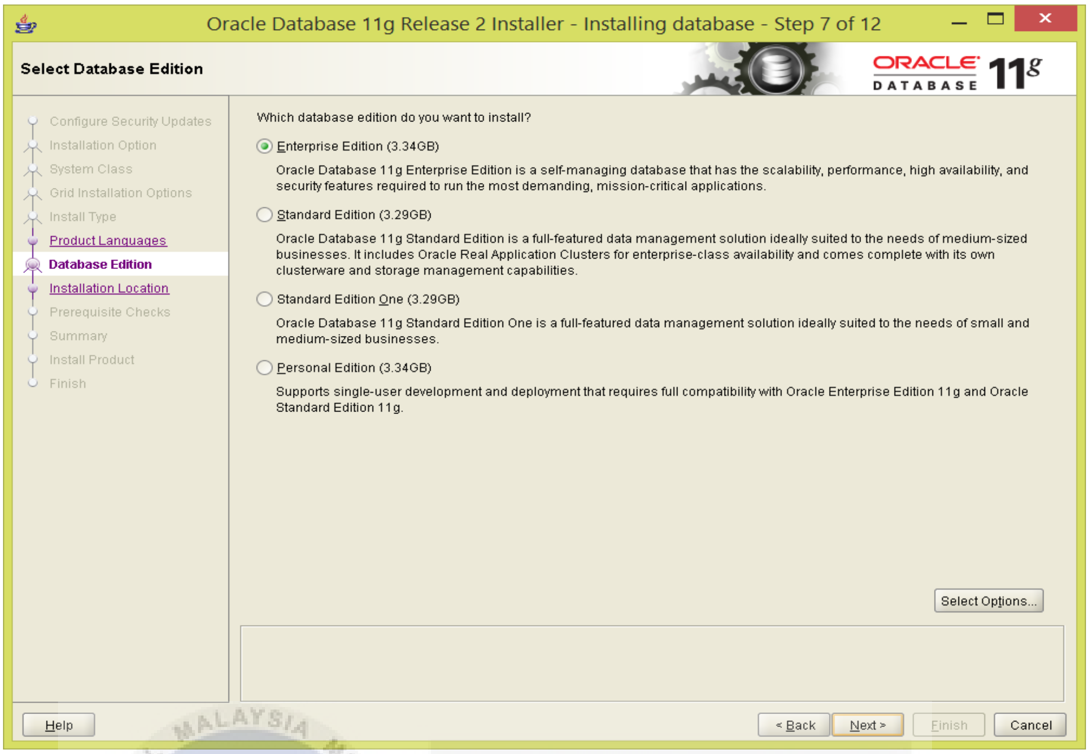
Available Languages:

Arabic
Bengali
Brazilian Portuguese
Bulgarian
Canadian French
Catalan
Croatian
Czech
Danish
Dutch
Egyptian
English (United Kingdom)
Estonian
Finnish
French
German
Greek
Hebrew
Hungarian
Icelandic
Indonesian
Italian
Japanese
Korean
Latin American Spanish

Selected Languages:

English

Figure 5.7: Installation of Oracle 11g database



Oracle Database 11g Release 2 Installer - Installing database - Step 7 of 12

Select Database Edition

Which database edition do you want to install?

● Enterprise Edition (3.34GB)

Oracle Database 11g Enterprise Edition is a self-managing database that has the scalability, performance, high availability, and security features required to run the most demanding, mission-critical applications.

○ Standard Edition (3.29GB)

Oracle Database 11g Standard Edition is a full-featured data management solution ideally suited to the needs of medium-sized businesses. It includes Oracle Real Application Clusters for enterprise-class availability and comes complete with its own clusterware and storage management capabilities.

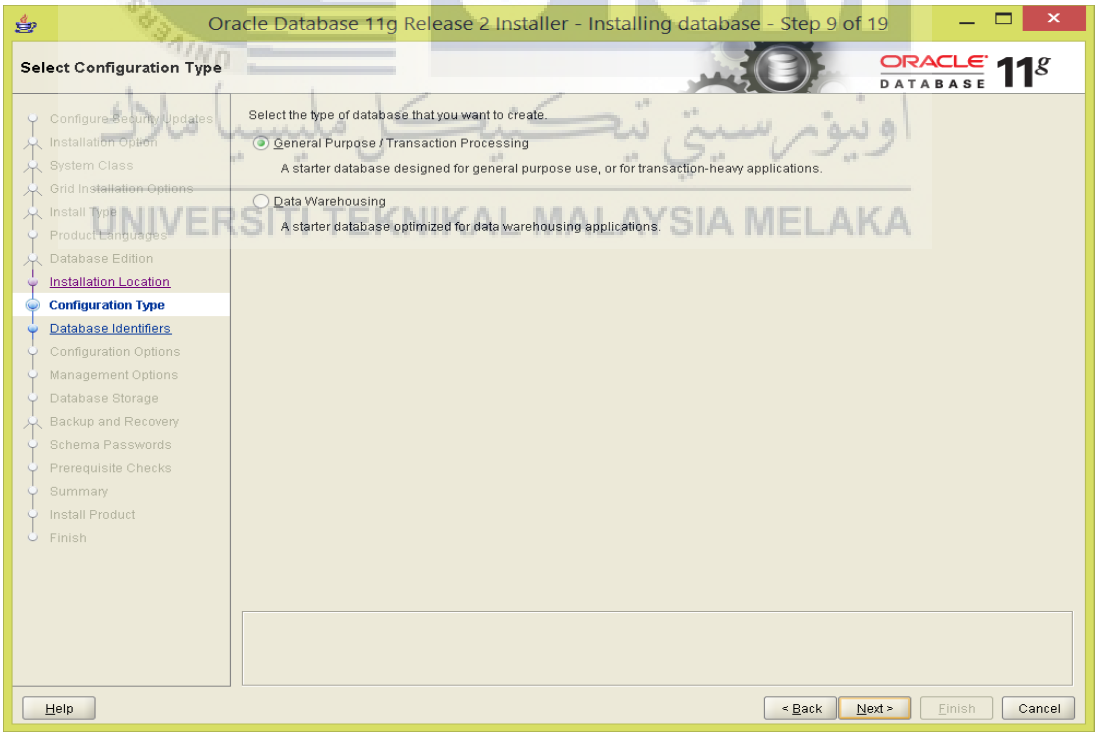
○ Standard Edition One (3.29GB)

Oracle Database 11g Standard Edition One is a full-featured data management solution ideally suited to the needs of small and medium-sized businesses.

○ Personal Edition (3.34GB)

Supports single-user development and deployment that requires full compatibility with Oracle Enterprise Edition 11g and Oracle Standard Edition 11g.

Figure 5.8: Installation of Oracle 11g database



Oracle Database 11g Release 2 Installer - Installing database - Step 9 of 19

Select Configuration Type

Select the type of database that you want to create.

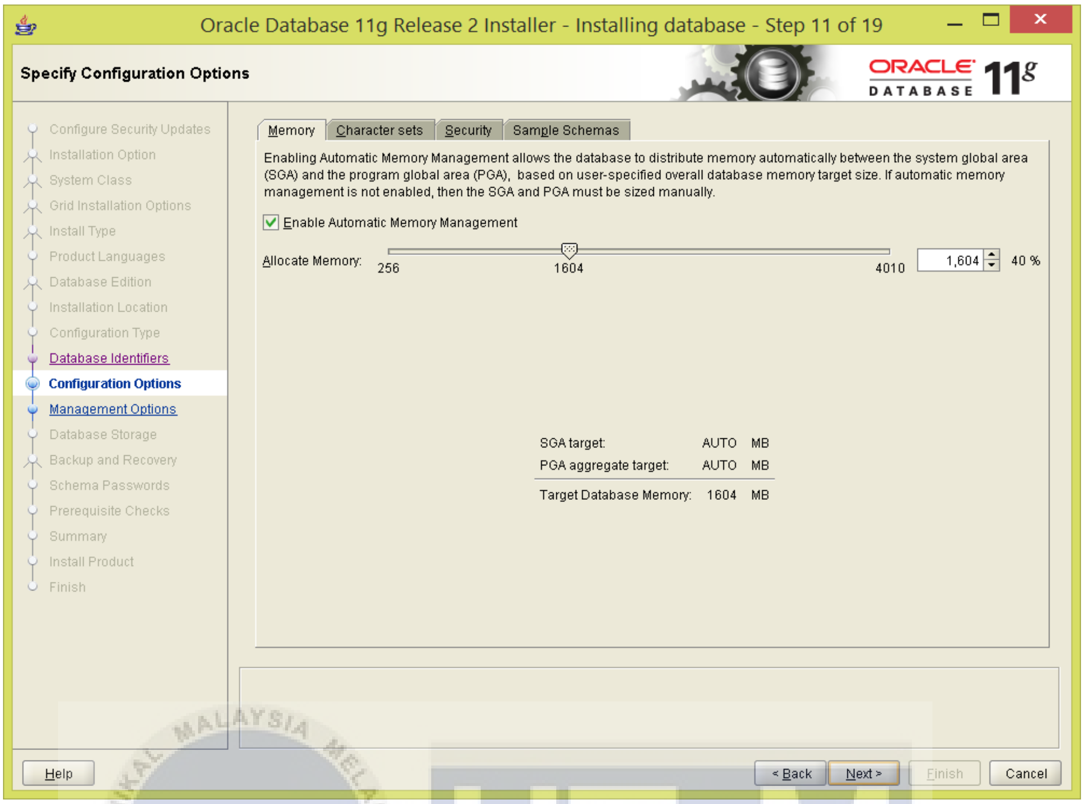
● General Purpose / Transaction Processing

A starter database designed for general purpose use, or for transaction-heavy applications.

○ Data Warehousing

A starter database optimized for data warehousing applications.

Figure 5.9: Installation of Oracle 11g database



Specify Configuration Options

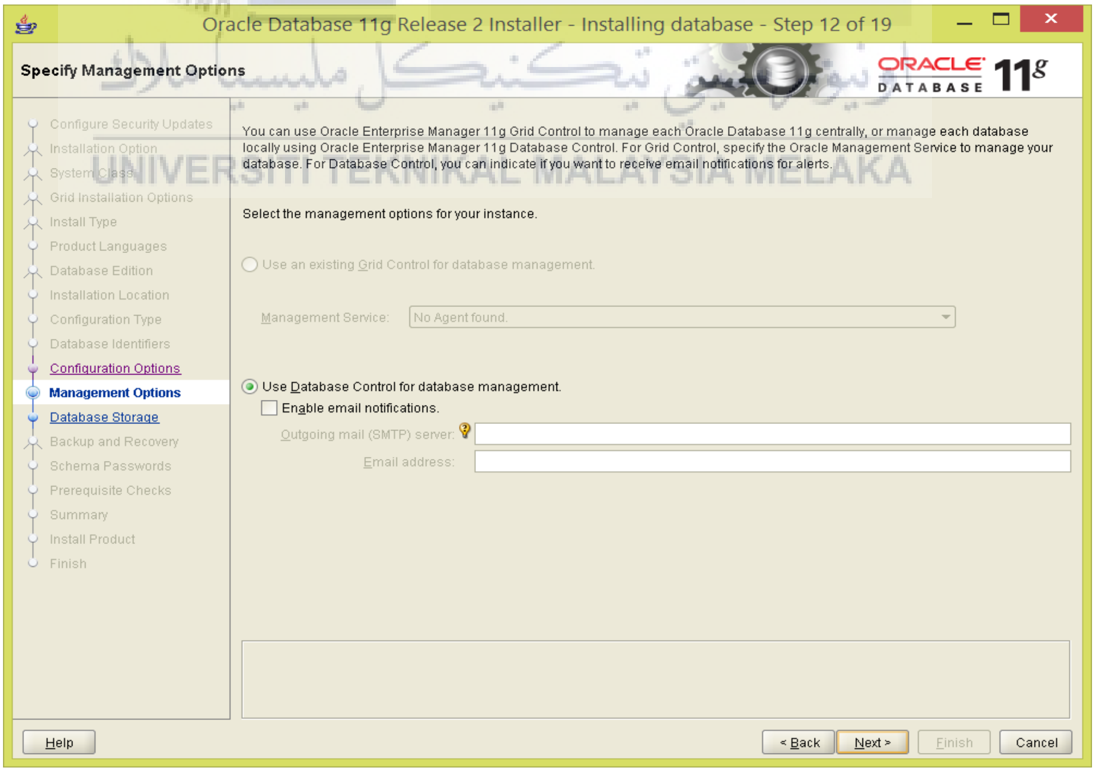
Enabling Automatic Memory Management allows the database to distribute memory automatically between the system global area (SGA) and the program global area (PGA), based on user-specified overall database memory target size. If automatic memory management is not enabled, then the SGA and PGA must be sized manually.

☑ Enable Automatic Memory Management

Allocate Memory: 1,604 40 %

SGA target:	AUTO	MB
PGA aggregate target:	AUTO	MB
Target Database Memory:	1604	MB

Figure 5.10: Installation of Oracle 11g database



Specify Management Options

You can use Oracle Enterprise Manager 11g Grid Control to manage each Oracle Database 11g centrally, or manage each database locally using Oracle Enterprise Manager 11g Database Control. For Grid Control, specify the Oracle Management Service to manage your database. For Database Control, you can indicate if you want to receive email notifications for alerts.

Select the management options for your instance.

○ Use an existing Grid Control for database management.

Management Service: No Agent found.

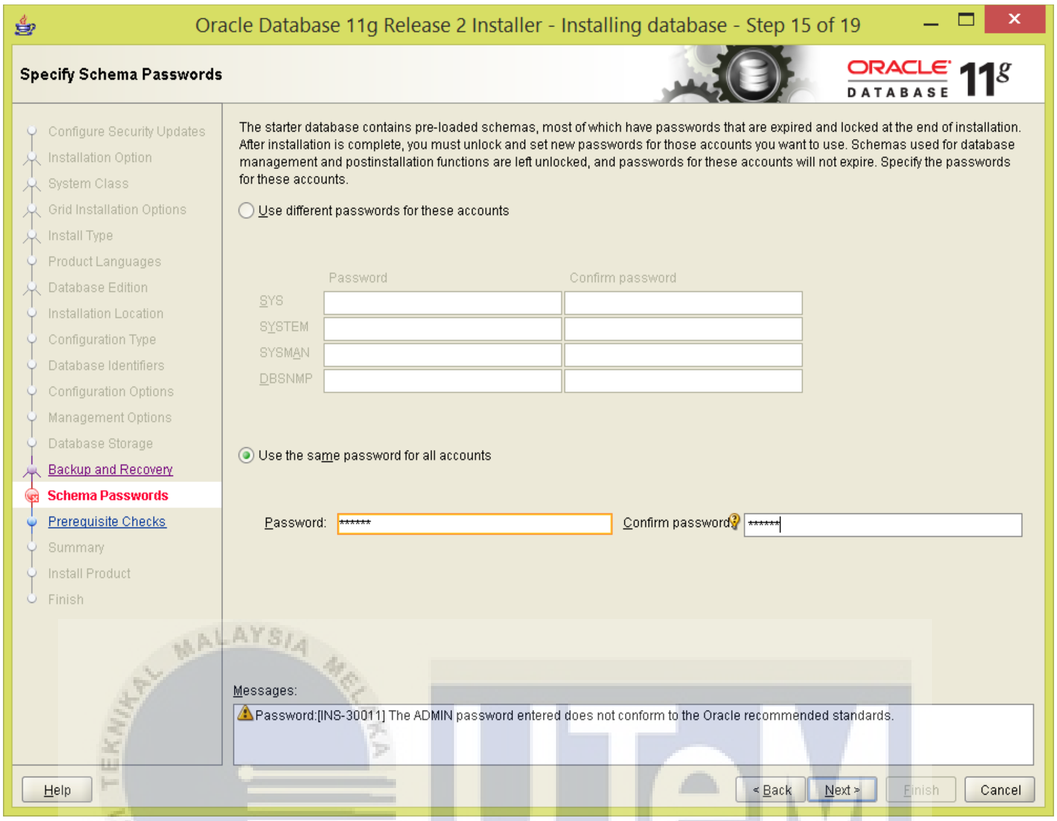
● Use Database Control for database management.

□ Enable email notifications.

Outgoing mail (SMTP) server:

Email address:

Figure 5.11: Installation of Oracle 11g database



Oracle Database 11g Release 2 Installer - Installing database - Step 15 of 19

Specify Schema Passwords

The starter database contains pre-loaded schemas, most of which have passwords that are expired and locked at the end of installation. After installation is complete, you must unlock and set new passwords for those accounts you want to use. Schemas used for database management and postinstallation functions are left unlocked, and passwords for these accounts will not expire. Specify the passwords for these accounts.

○ Use different passwords for these accounts

	Password	Confirm password
SYS		
SYSTEM		
SYSMAN		
DBSNMP		

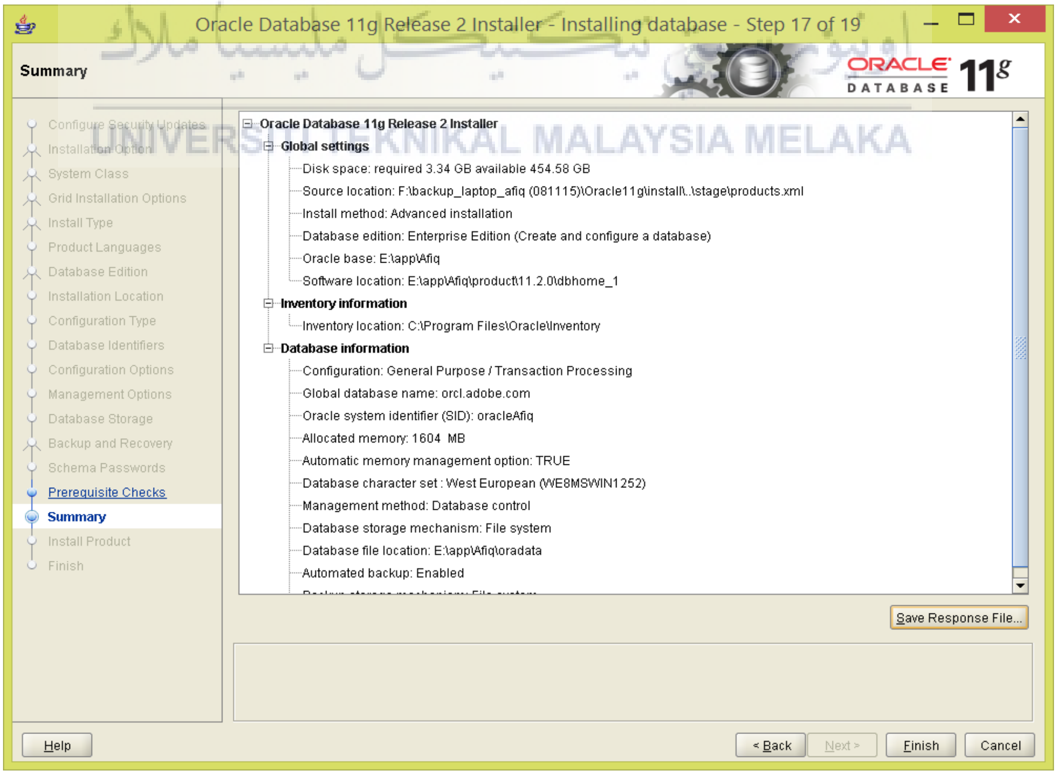
● Use the same password for all accounts

Password: ****** Confirm password: ******

Messages:

Password:[INS-30011] The ADMIN password entered does not conform to the Oracle recommended standards.

Figure 5.12: Installation of Oracle 11g database

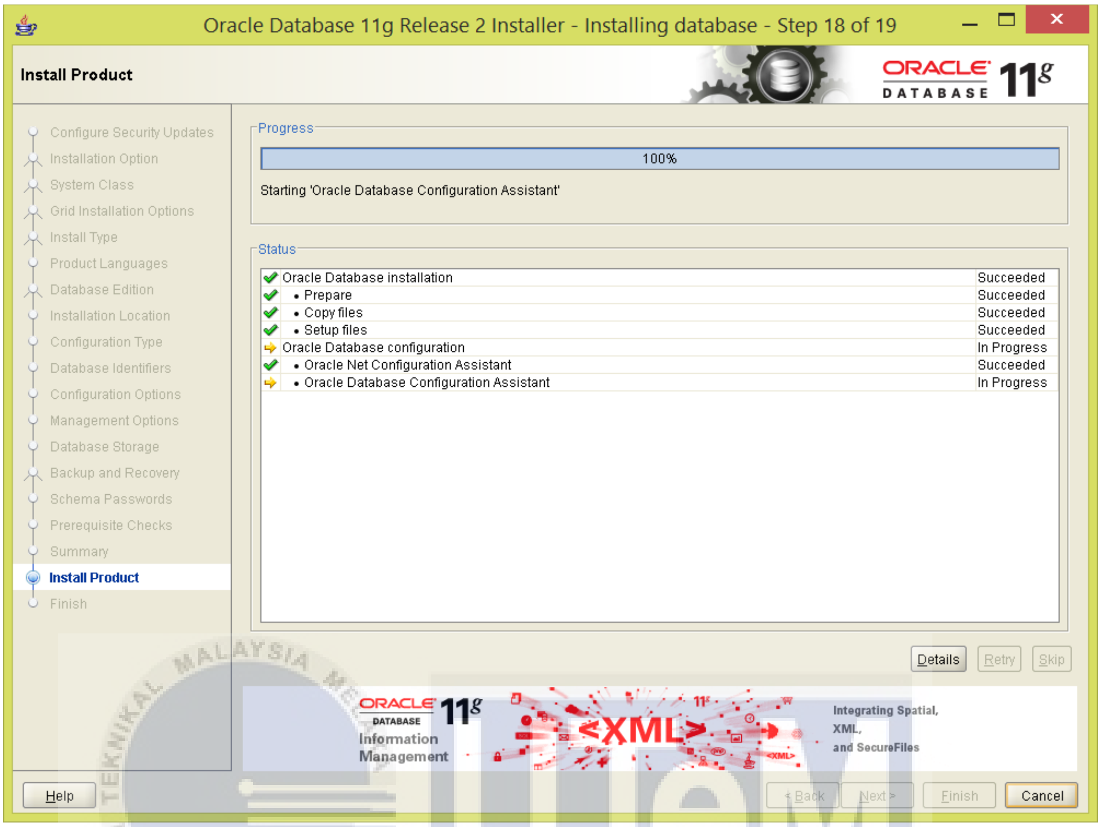


Oracle Database 11g Release 2 Installer - Installing database - Step 17 of 19

Summary

```
Oracle Database 11g Release 2 Installer
    Global settings
        Disk space: required 3.34 GB available 454.58 GB
        Source location: F:\backup_laptop_afiq (081115)\Oracle11g\install\..\stage\products.xml
        Install method: Advanced installation
        Database edition: Enterprise Edition (Create and configure a database)
        Oracle base: E:\app\Afiq
        Software location: E:\app\Afiq\product\11.2.0\dbhome_1
    Inventory information
        Inventory location: C:\Program Files\Oracle\Inventory
    Database information
        Configuration: General Purpose / Transaction Processing
        Global database name: orcl.adobe.com
        Oracle system identifier (SID): oracleAfiq
        Allocated memory: 1604  MB
        Automatic memory management option: TRUE
        Database character set : West European (WE8MSWIN1252)
        Management method: Database control
        Database storage mechanism: File system
        Database file location: E:\app\Afiq\oradata
        Automated backup: Enabled
        Backup storage mechanism: File system
```

Figure 5.13: Installation of Oracle 11g database



Oracle Database 11g Release 2 Installer - Installing database - Step 18 of 19

Install Product

Progress

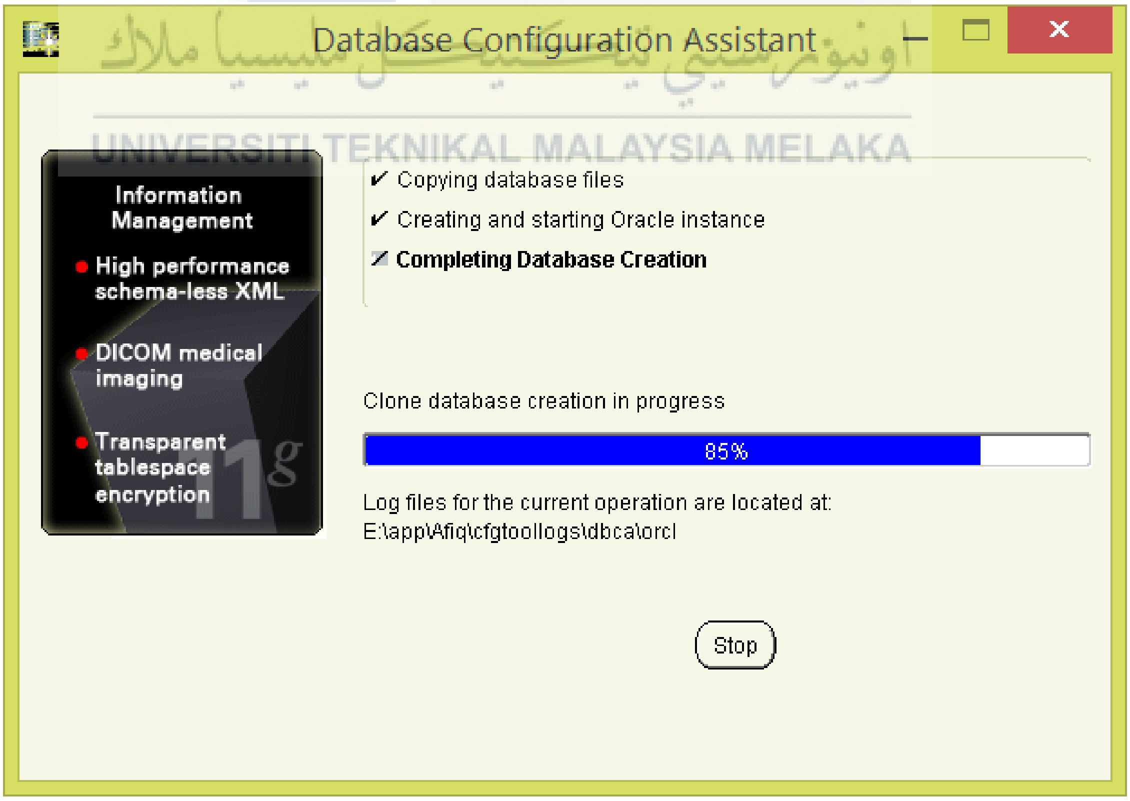
100%

Starting 'Oracle Database Configuration Assistant'

Status

✓ Oracle Database installation	Succeeded
✓ • Prepare	Succeeded
✓ • Copy files	Succeeded
✓ • Setup files	Succeeded
➜ Oracle Database configuration	In Progress
✓ • Oracle Net Configuration Assistant	Succeeded
➜ • Oracle Database Configuration Assistant	In Progress

Figure 5.14: Installation of Oracle 11g database



Database Configuration Assistant

Information Management
High performance schema-less XML
DICOM medical imaging
Transparent tablespace encryption
11g
Stop

✔ Copying database files

✔ Creating and starting Oracle instance

Completing Database Creation

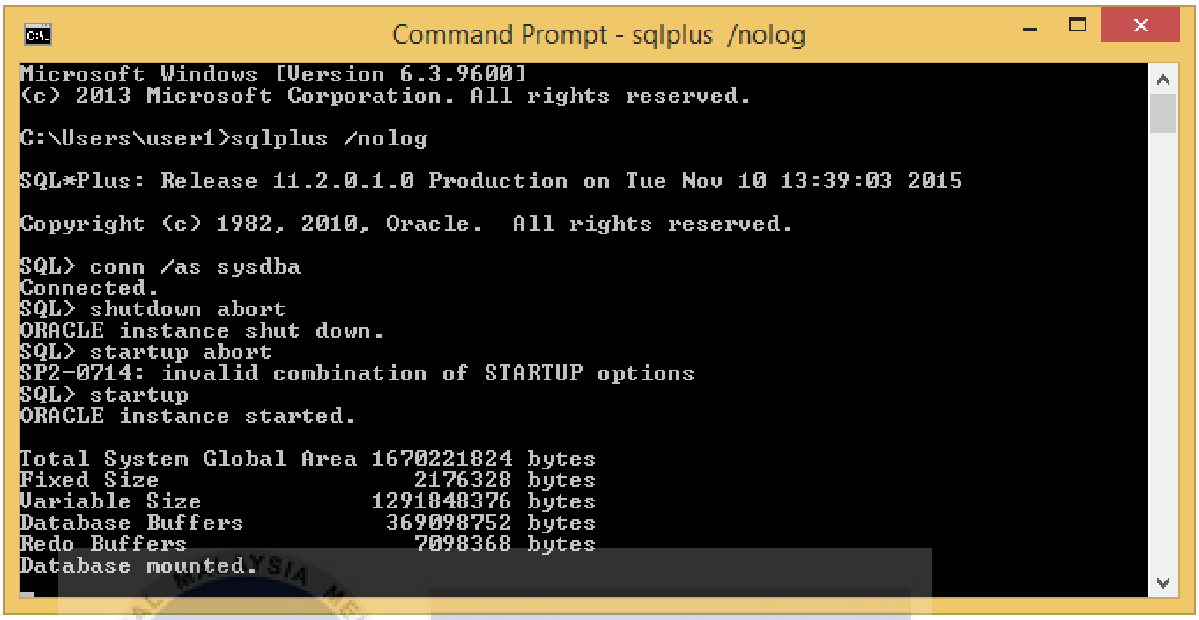
Clone database creation in progress

85%

Log files for the current operation are located at:
E:\app\Afiq\cfgtoollogs\dbca\orcl

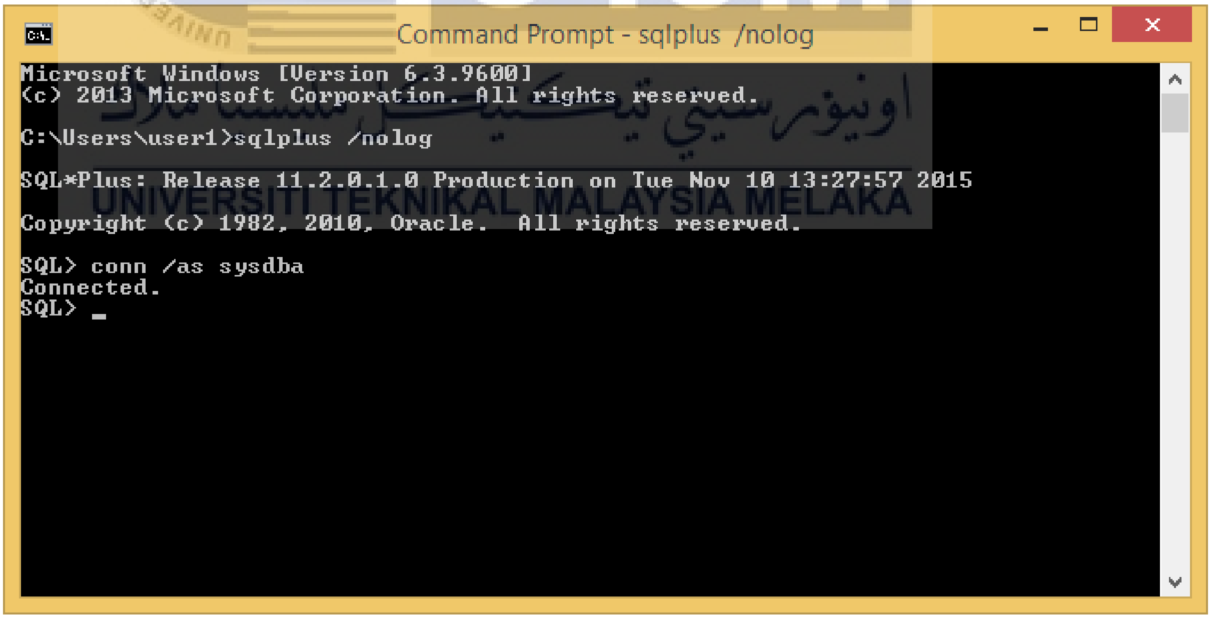
Figure 5.15: Installation of Oracle 11g database

ii. Service / Instance Up and login for the Database Administrator (DBA) is successful.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\user1>sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Tue Nov 10 13:39:03 2015

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn /as sysdba
Connected.
SQL> shutdown abort
ORACLE instance shut down.
SQL> startup abort
SP2-0714: invalid combination of STARTUP options
SQL> startup
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size                  2176328 bytes
Variable Size            1291848376 bytes
Database Buffers          369098752 bytes
Redo Buffers                7098368 bytes
Database mounted.
_
```

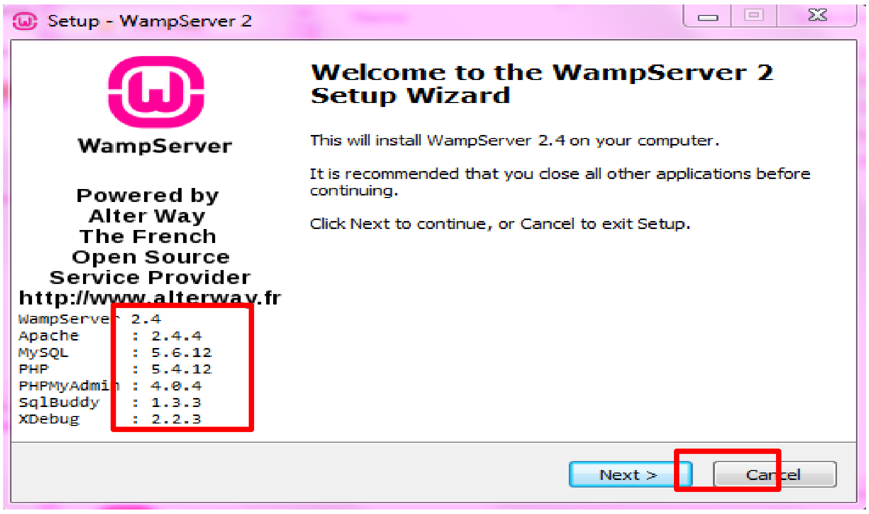
Figure 5.16: Instance start-up and shut-down of Oracle 11g database



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\user1>sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Tue Nov 10 13:27:57 2015

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn /as sysdba
Connected.
SQL> _
```

Figure 5.17: Admin login of Oracle 11g database

iii. Web Server Installation (WAMP)



Setup - WampServer 2

Welcome to the WampServer 2 Setup Wizard

This will install WampServer 2.4 on your computer.

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

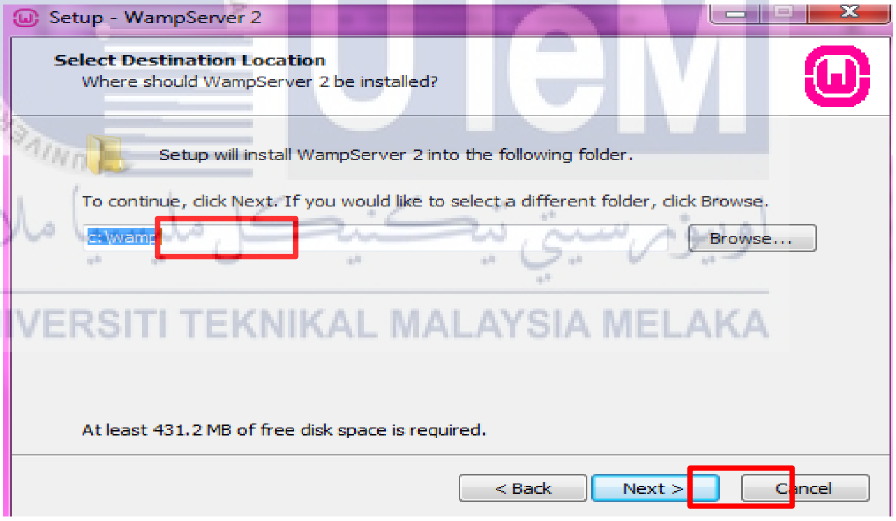
Click Next to continue, or Cancel to exit Setup.

WampServer

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

```
WampServer 2.4
Apache     : 2.4.4
MySQL      : 5.6.12
PHP        : 5.4.12
PHPMyAdmin : 4.0.4
SqlBuddy   : 1.3.3
XDebug     : 2.2.3
```

Figure 5.18: Run WAMP server exe file and click ‘Next’



Setup - WampServer 2

Select Destination Location

Where should WampServer 2 be installed?

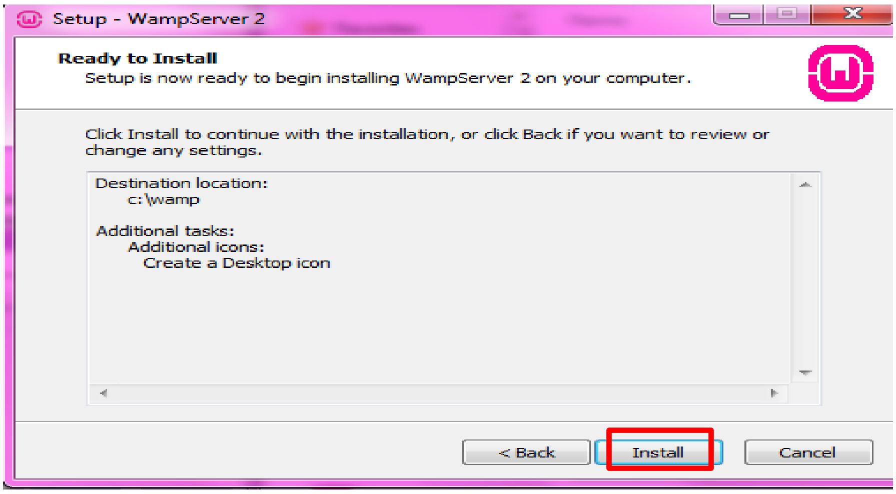
Setup will install WampServer 2 into the following folder.

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

c:\wamp

At least 431.2 MB of free disk space is required.

Figure 5.19: Agree to the license of WAMP Server before selecting the installation destination on windows machine. Install WAMP Server in C: driver and click ‘Next’



Setup - WampServer 2

Ready to Install

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

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

```
Destination location:
      c:\wamp

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

Figure 5.20: Allow the access to Apache server at SMTP server configuration dialog. Then, specify the SMTP server and the address mail that will be used by PHP when using the function mail ().

Complete the installation of WAMP Server 2.4 dialog with Apache, MySQL and PhpMyAdmin on the computer. Click button “Finish” to start WAMP Server together with other services.

5.3 Database Implementation

Database creation and database objects

Customer Table

```
create table customer(
customer_ID varchar2(20),
customer_fName varchar2(20) not null,
customer_lName varchar2(20) not null,
customer_Email varchar2(50) not null,
customer_telNo number not null,
customer_Address varchar2(100) not
null,
customer_Username varchar2(20) not
null,
customer_Password varchar2(20) not
null,
stamp number not null,
constraint customer_pk primary key
(customer_ID)
 );
```

Order Table

```
create table order(
order_ID varchar2(20),
grand_Payment number(7,2),
customer_ID varchar2(20) not null,
constraint order_pk primary key
(order_ID),
constraint order_fk foreign key
(customer_ID) references
customer_adp(customer_ID)
);
```

Menu Table

```
create table menu(
menu_ID varchar2(20),
menu_Name varchar2(50) not null,
menu_Price number(7,2) not null,
menu_Ingredient varchar2(100) not null,
constraint order_menu_pk primary key
(menu_ID)
);
```

Order_menu Table

```
create table order_menu(
order_menuID varchar2(20),
order_menuQuantity number not null,
menu_ID varchar2(20) not null,
menu_Name varchar2(50) not null,
menu_Price number(7,2) not null,
order_ID varchar2(20),
order_Date DATE not null,
action varchar2(20) not null,
customer_ID varchar2(20) not null,
constraint order_menu_pk1 primary key
(order_MenuID),
constraint order_menu_fk2 foreign key
(menu_ID) references menu_adp(menu_ID),
constraint order_menu_fk3 foreign key
(order_ID) references
order_adp(order_ID),
constraint order_menu_fk3 foreign key
(customer_ID) references
customer_adp(customer_ID)
 );
```

Payment Table

```
create table payment(
payment_ID varchar2(20),
payment_Date DATE not null,
payment_Evidence BLOB,
customer_ID varchar2(20) not null,
staff_ID varchar2(20),
status varchar2(20) not null,
constraint payment_pk primary key
(payment_ID),
constraint payment_fk foreign key
(customer_ID) references
customer_adp(customer_ID),
constraint payment_fk2 foreign key
(staff_ID) references
staff_adp(staff_ID)
);
```

Staff Table

```
create table staff(
staff_ID varchar2(20),
staff_fName varchar2(20) not null,
staff_lName varchar2(20) not null,
staff_Email varchar2(50) not null,
staff_telNo number not null,
staff_Address varchar2(100) not
null,
staff_Username varchar2(20) not
null,
staff_Password varchar2(20) not
null,
status number not null,
constraint staff_pk primary key
(staff_ID)
);
```

There are a total of six types of triggers and three types of stored procedures implemented in this NOOCM system. The use of each part of triggers and stored procedures implemented are being explained in the tables below:

Table 5.1: Types of triggers implemented with its usage

Types of triggers	Implemented in table	Use of trigger
Before insert	Customer, Staff, Payment, Order, Order_menu, Menu	Generating primary key automatically
After insert	Order_menu	To count total stamp received by customers each time they are making orders
Before update	Order_menu	To check if the order has been approved by staff, then customers cannot do any changes to it
After update	Payment	To capture the staff ID to indicates which staff has approved the payment made by customers
Before delete	Order_menu	To check if the order has been approved by staff, then customers cannot do any changes to it
After delete	Order_menu	To transfer all data into audit table after deletion

Table 5.2: Types of stored procedures implemented with its usage

Types of procedures	Implemented in table	Use of procedures
Stored procedure (DML)	Customer, Staff, Payment, Order, Menu	For viewing, inserting, updating, and deleting data
Stored procedure (Simple query)	Order_menu + Customer	To show orders which have been made by certain customers
Stored procedure (Complex query)	1) Order_menu + Menu + Customer	To show the total payment that customer needs to pay by grouping it according to the order date and customer ID
	2) Order_menu + Menu	To generate a monthly sales report by grouping the total sales according to the month
	3) Order_menu + Menu	To generate total sales sorted by each menu id and menu name

Trigger Before Insert

1) Auto generate for primary key in table Customer

```
create or replace TRIGGER CUSTOMER_ADP_TRIG
BEFORE INSERT ON CUSTOMER_ADP
FOR EACH ROW
BEGIN
SELECT 'NCC0'||CUSTOMER_ADP_SEQ.NEXTVAL INTO
:NEW.CUSTOMER_ID FROM DUAL;
END;
```

2) Auto generate for primary key in table Staff

```
create or replace TRIGGER STAFF_ADP_TRIG
BEFORE INSERT ON STAFF_ADP
FOR EACH ROW
BEGIN
SELECT 'NCS0'||STAFF_ADP_SEQ.NEXTVAL INTO
:NEW.STAFF_ID FROM DUAL;
END;
```

3) Auto generate for primary key in table Payment

```
create or replace TRIGGER PAYMENT_ADP_TRIG
BEFORE INSERT ON PAYMENT_ADP
FOR EACH ROW
BEGIN
SELECT 'NCOP0'||PAYMENT_ADP_SEQ.NEXTVAL INTO
:NEW.PAYMENT_ID FROM DUAL;
:NEW.PAYMENT_DATE:=to_char(sysdate,'DD-MON-YY');
END;
```

4) Auto generate for primary key in table Order

```
create or replace TRIGGER ORDER_ADP_TRIG
BEFORE INSERT ON ORDER_ADP
FOR EACH ROW
BEGIN
SELECT 'NCOR0'||ORDER_ADP_SEQ.NEXTVAL INTO
:NEW.ORDER_ID FROM DUAL;
END;
```

5) Auto generate for primary key in table Order_menu

```
create or replace TRIGGER ORDER_MENU_ADP_TRIG
BEFORE INSERT ON ORDER_MENU_ADP
FOR EACH ROW
BEGIN
SELECT 'NCOM0'||ORDER_MENU_ADP_SEQ.NEXTVAL INTO
:NEW.ORDER_MENUID FROM DUAL;
select menu_ID, menu_Price
into :new.menu_ID, :new.menu_Price
from menu_adp
where menu_Name = :new.menu_Name;
:new.order_Date := to_char(sysdate, 'DD-MON-YY');
:new.action := 'Pending';
END;
```

6) Auto generate for primary key in table Menu

```
create or replace TRIGGER MENU_ADP_TRIG
BEFORE INSERT ON MENU_ADP
FOR EACH ROW
BEGIN
SELECT 'NCME0'||MENU_ADP_SEQ.NEXTVAL INTO
:NEW.MENU_ID FROM DUAL;
END;
```

Trigger After Insert

```
create or replace trigger after_insert_order_menu
after insert on order_menu_adp
for each row
begin
update customer_adp set stamp = nvl(stamp,0)+1
where customer_id = :new.customer_id;
end;
```

Trigger Before Update

```
create or replace trigger before_update_order_menu
before update on order_menu_adp
for each row
begin
if(:new.action = 'Approved') then
raise_application_error(-20502,'Sorry, your order has been
approved by system Admin. You cannot make any changes.');
end if;
end;
```

Trigger After Update

```
create or replace TRIGGER after_update_payment
AFTER UPDATE OF STATUS ON payment_adp
FOR EACH ROW
DECLARE
v_staffid varchar2(20);
v_username varchar2(20);
BEGIN

SELECT staff_ID, username
into v_staffid, v_username
from staff_adp
where username = v_username;

update payment_adp
set staff_ID = v_staffid
where payment_ID = :old.payment_ID;
END;
```

Trigger Before Delete

```
create or replace trigger before_delete_order_menu
before delete on order_menu_adp
for each row
begin
if(:new.action = 'Approved') then
raise_application_error(-20502,'Sorry, your order has been
approved by system Admin. You cannot make any changes.');
end if;
end;
```

Trigger After Delete

```
create or replace trigger after_delete_order_menu
after delete on order_menu_adp
for each row
begin
INSERT INTO ORDER_MENU_AUDIT(ORDER_MENUID,
ORDER_MENUQUANTITY, MENU_ID, MENU_NAME, ORDER_DATE,
CUSTOMER_ID)
VALUES (:OLD.ORDER_MENUID, :OLD.ORDER_MENUQUANTITY,
:OLD.MENU_ID, :OLD.MENU_NAME, :OLD.ORDER_DATE,
:OLD.CUSTOMER_ID);
END;
```

Store Procedure (DML)

```
create or replace PROCEDURE INSERTcustomer(
customer_fName CUSTOMER_ADP.customer_fname%TYPE,
customer_lName CUSTOMER_ADP.customer_lname%TYPE,
customer_Email CUSTOMER_ADP.customer_email%TYPE,
customer_telNo CUSTOMER_ADP.customer_telno%TYPE,
customer_Address CUSTOMER_ADP.customer_address%TYPE,
customer_Username CUSTOMER_ADP.username%TYPE,
customer_Password CUSTOMER_ADP.password%TYPE)
IS
BEGIN
INSERT INTO CUSTOMER_ADP(CUSTOMER_FNAME, CUSTOMER_LNAME,
CUSTOMER_EMAIL, CUSTOMER_TELNO, CUSTOMER_ADDRESS, USERNAME,
PASSWORD)
VALUES (customer_fname, customer_lname, customer_email,
customer_telno, customer_address, customer_username,
customer_password);
COMMIT;
END;
```

```
create or replace PROCEDURE UPDATEcustomer(
v_customer_id CUSTOMER_ADP.customer_id%TYPE,
v_customer_first_name CUSTOMER_ADP.customer_fname%TYPE,
v_customer_last_name CUSTOMER_ADP.customer_lname%TYPE,
v_customer_email CUSTOMER_ADP.customer_email%TYPE,
v_customer_tel_no CUSTOMER_ADP.customer_telno%TYPE,
v_customer_address CUSTOMER_ADP.customer_address%TYPE,
v_customer_username CUSTOMER_ADP.username%TYPE,
v_customer_password CUSTOMER_ADP.password%TYPE)
IS
BEGIN
UPDATE CUSTOMER_ADP
SET customer_fname = v_customer_first_name, customer_lname
= v_customer_last_name, customer_email = v_customer_email,
customer_telno = v_customer_tel_no, customer_address =
v_customer_address, username = v_customer_username,
password = v_customer_password

WHERE customer_id =v_customer_id;
COMMIT;
END;
```

```
create or replace PROCEDURE DELETEcustomer(
v_customer_ID CUSTOMER_ADP.customer_ID%TYPE,
v_customer_first_name CUSTOMER_ADP.customer_fname%TYPE,
v_customer_last_name CUSTOMER_ADP.customer_lname%TYPE,
v_customer_email CUSTOMER_ADP.customer_email%TYPE,
v_customer_tel_no CUSTOMER_ADP.customer_telno%TYPE,
v_customer_address CUSTOMER_ADP.customer_address%TYPE,
v_customer_stamp CUSTOMER_ADP.stamp%TYPE
)
IS
BEGIN
DELETE CUSTOMER_ADP
WHERE customer_ID =v_customer_ID;
COMMIT;
END;
```

```
create or replace PROCEDURE DELETEcustomer1(
v_customer_ID CUSTOMER_ADP.customer_ID%TYPE,
v_customer_first_name CUSTOMER_ADP.customer_fname%TYPE,
v_customer_last_name CUSTOMER_ADP.customer_lname%TYPE,
v_customer_email CUSTOMER_ADP.customer_email%TYPE,
v_customer_tel_no CUSTOMER_ADP.customer_telno%TYPE,
v_customer_address CUSTOMER_ADP.customer_address%TYPE,
v_customer_stamp CUSTOMER_ADP.stamp%TYPE
)
IS
BEGIN
DELETE CUSTOMER_ADP
WHERE customer_ID =v_customer_ID;
COMMIT;
END;
```

```
create or replace procedure custselfdetails(
cust_name customer_adp.username%TYPE,rc out sys_refcursor)
as
begin
 open  rc for select * from customer_adp where
cust_name=username;
end;
```

```
create or replace procedure show_customer(myrc out
sys_refcursor) as
begin
    open myrc for select * from customer_adp;
end;
```

```
create or replace PROCEDURE INSERTstaff(
staff_fName STAFF_ADP.staff_fname%TYPE,
staff_lName STAFF_ADP.staff_lname%TYPE,
staff_Email STAFF_ADP.staff_email%TYPE,
staff_telNo STAFF_ADP.staff_telno%TYPE,
staff_Address STAFF_ADP.staff_address%TYPE,
staff_Username STAFF_ADP.username%TYPE,
staff_Password STAFF_ADP.password%TYPE,
staff_Status STAFF_ADP.status%TYPE)
IS
BEGIN
INSERT INTO STAFF_ADP(STAFF_FNAME, STAFF_LNAME,
STAFF_EMAIL, STAFF_TELNO, STAFF_ADDRESS, USERNAME,
PASSWORD, STATUS)
VALUES (staff_fname, staff_lname, staff_email, staff_telno,
staff_address, staff_username, staff_password,
staff_status);
COMMIT;
END;
```

```
create or replace PROCEDURE UPDATEstaff(
v_staff_id STAFF_ADP.staff_id%TYPE,
v_staff_first_name STAFF_ADP.staff_fname%TYPE,
v_staff_last_name STAFF_ADP.staff_lname%TYPE,
v_staff_email STAFF_ADP.staff_email%TYPE,
v_staff_tel_no STAFF_ADP.staff_telno%TYPE,
v_staff_address STAFF_ADP.staff_address%TYPE,
v_staff_username STAFF_ADP.username%TYPE,
v_staff_password STAFF_ADP.password%TYPE)
IS
BEGIN
UPDATE STAFF_ADP
SET staff_fname = v_staff_first_name, staff_lname =
v_staff_last_name, staff_email = v_staff_email, staff_telno
= v_staff_tel_no, staff_address = v_staff_address, username
= v_staff_username, password = v_staff_password
WHERE staff_id =v_staff_id;
COMMIT;
END;
```

```
create or replace PROCEDURE DELETEstaff(
v_staff_ID STAFF_ADP.staff_ID%TYPE,
v_staff_first_name STAFF_ADP.staff_fname%TYPE,
v_staff_last_name STAFF_ADP.staff_lname%TYPE,
v_staff_email STAFF_ADP.staff_email%TYPE,
v_staff_tel_no STAFF_ADP.staff_telno%TYPE,
v_staff_address STAFF_ADP.staff_address%TYPE,
v_staff_username STAFF_ADP.username%TYPE,
v_staff_password STAFF_ADP.password%TYPE,
v_staff_status STAFF_ADP.status%TYPE
)
IS
BEGIN
DELETE STAFF_ADP
WHERE staff_ID =v_staff_ID;
COMMIT;
END;
```

```
create or replace procedure show_staff(myrc out
sys_refcursor) as
begin
    open myrc for select * from staff_adp;
end;
```

```
create or replace PROCEDURE UPDATEstaffADMIN(
v_staff_id STAFF_ADP.staff_id%TYPE,
v_staff_first_name STAFF_ADP.staff_fname%TYPE,
v_staff_last_name STAFF_ADP.staff_lname%TYPE,
v_staff_email STAFF_ADP.staff_email%TYPE,
v_staff_tel_no STAFF_ADP.staff_telno%TYPE,
v_staff_address STAFF_ADP.staff_address%TYPE,
v_staff_username STAFF_ADP.username%TYPE,
v_staff_password STAFF_ADP.password%TYPE,
v_staff_status STAFF_ADP.status%TYPE)
IS
BEGIN
UPDATE STAFF_ADP
SET staff_fname = v_staff_first_name, staff_lname =
v_staff_last_name, staff_email = v_staff_email, staff_telno
= v_staff_tel_no, staff_address = v_staff_address, username
= v_staff_username, password = v_staff_password, status =
v_staff_status
WHERE staff_id =v_staff_id;
COMMIT;
END;
```

```
create or replace procedure staffselfdetails(
staff_name staff_adp.username%TYPE,rc out sys_refcursor)
as
begin
 open  rc for select * from staff_adp where
staff_name=username;
end;
```

```
create or replace PROCEDURE INSERTordermenu(
order_menuquantity ORDER_MENU_ADP.order_menuquantity%TYPE,
menu_name ORDER_MENU_ADP.menu_name%TYPE,
customer_ID ORDER_MENU_ADP.customer_id%TYPE)
IS
BEGIN
INSERT INTO
ORDER_MENU_ADP(ORDER_MENUQUANTITY,MENU_NAME,CUSTOMER_ID)
VALUES (order_menuquantity,menu_name,customer_id);
COMMIT;
END;
```

```
create or replace PROCEDURE DELETEordermenu(
v_order_menu_id ORDER_MENU_ADP.order_menuid%TYPE,
v_order_menu_quantity
ORDER_MENU_ADP.order_menuquantity%TYPE,
v_menu_name ORDER_MENU_ADP.menu_name%TYPE,
v_order_date ORDER_MENU_ADP.order_date%TYPE,
v_customer_id ORDER_MENU_ADP.customer_id%TYPE,
v_status ORDER_MENU_ADP.action%TYPE)
IS
BEGIN
DELETE ORDER_MENU_ADP
WHERE order_menuID =v_order_menu_ID;
COMMIT;
END;
```

```
create or replace PROCEDURE UPDATEordermenu(
v_order_menu_id ORDER_MENU_ADP.order_menuid%TYPE,
v_order_menu_quantity
ORDER_MENU_ADP.order_menuquantity%TYPE,
v_menu_id ORDER_MENU_ADP.menu_id%TYPE,
v_menu_name ORDER_MENU_ADP.menu_name%TYPE,
v_order_date ORDER_MENU_ADP.order_date%TYPE,
v_customer_id ORDER_MENU_ADP.customer_id%TYPE,
v_status ORDER_MENU_ADP.action%TYPE)
IS
BEGIN
UPDATE ORDER_MENU_ADP
SET order_menuquantity = v_order_menu_quantity, menu_id =
v_menu_id, menu_name = v_menu_name, order_date =
v_order_date, customer_id = v_customer_id, action =
v_status
WHERE order_menuid =v_order_menu_id;
COMMIT;
END;
```

```
create or replace PROCEDURE UPDATEordermenu_forcust(
v_order_menu_id ORDER_MENU_ADP.order_menuid%TYPE,
v_order_menu_quantity
ORDER_MENU_ADP.order_menuquantity%TYPE,
v_menu_name ORDER_MENU_ADP.menu_name%TYPE,
v_order_date ORDER_MENU_ADP.order_date%TYPE,
v_customer_id ORDER_MENU_ADP.customer_id%TYPE,
v_status ORDER_MENU_ADP.action%TYPE)
IS
BEGIN
UPDATE ORDER_MENU_ADP
SET order_menuquantity = v_order_menu_quantity, menu_name =
v_menu_name, order_date = v_order_date, customer_id =
v_customer_id, action = v_status
WHERE order_menuid =v_order_menu_id;
COMMIT;
END;
```

```
create or replace PROCEDURE DELETEcustorderdetails(
v_order_menuid ORDER_MENU_ADP.order_menuid%TYPE,
v_order_menuquantity
ORDER_MENU_ADP.order_menuquantity%TYPE,
v_menu_id ORDER_MENU_ADP.menu_id%TYPE,
v_menu_name ORDER_MENU_ADP.menu_name%TYPE,
v_order_date ORDER_MENU_ADP.order_date%TYPE,
v_customer_id ORDER_MENU_ADP.customer_id%TYPE)
IS
BEGIN
DELETE ORDER_MENU_ADP
WHERE order_menuid =v_order_menuid;
COMMIT;
END;
```

```
create or replace procedure show_custorderdetail(myrc out
sys_refcursor) as
begin
    open myrc for select * from order_menu_adp;
end;
```

```
create or replace PROCEDURE INSERTmenu(
menu_name MENU_ADP.menu_name%TYPE,
menu_price MENU_ADP.menu_price%TYPE)
IS
BEGIN
INSERT INTO MENU_ADP(MENU_NAME, MENU_PRICE)
VALUES (menu_name, menu_price);
COMMIT;
END;
```

```
create or replace PROCEDURE UPDATEmenu(
v_menu_id MENU_ADP.menu_id%TYPE,
v_menu_name MENU_ADP.menu_name%TYPE,
v_menu_price MENU_ADP.menu_price%TYPE)
IS
BEGIN
UPDATE MENU_ADP
SET menu_name = v_menu_name, menu_price = v_menu_price
WHERE menu_id =v_menu_id;
COMMIT;
END;
```

```
create or replace PROCEDURE DELETEmenu(
v_menu_id MENU_ADP.menu_id%TYPE,
v_menu_name MENU_ADP.menu_name%TYPE,
v_menu_price MENU_ADP.menu_price%TYPE)
IS
BEGIN
DELETE MENU_ADP
WHERE menu_id =v_menu_id;
COMMIT;
END;
```

```
create or replace procedure show_menu(myrc out
sys_refcursor) as
begin
    open myrc for select * from menu_adp;
end;
```

```
create or replace PROCEDURE UPDATEpaymentreceipt(
v_payment_id PAYMENT_ADP.payment_id%TYPE,
v_payment_date PAYMENT_ADP.payment_date%TYPE,
v_customer_id PAYMENT_ADP.customer_id%TYPE,
v_status PAYMENT_ADP.status%TYPE)
IS
BEGIN
UPDATE PAYMENT_ADP
SET payment_date = v_payment_date, customer_id =
v_customer_id, status = v_status
WHERE payment_id =v_payment_id;
COMMIT;
END;
```

Stored Procedure (Simple Query - Join of at least 2 tables)

```
create or replace procedure show_order_menu(myrc out
sys_refcursor, v_customer_id in varchar2) as
begin
open myrc for  select O.order_menuid, O.order_menuquantity,
O.menu_name, O.order_date, C.customer_id, O.action
from order_menu_adp O, customer_adp C
where O.customer_id=C.customer_id
and C.customer_id=v_customer_id
order by O.order_date desc;
end;
```

Stored Procedure (Complex Query – Join of at least 2
Tables with grouping, aggregate, outer joins)

```
create or replace procedure show_total_payment(myrc out
sys_refcursor, v_customer_id in varchar2) as
begin
open myrc for select C.customer_id, O.order_date,
SUM(O.order_MenuQuantity*O.menu_Price) as TOTAL_PAYMENT
from order_Menu_adp O, customer_adp C
where C.customer_id = O.customer_id
and C.customer_id = v_customer_id
group by order_date, C.customer_id
order by O.order_date desc;
end;
```

```
create or replace procedure total_order_report (myrc out
sys_refcursor) as
begin
open myrc for select TO_CHAR(O.order_date,'mm-yyyy'),
SUM(O.order_MenuQuantity*M.menu_Price) as TOTAL_SALES
from order_Menu_adp O, menu_adp M
where O.menu_id = M.menu_id
group by TO_CHAR(O.order_date,'mm-yyyy')
order by TO_CHAR(O.order_date,'mm-yyyy') desc;
end;
```

```
create or replace procedure total_menu_sales (myrc out
sys_refcursor) as
begin
open myrc for select M.menu_name as MENU_NAME,
SUM(O.order_menuQuantity) as TOTAL_MENU_SALES
from menu_adp M, order_menu_adp O
where M.menu_id = O.menu_id
group by M.menu_name;
end;
```

5.4 Conclusion

As a conclusion, Ingram (n.d) stated that this implementation phase is very important to produce strategic planning process because it can reveal new issues and challenges that planners may not have anticipated, ultimately resulting in more refined strategies, products and processes. Putting projects into action is the vital step linking the planning stage to the final achievement of strategic goals, which makes project implementation of the utmost importance to strategic planning efforts.

For the next chapter (Chapter 6), testing will be carried out to test the functionality of this system into details. Both back end (to test the database object created) and front end (to test the integration of system interface with the back end) will be done together with the functionality test.

CHAPTER VI

TESTING

6.1 Introduction

Testing shows the presence, not the absence of bugs (Edsger W. Dijkstra, n.d). Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product to meet the business and technical requirements that guided its design and development.

Altogether, there are five essentials for software testing:

1) A test strategy that tells what types of testing and the amount of testing will work best at finding the defects that are lurking in the software
2) A testing plan of the actual testing tasks that will need to be executed to carry out the strategy implemented
3) Test cases that have been prepared in advance in the form of detailed examples which will use to check either the software will actually meet its requirements
4) Test data consisting of both input test data and database test data to use while executing the test cases, and

5) A test environment which will be used in carrying out the testing.

In this project, the testing strategies to be adopted are black-box and white-box testing. Black-box testing are performed using software interfaces to ensure that they work as expected while white-box testing looks inside the software that is being tested (coding) and uses that knowledge as part of the testing process. White-box testing requires internal knowledge of the system and programming skills.

6.2 Test Plan

Test plan is the project plan for the testing work to be done. Most of the test plans do not addressed into details. In this project test plan, it consists of test organization, test environment and test schedule. Test organization is to describe the person involved in the testing procedure while the test environment is to describe the environment of testing to be carried out, and to define hardware, firmware configurations, preparations and training prior to testing. Test schedule is to define how many cycles and duration of the test to be conducted.

6.2.1 Test Organization

Test organization represents the group of people who will be carrying out the test procedures. Good testing procedures will involve different type of people from different background because few perspectives could be produced due to their knowledge in Information Technology. In this NOOCM system, it will be tested by the software developer and a client.

Software developer is the person who is held responsible in developing this NOOCM system while the client is the person who is targeted to be the end user. Table 6.1 shows the list of tester and their responsibilities.

Table 6.1: List of tester and their responsibilities

Tester ID	Title / Position	Responsibilities
Tester 1	System Developer	Reviewing current systems, presenting ideas for system improvements, producing detailed specifications and writing the programme codes, testing the product in controlled and in real situations before going live, preparation of training manuals for users, and maintaining the systems once they are up and running.
Tester 2	Client	Act as the end user to obtain some feedbacks. Those feedbacks will be used in order to improve the system for future use.

6.2.2 Test Environment

A test environment is the testing setup that includes the software and hardware which helps to perform the test procedures in this NOOCM system. Testing procedure consists of each and every possible conditions and circumstances that might occur during transaction events in a system.

6.2.2.1 Environment Setup

The environment setup describes the environment of testing which will be carried out consistently throughout the system testing, the testing modules involve, and the software and hardware used.

Table 6.2: Environment of testing in NOOCM System

Environment Specification	Description
Operating System	Windows 8.1
Processor	Intel Core i5
Random Access Memory (RAM)	4 GB
Database	Oracle 11g
Server	Apache Web Server
Server-Scripting	PHP

Table 6.3: NOOCM System Testing Module

Testing Module	i. User Registration ii. Login (User Authentication) iii. Purchase Order iv. Payment v. Order History vi. Search Cookies vii. Import Order

Table 6.4: Software used in NOOCM System Testing

Software	i. Windows 8.1 ii. Oracle 11g (Full Edition Release 2) iii. WAMP Server (Apache) iv. Adobe Dreamweaver CS3 v. Microsoft Office Word 2010 vi. Google Chrome (Web Browser)

Table 6.5: Hardware used in NOOCM System Testing

Hardware	i. Laptop ii. Mouse iii. Keyboard iv. Printer v. External Disk vi. USB (Pen-Drive)

6.2.3 Test Schedule

Test schedule is the methodology of testing all the information collected in the length of time which has been set. The target behind test schedule is to ensure each and every testing activity by whom and when those have been perform. Since the term of undertaking headway for around 6 month, the datebook will go about as an assistant for developer to test on the accurate time which had been set. Table 5 underneath shows the test timetable for developer of NOOCM System.

Table 6.6: Test Schedule of NOOCM System Testing

Testing Module	Test Activity	Duration (Day(s))	Cycle (Time(s))
User Registration	Error handling test and integration test	1	7
Login (User Authentication)	Error handling test and integration test	1	7
Purchase Order	Error handling test and integration test	3	10
Payment	Error handling test and integration test	3	15
Order History	Error handling test and integration test	2	10
Search Cookies	Error handling test and integration test	2	10
Import Order	Error handling test and integration test	3	7

6.3 Test Strategy

Test strategy is planned to facilitate the testing process. Test strategy outlines the testing approach and everything else that surrounds it. It is different from the test plan, in the sense that a test strategy is only a sub set of the test plan. It is a hard core test document that is to an extent generic and static. In this system, the types of test strategy selected are the black-box and white-box testing.

Black-box testing (also known as functional testing) treats software under test as a black-box without knowing its internals. Tests are using software interfaces and trying to ensure that they work as expected. As long as functionality of interfaces remains unchanged, tests should pass even if internals are changed. Tester is aware of what the program should do but does not have the knowledge of how it does it.

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) looks inside the software that is being tested and uses that knowledge as part of the testing process. If, for example, exception is thrown under certain conditions, test might want to reproduce those conditions. White-box testing requires internal knowledge of the system and programming skills.

Table 6.7: Black-box VS White-box Testing

Black-box	White-box
Testing the application based on its behaviour	Testing the application by using internal structure (coding) of application
Also known as "Functional Testing"	Also known as "Structural Testing" or "Glass Box Testing"
Testers involve in this type of testing	Developer and testers will involve in this type of testing

6.3.1 Classes of Test

There are many classifications of test that could be carried out in the testing process. In this case, two types of test were chosen which are error handling test and integration test.

i. Error Handling Test

Error handling test was utilized to validate only correct and accurate data from the client (e.g. No null values are allowed in any form). Error message will popped up on the screen to inform user regarding on their wrong or null input.

ii. Integration Test

This test is to ensure that this system captures data into the database correctly based on what user has keyed in. If the process is successful, it shows that the system is well integrated with the database.

6.4 Test Design

There are two parts of test design which are test description and test data. Test description is the test case identification, test cases and expected result for each testing module which are being designed and documented. As for the test data, it includes the real life or synthetic data that will be selected.

6.4.1 Test Description

This part will explain on each test module by describing their test case description, testing type and expected result.

6.4.1.1 Test Module 1: User Registration

Table 6.8: Test Description of User Registration Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_01-1	Field is blank for first name, last name, email, telephone number, address, username, or password	Error Handling Test	'Please fill in the form completely' message will be popped up
NOOCMS_01-2	Valid input for each field	Unit testing / Integration	New user's information is successfully recorded into the database

6.4.1.2 Test Module 2: Login (User Authentication)

Table 6.9: Test Description of Login Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_02-1	Invalid username and password	Error Handling Test	'Username or Password is invalid' message will be popped up

NOOCMS_02-2	Valid input for each field	Unit testing / Integration	User can log into the system successfully

6.4.1.3 Test Module 3: Purchase Order

Table 6.10: Test Description of Purchase Order Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_03-1	Quantity field is blank	Error Handling Test	'Please fill in the quantity required' message will be popped up
NOOCMS_03-2	Valid input for each field	Unit testing / Integration	New order will be successfully recorded into the database

6.4.1.4 Test Module 4: Payment

Table 6.11: Test Description of Payment Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_04-1	No file chosen to be uploaded	Error Handling Test	'Couldn't insert the new product' error will be shown

NOOCMS_04-2	Valid input for image chosen	Unit testing / Integration	New payment will be successfully recorded into the database

6.4.1.5 Test Module 5: Order History

Table 6.12: Test Description of Order History Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_05-1	Order status is currently in 'Approved' state	Error Handling Test	'Order quantity cannot be updated because Admin has approved this order' message will be popped up
NOOCMS_05-2	Order status is currently in 'Pending' state	Unit Testing / Integration	Old order menu quantity will be replaced by the new order menu quantity chose by user through the system

6.4.1.6 Test Module 6: Search Cookies

Table 6.13: Test Description of Search Cookies Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_06-1	User enter keyword that does not exists	Error Handling Test	Nothing will be displayed (blank table)
NOOCMS_06-2	User enter keyword which exists	Unit Testing / Integration	All data that matches the keyword inserted will be displayed

6.4.1.7 Test Module 7: Import Order

Table 6.14: Test Description of Import Order Module

Test Case ID	Description	Testing Type	Expected Result
NOOCMS_07-1	No file chosen to be uploaded	Error Handling Test	'Filename cannot be empty' error message will be shown
NOOCMS_07-2	Invalid file is chosen to be uploaded	Error Handling Test	'No data found' error will be shown

NOOCMS_07-3	Valid file is chosen to be uploaded	Unit Testing / Integration	New order menu data from the file will be inserted into the database

6.4.2 Test Data

This part will explain on the real life or synthetic data that were selected to be tested.

6.4.2.1 Test Module 1: User Registration

Table 6.15: Test Data of User Registration Module

Test Data ID	TD_01-1	TD_01-2
Test Case ID	NOOCMS_01-1	NOOCMS_01-2
User registration form	User does not fill up the fields completely First name: Razak Last name: Email: razak@yahoo.com Telephone No: 0122820170 Address: No 2, Jalan Setia 5, Taman Sentosa, 41200 Klang, Selangor Username: Razak Password: razak123	User fills up the fields completely First name: Razak Last name: Hussein Email: razak@yahoo.com Telephone No: 0122820170 Address: No 2, Jalan Setia 5, Taman Sentosa, 41200 Klang, Selangor Username: Razak Password: razak123

Test Data Result	New user registration fail due to insufficient of data during registration	New user's information is successfully being recorded into the database

6.4.2.2 Test Module 2: Login (User Authentication)

Table 6.16: Test Data of Login Module

Test Data ID	TD_02-1	TD_02-2
Test Case ID	NOOCMS_02-1	NOOCMS_02-2
Username	Zulaikha	Zulaikha
Password	zuzu11	zuzu111
Test Data Result	Login fail, due to invalid username or password	User logged in successfully

6.4.2.3 Test Module 3: Purchase Order

Table 6.17: Test Data of Purchase Order Module

Test Data ID	TD_03-1	TD_03-2
Test Case ID	NOOCMS_03-1	NOOCMS_03-2
Purchase Order Form	User does not fill up the fields completely Quantity: Menu name: Mazola Customer ID: NCC063	User fills up the fields completely Quantity: 3 Menu name: Mazola Customer ID: NCC063
Test Data Result	New order detail fail to be recorded due to insufficient of data during ordering	New order's information is successfully being recorded into the database

6.4.2.4 Test Module 4: Payment

Table 6.18: Test Data of Payment Module

Test Data ID	TD_04-1	TD_04-2
Test Case ID	NOOCMS_04-1	NOOCMS_04-2
Payment Form	User does not choose any image file to be uploaded Image file name: Customer ID: NCC063	User chose an image file to be uploaded Image file name: resitatm.jpg Customer ID: NCC063
Test Data Result	New payment detail fail to be recorded due to insufficient of data during uploading	New payment detail is successfully being recorded into the database

6.4.2.5 Test Module 5: Order History

Table 6.19: Test Data of Order History Module

Test Data ID	TD_05-1	TD_05-2
Test Case ID	NOOCMS_05-1	NOOCMS_05-2
Order History Form (Updating)	If order status is currently in ‘Approved’ state and user tries to update the order menu quantity	If order status is currently in ‘Pending’ state and user tries to update the order menu quantity
Test Data Result	Updating not successful because order has been approved by Admin	Order menu quantity is updated successfully into the database

6.4.2.6 Test Module 6: Search Cookies

Table 6.20: Test Data of Search Cookies Module

Test Data ID	TD_06-1	TD_06-2
Test Case ID	NOOCMS_06-1	NOOCMS_06-2
Search Cookies Column	User enters ‘chocolate’ as a keyword Keyword: chocolate	User enters ‘chocOlate’ which contains spelling error Keyword: chocOlate
Test Data Result	All menu which contains the keyword ‘chocolate’ will be listed in the table	Nothing will be displayed (blank table)

6.4.2.7 Test Module 7: Import Order

Table 6.21: Test Data of Import Order Module

Test Data ID	TD_07-1	TD_07-2	TD_07-3
Test Case ID	NOOCMS_07-1	NOOCMS_07-2	NOOCMS_07-2
Import Order form	User does not choose any file to be uploaded Choose file:	User chose invalid file type to be uploaded Choose file: resitatm.jpg	User chose valid file type to be uploaded Choose file: order_menu.csv

Test Data Result	Nothing will be imported	Data cannot be imported because invalid file has been chosen	New menu orders will be successfully inserted into the database

6.5 Test Results and Analysis

This part will explain each module's test case results which consists of the description and whether it is a success or failure.

6.5.1 Test Module 1: User Registration

Table 6.22: Test Results and Analysis of User Registration Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_01-1	TD_01-1	11/7/2016	Some field is blank	√	
NOOCMS_01-2	TD_01-2	11/7/2016	All fields were filled	√	

6.5.2 Test Module 2: Login (User Authentication)

Table 6.23: Test Results and Analysis of Login Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_02-1	TD_02-1	12/7/2016	Invalid username and password	√	
NOOCMS_02-2	TD_02-2	12/7/2016	Valid username and password	√	

6.5.3 Test Module 3: Purchase Order

Table 6.24: Test Results and Analysis of Purchase Order Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_03-1	TD_03-1	13/7/2016 – 14/7/2016	Some field is blank	√	
NOOCMS_03-2	TD_03-2	15/7/2016	All fields were filled	√	

6.5.4 Test Module 4: Payment

Table 6.25: Test Results and Analysis of Payment Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_04-1	TD_04-1	16/7/2016 – 17/7/2016	Some field is blank	√	
NOOCMS_04-2	TD_04-2	18/7/2016	All fields were filled	√	

6.5.5 Test Module 5: Order History

Table 6.26: Test Results and Analysis of Order History Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_05-1	TD_05-1	19/7/2016	Updating order menu quantity when the order status is currently in 'Approved' state	√	
NOOCMS_05-2	TD_05-2	20/7/2016	Updating order menu quantity when the order status is currently in 'Pending' state	√	

6.5.6 Test Module 6: Search Cookies

Table 6.27: Test Results and Analysis of Search Cookies Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_06-1	TD_06-1	21/7/2016	User enters unmatched keyword	√	
NOOCMS_05-2	TD_05-2	22/7/2016	User enters matched keyword	√	

6.5.7 Test Module 7: Import Order

Table 6.28: Test Results and Analysis of Import Order Module

Module / Component			Result		
Test Case ID	Test Data ID	Testing Date	Description	Success	Fail
NOOCMS_07-1	TD_07-1	23/7/2016	User did not choose any file to be uploaded	√	
NOOCMS_07-2	TD_07-2	24/7/2016	User chose invalid file to be uploaded	√	
NOOCMS_07-3	TD_07-3	25/7/2016	User chose valid file to be uploaded	√	

Based on the test results obtained, it can be analysed that each module has achieved the goals targeted based on the test module and test data planned.

6.6 Conclusion

Software testing is necessary because we all make mistakes. Some of those mistakes are not very important, but some of them are dangerous. We need to check everything that we have produced because things can always go wrong. Humans make mistakes all the time.

Since we assume that our system might have some errors, hence we really need to check our own work. However, some mistakes came from bad assumptions and blind spots; therefore we might make the same mistake when checking our own work. We might not notice the flaws in what we have done. Ideally, getting someone else to check our work is the best way to overcome this problem, because another person is more likely to spot the flaws. They are called the tester.

Software testing is very important because of the following reasons; it is required to point out the defects and errors that were made during the development phases. It is essential since it make sure customer’s satisfaction in the application. It is very important to ensure the quality of the application because quality of product delivered to the customers will help in gaining their confidence. Besides that, testing is necessary in order to lower the maintenance cost. It is important to ensure that the application should not result into many failures because it can be very expensive in the future or in the later stages of the development. Last but not least, testing is required to stay in the business.

For the last chapter (Chapter 7), a conclusion will be explained by pointing out the strengths and weaknesses of this NOOCM system through observations, the propositions for any further improvements that might be carried out in the future, and any project contribution.

CHAPTER VII

CONCLUSION

7.1 Introduction

This last chapter will elaborate on a few important things such as the weaknesses and strengths of this Nissa's Online Ordering Cookies and Marketing System (NOOCMS), the propositions for some improvement that could be made regarding on the weaknesses stated, the contribution to any party that has involved in the development of this system either directly or indirectly, and a conclusion that will conclude whether this project has met its objectives conclusively or the other way round.

7.2 Observation On Weaknesses And Strengths

In each and every system built by human, there must be some weaknesses rather than just strengths. The strengths and weaknesses in this Nissa's Online Ordering Cookies and Marketing System (NOOCMS) are as listed below:

7.2.1 Strengths

i. It is developed in a web-based environment which will be very convenient for customers to view the website anywhere; provided that Internet is available.

ii. All data regarding on the customers, staff, orders, payments, and sales report is kept in a proper database compared to a manual system which is not applicable in this era of information technology anymore.

iii. The system is built with different level of user access; customer, admin, and staff. This is to divide and control user's access into the system. Each user will have different kind of access when they logged into the system.

iv. Staff can import customer's orders from Excel Spread Sheet straight into the database. This will accommodate them by allowing the migration of data from file-based system to database. They can even export orders (for backup) from database into Excel Spread Sheet if needed.

7.2.2 Weaknesses

i. There are no backup and recovery available for the data captured in each transaction. This will be the critical part of the security issue if all data went missing when problems occurred.

ii. The payment receipt uploaded by customers cannot be deleted nor updated and it will grow from time to time. This will make it hard for customer or staff to keep track of the payment history as they will have to scroll down looking for the perfect match.

7.3 Propositions For Improvement

Regarding on the weaknesses stated above, there are some improvements that could be made for the future use. For example, in order to strengthen the security issue, full backup or incremental backup can be implemented in NOOCMS daily transactions made throughout the system. This backup is to ensure that all data could be recover and gained back if something went wrong.

Moving to the next weakness of this system, as for the payment receipt, a link should be made between the payment and order entity so that the order id could be tracked to distinguish which order is belong to that certain payment made. In order to implement this improvement, the foreign key must be taken into consideration because to capture the foreign key automatically into a table, it is not as easy as it look. Hence, it must be properly planned.

7.4 Project Contribution

I would like to thank some of the website owner which I have been referring to while carrying out this project. The websites are the Rockland Bakery and D’Mahligai Scarf. Both website are online shopping sites. These websites have helped me a lot by giving me new ideas and inspires me on the way to make a website more presentable.

Besides that, I would like to express my gratitude to my supervisor, Dr Nurul Akmar binti Emran because without her help and support, I would not be able to achieve what I have been achieving in this project. Thank you.

7.5 Conclusion

To put it into a nutshell, this system has met its objectives. The objectives of this project are to develop a system that could allow customers to place their orders through the Internet by using a web based environment. Next, is to accommodate NOOCMS's admin to save and keep track all customer's information, orders and sales into a proper system with a database. Lastly, is to allow NOOCMS's admin to view accurate sales report without data redundancy by enforcing key constraints.

This project is being built using SDLC methodology. It has completed all stages of planning, analysis, design, implementation and testing. After this system has been completed, a conclusion could be made is that, this system has been successfully developed and has met all requirements mentioned in earlier stage of the system development. However, there are still some weaknesses existed in some parts of the system which needs to be improvised for future use. These weaknesses exist due to the time constraint faced during the development. Nevertheless, as long as this system has achieved all objectives stated, it means that Nissa's Online Ordering Cookies and Marketing System (NOOCMS) is a success and will be very useful to the end user.

REFERENCES

Anonymous (July, 2015). Analysis in Instructional Design [online]

Available at: http://www.nwlink.com/~donclark/hrd/sat2.html

[Accessed on 6th of April 2016]

Anonymous (March, 2016). What Is The Difference Between Test Plan and Test Strategy? [online]
Available at: http://www.softwaretestinghelp.com/difference-between-test-plan-test-strategy-test-case-test-script-test-scenario-and-test-condition/
[Accessed on 22nd of July 2016]

Anonymous (n.d). Phases of System Development Life Cycle [online]

Available at: http://oer.nios.ac.in/wiki/index.php/Phases_of_System_Development_Life_Cycle

[Accessed on 16th of April 2016]

Anonymous (n.d). Developing Architecture View [online]

Available at: http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap31.html

[Accessed on 16th of April 2016]

Anonymous (n.d). The Benefits of Web Based Application [online]

Available at:

http://www.magicwebsolutions.co.uk/blog/the-benefits-of-web-based-applications.htm

[Accessed on 20th of April 2016]

Anonymous (n.d). D'Mahligai Scarf [online]

Available at: http://www.dmahligaiscarf.com/

[Accessed on 25th of April 2016]

Anonymous (n.d). Rockland Bakery [online]

Available at: http://www.rocklandbakery.com/

[Accessed on 25th of April 2016]

Anonymous (n.d). Third Normal Form (3NF) [online]

Available at:

https://www.1keydata.com/database-normalization/third-normal-form 3nf.php

[Accessed on 25th of April 2016]

Anonymous (n.d.). Black-box vs White-box Testing [online]

Available at: https://technologyconversations.com/2013/12/11/black-box-vs-white-box-testing/

[Accessed on 20th of July 2016]

Anonymous (n.d.). What is the purpose and importance of test plans in software testing? [online]

Available at: http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/

[Accessed on 21st of July 2016]

Anonymous (n.d.). Software Developer Job Description [online]

Available at: http://www.totaljobs.com/careers-advice/job-profile/it-jobs/software-developer-job-description

[Accessed on 21st of July 2016]

Anonymous (n.d.). Why Is Software Testing Necessary? [online]

Available at: http://istqbexamcertification.com/why-is-testing-necessary/

[Accessed on 22nd of July 2016]

Chandra Vennapoosa (Dec, 2006). Advantages of Using Oracle [online]

Available at:

http://www.learn.geekinterview.com/database/oracle/advantages-of using-oracle.html

[Accessed on 26th of April 2016]

Database Systems: Design, Implementation, & Management, 5th Edition (Rob & Coronel) (n.d.). How does database design relate to the SDLC? [online]

Available at: http://pkirs.utep.edu/

[Accessed on 27th of March 2016]

David Ingram (n.d.). Why Is The Implementation of Projects Important? [online]

Available at: http://smallbusiness.chron.com/implementation: projects-important-strategic-planning-project-manager-18215.html

[Accessed on 26th of May 2016]

David Lile Brown (n.d.). The Five Essentials for Software Testing [online]

Available at: https://www.isixsigma.com/industries/software-it/five essentials-software-testing/

[Accessed on 20th of July 2016]

Endeavour (n.d.). What are the seven disadvantages to a manual system [online]

Available at: http://www.endeavour.co.nz/7-disadvantages-manual-system

[Accessed on 23rd of March 2016]

Margaret Rouse (n.d.). Implementation Definition [online]

Available at: http://searchcrm.techtarget.com/definition/implementation

[Accessed on 26th of May 2016]

Mike Griffiths (March, 2009). Non-functional requirements minimal checklist [online]

Available at:

http://leadinganswers.typepad.com/leading_answers/2009/03/nonfunctional-requirements-minimal-checklist.html

[Accessed on 7th of April 2016]

Sybite Technologies (n.d.). Development Approach [online]

Available at: http://www.sybite.com/approach.html

[Accessed on 27th of March 2016]