DENTAL CLINIC MANAGEMENT SYSTEM

WAN NORAQILAH BINTI A. RAZAK

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: DENTAL CLINIC MANAGEMENT SYSTEM

SESI PENGAJIAN:

Saya WAN NORAQILAH BINTI A. RAZAK

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

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

____ SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
____ TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
/	TIDAK TERHAD

(WAN NORAQILAH A.RAZAK)

Alamat tetap:

NO 17, JALAN KEMPAS INDAH
8/4, TAMAN KEMPAS INDAH,
81300, SKUDAI, JOHOR

Tarikh: 6/9/2017

Nama Penyelia:
Dr. Noraswaliza Binti Abdullah

Tarikh: 6/9/2017

DENTAL CLINIC MANAGEMENT SYSTEM

WAN NORAQILAH BINTI A. RAZAK

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2017

DECLARATION

I hereby declare that this project report entitled

DENTAL CLINIC MANAGEMENT SYSTEM

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

STUDENT : _____ Date: 6/9/2017

(WAN NORAQILAH BINTI A.RAZAK)

I hereby declare that I have read this project 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: 6/9/2017

(DR. NORASWALIZA BINTI ABDULLAH)

DEDICATION

This report is dedicated to my parents, who taught me that even the largest task can be accomplished if it is done one step at a time and for their support. To my friends, who helping me out in completing this project and Dr. Noraswaliza Binti Abdullah for her guidance throughout completing this project.

ACKNOWLEDGEMENTS

Assalamualaikum wbt. first, I would like to express the deepest appreciation to my supervisor, Dr Noraswaliza Binti Abdullah for the supervision, guidance and support that she gave throughout completing this project. Without her guidance and persistent help this system would not have been possible.

In addition, I would like to thank my friends for helping me out in completing this system and for their support. I would like to thank the university for guiding me throughout this semester in completing the PSM1 and PSM2.Lastly, I would to express my deepest appreciation to my Advanced Database Programming Language lecturer, Encik Yahya Bin Ibrahim for his guidance and help throughout his class. He helped us on creating trigger and stored procedure for our project. Finally, thanks to all who are directly involved or not.

ABSTRACT

Dental Clinic Management System(DCMS) is a system that records, organizes and maintains all dental office records. The system maintains individual patient account ledgers for each treatment and includes appointment schedule. There are some problems faced by the user, firstly is data redundancy. The information of the patient can be redundant as if there are patients that visited more than once. Besides, difficulty in keeping track of new appointments and patient's records. It is also hard for the dentist to make review or analyze patient's progress and mistakes may happen when writing the prescriptions. This project embarks on the following objectives, it is build to prevent data redundancy of patient's record, to keep in tracks of new appointments and patient record. It helps to provide an efficient workflow and improved accessibility to patient data and to avoid any mistakes when giving out the prescription to the patients. A survey at Klinik Rabbisyfina in Johor Bahru has been conducted and there are several requirements that have been gathered by interviewing the staff and dentist, Dr. Ahmad Bin Ali. This project aims at creating a system that able to keep track of a database for clinic such as the information about the patient by having a computerized patient registration and systematic patient's record which allow patient to update their personal information online and able to view their medical record, appointment schedule, previous treatment records with digital image of dental chart, prescriptions of medicine and it can generate receipt for every transaction and to create report required by the management of the clinic. This system provides a smooth and well-organized system for dental clinic including the doctor, staff and patients. Thus, it helps the dental clinic management to manage the dental clinic

ABSTRAK

Dental Clinic Management System(DCMS) merupakan sistem yang merekod, menyusun dan menyimpan rekod berkaitan klinik pergigian tersebut. Sistem ini menyimpan rekod setiap pesakit termasuk jadual temuduga pesakit terssebut. Terdapat beberapa masalah yang diihadapi oleh pengguna sekiranya mereka tidak menggunakan system. Pertama, data yang berlebihan. Rekod pesakit akan mejadi banyak sekiranya pesakit tersebut dating ke klinik lebih dari sekali. Selain itu, temuduga pesakit yang baru sukar untuk disimpan atau rekod temuduga tersebut boleh hilang. Tambahan pula, sukar untuk doctor untuk melakukan analisis berkaitan progress pesakit tersebut dan kesilapan juga boleh berlaku semasa menulis preskripsi ubat. Oleh itu, sistem ini dapat menmbantu doctor untuk mengelakkan berlakunya data berlebihan, memudahkan untuk membuat temuduga dan membantu melancarkan process klinik pergigian tersebut. Kajian telah dilakukan Klinik Rabbisyfina di Johor Bahru. Terdapat beberapa maklumat yang telah dikumpulkan dengan menemubual pekerja dan doctor gigi, Dr. Ahmad Bin Ali. Tujuan utama projek ini adalah untuk membina system yang boleh menyimpan data untuk klinik seperti rekod pesakit yang membolehkan pesakit untuk mengemaskini data mereka secara online dan boleh melihat rekod kesihatan dengan adanya carta gigi dan rekod temu janji serta preskripsi ubat. Malah, system ini mampu mencetak resit untuk kegunaan laporan. Sistem ini merupakan system yang teratur dan mudah digunakan di klinik gigi. Tambahan pula, ia dapat membantu doktor dan pekerja menguruskan klinik gigi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DCMS	-	Dental Clinic Management System
ERD	-	Entity Relationship Diagram
DFD	-	Data Flow Diagram

CHAPTER I

INTRODUCTION

1.1 Project Background

Dental Clinic Management System(DCMS) is a system that records, organizes and maintains patients records in a dental clinic. DCMS software provides easy access of the data in advanced and ordered fashion. The system maintains individual patient treatment record including appointment schedule. It is designed and built to meet all the requirements of a dental clinic and to help the dentist in managing their clinics effectively. With just a few clicks the user can schedule, record treatment plans easier and faster.

This project aims at creating a system which will automate the process in DCMS. It is possible to keep track of a database for clinic such as the information about the patient, appointment schedule, previous treatment records with digital image of dental chart, prescriptions of medicine and it can generate receipt for every transaction and to create report required by the management of the clinic. Patient records are used by the dentists to organize the records of the patients in their practice.

Before the patient reaches the front desk, staff has prepared a walkout statement in which treatment plan fees have been entered. As the last patient leaves, staff can access

reports to make sure that day has been efficient and financially productive. In a single click, they can quickly schedule a follow-up appointment. The software offers automated reports to close the day. Moreover, if the patient wants to set up an appointment, they must choose dental treatment and services such as Oral Prophylaxis (Cleaning), Restoration (Filling), Prosthodontics (Dentures, Crown, Bridged), Surgery (Extraction/Impaction), Endodontic (Root canal treatment), and Cosmetics (Tooth Whitening) and other related services. DCMS able to save time and streamline workflow.

1.2 Problem Statements

These are the problems faced by user:

i. Data Redundancy
   The information of the patient can be redundant because there are patients that visited more than once.

ii. Difficult in keeping track of new appointments and patient records
   All appointments are paper-based and the tendency of data lost is high

iii. It is hard for the dentist to make review or analyze patient's progress
   They need to find patients records and it is time consuming.

iv. Mistakes may happen when writing the prescriptions
   Dentist may make mistake when writing the prescription and their illegible handwriting can create confusion.

1.3 Objectives

This project embarks on the following objectives:

i. To prevent data redundancy of patient's record

ii. To keep tracks of new appointments and patient record by replacing the current system with a computerized system to provide better management.

iii. To improved accessibility to patient data. The dentist can automatically record patient's treatment by clicking on the dental treatment chart. By creating this chart, dentist has all the information that they need to access patient's dental health, in one place in a simple format. They will update patient chart every time they have a dental appointment so they can track the progress of patient's dental health.

iv. To ease the process of giving the prescription and calculating the total payment to the patients by clicking all the dosage and prescription on the computer to be sent to the staff.

1.4 Scope

i. Login and Logout

There are three type of users which are doctor, staff and patient. User need to sign up before login in to the system by entering username and matching password.

ii. Registration

Registration of a new patient is made on computer by staff. All the records regarding patient's information is save into the database.

iii. Appointment

Doctor can set an appointment for each patient on the system and patient can request an appointment based on the doctor's availability.

iv. Dental Imaging Treatment

Treatment data on each patient will be uploaded on the dental image, so that it can be easily viewed by the doctor or patience.

v. Report

Staff can analyze and print sales report.

1.5 Project significance

This system provides a smooth and well-organized system for the dental clinic including the doctor, staff and patients. Thus, it helps the dental clinic management to manage the dental clinic.

1.6 Expected Output

i. A computerized patient registration and systematic patient's record which allow patient to update their personal information online and able to view their medical record.

ii. Well-organized and smooth treatment by having a dental treatment chart.

iii. Accurate Sales Report in tabular format which can be printed through the system.

1.7 Conclusion

With dental clinic management system, the process gets much faster and more efficient than traditional way. DCMS is specially designed to let the clinic staff to have a high efficiency management tools, computerized and systematic patients' records, and detail of treatment records. This system also provides appointment feature, which allow staffs, dentist and patients to view the appointment that already made by dentist and patient. It also can increase the profitability of the organization. The methodology that will be used will be explained in the next chapter.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

In this chapter, the methodology that is used which is Incremental Development method and the project milestone for this system starting from the week 1 until week 14 will be explained. There are also expected documents and dates provided in the project schedule.

2.2 Project Methodology

In this section, the research methodology will be explained in detail regarding the methodology that is used in the development of Dental Clinic Management System(DCMS). The fundamental for this project is to develop a management system that can be implemented and integrated in dental clinics. This project will be conducted based on the Database Lifecycle Method (DBLC) as illustrated in Figure 2.2. DBLC contains six phases which are initial study, database design, implementation and loading, testing and evaluation, operation, maintenance and evolution.

A survey has been conducted at Klinik Rabbisyfina in Johor Bahru and there are several requirements that have been gathered by interviewing the staff and dentist. Dr. Ahmad Bin Ali, the dentist at the clinic, had explain the system that is used at Klinik Rabbisyfina. Currently, the clinic does not have any computerized system as they are still using a manual system.

Initial study is the first step in DBLC. It analyses the company situation, define problems and constraints, define objectives of the organization and define the scope and boundaries to make sure the database is created exactly as specified. During this phase, the objective of the project is determined and the requirements to produce the product are considered. The analysis of the dental clinic has been figured out. The requirements have been collected by interviewing the dentist and the staff at Klinik Rabbisyfina. The schedule and milestone of this system has been stated as illustrated in Figure 2.1. All the requirements have been classified into two types of requirements which are functional and non-functional requirement. The functional requirement is based on the scope that has been identified during planning phase.

During the database design phase, the conceptual design is created by creating an abstract data structure that attempts to model real-world objects. Then, it is translated into logical design which is a DBMS dependent. Physical design is the process of selecting data storage and data characteristics of the database.

Conceptual design is used to determine end-user views, outputs and transaction requirements. During this phase, Entity Relationship diagram(ERD) has been drawn as shown in figure 4.2. Selection of DBMS which is Oracle SQL Developer has been selected for this project.

The logical design includes defining the tables, columns, relationship and constraints of the DCMS. The purpose of logical design is to ensure the model supports user requirements. DCMS database system is logically explained in this stage by specifying data and query based on user requirements. Physical design is the last step and it is the process of defining tables, indexes, roles and access control.

The third step is implementation and loading. In this phase, the DBMS is installed and the data is loaded into the database tables. In this phase, conceptual, logical and physical design is implemented.

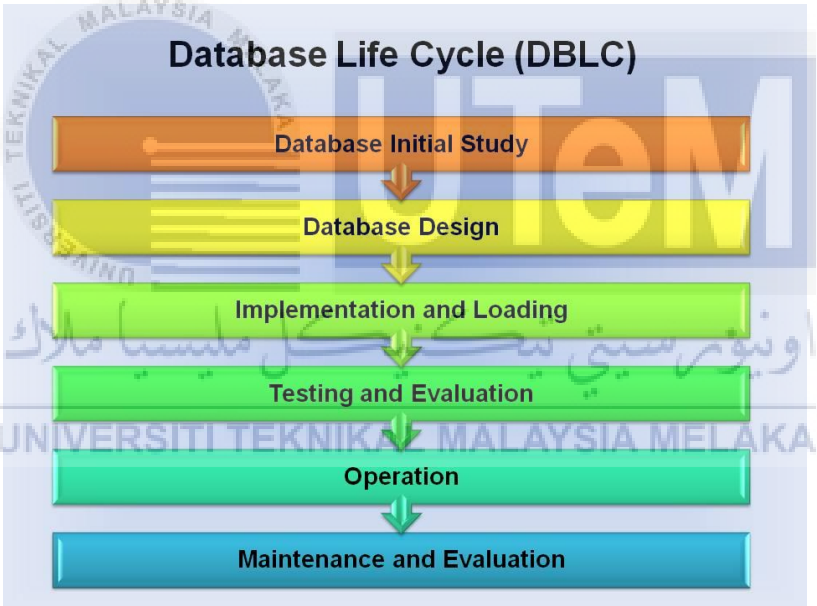


Figure 2.1 Database Lifecycle Method (DBLC)

In testing and evaluation phase, it requires the database to be tested by testing the integrity, physicsl security, password security and access rights. The testing and evaluation phase occurs in parallel with applications programming. During the test phase, all aspects of the system are tested for functionality and performance. The DCMS database is tested to ensure that it maintains the integrity and security of the

data. Data integrity is enforced by the DBMS through the proper use of primary and foreign key rules. In database testing, physical security allows, password security, access database tables. rights and data encryption must be checked. This phase is used ensure that it maintains the integrity and security of the data.

The DCMS database performance also need to be evaluated, security standards set up, backup and recovery procedures such as full backup, differential backup and transaction log backup are put in place and data integrity enforcement. The operation phase is the phase where the database has passed the evaluation stage, it is considered to be operational. In operation phase, problems that could not have been foreseen during this phase begin to surface. All the changes occur will lead to the last phase, maintenance and evolution. In maintenance and evolution phase, there are some routine maintenance activities within the database need to be prepared for example, baackup and recovery.

2.3 Project Schedule and Methodology

Table 2.1: Project Schedule and Milestone

Requirement	**Expected Documents/ Output**	**Dates(Weeks)**
Initial Planning	Objectives, Scope, Project Significance	6/3/2017 – 10/3/2017 (Week 2- Week 4)
Project Planning	Project Methodology, Project Schedule and Milestones	13/3/2017 – 17/3/2017 (Week 4- Week 6)

Analysis	Functional Requirements, Non-functional Requirements	20/3/2017 – 24/3/2017 (Week 6-Week 7)
Design	ERD, Context Diagram, Data Flow Diagram(DFD), Graphical User Interface(GUI)	27/3/2017-31/3/2017 (Week 7)
Implementation	Create database, triggers, stored procedures, system implementation	13/7/2017 – 9/8/2017
Testing	Create test testing, test data and test result	15/4/2017 – 19/5/2017
Presentation	Presentation of PSM2	16/8/2017

Table 2.2 : Gantt Chart of DCMS

Activity	Duration														
	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15
Proposal Writing Report															
Topic Discussion															
Problem Statement															
Objectives															
Scope															
Project significant															
Conclusion															
Submission using PSM online system															
Proposal Correction															
Proposal Presentation															
Analysis															
Analysis of Problem															
System Architecture															
Modules/Functions															
Dataset															
Report Writing															
Design															
ERD															
Data Dictionary															
User Interface															

Implementation														
System Implementation														
Prepare Report														
First Presentation														
Project Demo														
PSM1 Report														
Final Presentation														
Presentation														
Final Project Demo														

2.4 Conclusion

There are many reasons why Database Lifecycle Method has been used for this project. DBLC Method is a very efficient to deal with a project that change often and response need to be quick. Furthermore, DBLC method is a top down approach that systematically implement and maintain a database. Therefore, it is suitable to be used for this project.

CHAPTER III

ANALYSIS

3.1 Introduction

The analysis phase involves gathering requirements for the system. At this stage, business needs are studied with the intention of making business processes more efficient. The system analysis phase focuses on what the system will do in an effort that views all stakeholders, as viable sources of information. In the analysis phase, a significant amount of time is spent talking with stakeholders and reviewing the stakeholder's input. Within this analysis phase, requirements are collected at Klinik Rabbisyfina in Johor Bahru using interview method.

3.2 Problem analysis

Based on the current system, the dental clinic still using the traditional method. Traditional method is the process that is being done manually, for example, papers, files are still use for patient registration, medicine prescription and treatment record. Manual method has a lot of disadvantages as it takes a longer time to retrieve the information, and mistakes might happen when writing the prescription or misplaced the files.

Figure 3.1 illustrates the flowchart of appointment set by patient on current system. Based on the flowchart, patient need to go to the counter and the staff will check whether they are existing patient or new patient. If they are new patient, the patient need to fill in their details in registration form manually and the information will only keep in files. If the existing patient set an appointment through phone, the need to choose date and time. The appointment can only be set based on the dentist availability.

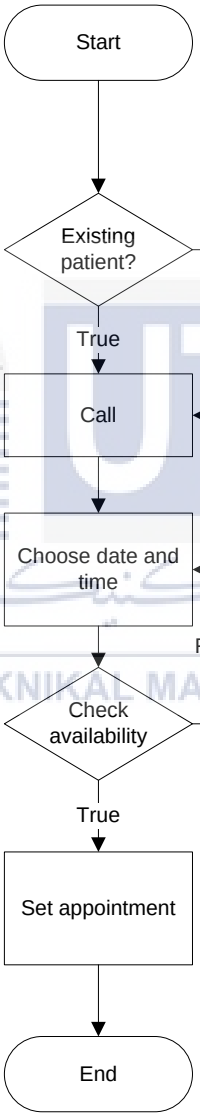


Figure 3.1 Flowchart of Patient based on current system

Based on figure 3.2, flowchart of dentist on current system, the dentist need to write down the patient name and treatment of each patient for the medical record. After the treatment, the dentist will write the medicine prescription and give to the staff. If the patient need to have follow-up appointment, dentist will choose the available date and time.

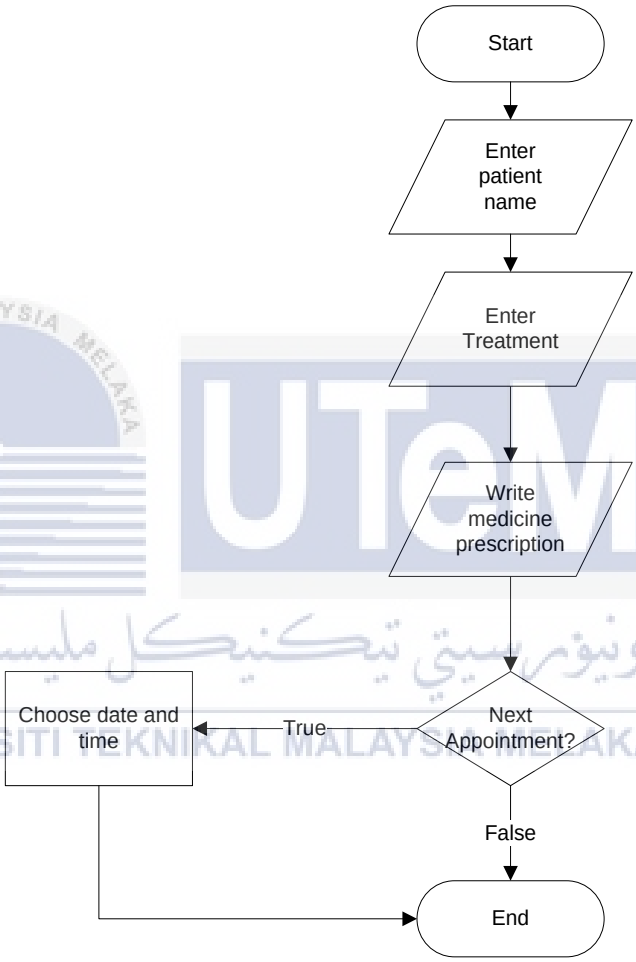


Figure 3.2 Flowchart of Dentist based on Current System

Figure 3.3 illustrates the flowchart of the payment process based on the current system after treatment. Staff will receive medicine prescription from the dentist and they will give out the medicine to the patient. The payment will be calculated based on the treatment and prescribe medicine.

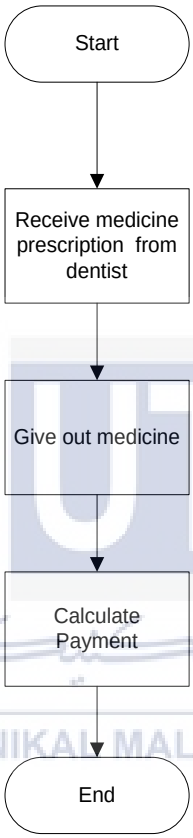


Figure 3.3: Flowchart of Payment Process based on Current System

3.3 The proposed improvements/solutions

Dental charting is a neatly designed interface that will be used in this system. It helps the dentist to mark treatments or disease on a dental chart. It helps them to enter the details very quickly and efficiently.

Dental appointments help to save a lot of time. With appointment module, the doctor can easily give appointments on available time slots to patients. This system also allows user to generate reports in few clicks. Staff won't spend hours manually formatting spreadsheets and consolidating and collating reports.

Figure 3.4 illustrates the flowchart of patient. Patient need to sign-up before they can access to the system. If they have sign-up, they need to login by entering correct username and password. Patient can set an appointment by checking on dentist availability. They need to choose date and time for appointment.

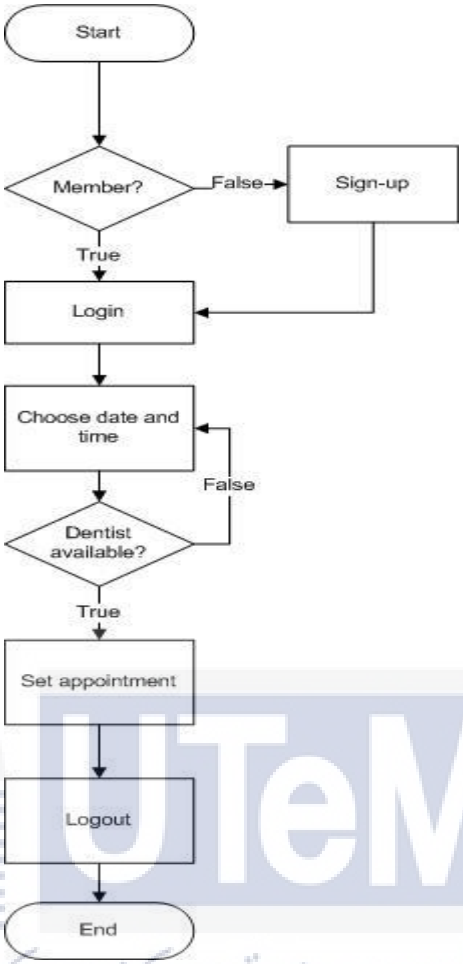


Figure 3.4 Flowchart of Patient based on Proposed System

In figure 3.5, the figure shows the flowchart of dentist. Dentist need to login before access to the system. The dentist need to enter patient name for treatment and after the treatment is done, dentist will enter the treatment records and medicine prescription. If the patient need follow-up check-up, dentist can set an appointment by choosing the date and time.

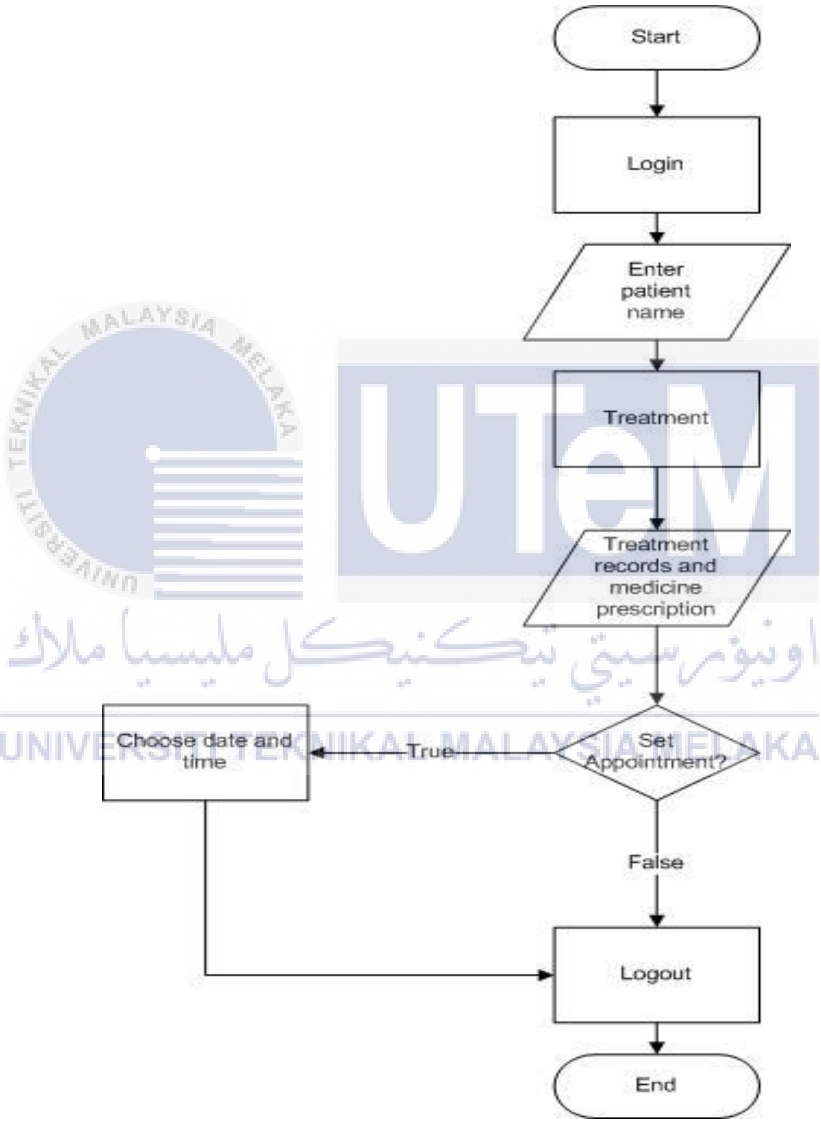


Figure 3.5 Flowchart of Dentist based on Proposed System

Figure 3.6 illustrates the flowchart of staff. Staff need to login before access to the system. Staff will save the date and time of visit patient. After the treatment by the dentist is done, staff will receive the medicine prescription to give out to the patient. They need to enter the patient name for payment process. Staff able to print receipt. Other than that, staff able to generate reports.

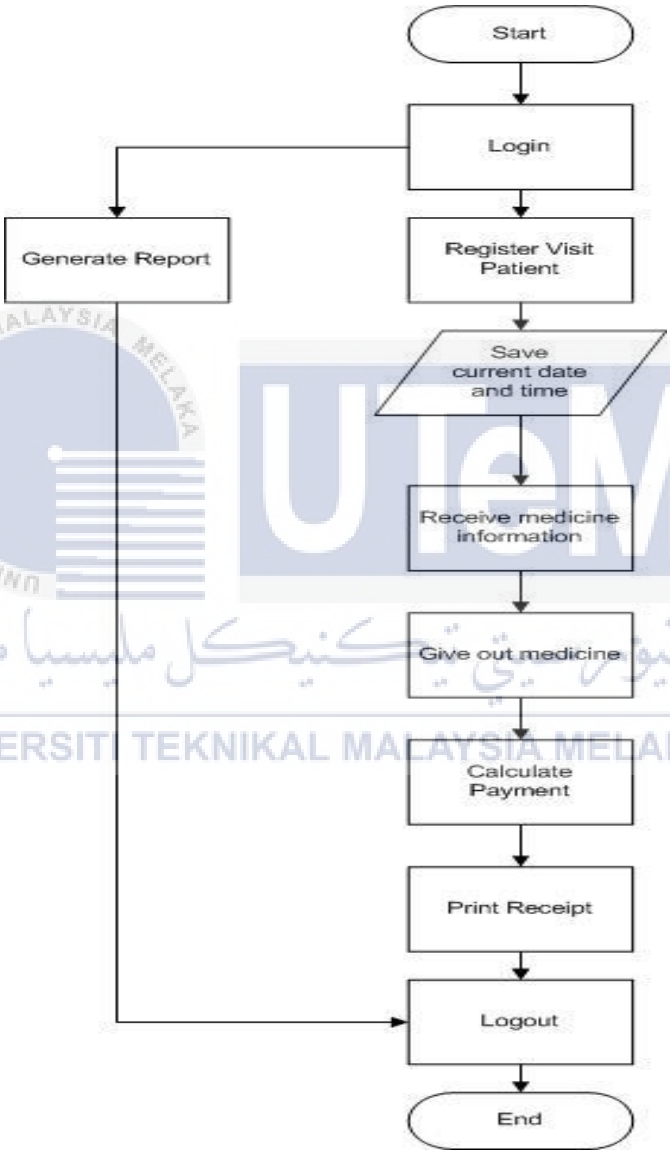


Figure 3.6 Flowchart of Staff based on Proposed System

3.4 Requirement analysis of the to-be system

There are two types of requirement analysis which are functional and non-functional requirement.

3.4.1 Functional Requirement (Process Model)

Figure 3.7 shows the context diagram of DCMS. The circle represents an entity, rectangle represent an external entity and each directional line indicates the flow of the process. Appointment selection is made by patient and dentist and this process flows from the user to the system. Both users receive the appointment notification. Dentist set available schedule and prescription of each patient to the system. Staff able to retrieve registered patient information and medicine information from the system. Staff will save the payment to the system.

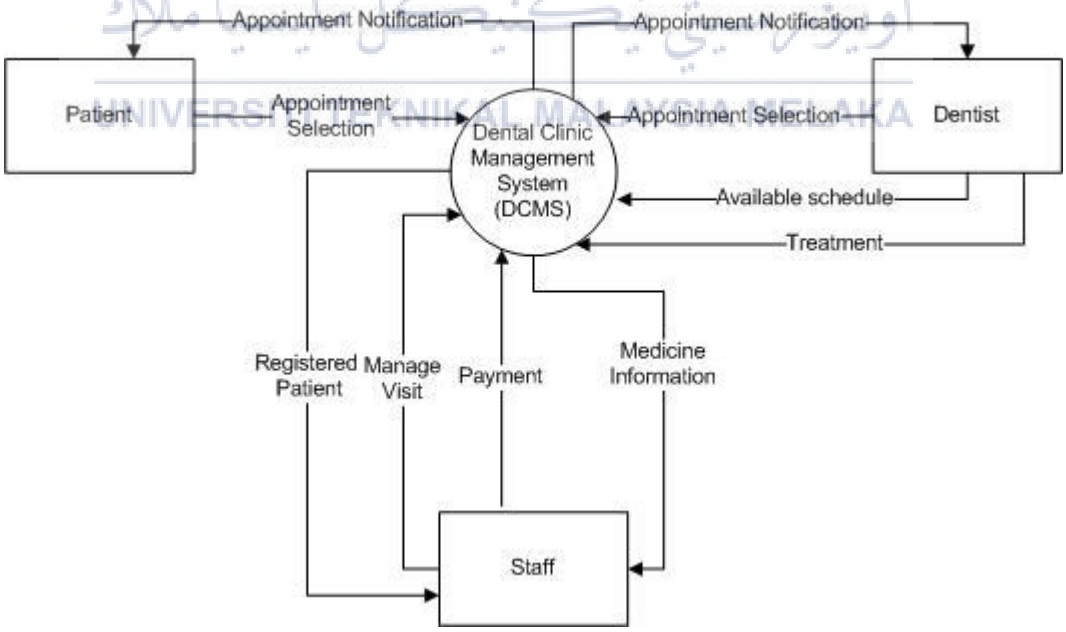


Figure 3.7 Context Diagram of DCMS

Data Flow Diagram

Figure 3.8 shows the parent process of 'DCMS' which will be described in detail in Figure 3.8. The external entity of this system are staff, patient and dentist. In login process, staff, patient and dentist need to enter their email and password and the system will verify the data by accessing into the data store. Process 2 shows the flow diagram of manage appointment process. Patient and dentist need to enter the appointment information such as date and time. The appointment information will be saved in the appointment table. User will get the appointment status from the appointment table. In visit process, when the patient visits the clinic, the dentist will get the visit information of each patient. Treatment is the fourth process of level 0. Dentist will enter the treatment information of the patient and that information will be saved in v_treatment table. Dentist can retrieve data in medicine table during treatment process. In the fifth process, which is the compute total payment and print receipt process, staff will calculate the total payment and it will be saved in payment table. Report process is the process where the staff will generate reports.

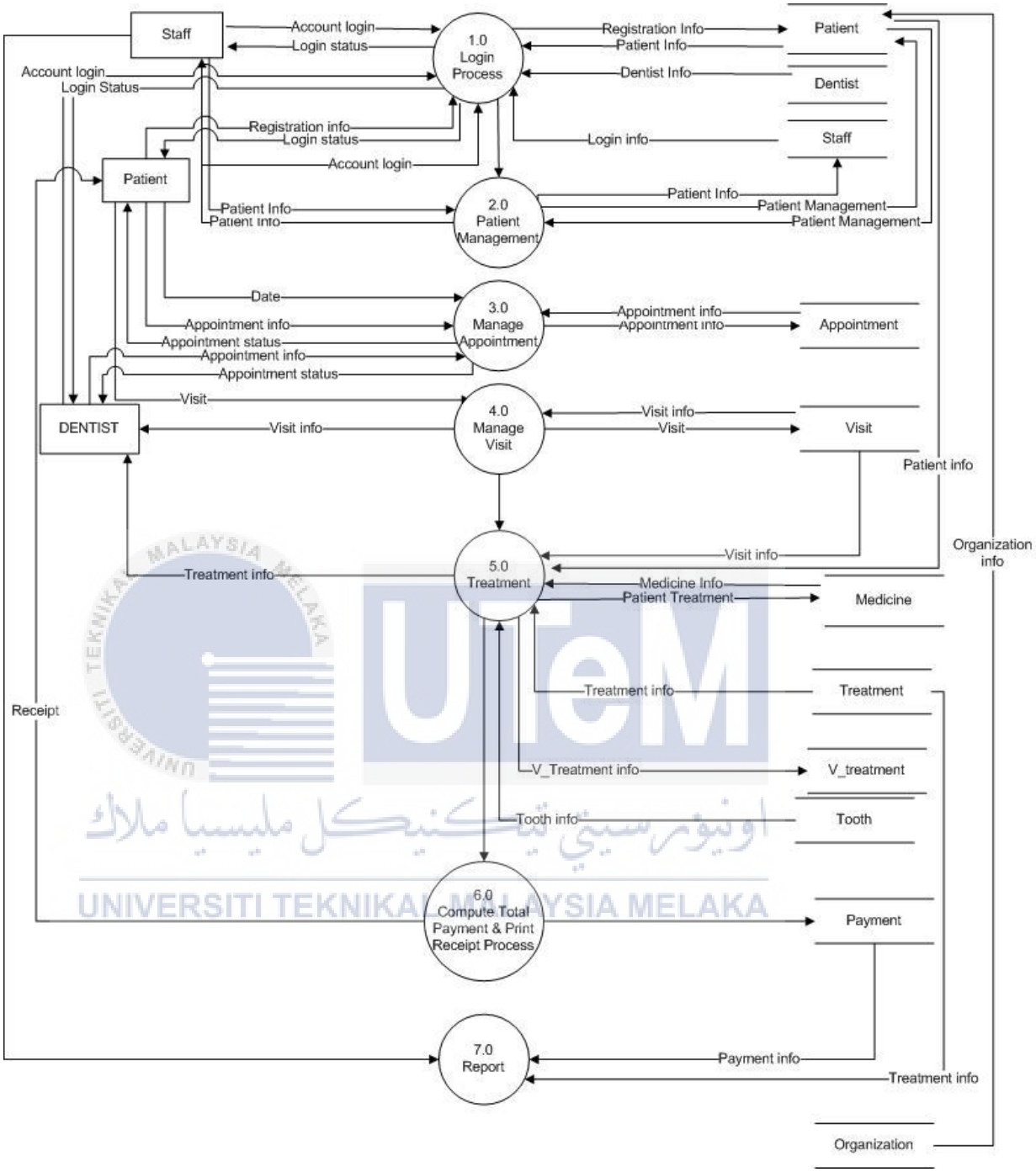


Figure 3.8 DFD of Level 0

i. Appointment Selection

The appointment calendar used to set an appointment. Appointment can be made by patient and dentist. When a patient creates an account, he/she will be assigned to the appointment scheduler. They can set an appointment based on the doctor's availability by clicking on the date and time.

ii.Appointment Notification

Patient and dentist will receive appointment notification. The appointment notification received by patient is based on dentist availability while dentist is notified by the date and time that has been chosen by patient.

iii.Available Schedule

Patient can make an appointment based on dentist availability as shown on the appointment status.

iv.Treatment

Graphical treatment chart is an interactive graphic experience with tooth numbering. It facilitates the doctor to enter treatment or defect for each patient in organized manner by clicking on the check button of the tooth number.

v.Registered Patient

Registration information of new patient is saved on the computer and patient can also update their personal information.

vi.Manage Visit

Staff will save the visit date and time of each patients and it will be saved into the database.

vii.Payment

Payment is calculated automatically based on the medicine and treatment of each patient

viii.Medicine Information

Staff can view medicine information including the current quantity of the medicine. Staff will receive the prescribed medicine for patient.

Figure 3.9 shows the level 1 of login process. In level 1.1 process, user need to enter the correct username and password and the verification of data the system will verify the data by accessing into the 'User account' data store. If it is an unverified account, user need to enter the correct information. Patient can update their information as shown in the level 1.3.

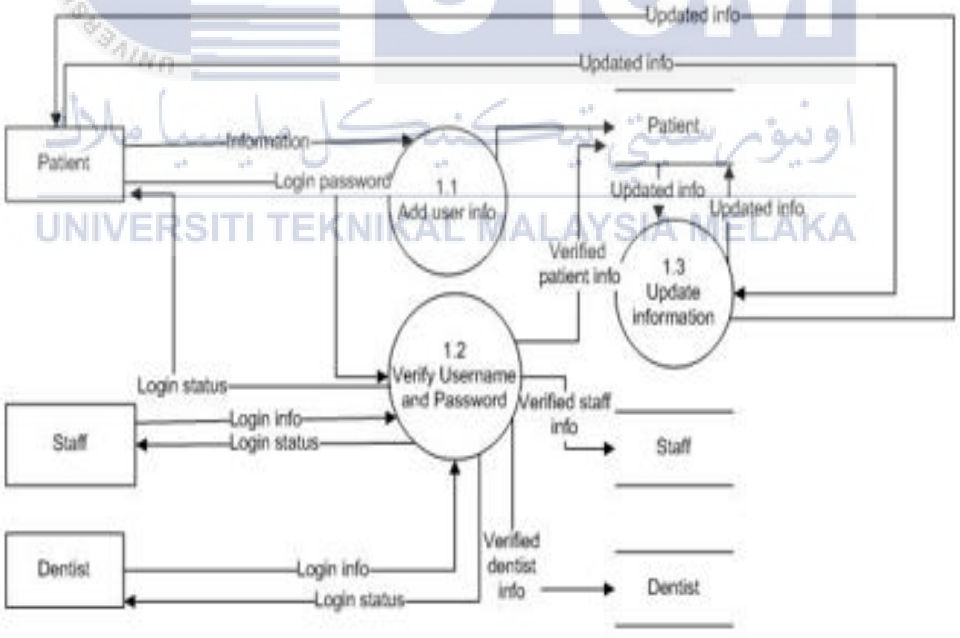


Figure 3.9 DFD of Level 1 of Login

In figure 3.10, patient need to enter date and time and the system will verify the selection by checking on the available slot of the dentist schedule. The dentist can also set an appointment and the dentist will get the appointment status from the appointment table. Staff able to view the all the appointments.

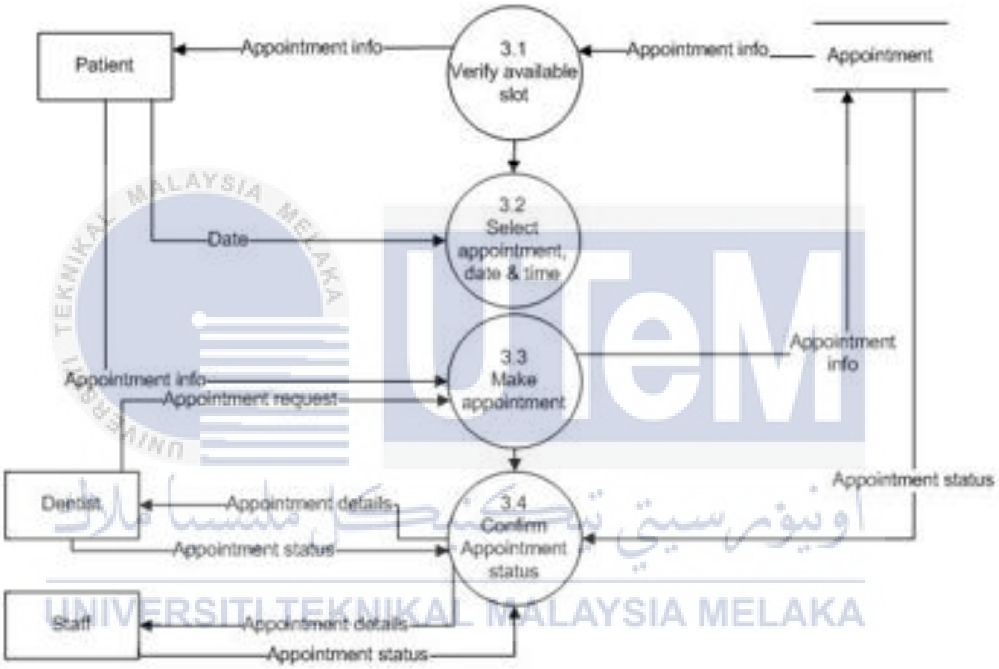


Figure 3.10 DFD of Level 1 of Manage Appointment

Figure 3.11 shows the data flow diagram of level 1 of process 5. Level 5.1 shows the insert treatment details process, where the patient able to view their medical record by retrieving the data from v_treatment table. While, dentist will save the patient treatment information and medicine prescription. The medicine prescription will be saved on v_medicine table and all these data will be saves on visit table. Dentist are able to view the treatment information from the visit and v_medicine table.

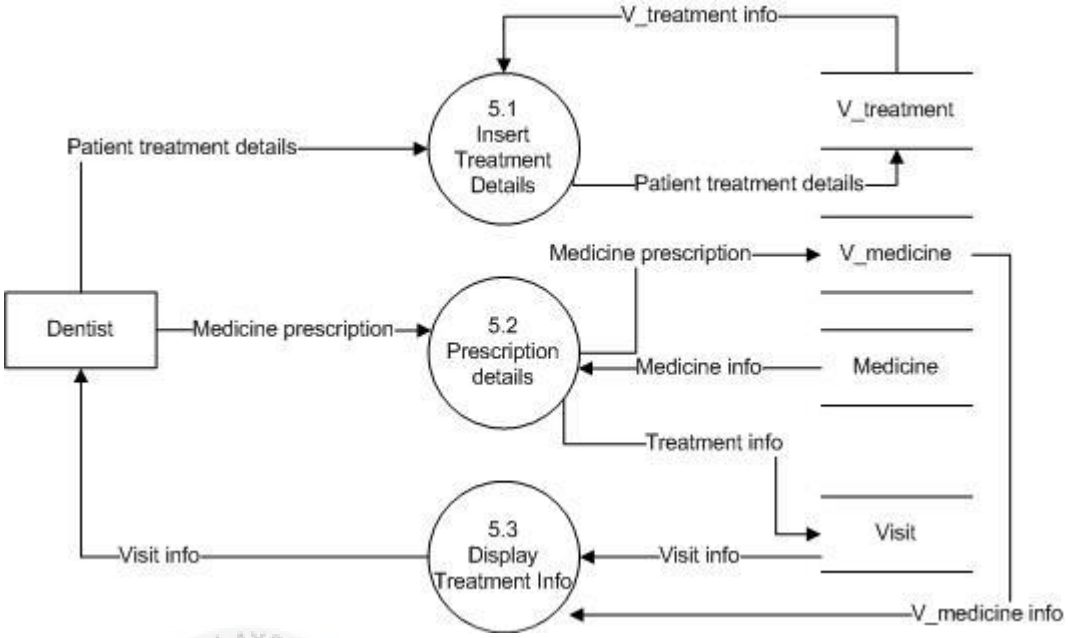


Figure 3.11 DFD of Level 1 of Treatment

Figure 3.12 is the level 1 of process 6. This process shows the flow diagram of payment process. Total payment information saved on payment table. The treatment data retrieved from v_treatment, v_medicine, treatment, medicine, patient and visit table. Patient will get the receipt and the information on the receipt is retrieved from the payment table.

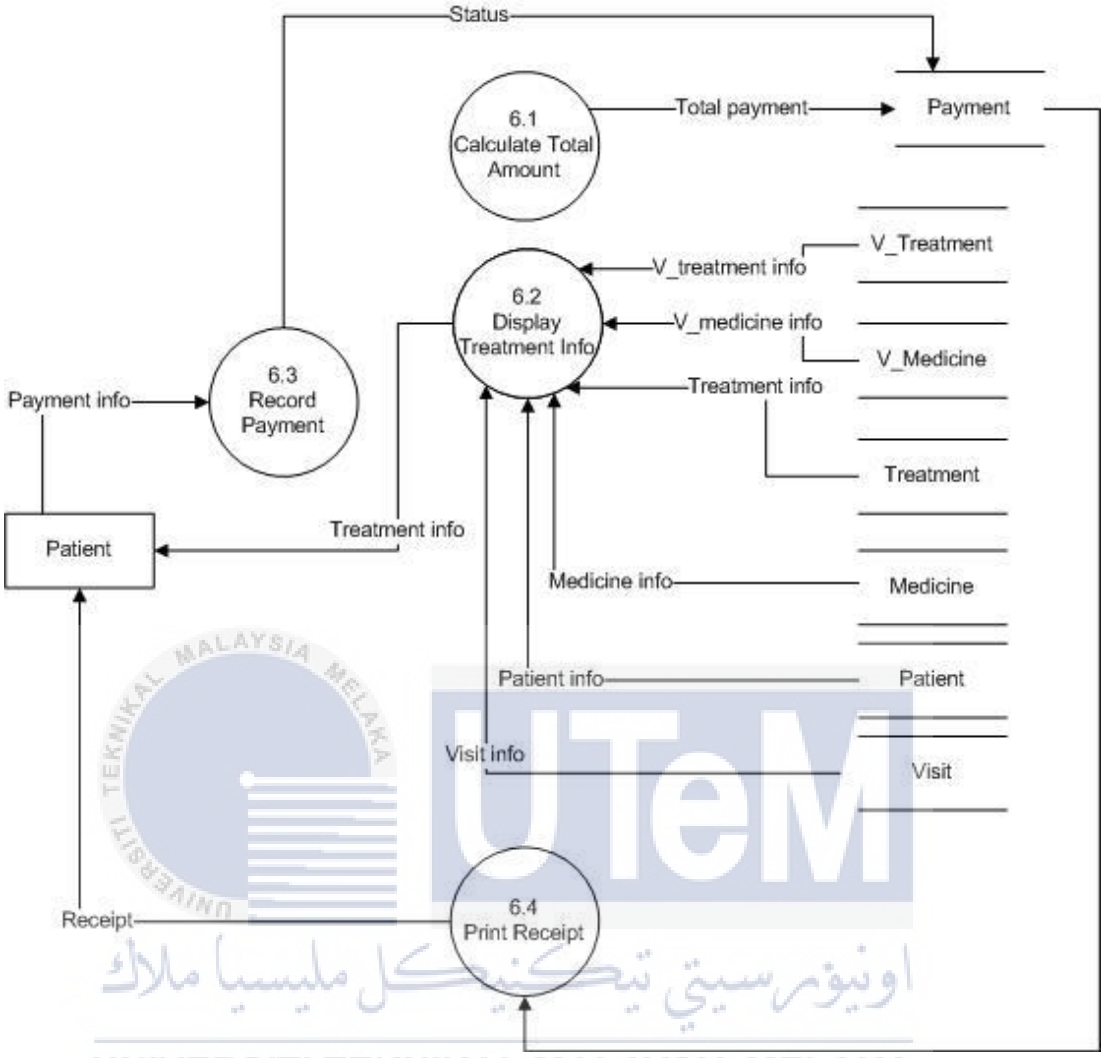


Figure 3.12 DFD of Level 1 of Compute Total Payment & Print Receipt

3.4.2 Non-functional Requirement

i. Security

Login Identification:

The system requires the patient to identify himself /herself

using their username and correct password. Any user who uses the system shall have a Logon ID and Password.

Staff Rights:

Staff shall be able to view all patient information, add new payment to system but shall not be able to modify any information in it. Staff can also view and analyse sales report.

ii. Performance Requirements:

Response Time:

The system shall give responses in 1 second after checking the patient's information.

Capacity:

The System must support 1000 people at a time.

User-interface:

The user-interface screen shall respond within 5 seconds.

iii. Maintainability:

Back-Up:

The system shall provide the capability to back-up the Data

Availability:

The system shall be available all the time

3.4.3 Other Requirements

This system uses PHP programming language and Oracle SQL Developer as the database system. PHP is a server-side scripting language designed primarily for web development but also

used as a general-purpose programming language. Oracle SQL Developer is a free integrated development environment that simplifies the development and management of Oracle Database in both traditional and Cloud deployments

3.5 Conclusion

In analysis phase, there are several requirements that has been gathered. The design phase will be explained in the next chapter including Entity Relationship Diagram, logical and physical design.

CHAPTER IV

DESIGN

4.1 Introduction

In this chapter, the design of this system will be explained. The complete Entity Relationship Diagram(ERD) diagrams with relationship cardinality is provided including the business rule. Design phase focuses on the logical and physical design.

4.2 System Architecture Design

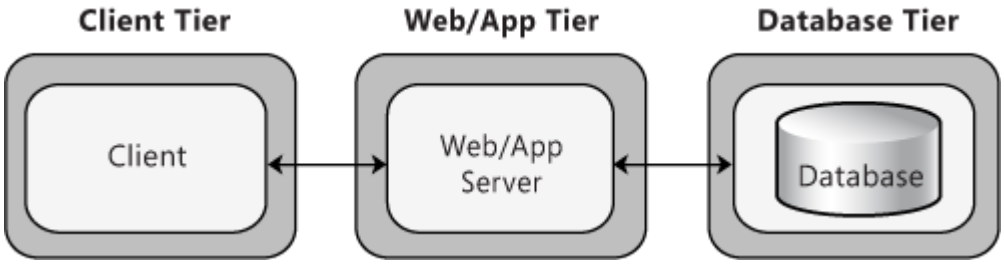


Figure 4.1: System Architecture Design of DCMS

In a 3-tier design as shown in the figure above, the client interacts with application software deployed on a separate server, and the application server interacts with a database that is located on another server. This is a very common pattern for most Web applications and Web services, and sufficient for most general scenarios. Based on the system, the patient, staff and dentist are the clients. The clients will interact or use the system as it is on the web/app tier. In web/app tier or also called middle tier, control the application functionality by performing detailed processing. For example, user can login, sign-up, save the visit information and calculate payment. In database tier, all the data that the user receive or retrieve is connected to the Oracle SQL Developer database.

4.3 Database Design

There are three types of database design will be explained in this section which are conceptual, logical and physical designs.

4.3.1 Conceptual Design

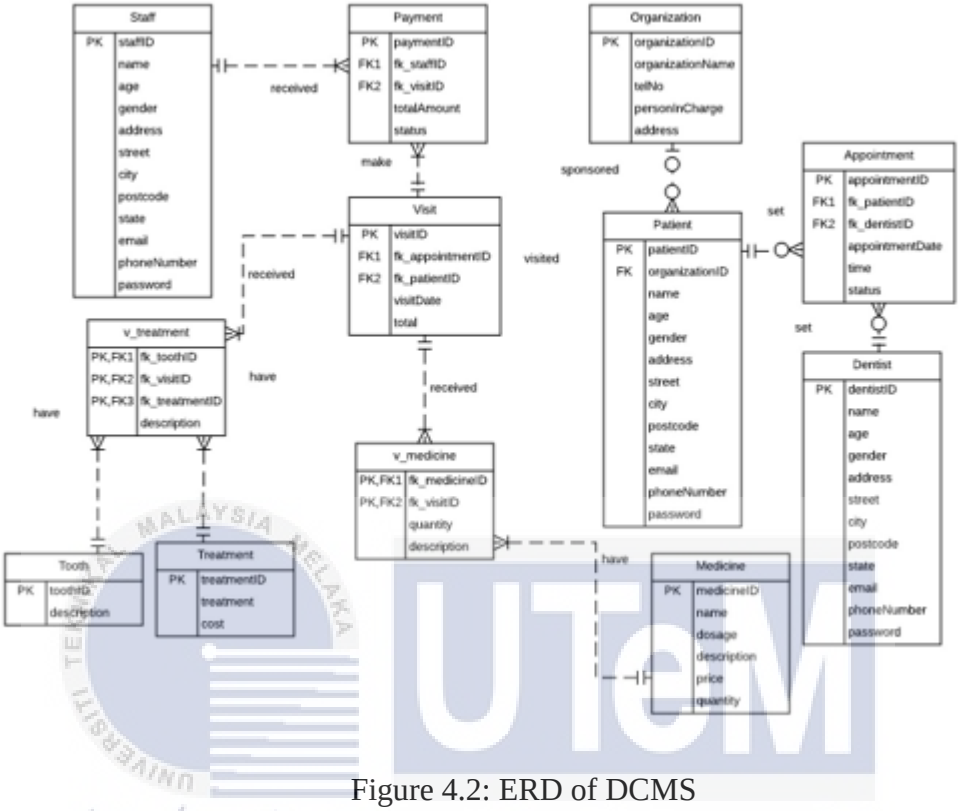


Figure 4.2: ERD of DCMS

Business Rule:

- A staff can a receive payment from one or many patients. A payment is received by a staff
- A patient can either request more than one appointments or did not make any appointment, an appointment can be requested by a patient.
- A dentist can either set more than one appointments or did not make any appointment of each patient, an appointment can be requested by a dentist.
- A patient may be sponsored by an organization
- An organization related to many patients to be served by the clinic.
- A visit is made by a patient
- A patient may visit the dental clinic once or more than one

- V_medicine represents the medication of each patient and a medicine can have one or many v_medicine
- A v_medicine related to a medicine
- V_treatment table save the medical record of each patient visit.
- A visit patient can have many treatments.
- One v_treatment is related for a visit.
- Each treatment, can be listed for one or many treatment of the patient
- A treatment received by patient is related to treatment.
- A tooth is related for many visit treatments.
- A visit treatment may have one tooth.
- Each visit generates one payment.
- Visit table stored information of patient such as treatment and medication treatment of each patient

4.3.2 Logical Design

Data Dictionary:

Table 4.1 Data Dictionary

Entity	Attribute	Description	Datatype	Range	PK/FK	Table
Staff	staffID	Primary key for staff	Varchar2	8	Pk	
	name	Name of staff	Varchar2	100		
	age	Age of staff	Number	3		
	email	Email of staff	Varchar2	50		
	address	Address of staff	Varchar2	100		
	street	Street of staff	Varchar2	100		
	gender	Gender of staff	Varchar2	6		
	phoneNumber	Phone number of staff	Varchar2	12		

	password	Password of staff	Varchar2	20		
	state	State of staff	Varchar2	100		
	postcode	Postcode of staff	Varchar2	5		
	city	City of staff	Varchar 2	100		
Payment	paymentID	Primary key of payment	Varchar2	8	Pk	
	Fk_staffID	Fk referencing staffID	Varchar2	8	Fk	Staff
	Fk_visitID	Fk referencing visitID	Varchar2	8	Fk	Visit
	totalAmount	Total amount received	Varchar2	8		
	status	Status of payment	Varchar2	10		
Patient	patientID	Primary key of patient	Varchar2	8	Pk	
	Fk_organizationID	Fk referencing organizationID	Varchar2	8	Fk	Organization
	name	Name of patient	Varchar2	100		
	age	Age of patient	number	-		
	email	Email of patient	Varchar2	50		
	address	Address of patient	Varchar2	100		
	street	Street of patient	Varchar 2	100		
	gender	Gender of patient	Varchar2	6		
	phoneNumber	Phone number of patient	Varchar2	12		
	password	Password of patient	Varchar2	20		

	state	State of patient	Varchar2	100		
	postcode	Postcode of patient	Varchar2	5		
	city	City of patient	Varchar 2	100		
Appointment	appointmentID	Primary key of appointment	Varchar2	8	Pk	
	Fk_patientID	FK referencing patientID	Varchar2	8	Fk	Patient
	Fk_dentistID	Fk referencing dentistID	Varchar2	8	Fk	Dentist
	appointmentDate	Date of appointment	Timestamp	-		
	status	Status of appointment	Varchar2	8		
Dentist	dentistID	Primary key of dentist	Varchar2	8	Pk	
	name	Name of dentist	Varchar2	100		
	age	Age of dentist	number	2		
	email	Email of dentist	Varchar2	50		
	address	Address of dentist	Varchar2	100		
	gender	Gender of dentist	Varchar2	6		
	phoneNumber	Phone number of dentist	Varchar2	12		
	password	Password of patient	Varchar2	20		
	postcode	Postcode of dentist	Varchar2	5		
	State	State of dentist	Varchar2	100		

Visit	visitID	Primary key of visit	Varchar2	8	Pk	
	Fk_appointmentID	Fk referencing appointmentID	Varchar2	8	Fk	Appointment
	Fk_patientID	Fk referencing patientID	Varchar2	8	Fk	Patient
	visitDate	Date of visit	Timestamp	-		
V_medicine	Fk_visitID	Fk referencing visitID	Varchar2	8	Pk,Fk	Visit
	Fk_medicineID	Fk referending medicineID	Varchar2	8	Pk,Fk	Medicine
	\Quantity	Quantity of medicine given to the patient	Number	3		
	Description	Description of prescribed medicine	Varchar2	500		
Medicine	medicineID	Primary key of medicine	Varchar2	8	Pk	
	name	Name of medicine	Varchar2	100		
	dosage	Dosage of medicine	Varchar2	10		
	description	Description of medicine	Varchar2	500		
	price	Price of medicine	Varchar2	10		
	quantity	Quantity of medicine	Number	3		

V_Treatment	Fk_visitID	Fk referencing visit	Varchar2	8	Pk,Fk	Visit

	Fk_treatmentID	Fk referencing treatmentID	Varchar2	8	Pk,Fk	Treatment
	Fk_toothID	Tooth of patient	Varchar2	100	Pk,Fk	Tooth
	Description	Description of patient treatment	Varchar2	500		
Treatment	treatmentID	Id of each treatment	Varchar 2	8	Pk	
	treatment	Treatment provided by the dental clinic	Varchar2	100		
	cost	Cost of each treatment	Number	12,2		
Organization	organizationID	Pk referencing organization	Varchar 2	8	Pk	
	organizationName	Organization name of organization	Varchar 2	100		
	telNo	Telephone number of organization	Varchar 2	12		
	personInCharge	Name of the person in charge	Varchar 2	100		
	address	Address of organization	Varchar 2	500		
Tooth	toothID	Primary key of tooth	Varchar 2	8	Pk	
	description	Description of tooth numbering	Varchar2	500		

Data Normalization

The normalization is displayed using relational schema.

i. Payment

1st Normalization:

Payment(patientID,name,visitID,visitDate,treatmentID,treatment,cost,toothID,description,medicineID,name,dosage,description,price,quantity,staffID,name,totalAmount,status)

2nd Normalization:

Patient(patientID,name)

Visit(visitID,visitDate)

Treatment(treatmentID,treatment,cost)

Tooth(toothID,description)

Medicine(medicineID,name,dosage,description,price,quantity)

Staff(staffID,name,totalAmount,status)

3rd Normalization:

Payment(paymentID,staffID,visitID,totalAmount,status)

ii. Visit

1st Normalization:

Visit(appointmentID,appointmentDate,time,patientID,name,treatmentID,treatment,cost,medicineID,name,dosage,description,price,quantity,toothID,description,visitDate)

2nd Normalization:

Appointment(appointmentID,appointmentDate,time)

Patient(patientID,name)

Treatment(treatmentID,treatment,cost)

Tooth(toothID,description)

Medicine(medicineID,name,dosage,description,price,quantity)

3rd Normalization:

Visit(visitID,appointmentID,patientID,visitDate)

4.3.4 Query Design

Select, insert and update subqueries has been used in this system and join queries of two tables which are appointment and patient table. Besides, insert and update stored procedure was used in this system too. Aggregate queries used to get the total amount of payment by using sum functions while count was used to get the total number of patient per month.

List of the queries in the system including aggregation, join and subqueries:

Table 4.2 List of Queries

Type of Query	Query	Description
Simple Query	Simple query used to select all from medicine	It used for the dentist to view list of medicine from medicine table including the price, description and current quantity
Simple Query	Simple query used to select patient ID and name from patient	It is used to display list of patient name for the dentist to make appointment

Simple Query	Simple query statement that select all from medicine	It is used for the staff to view data from medicine table
Join Query	Join query that is used to select visit ID from visit table and patient ID, and name from patient table	To display patient name based on the visiting time and date during patient registration
Join Query	Join query that is used to select appointment date and status from appointment table and patient ID from patient table order by appointment date	To display appointment date, time and status of the patient according to the date on patient page
Join Query	Join query that is used to select appointment date and status from appointment table and dentist ID from patient table order by appointment date	To display appointment date, time and status of the patient on dentist page according to the date
Join Query	Join query that is used to select all from visit table and patient ID from patient table by using sysdate	To display list of visiting patients according to visit date
Join Query	Join query between visit and patient table by having where clause where patient ID equivalent to the foreign key of patient ID in visit table order by appointment date	To display list of visiting patient name based on the date of treatment

Join Query	Join query that is used to select patient name from patient table and appointment ID, date, status from appointment table by having where clause which is appointment date is more than sysdate and order by appointment date	To display list of appointment according to the current date
Subquery	Select all information from dentist table where email is username and password is password	It is a query that is used to validate dentist id and password
Subquery	Select all information from staff table where email is username and password is password	It is a query that is used to validate staff id and password
Subquery	Select all information from patient table where email is username and password is password	It is a query that is used to validate patient id and password
Subquery	Select all information from patient table where email is patient ID	To display data of the patient from patient table for update
Subquery	Select all from patient by having where clause which is name is not null	To display all information of the patient during patient registration
Aggregate Query	Select sum from v_treatment and treatment	It is used to calculate cost of

		the treatment of the patient
Aggregate Query	Select sum from v_medicine and medicine	This select statement is used to calculate the total price of treatment and medicine
Aggregate Query	Select count all from patient where email is username and password is password	This query is used to know the existence of user ID

4.3.3 Physical Design

4.3.4.1 Selection of DBMS

Oracle SQL Developer is the database chosen for this system. Oracle SQL Developer is a free integrated development environment that simplifies the development and management of Oracle Database in both traditional and Cloud deployments.

4.3.4.2 Stored Procedure and Triggers

Insert, update and select stored procedure has been used in this system. Stored procedures allow faster execution and it can be used as a security mechanism. It also helps to execute SQL statement with a single call. Sequences statement is made for

auto-increment on every primary key of each table. There are several triggers that has been used and there are before insert and after insert triggers. Before insert trigger of auto-increment was used on all tables, status of appointment of each selection and trigger of 'No appointment can be made on Sunday' is triggered on appointment table. After insert trigger is used on medicine table as it will calculate current quantity of medicine.

Table 4.3 Stored Procedure used in Patient Module

Module	Stored Procedure	Description
Patient	Insert	Insert information of patient into patient table
	Insert	Insert appointment date and time into appointment table by patient
	Update	Update patient information on patient table

Table 4.4 Stored Procedure used in Dentist Module

Module	Stored Procedure	Description
Dentist	Insert	Insert appointment date and time into appointment table by patient
	Insert	Insert fk toothID, fk treatment ID, fk visit ID and description into v_treatment table

Table 4.5 Stored Procedure used in Staff Module

Module	Stored Procedure	Description

Staff	Insert	Insert fk visitID, fk staffID, totalAmount and status into payment table

Table 4.6 Triggers of DCMS

Trigger Name	Type	Description
Appt_status	Before insert	Appointment status was set 'pending' whenever user choose date and time
Auto_increment	Before insert	Auto increment of dentistID
Auto_increment_appt	Before insert	Auto increment of appointmentID
Auto_increment_medicine	Before insert	Auto increment of medicineID
Auto_increment_patient	Before insert	Auto increment of patientID
Auto_increment_payment	Before insert	Auto increment of paymentID
Auto_increment_Staff	Before insert	Auto increment of staffID
Auto_increment_tooth	Before insert	Auto increment of toothID
Auto_increment_treatment	Before insert	Auto increment of treatmentID
Auto_increment_visit	Before insert	Auto increment of visitID
Auto_increment_vmedicine	Before insert	Auto increment of vmedicineID
Auto_increment_vtreatment	Before insert	Auto increment of vtreatmentID
Backup_patient	After insert or update	Insertion of new patient information or updates will be inserted or updated on patient_backup table
Med_qty	After insert	Medicine quantity will be deducted by 1 on medicine table,

		whenever the dentist click on the medicine button. Staff able to view the current medicine quantity.
No_appt	Before insert or update	This trigger disallow user from setting an appointment on Sunday

4.3.4.3 Security Mechanism

There are several security mechanisms that has been implemented in this system. Each user is identified by username and password. A user can set their own password and only the matching password allowed to access the system.

4.4 Graphical User Interface (GUI) Design

4.4.1 Navigation Design

Figure 4.1 shows the navigation design of DCMS. Users need to login before access to the system. Patient need to sign-up and they can update their profile and set an appointment. A staff is able to view appointment list, register visit patient, view patient profile, view medicine inventory, receive payment and generate reports. A dentist is able to set appointment, do treatment on patient and give out medicine prescription.

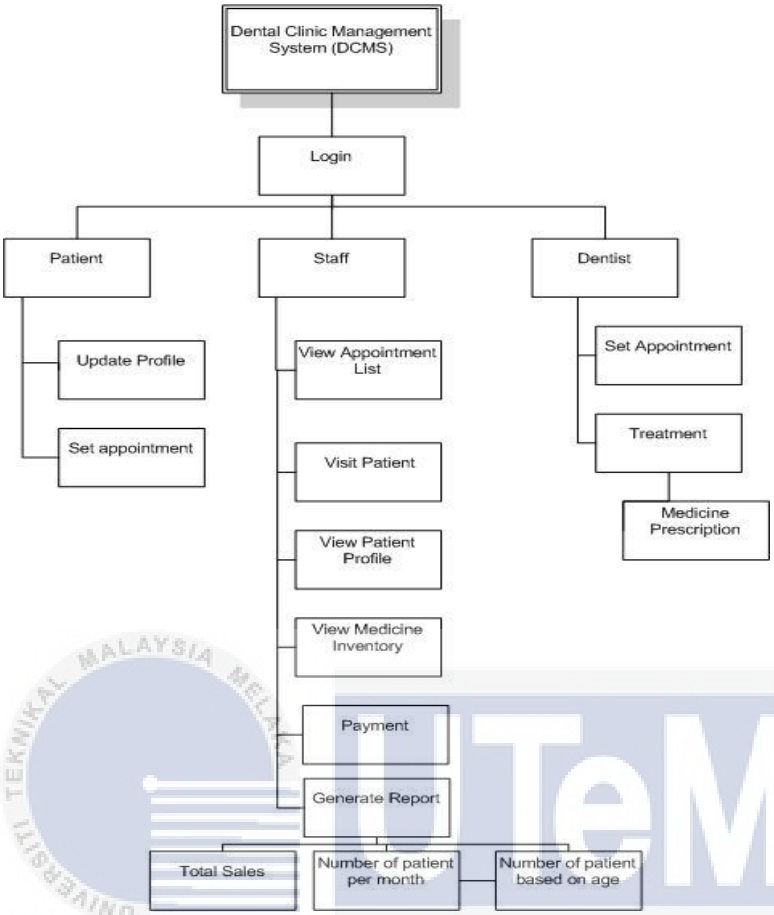


Figure 4.3 Navigation Design of DCMS

4.4.2 Input Design

Figure 4.4 shows login interface of DCMS. There are three users which are dentist, staff and patient. Table 4.7 shows the input design of login interface.

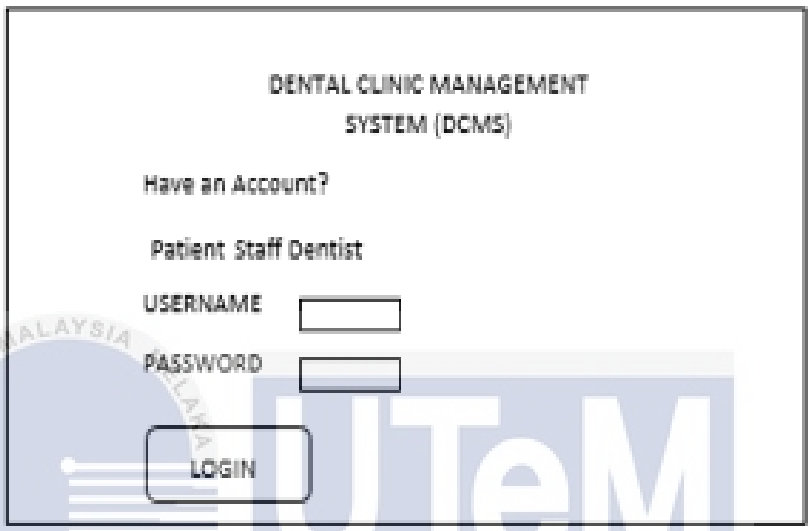


Figure 4.4: Login Interface

Table 4.7 Input Design of Login Interface

Attribute	**Data type**	**Data(eg)**	**Validation**
Username	Varchar2(50)	wan@gmail.com	Required
Password	Varchar2(20)	123abc	Required

Figure 4.5 shows the registration patient interface.

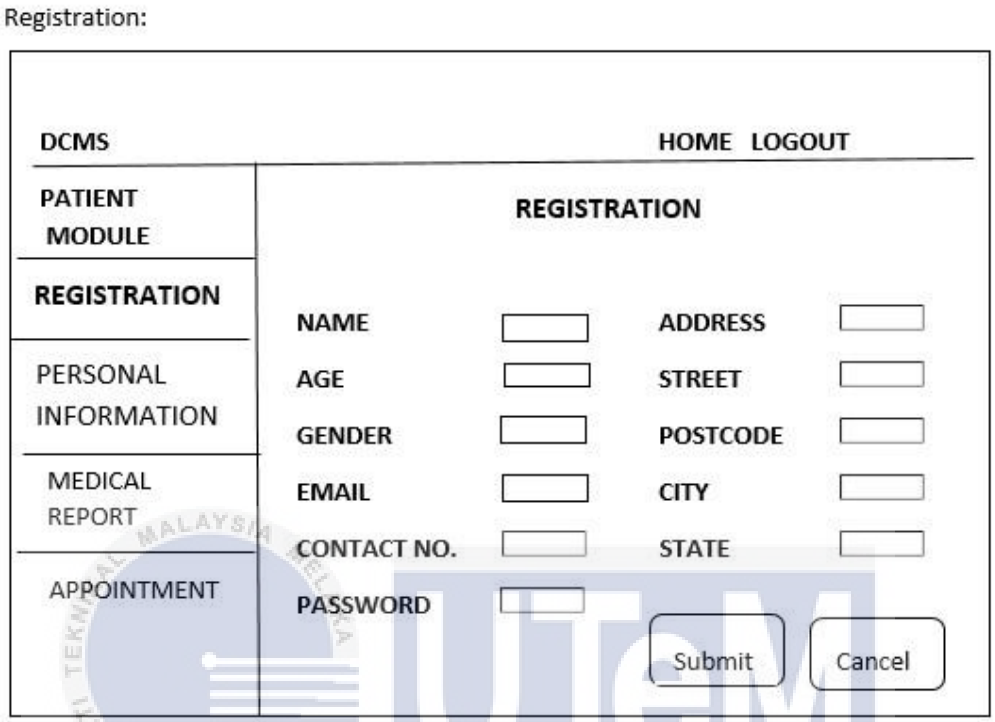


Figure 4.5: Registration Interface

Table 4.8 Input Design of Registration Interface

Attribute	**Data type**	**Data(eg)**	**Validation**
Name	Varchar2(100)	Wan Noraqilah	Required
Age	Number	22	Required
Email	Varchar2(50)	wan@gmail.com	Required
Address	Varchar2(100)	Bukit Beruang	Required
Gender	Varchar2(6)	Female	Required
Phone Number	Varchar2(12)	0123344567	Required
Password	Varchar2(20)	123abc	Required
State	Varchar2(100)	Melaka	Required
Street	Varchar2(100)	Jalan Bukit Beruang Bestari	Required
City	Varchar2(100)	Bukit Beruang	Required
Postcode	Number	79539	Required

Figure 4.6 shows personal information interface. It is used for the patient to update their personal information.

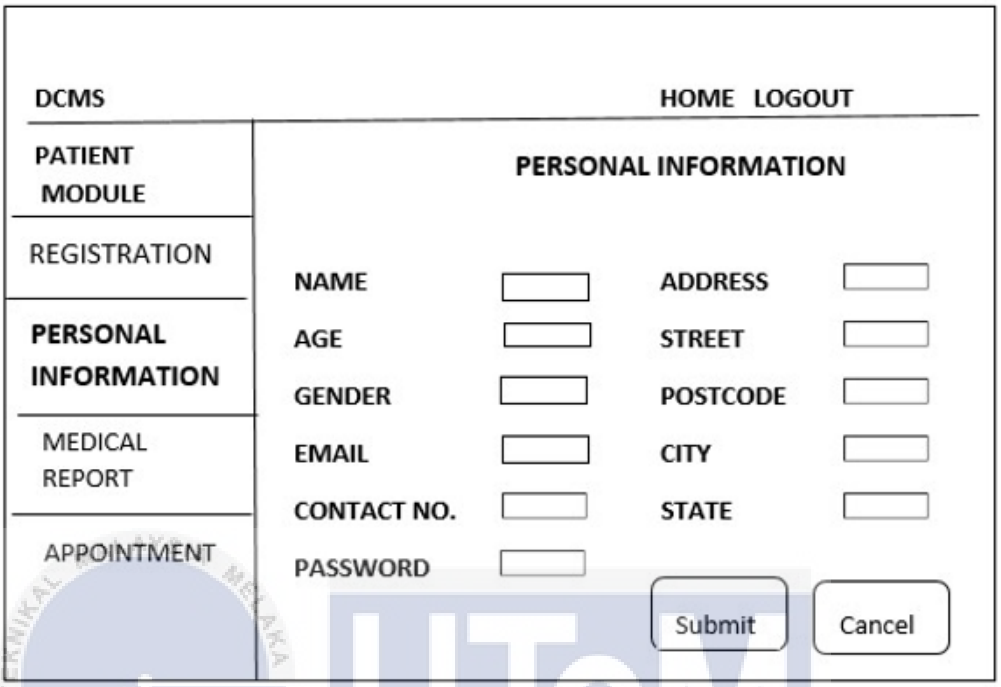


Figure 4.6: Personal Information Interface

Table 4.9 Input Design of Personal Information Interface

Attribute	**Data type**	**Data(eg)**	**Validation**
Name	Varchar2(100)	Wan Noraqilah	Required
Age	Number	22	Required
Email	Varchar2(50)	wan@gmail.com	Required
Address	Varchar2(100)	Bukit Beruang	Required
Gender	Varchar2(6)	Female	Required
Phone Number	Varchar2(12)	0123344567	Required
Password	Varchar2(20)	123abc	Required
State	Varchar2(100)	Melaka	Required
Street	Varchar2(100)	Jalan Bukit Beruang Bestari	Required
City	Varchar2(100)	Bukit Beruang	Required
Postcode	Number	79539	Required

Figure 4.7 shows treatment interface used by dentist.

Figure 4.7: Treatment Interface

Table 4.10: Input design of Treatment Interface

Attribute	**Data type**	**Data(eg)**	**Validation**
Treatment	Varchar2(100)	Bridged	Required
Patient Name	Varchar2(100)	Wan	Required
Tooth	Varchar2(100)	1-1 incisors	Required
Medicine	Varchar2(100)	Amoxilin Suspension	Required

Figure 4.8 shows medicine interface on staff module

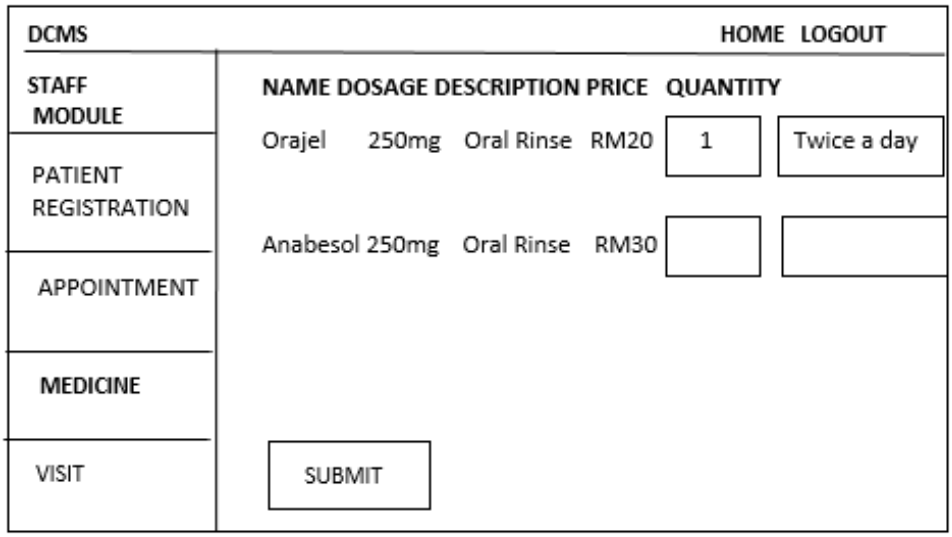


Figure 4.8: Medicine Interface on Staff Module

Table 4.11: Input design of Medicine Interface on Staff Module

Attribute	Data type	Data(eg)	Validation
Quantity	Number	1	Required
Description	Varchar2(500)	Twice a day	Required

Figure 4.9 shows appointment interface

Figure 4.9: Appointment Interface

Table 4.12: Input design of Appointment Interface

Attribute	Data type	Data	Validation
Date	Timestamp	20/6/2017 10:00 a.m.	Required

Figure 4.10 shows payment interface on staff module.

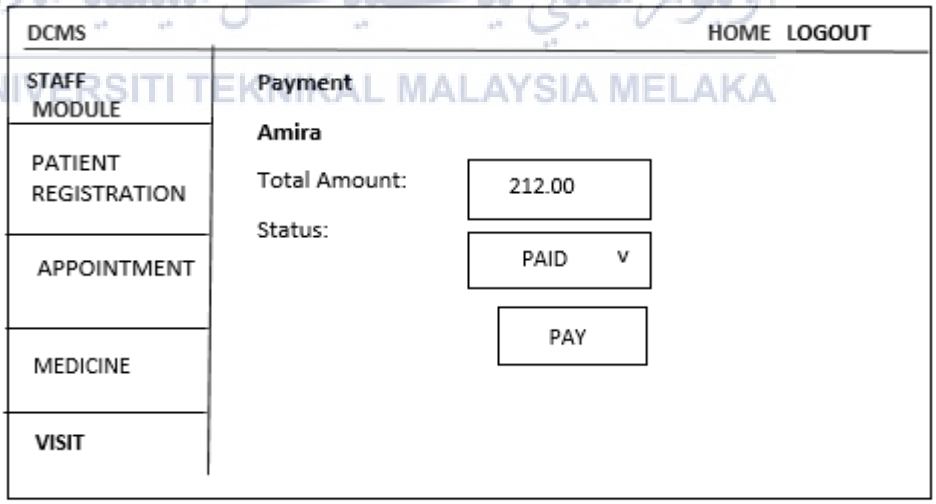


Figure 4.10: Payment Interface

Table 4.13: Input design of Payment Interface

Attribute	Data type	Data(eg)	Validation

Status	Varchar2(10)	Paid	Required

4.4.3 Output Design

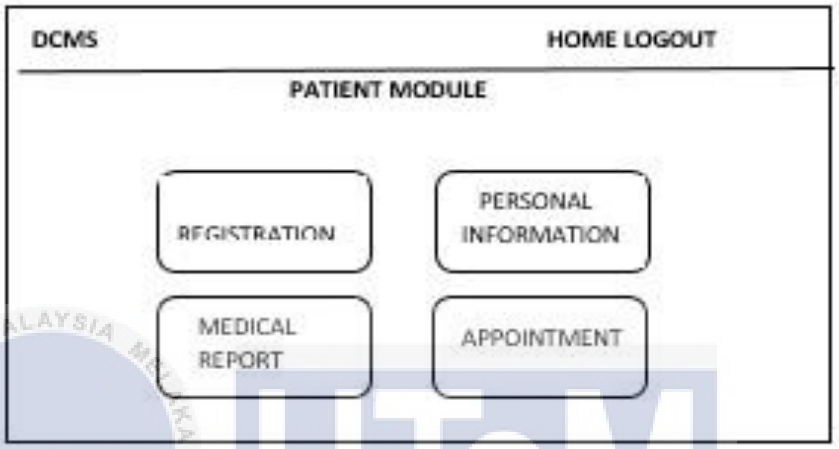


Figure 4.11: Patient Module Interface

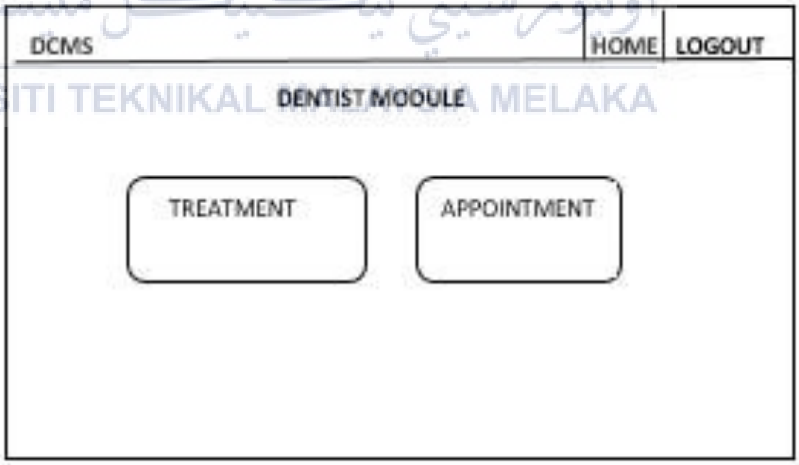


Figure 4.12: Dentist Module Interface

Staff Module:

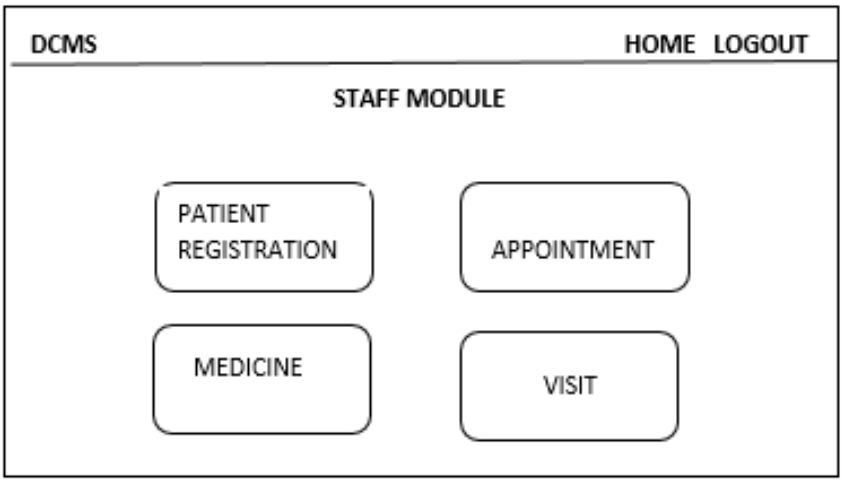


Figure 4.13: Staff Module Interface

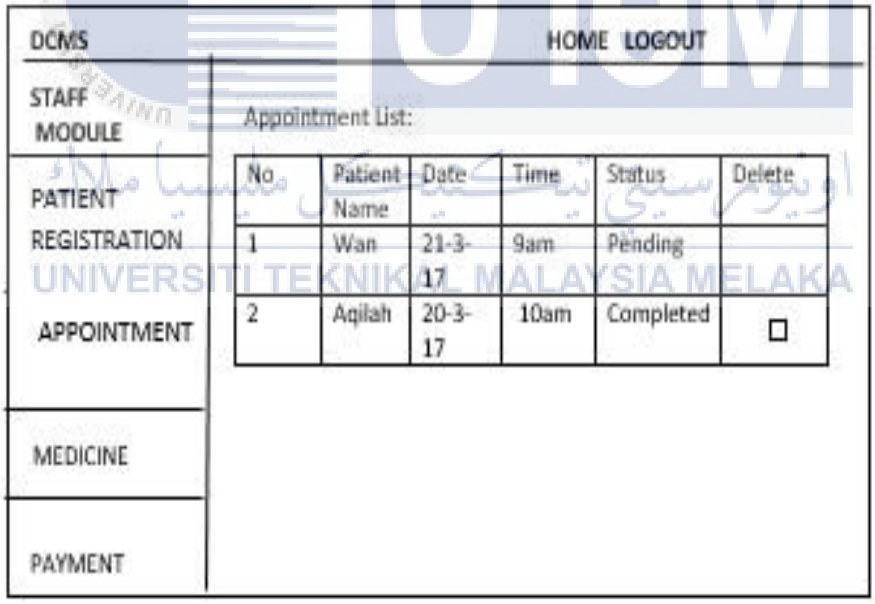


Figure 4.14: Appointment Interface on Staff Module

List of Medicine:

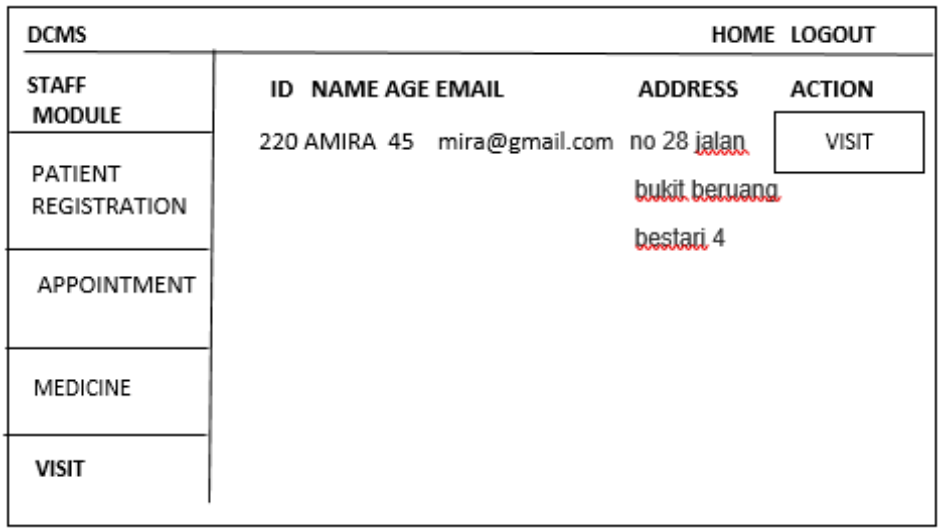


Figure 4.16: Visit Interface on Staff Module

Payment:

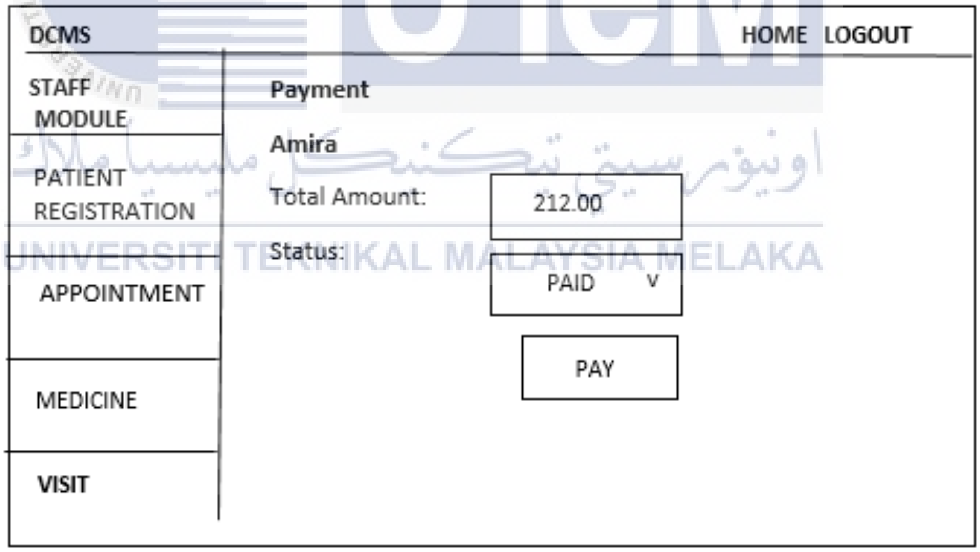


Figure 4.17: Payment Interface on Staff Module

Payment Receipt:

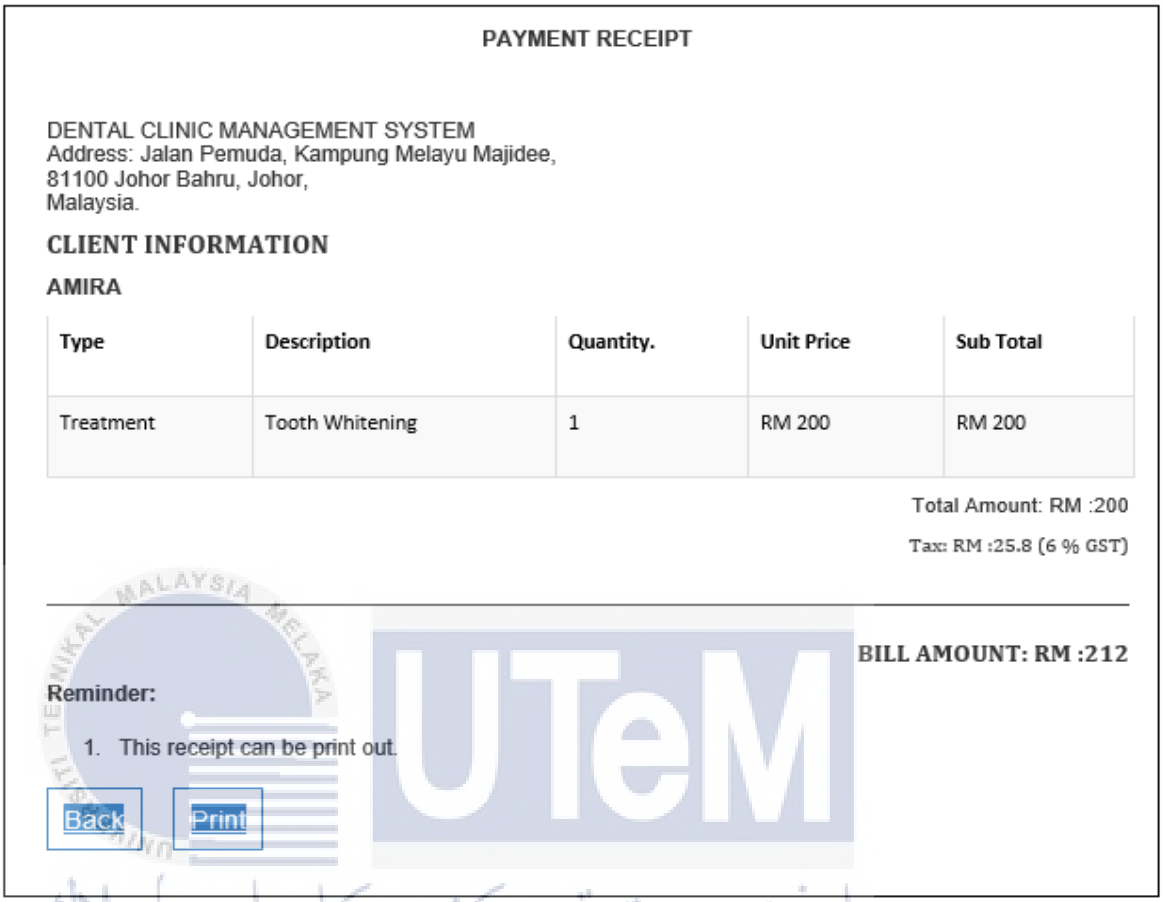


Figure 4.18: Payment Receipt Interface on Staff Module

Report

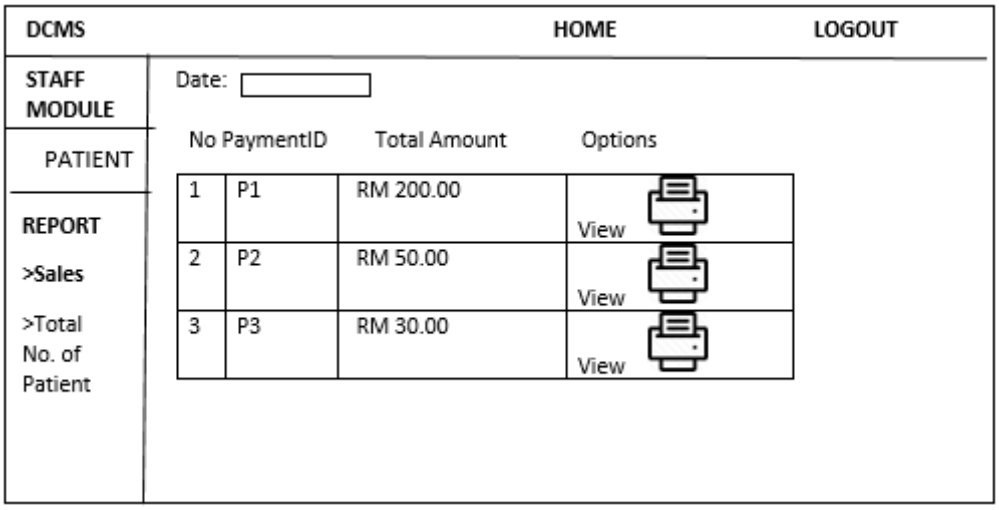


Figure 4.19: Report Interface on Staff Module

4.5 Conclusion

In this chapter, the design stage of this system has been explained. Design includes the ERD, query design, stored procedure, triggers, security mechanism and the user interface. For the next chapter, implementation phase will be explained in more details.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

In this chapter, the implementation of the system will be explained. Implementation includes the installation of Oracle SQL Developer, DDL statements and the implementation of stored procedures and triggers. All the stored procedures and triggers will be shown in this chapter.

5.2 System Development Environment setup

DCMS used Hypertext Preprocessor(PHP) Language, Oracle SQL Developer database and xampp version 3.2.2 that act as a server to integrate connection between database and interfaces of the system.

5.2.1 Server Development Environment setup

Step 1: Anti-virus must be disabled as it can cause some XAMPP components to behave erratically.

Step **2:** To begin installation, Open/double-click .exe file On opening the file, a XAMPP set up wizard will initiate

Figure 5.1: Xampp Installation

Step 3: Choose the components by clicking on the checkbox. Then, click 'Next'.

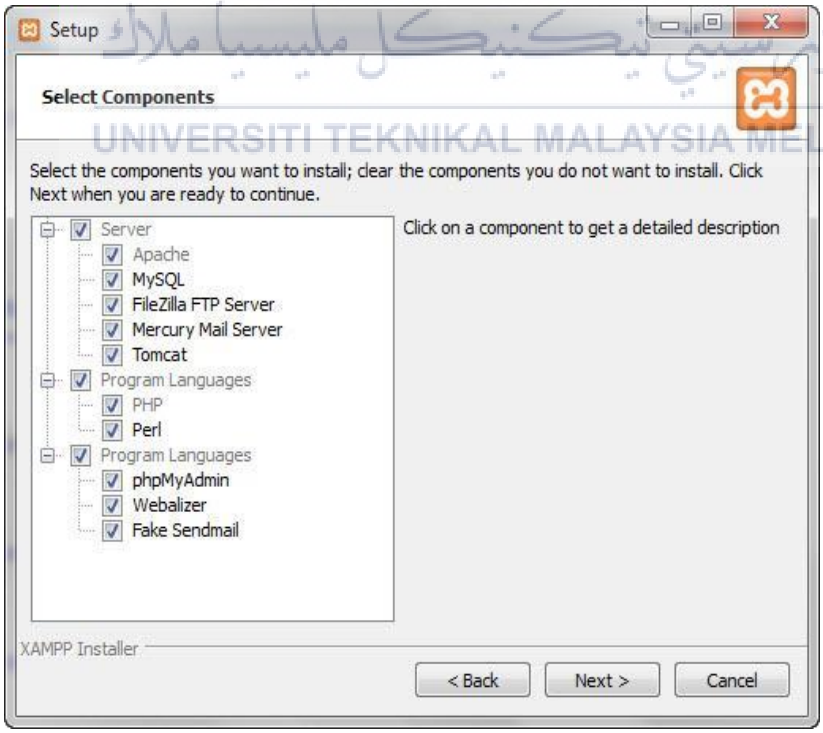


Figure 5.2: Select Component of Xampp Installation

Step 3: Click 'Browse' and choose Local Disk(C) **to** install the XAMPP. Then, click 'Next'.

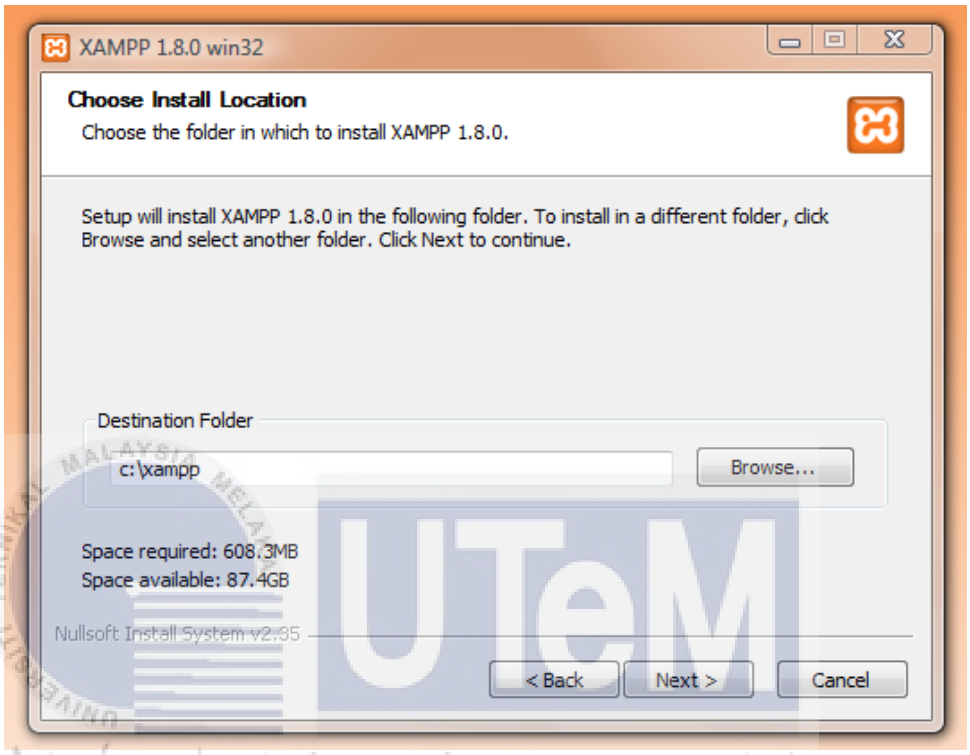


Figure 5.3: Choose Install Location of Xampp Installation

Figure below shows the Xampp folder that has been saved on Local Disk(C.)

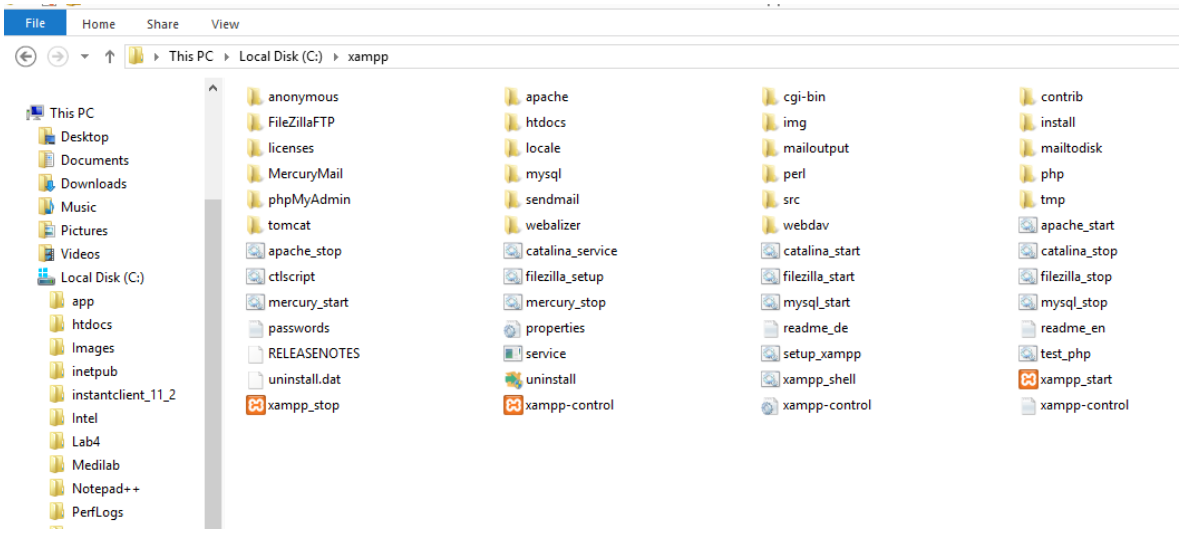


Figure 5.4: Xampp Folder on Local Disk(C)

Step 4: This is the installing page and it takes less than 10 minutes.

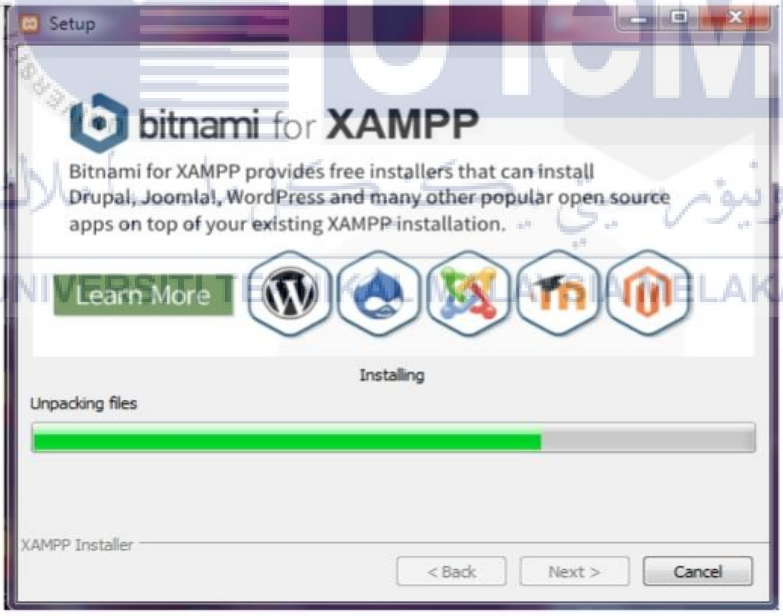


Figure 5.5: Installing Xampp

Step 5: On successful completion of Installation, following screen will appear. Then, click ‘Finish’.

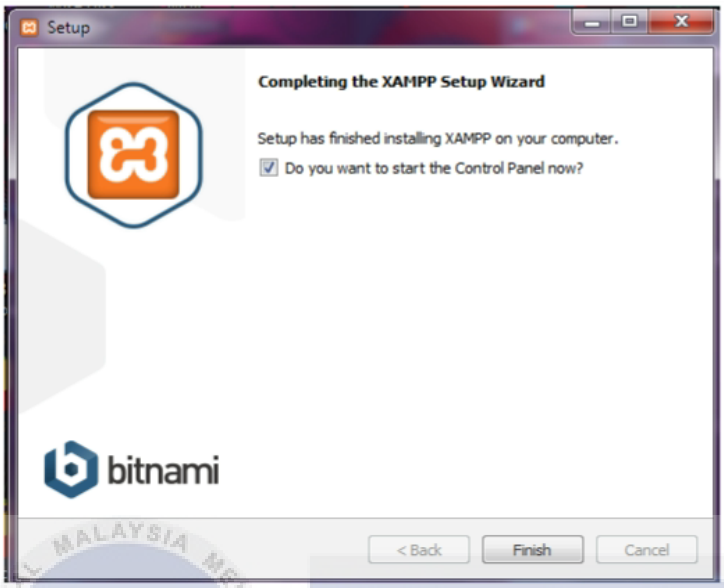


Figure 5.6: Completion of Xampp Installation

Step 6: Then the control panel screen will automatically launch. Click on ‘Apache’ to continue.

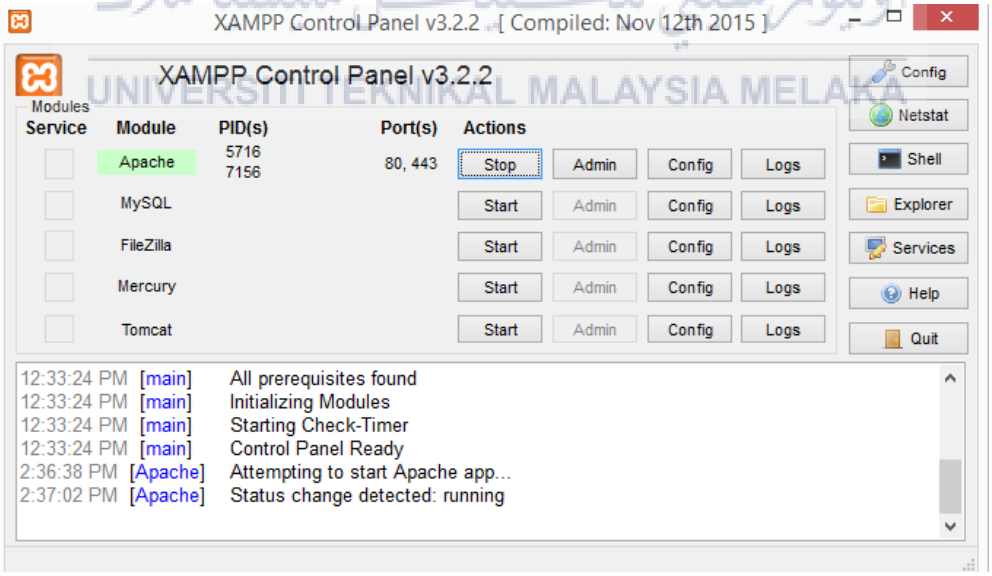


Figure 5.7: Xampp Control Panel v3.2.

5.2.2 Database Development Environment setup

Installation of Oracle 11g Enterprise Edition

Step 1: Go to the following site to download Oracle database 11g for Windows, http://www.oracle.com.

Step 2: Click free download and unachieved the zipped download file.

Step 3: Double click on setup.exe from unachieved folder of oracle database source.

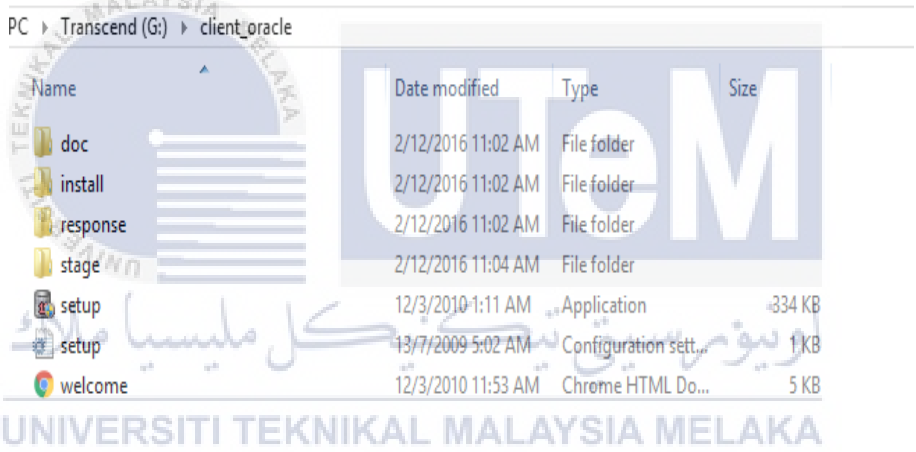


Figure 5.8: Oracle Database 11g Folder

Step 4: Email is needed for security updates and user can uncheck the box. Then, click Next.

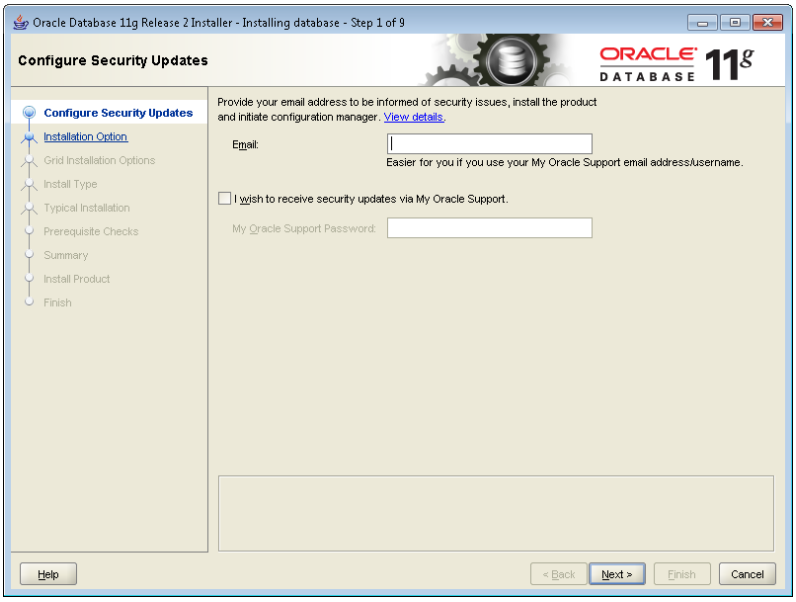


Figure 5.9: Oracle 11g Database Installation

Step 5: Tick on 'create and configure a database' radio button and click Next to continue.

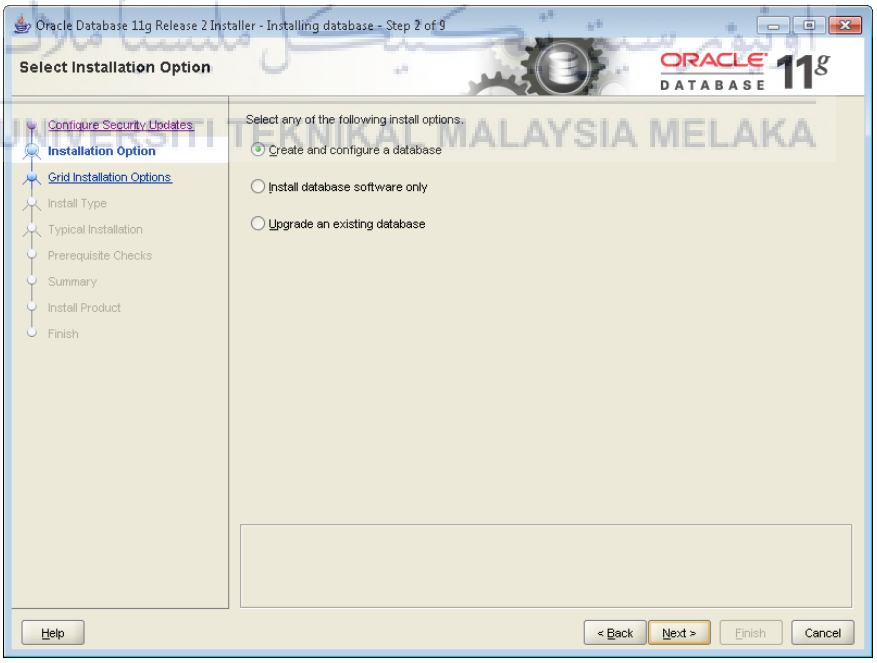


Figure 5.10: Oracle 11g Database Installation Option

Step 6: Tick on ‘desktop class’ then click Next.

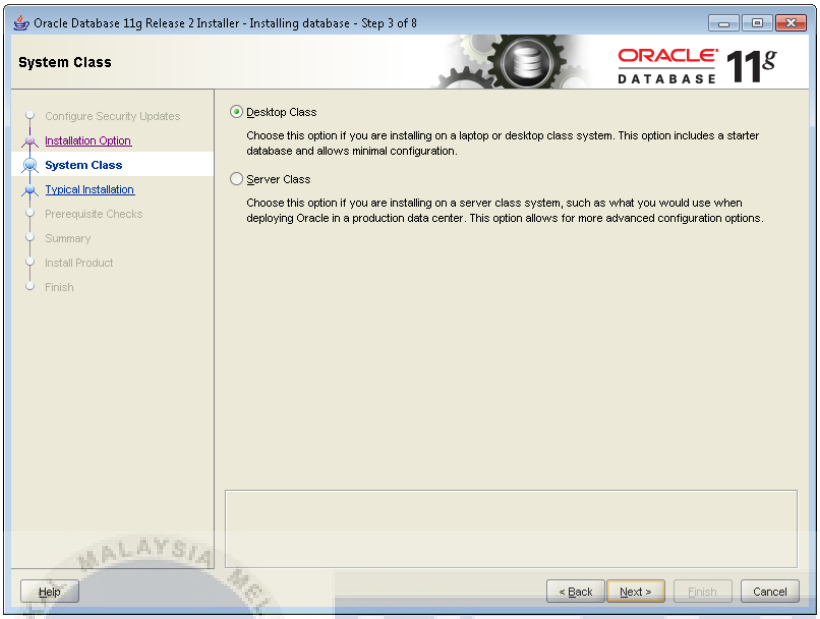


Figure 5.11: Oracle 11g Database System Class

Step 7: Enter the global database name as ‘orcl’ then enter password twice. Click Next button to continue.

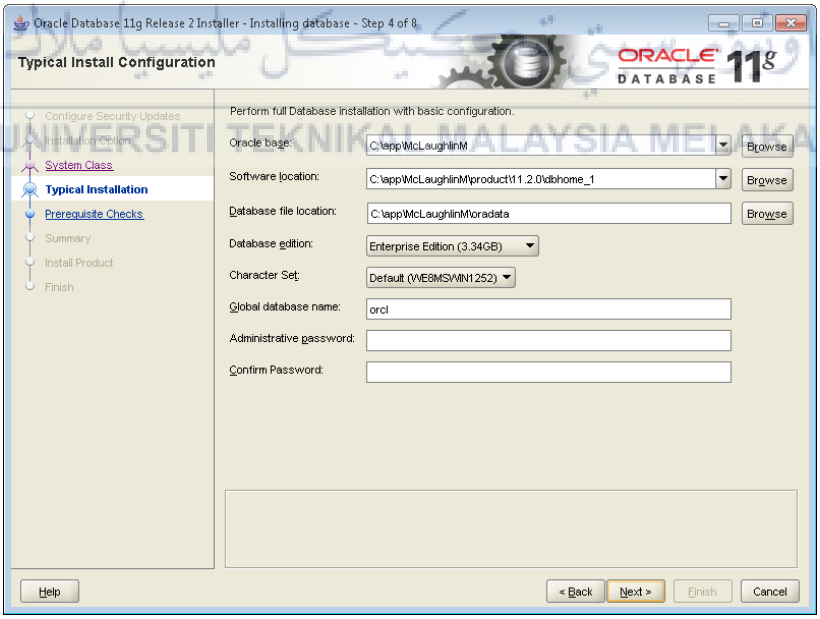


Figure 5.12: Typical Installation of Oracle 11g Database

Step 8: This is a progress bar screen that checks for prerequisites. It takes less than a minute to run but can take two or so. Then, it will proceed to the summary page.

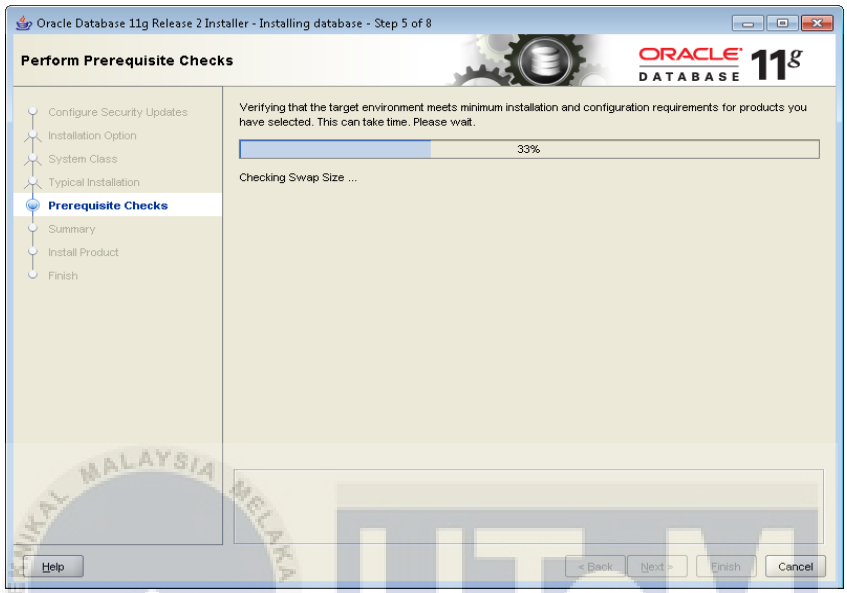


Figure 5.13: Prerequisite Checks of Oracle 11g Database

Step 9: The summary shows all the selected values for the installation. Click Finish button to proceed.

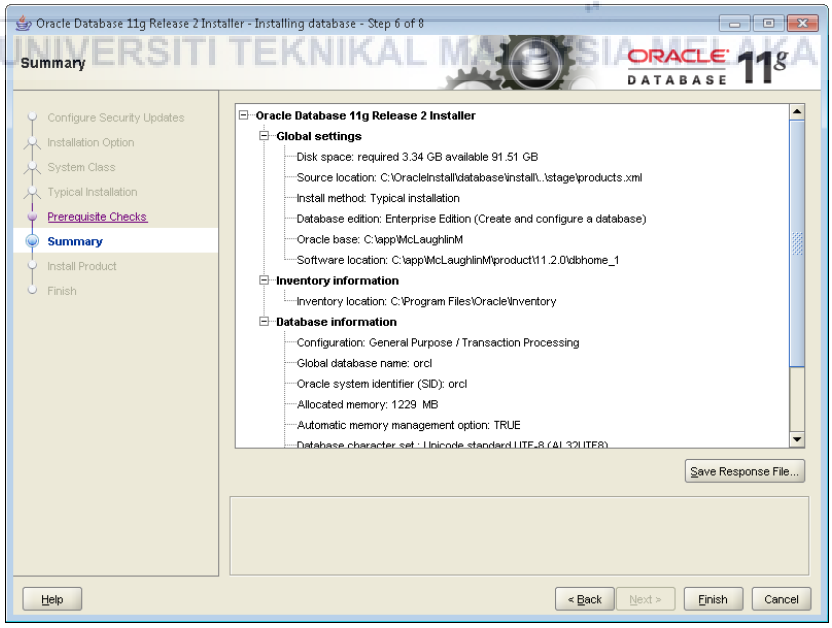


Figure 5.14: Summary Installation of Oracle

Step 10: This is the main progress bar and it normally takes 5 to 10 minutes.

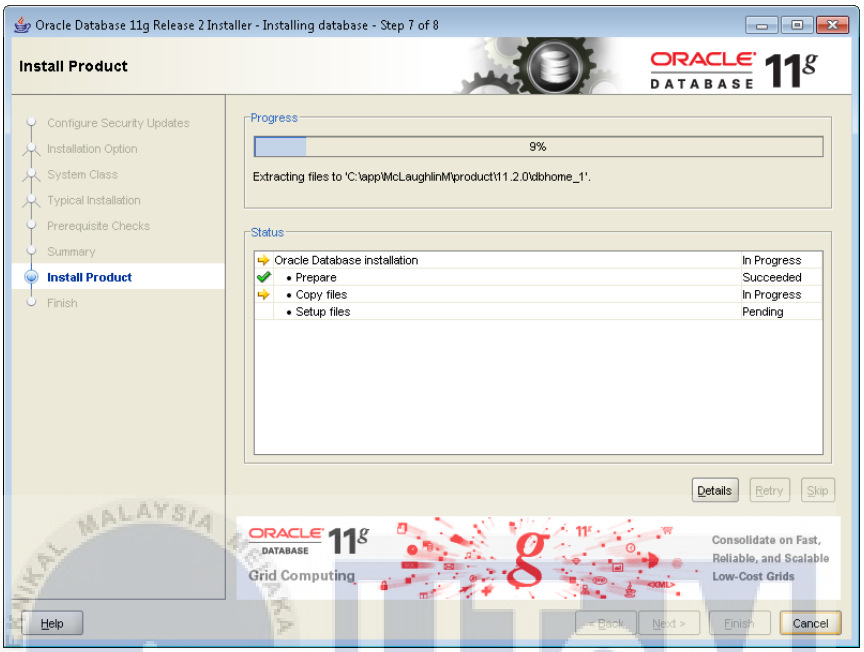


Figure 5.15: Install Product of Oracle 11g Database

Step 11: The popup screen requires the authorization of the Java runtime to call home. Click Allow access to continue.

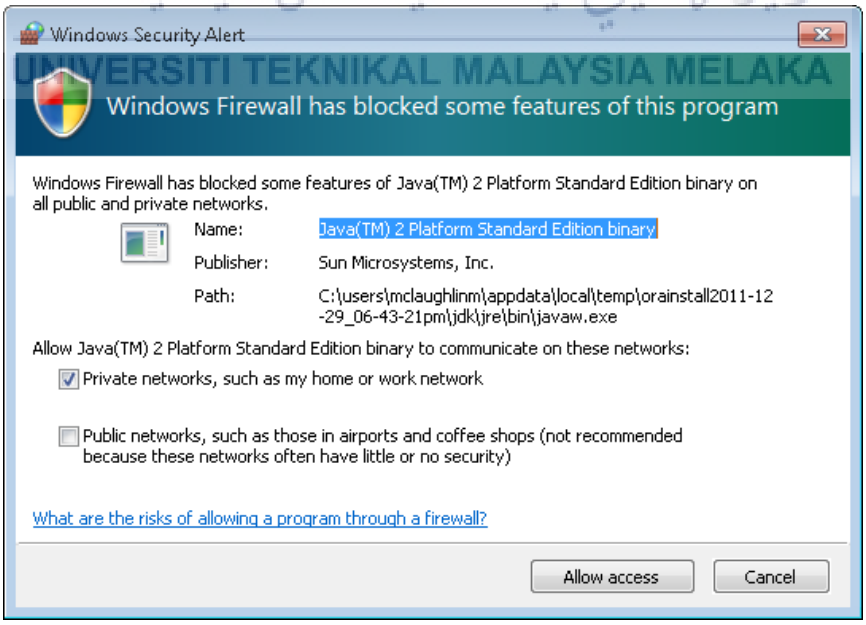


Figure 5.16: Windows Security Alert

Step 12: On this page, failures may occur as the progress bar configuring the instance. If there is no failure, it will proceed to the next page.

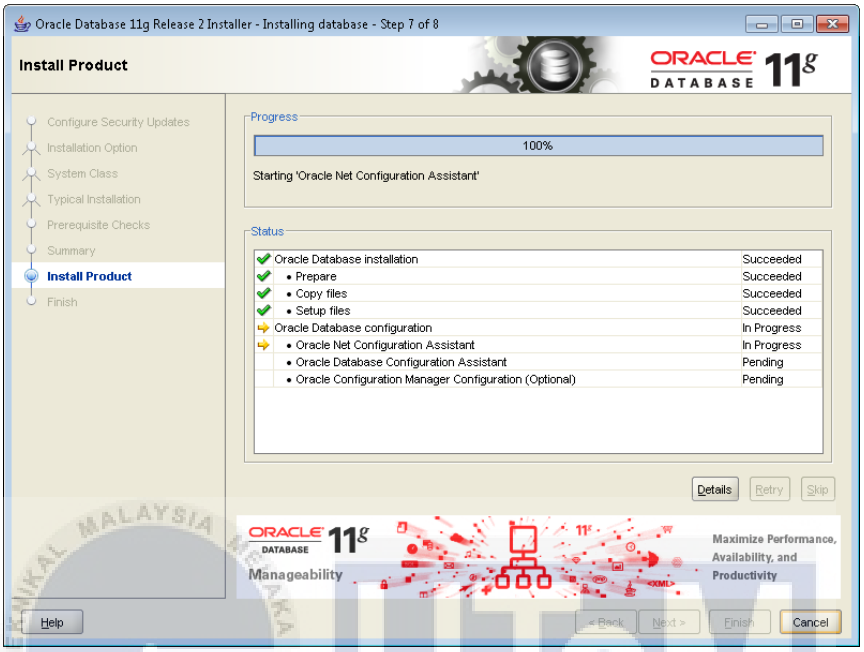


Figure 5.17: Configuring the Instance of Oracle 11g Database

Step 13: This progress bar is launched by the Oracle Database Configuration Assistant and it takes a few minutes.

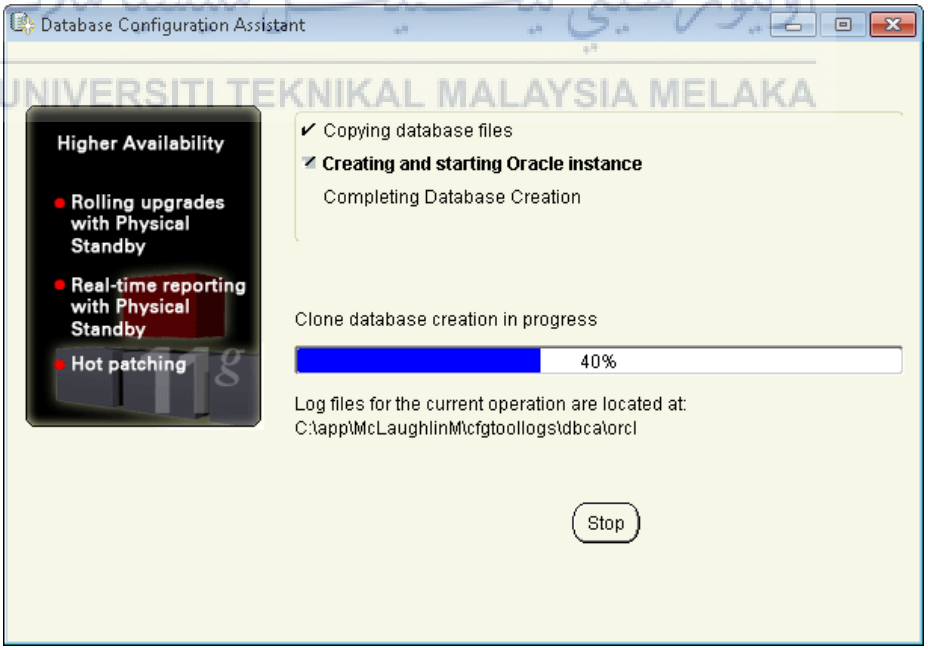


Figure 5.18: Database Configuration Assistant

Step 14: The second popup requires the authorization of the Java runtime to call home. Then, click Allow access to proceed.

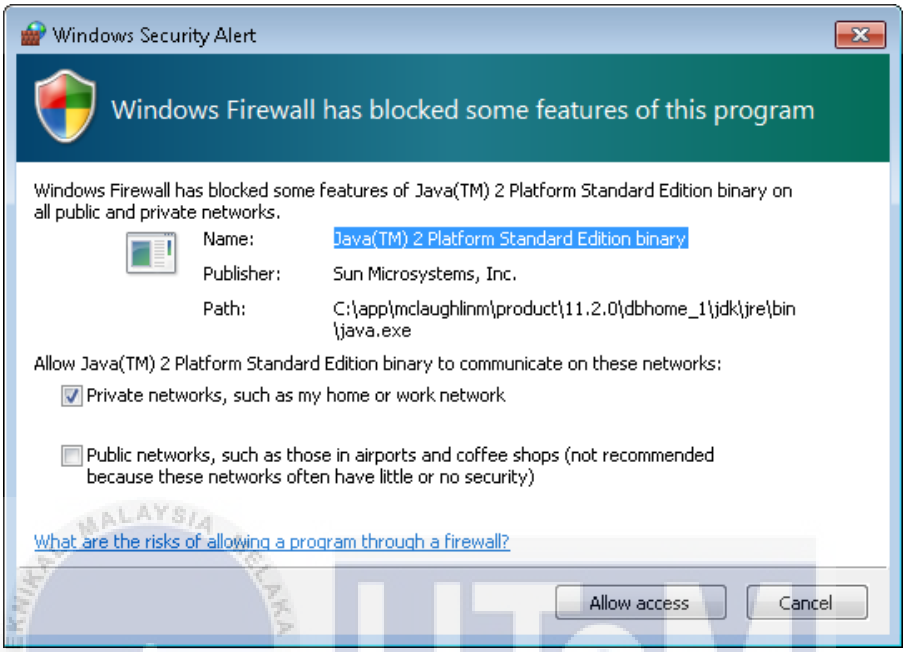


Figure 5.19: Windows Security Alert

Step 15: The third popup asks whether the user want to open the other scheme. Click OK button to proceed.

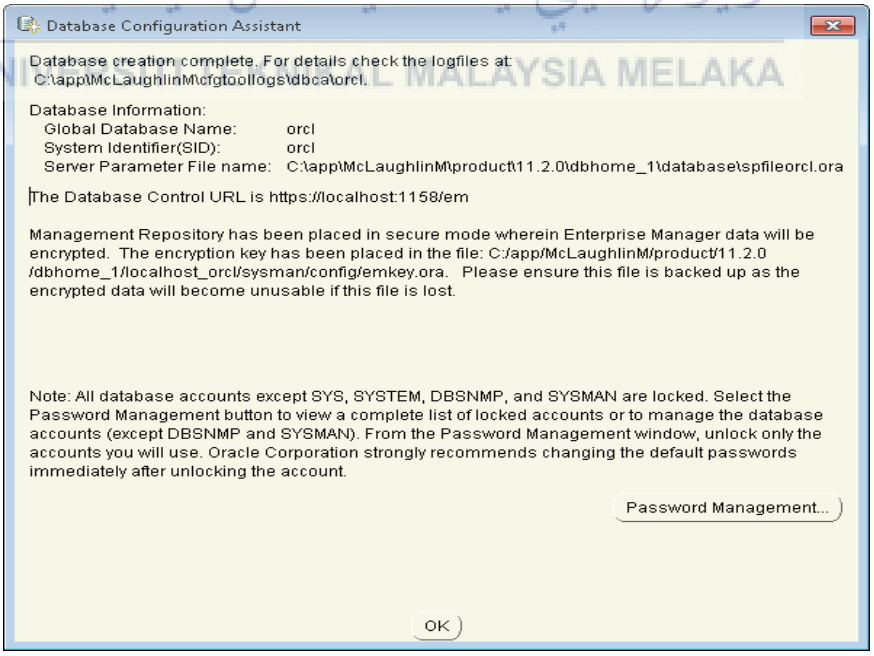


Figure 5.20: Database Configuration Assistant

Step 16: This page shows that the network and database instances are configured and it also run the OEM(Oracle Enterprise Manager) installation.

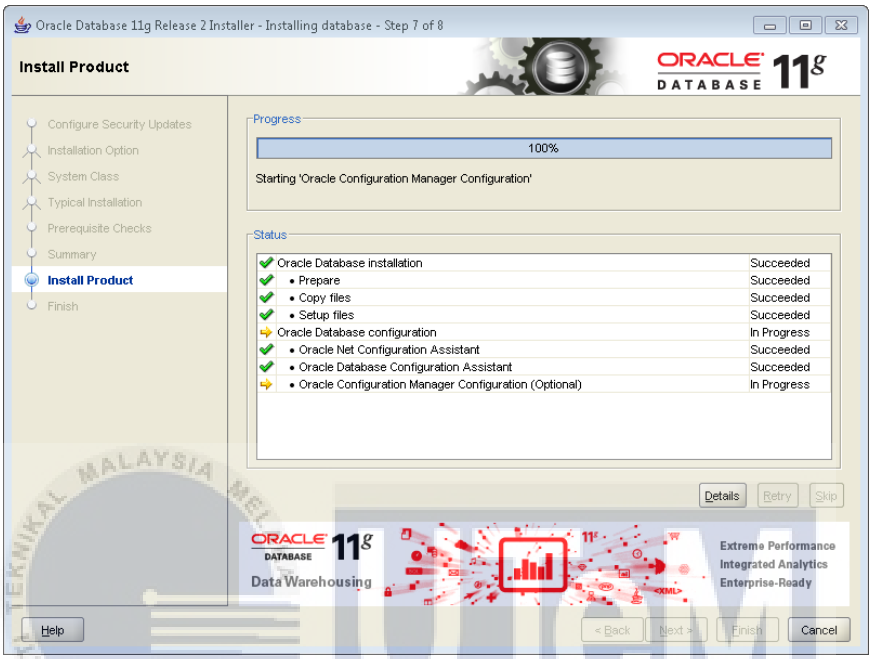


Figure 5.21: Completion of Installation of Oracle 11g Database

Step 17: Open SQL Developer

Figure 5.22: Oracle SQL Developer

5.2.3: Managing database objects using Oracle SQL Developer

Step 1: Enter service name for the Connection Name, then click Test.

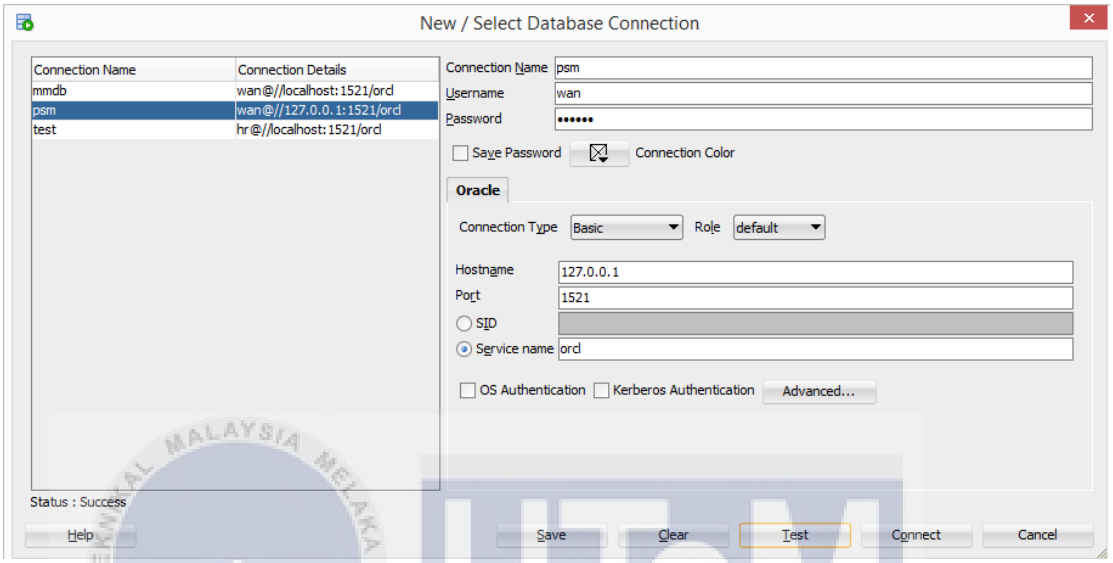


Figure 5.23: Create New Connection

Step 2: The status of the connection was tested successfully. The connection was not saved however. To save the connection, click Connect.

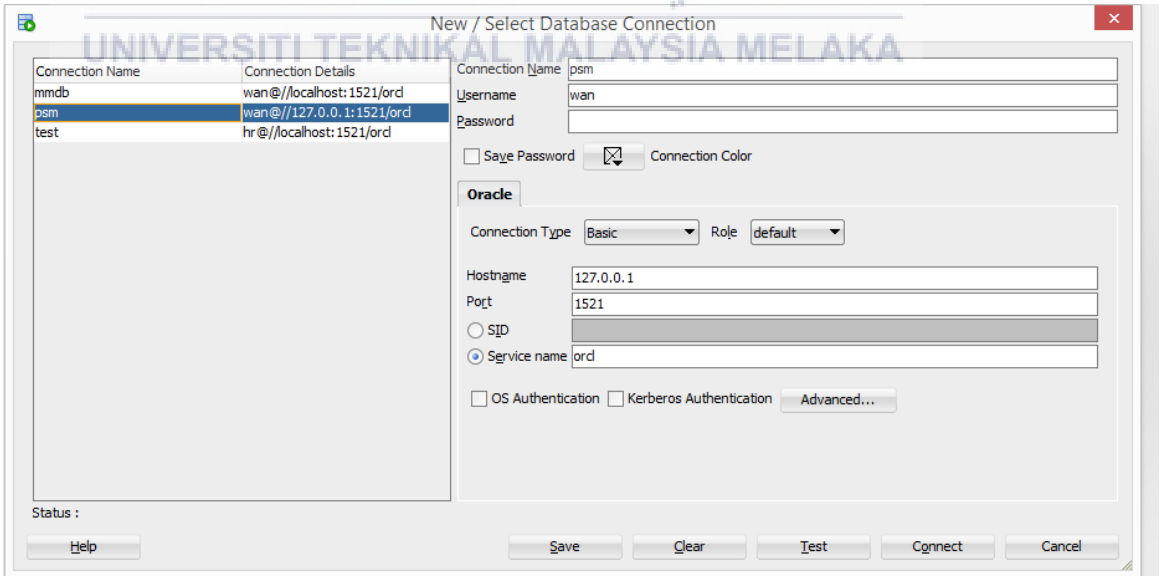


Figure 5.24: Test New Database Connection

Step 3: In the Connections tab, right-click Connection and select 'New Database Connection'.

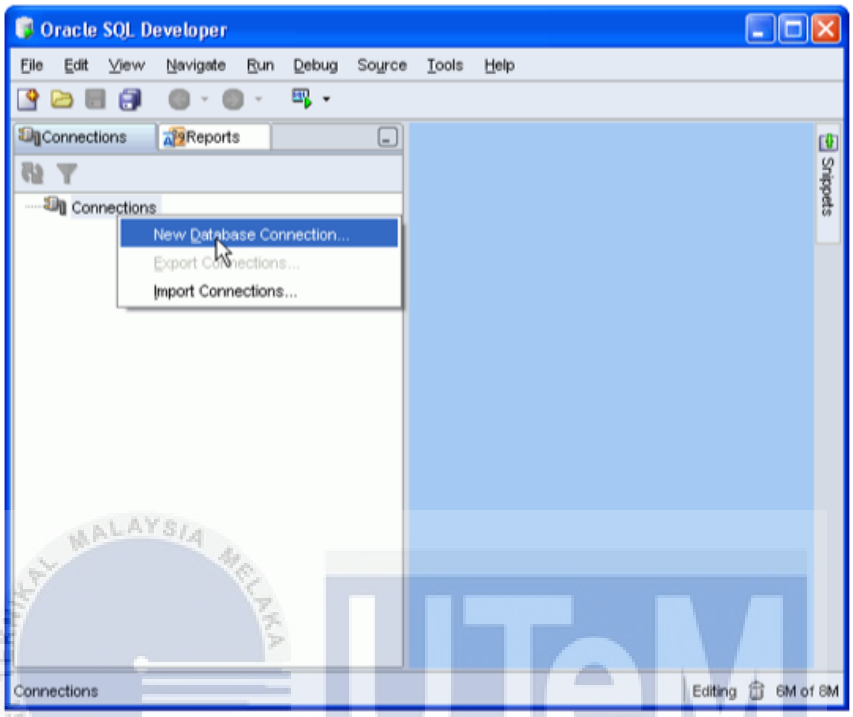


Figure 5.25: Create New Database Connection

Step 4: The connection was saved and the database is in the list. Expand **PSM**.

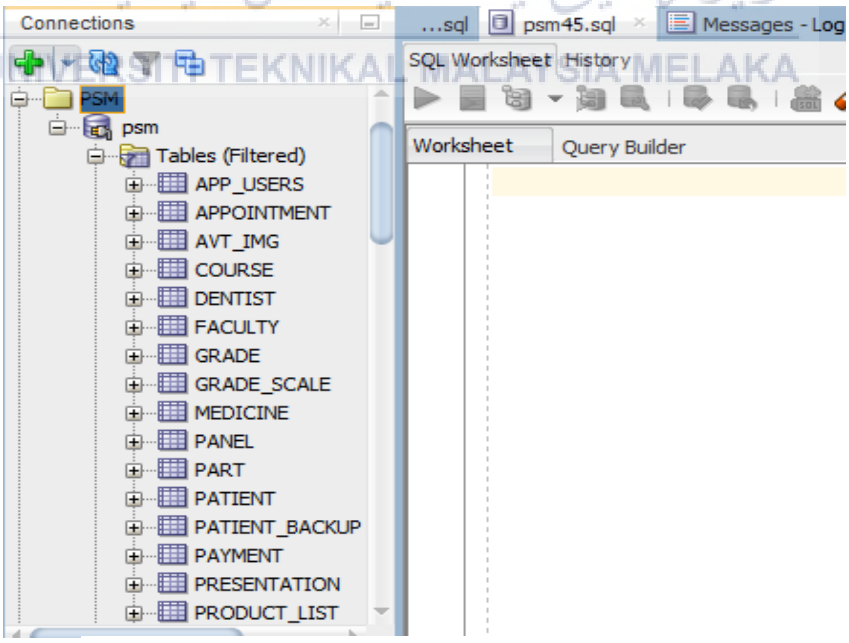


Figure 5.26: List of Table in PSM Connection

Step 5: Select the table to view the table definition.

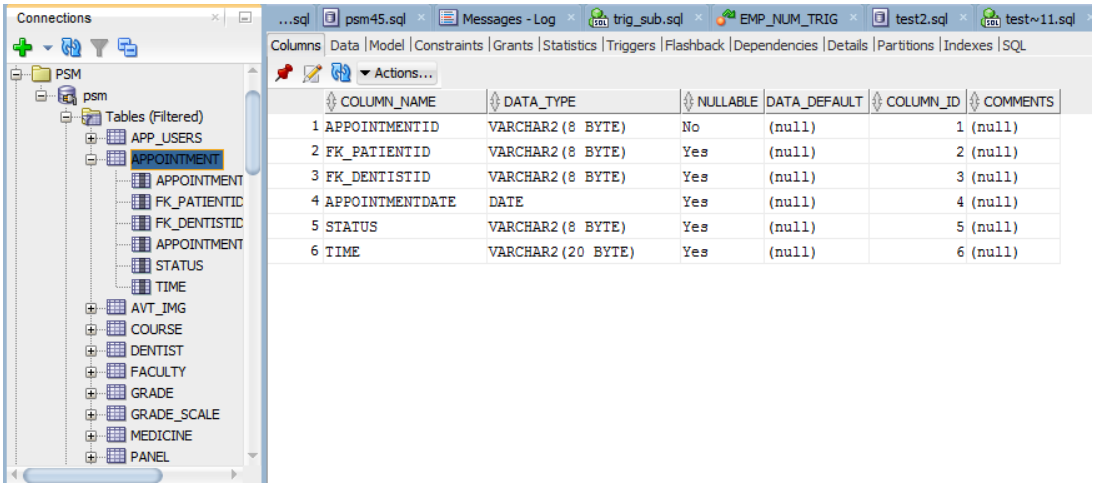


Figure 5.27: Appointment Table in PSM

5.2.4 PHP connection with Oracle SQL Developer

Step 1: Review the code in connect.php The oci_connect() function contains the username, the password and the connection string. The oci_close() function closes the connection. Any standard connections not explicitly closed will be automatically released when the script ends.

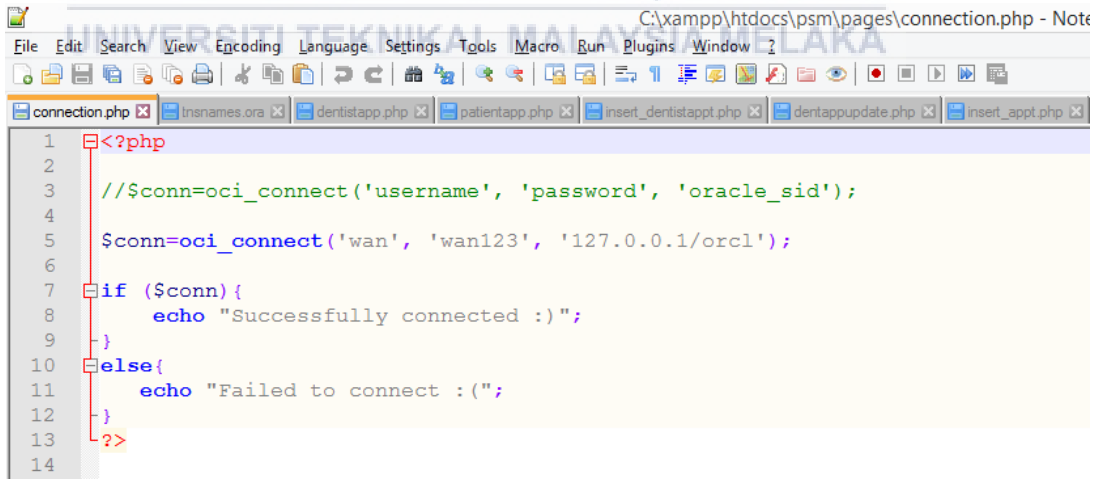


Figure 5.28: Connection.php coding

Step 2: Review the code in tnsnames.ora. The hostname '127.0.0.1' is put on host while port is replaced to 1521 port based on the oracle port.

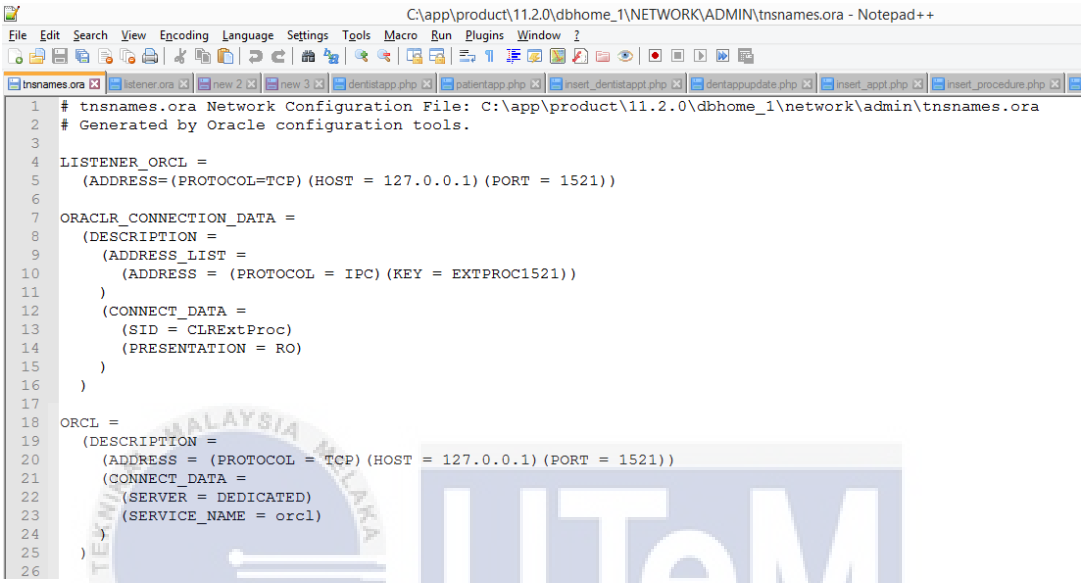


Figure 5.29: tnsnames.ora coding

Step 3: Review the code in listener.ora. The hostname '127.0.0.1' is put on host while port is replaced to 1521 port based on the oracle port.

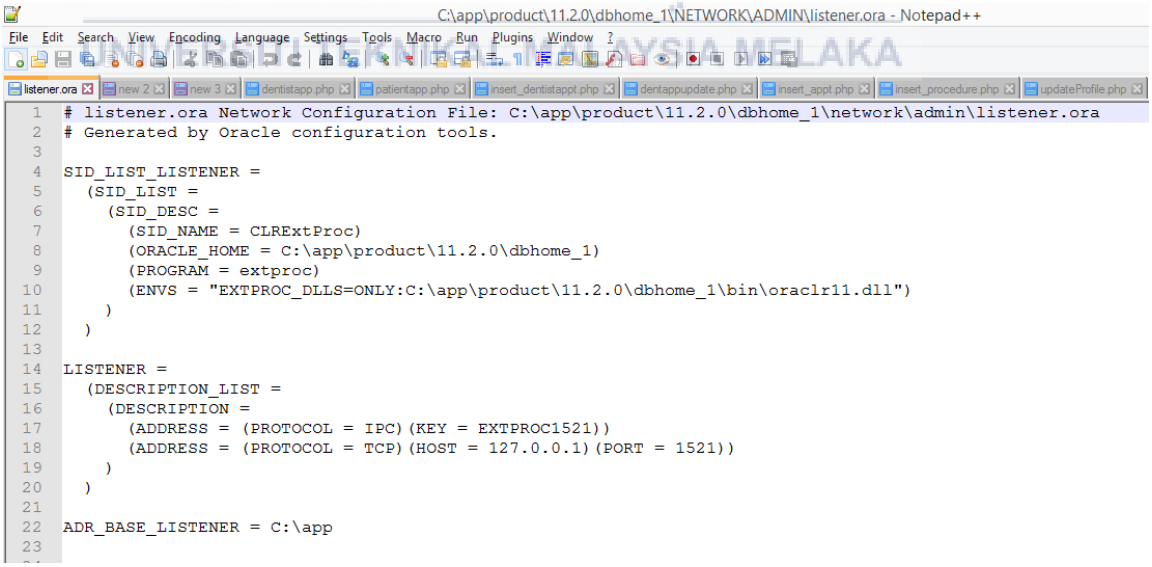


Figure 5.30: listener.ora coding

Step 4: Open a Web browser and enter the following URL to display the output:

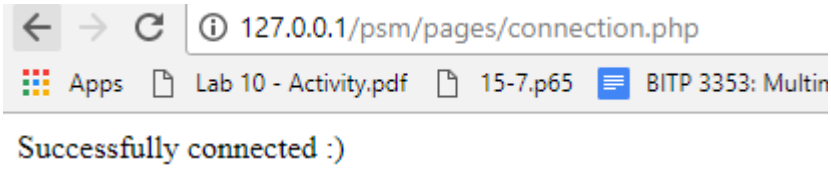


Figure 5.31: Connection on web browser

5.3 Database Implementation

5.3.1: Data Definition Language(DDL)

Create Statement:

i. Appointment Table:

```
create table appointment
(
appointmentID varchar2(8) primary key not null,
fk_patientID references patient(patientID),
fk_dentistID  references dentist(dentistID),
appointmentDate date,
time timestamp ,
status varchar2(8)
);
```

ii. Dentist Table:

```
create table dentist(
dentistID varchar2(8) primary key not null,
name varchar2(100),
age number,
email varchar2(50),
address varchar2(100),
street varchar2(100),
city varchar2(100),
postcode number,
```

```
state varchar2(100),
gender varchar2(6),
phoneNumber varchar2(12),
password varchar2(20)
);
```

iii. Medicine Table:

```
create table medicine
(
medicineID varchar2(8) primary key not null,
name varchar2(100),
dosage varchar2(10),
description varchar2(500),
price varchar2(10),
quantity number
);
```

iv. Organization Table:

```
create table organization
(
organizationID  varchar2(8)not null primary key,
organizationName varchar2(100)
telNo varchar2(10),
personInCharge varchar2(100),
address varchar2(500)
);
```

v. Patient Table:

```
create table patient(
patientID varchar2(8) primary key not null,
fk_panelID references panel(panelID),
name varchar2(100),
age number,
email varchar2(50),
gender varchar2(6),
address varchar2(100),
street varchar2(100),
city varchar2(100),
postcode number,
state varchar2(100),
phoneNumber varchar2(12),
password varchar2(20)
);
```

vi. Payment Table:

```
create table payment (
```

```
    paymentID varchar2(8) primary key not null,
    fk_staffID references staff(staffID),
    fk_visitID  references visit(visitID),
    totalAmount varchar2(8),
    status varchar2(10));
```

vii. Staff Table:

```
  Create table staff (
  staffID varchar2(8) primary key not null,
  name varchar2(100),
  age number,
  email varchar2(50),
  address varchar2(100),
  street varchar2(100),
  city varchar2(100),
  postcode number,
  state varchar2(100),
  gender varchar2(6),
  phoneNumber varchar2(12),
  password varchar2(20)
  );
```

viii. Tooth Table:

```
   create table tooth (
   toothID varchar2(100) primary key not null,
   description varchar2(500)
   );
```


ix. Treatment Table:

```
 create table treatment
 (
 treatmentID varchar2(8) primary key not null,
 treatment varchar2(100),
 cost number (12,2)
 );
```

x. Visit Table:

```
 create table visit
 (
 visitID varchar2(8) primary key not null,
 fk_appointmentID references appointment(appointmentID),
 fk_patientID  references patient(patientID),
 visitDate date,
 time timestamp,
 mc varchar2(50)
```

```
);
```

xi. V_medicine Table:

```
CREATE TABLE V_MEDICINE
(
fk_medicineID varchar2(8) not null,
fk_visitID varchar2(8) not null,
quantity number (30,0),
description varchar2(200),
CONSTRAINT v_medicine_pk PRIMARY KEY
(
  fk_medicineID,
  fk_visitID
);
```

xii. V_treatment Table:

```
    create table v_treatment
(
fk_toothID varchar2(8) not null,
fk_visitID varchar2(8) not null,
fk_treatmentID varchar2(8) not null,
description varchar2(200)
CONSTRAINT v_treatment_pk PRIMARY KEY
(
  fk_toothID,
  fk_visitID,
  fk_treatmentID
);
```


Alter Statement:

i. Staff Table:

```
alter table staff
add postcode varchar2(5),
add street varchar2(100),
add city varchar2(100);
```

ii. Dentist: Table:

```
alter table dentist
add street varchar2(100),
add city varchar2(100);
```

iii. Patient Table:

```
alter table patient
add street varchar2(100),
add city varchar2(100);
```

iv. Medicine Table:

```
alter table medicine
modify price varchar2(10);
```

v. Medicine Table:

```
alter table medicine
add quantity int;
```

5.3.2: Data Manipulation Language(DML)

Insert Statement

i. Insert statement is used to insert data of visiting patient into visit table.

```
INSERT INTO VISIT (FK_PATIENTID, VISITDATE)
VALUES('$patient_id',sysdate)
```

Update Statement

i. Update statement is used to update the appointment status for new appointment

```
UPDATE appointment SET STATUS='approve' WHERE
APPOINTMENTID='$latestID'
```


Select Statement

i. Select statement used to check the existence of user email and password in patient table

```
select count(*) from patient where email='$username' and
password='$password'
```

ii. It is a query that is used to validate patient email and password

```
select * from patient where email='$username' and password='$password'
```

iii. It is a query that is used to validate staff email and password

```
select * from staff where email='$username' and password='$password'
```

iv. It is a query that is used to validate dentist email and password

select * from dentist where email='$username' and password='$password'

v. Select statement is used to display data of the patient from patient table for update

select * from patient where email = '$patientID'

vi. Select statement is used for the patient to view list of appointments according to date

select TO_CHAR(a.appointmentdate, 'dd/MM/yyyy HH:MI AM'),a.status from APPOINTMENT a, patient p where a.fk_patientId = p.patientId and p.patientId='$PATIENDID' ORDER BY a.appointmentdate'

vii. Select statement is used to display list of patient name for the dentist to make appointment

SELECT PATIENTID, NAME FROM PATIENT

viii. Select statement is used for the dentist view list of appointment where the status is pending and according to date

select a.appointmentid,d.dentistId,TO_CHAR(a.appointmentdate, 'dd/MM/yyyy HH:MI AM'),a.status from APPOINTMENT a, dentist d where a.status='pending' ORDER BY a.appointmentdate DESC

ix. Select statement is used for the dentist to view list of appointments according to date and status is not pending

select a.appointmentid,d.dentistId,TO_CHAR(a.appointmentdate, 'dd/MM/yyyy HH:MI AM'),a.status from APPOINTMENT a, dentist d where a.fk_dentistId = d.dentistId and a.status<>'pending' ORDER BY a.appointmentdate DESC

x. Select statement is used to display list of visiting patient name based on the date of treatment

SELECT v.VISITID,p.PATIENTID,p.NAME FROM VISIT v,PATIENT p WHERE p.PATIENTID=v.FK_PATIENTID

AND
TO_CHAR(v.VISITDATE,'DD/MM/YYYY')=TO_CHAR(sysdate,'DD/MM/YYYY')

xi. Select statement is used for the staff to view list of medicine from medicine table

SELECT * FROM medicine

xii. Select statement is used for the dentist to view list of medicine from medicine table

SELECT * FROM medicine

xiii. Select statement is used to view list of patient name from patient table by staff for patient registration

SELECT * FROM patient where NAME is not null

xiv. Select statement is used to display list of visiting patients according to date

SELECT * FROM VISIT v,PATIENT p WHERE p.PATIENTID=v.FK_PATIENTID AND TO_CHAR(v.VISITDATE,'DD/MM/YYYY')=TO_CHAR(sysdate,'DD/MM/YYYY')

xv. Select statement is used for the staff to view list of recent appointment

select n.name,a.appointmentID,TO_CHAR(a.appointmentdate, 'dd/MM/yyyy HH:MI AM'), a.status from patient n, appointment a where a.fk_patientID = n.patientID and a.appointmentdate>sysdate order by a.appointmentDate

xvi. This select statement is used to display patient name on the receipt

SELECT p.name FROM VISIT v,PATIENT p WHERE p.PATIENTID=v.FK_PATIENTID AND v.visitid=$visitid

xvii. It is used to display patient treatment and cost

SELECT T.TREATMENT,T.COST FROM V_TREATMENT V,TREATMENT T WHERE V.FK_VISITID=$visitid AND T.TREATMENTID=V.FK_TREATMENTID

xviii. This select statement is used to display medicine description, quantity and price of the patient

SELECT M.DESCRIPTION,V.QUANTITY,M.PRICE FROM V_MEDICINE V,MEDICINE M WHERE V.FK_VISITID=$visitid AND M.MEDICINEID=V.FK_MEDICINEID

xix. Select statement to display patient information from patient table which is not null

SELECT * FROM patient where NAME is not null

xx. This select statement is used to display all data from visit table of the patient based on the patient ID

SELECT * FROM VISIT v,PATIENT p WHERE p.PATIENTID=v.FK_PATIENTID AND

p.PATIENTID='$patient_id' AND TO_CHAR(v.VISITDATE,'DD/MM/YYYY')=TO_CHAR(sysdate,'DD/MM/YYYY')

xxi. It is used to calculate cost of the treatment of the patient

SELECT SUM(T.COST) FROM V_TREATMENT V,TREATMENT T WHERE V.FK_VISITID=$visitid AND T.TREATMENTID=V.FK_TREATMENTID

xxii. This select statement is used to calculate the total price of treatment and medicine

SELECT SUM(V.QUANTITY*M.PRICE) FROM V_MEDICINE V,MEDICINE M WHERE V.FK_VISITID=$visitid AND M.MEDICINEID=V.FK_MEDICINEID

xxiii. Select statement that is used to display patient information order by id

SELECT * FROM patient ORDER BY PATIENTID

xxiv. Select statement that is used to know date that has been choose by patient to display error message of no_appt trigger where the patient cannot choose appointment date on sunday

SELECT to_char(to_date('$appointmentDate','yyyy-MM-dd HH24:MI SS') ,'Day') from dual"

5.3.3: Data Retrieval

5.3.3.4: Stored Procedures

i. Insert:
This is an insert stored procedure on patient table. All the information filled in by the patient will be inserted into patient table.

```
create or replace procedure patient_proc(
p_patientID patient.patientID%TYPE,
p_fk_panelID patient.fk_panelID%TYPE,
p_name patient.name%TYPE,
p_age patient.age%TYPE,
p_email patient.email%TYPE,
p_address patient.address%TYPE,
p_gender patient.gender%TYPE,
p_phoneNumber patient.phoneNumber%TYPE,
p_password patient.password%TYPE,
p_state patient.state%TYPE,
p_postcode patient.postcode%TYPE)
is
begin
insert into
(patientID,fk_panelID,name,age,email,address,gender,phoneNumber,passwor
d,state,postcode)
values
(p_patientID,p_fk_panelID,p_name,p_age,p_email,p_address,p_gender,p_ph
oneNumber,p_password,p_state,p_postcode);
end;
```

ii. Insert:
Appt_proc is a insert procedure that is used to insert appointment date and time into appointment table by patient.

```
create or replace procedure appt_proc(
p_fk_patientID in APPOINTMENT.fk_patientID%type,
p_appointmentDate in APPOINTMENT.APPOINTMENTDATE%type)
is
begin
insert into appointment(fk_patientID,appointmentDate) values
(p_fk_patientID,p_appointmentDate);
end;
```

iii. Insert:
Insert stored procedure is used to insert payment information into payment table.

```
create or replace procedure payment_proc(
p_fk_staffID in payment.fk_staffID%type,
p_fk_visitID in payment.fk_visitID%type,
p_totalAmount in payment.totalAmount%type,
p_status in payment.status%type
)
is
begin
insert into payment(fk_staffID,fk_visitID,totalAmount,status) values
(p_fk_staffID,p_fk_visitID,p_totalAmount,p_status);
end;
```

iv. Insert:
Dentist_proc is a insert procedure that is used to insert appointment date and time into appointment table by dentist.

```
create or replace procedure dentist_proc(
p_fk_dentistID in APPOINTMENT.fk_dentistID%type,
p_fk_patientID in APPOINTMENT.fk_patientID%type,
p_appointmentDate in APPOINTMENT.APPOINTMENTDATE%type)
is
begin
insert into appointment(fk_dentistID,fk_patientID,appointmentDate) values
(p_fk_dentistID,p_fk_patientID,p_appointmentDate);
end;
```


v. Insert:
Vtreatment_proc is used to insert data into vtreatment table

```
create or replace procedure vtreatment_proc(
p_fk_toothID in v_treatment.fk_toothID%type,
p_fk_visitID in v_treatment.fk_visitID%type,
p_fk_treatmentID in v_treatment.fk_treatmentID%type,
p_description in v_treatment.description%type
)
is
begin
insert into v_treatment(fk_toothID,fk_visitID,fk_treatmentID,description)
values (p_fk_toothID,p_fk_visitID,p_fk_treatmentID,p_description);
end;
```

vi. Update:

Update stored procedure is used to update patient information on patient table

```
create or replace PROCEDURE updatePatient(

    p_patientID IN patient.patientID%TYPE,
    p_name IN patient.name%TYPE,
    p_age IN patient.age%TYPE,
    p_email IN patient.email%TYPE,
    p_address IN patient.address%TYPE,
    p_phoneNumber IN patient.phoneNumber%TYPE,
    p_password IN patient.password%TYPE,
    p_gender IN patient.gender%TYPE,
    p_state IN patient.state%TYPE,
    p_postcode IN patient.postcode%TYPE)

AS
BEGIN

  UPDATE patient
  SET name = p_name,
  age = p_age,
  email = p_email,
  address = p_address,
  phoneNumber = p_phoneNumber,
  password = p_password,
  gender = p_gender,
  state = p_state,
  postcode = p_postcode
  where email = p_email;

END;
```


5.3.3.5 Triggers

i. Before insert:
This trigger will set an appointment status as ‘pending’ for new appointment.

```
create or replace trigger appt_status
before insert on appointment
for each row
begin
:new.status:= 'pending';
end;
```

ii. Before insert:

This is an auto increment trigger for dentist table.

```
create or replace trigger auto_increment
before insert on dentist
for each row
declare
begin
if ( :new.dentistID is null)
then
:new.dentistID := dentist_seq.nextval;
end if;
end;
```

iii. Before insert:
This is an auto increment trigger for appointment table.

```
create or replace trigger auto_increment_appt
before insert on appointment
for each row
declare
begin
if ( :new.appointmentID is null)
then
:new.appointmentID := appointment_seq.nextval;
end if;
end;
```


iv. Before insert:
This is an auto increment trigger for medicine table.

```
create or replace trigger auto_increment_medicine
before insert on medicine
for each row
declare
begin
if (:new.medicineID is null)
then
:new.medicineID := med_seq.nextval;
end if;
end;
```

v. Before insert:
This is an auto increment trigger for patient table.

```
create or replace trigger auto_increment_patient
before insert on patient
```

```
for each row
declare
begin
if ( :new.patientID is null)
then
:new.patientID := patient_seq.nextval;
end if;
end;
```

vi. Before insert:
This is an auto increment trigger for payment table.

```
create or replace trigger auto_increment_payment
before insert on payment
for each row
declare
begin
if ( :new.paymentID is null)
then
:new.paymentID := payment_seq.nextval;
end if;
end;
```

vii. Before insert:
This is an auto increment trigger for staff table.

```
create or replace trigger auto_increment_staff
before insert on staff
for each row
declare
begin
if ( :new.staffID is null)
then
:new.staffID := staff_seq.nextval;
end if;
end;
```

viii. Before insert:
This is an auto increment trigger for tooth table.

```
create or replace trigger auto_increment_tooth
before insert on tooth
for each row
declare
begin
if ( :new.toothID is null)
```

```
then
:new.toothID := tooth_seq.nextval;
end if;
end;
```

ix. Before insert:
This is an auto increment trigger for treatment table.

```
create or replace trigger auto_increment_treatment
before insert on treatment
for each row
declare
begin
if ( :new.treatmentID is null)
then
:new.treatmentID := treatment_seq.nextval;
end if;
end;
```

x. Before insert:
This is an auto increment trigger for visit table.

```
create or replace trigger auto_increment_visit
before insert on visit
for each row
declare
begin
if ( :new.visitID is null)
then
:new.visitID := visit_seq.nextval;
end if;
end;
```

xi. Before insert:
This is an auto increment trigger for v_medicine table.

```
create or replace trigger auto_increment_vmedicine
before insert on v_medicine
for each row
declare
begin
if ( :new.v_medicineID is null)
then
:new.v_medicineID := v_medicine_seq.nextval;
end if;
```

```
end;
```

xii. Before insert:
This is an auto increment trigger for v_treatment table.

```
create or replace trigger auto_increment_vtreatment
before insert on v_treatment
for each row
declare
begin
if ( :new.v_treatmentID is null)
then
:new.v_treatmentID := v_treatment_seq.nextval;
end if;
end;
```

xiii. Before insert or update:
No_appt trigger is a trigger that disallow user from choosing an appointment on Sunday.

```
create or replace trigger no_appt
before insert or update on appointment
for each row
begin
if(trim(to_char(:new.appointmentDate ,'Day')) = 'Sunday')
then
raise_application_error(-20100,'No appointment can be made on Sunday');
end if ;
end;
```


xiv. After insert:
This trigger will automatically deduct the medicine quantity by 1 whenever the dentist clicks on the medicine button. Staff are able to view the current quantity.

```
create or replace trigger med_qty
after insert on V_MEDICINE
for each row
declare
begin
  update MEDICINE set quantity = (quantity - 1) where MEDICINEID =
:new.FK_MEDICINEID;
end;
```

xv. After insert or update:
This trigger will automatically insert patient information into patient_backup table and it will update any changes made on patient table on the backup table.

```
create or replace trigger backup_patient
after insert or update on patient
for each row
declare
begin
insert into patient_backup values
(:new.patientID,:new.fk_panelID,:new.name,:new.age,:new.email,
:new.address,:new.gender,:new.phoneNumber,:new.password,:new.state,:new
.postcode);
DBMS_OUTPUT.PUT_LINE('Record successfully inserted into
patient_backup table');
END;
```

5.4 Conclusion

In this chapter, the installation steps, database creation and the database implementation has been explained. For the next chapter, testing phase will be explained in detail.

CHAPTER VI

TESTING

6.1 Introduction

In this chapter, the testing of the system will be explained. Testing phase includes the test plan, test strategy, test design and test result and analysis. Testing strategy that is adopted for this project is Dynamic Testing.

6.2 Test Plan

Test plan is a document or subproject plan for the testing part of a project. It contains the detailed procedures that determine the scope, approach, resources and schedule of all testing activities.

6.2.1 Test Organization

There are three persons involved in the testing phase. The system developer is the one that build the system while software tester and client will analyse and evaluate the software product and to detect defects. The detailed about the tester will be explained in the table below.

Table 6.1: Test Organization of DCMS

Name	Roles	Responsibilities
Wan Noraqilah Binti A.Razak	System Developer	Build the system and execute integration and component test.
Dr. Noraswaliza Binti Abdullah	Software Tester	Analyse the functionality of the system and monitor the system performance.
Dr. Ahamad Bin Ali	Client	Analyse the functionality and requirement of the system.

6.2.2 Test Environment

The testing is being done at Faculty of Information Communication and Technology(FICTS) of UTeM. The specification of system configuration that has been used for this project is shown in the table below.

Table 6.2: System Configuration of DCMS

System Configuration	Specification
Operating System	Windows 8.1
Database	Oracle SQL Developer
Server	Xampp v3.2.2
Web Browser	Google Chrome
Programming Language	Hypertext Preprocessor(PHP) Language

6.2.3 Test Schedule

Test schedule is a record of activities tor tasks with estimated start and end dates. Test schedule of DCMS system is shown on the table below.

Table 6.3: Test Schedule of DCMS

Activities	**Description**	**Start Date**	**Finish Date**	**Duration**
Unit testing	It is used to test the codes and functions of the system	27/3/2017	18/5/2017	2 months
Integration testing	It is used to verify combined functionality after integration	9/4/2017	15/4/2017	1 week
System testing	It is used to evaluate the system's compliance with specified requirements	16/4/2017	30/4/2017	2 weeks
Acceptance testing	It is used to test if the requirements of a specification are met	1/5/2017	14/5/2017	2 weeks

6.3 Test Strategy

Dynamic Testing is used for this system. It involves the execution of the software of a component or system. Dynamic testing is divided into two different methods which are Black-box and White-box. Black-box testing used to test the functional and non-functional of the system while White-box is used to test the internal structure of the system. The details of the methods is explained in the table 6.4 below.

Table 6.4: Test Strategy of DCMS

Approaches	Explanation
Black-box Testing	Black-box test either functional or non-functional without reference to the internal structure of the component or system. It focuses on the output generated in response to selected inputs and execution conditions
White-box Testing	White-box test based on an analysis of the internal structure of the component or system

6.4 Test Design

6.4.1 Test Description

i. Login

Login module involved patient, dentist and staff. This module allows authorized user with correct email address and matching password to log into the system. Table 6.5, 6.6 and 6.7 shows the test case of login for patient, dentist and staff.

Table 6.5: Test Case of Login for Patient

Test Case ID	Description	Action	Expected Output
TC P/01	Patient left the password text field blank	Password left blank	Error: Please fill out this field

TC P/02	Patient left the email text field blank	Email address is left blank	Error: Please fill out this field
TC P/03	Patient insert incorrect email address or password	All input inserted but misspelled	Error: Incorrect email or password!
TC P/04	Hidden password is used	Hidden password is used when user enter the password	Successfully login!
TC P/05	Patient insert correct email and password	All input inserted and spelled correctly	Successfully login!

Table 6.6: Test Case of Login for Dentist

Test Case ID	Description	Action	Expected Output
TC D/01	Dentist left the password text field blank	Password left blank	Error: Please fill out this field
TC D/02	Dentist left the email text field blank	Email address is left blank	Error: Please fill out this field

TC D/03	Dentist insert incorrect email or password	All input inserted but misspelled	Error: Incorrect email or password!
TC D/04	Hidden password is used	Hidden password is used when user enter the password	Successfully login!
TC D/05	Dentist insert correct email and password	All input inserted and spelled correctly	Successfully login!

Table 6.7: Test Case of Login for Staff

Test Case ID	Description	Action	Expected Output
TC S/01	Staff left the password text field blank	Password left blank	Error: Please fill out this field
TC S/02	Staff left the email text field blank	Email address is left blank	Error: Please fill out this field
TC S/03	Staff insert incorrect email or password	All input inserted but misspelled	Error: Incorrect email or password!

TC S/04	Hidden password is used	Hidden password is used when user enter the password	Successfully login!
TC S/05	Staff insert right email and password	All input inserted and spelled correctly	Successfully login!

ii. Patient Registration:

Table 6.8 shows the input that need to be fill-in in patient registration module.

Table 6.8: Test Case of Patient Registration

Test Case ID	Description	Action	Expected Output
TC PR/01	All text fields, radio buttons and button are active	No input data	Able to enter data in all input fields
TC PR/02	All fields marked with asterisk (*)	Patient fill-in data in the marked field	Show message" Please enter values in the required field"
TC PR/03	Interface verification	Use tab to navigate this page	Able to navigate in sequence

TC PR/04	Patient did not put @ symbol	Patient enter email address without @ symbol	Error: Please enter valid email!
TC PR/05	Patient did not put dot symbol in email address	Patient fill-enter email address without dot symbol	Error: Please enter valid email!
TC PR/06	System uses hidden password	Patient enter password	Successfully login!
TC PR/07	System validate full name	Patient enter alphanumeric name	Error: Please enter valid name!
TC PR/08	System validate length of password inserted by the patient	Patient enter password less than 8	Error: Password should be at least 8!
TC PR/09	System validate length of password inserted by the patient	Patient enter password more than 8	Error: Password is too long!
TC PR/10	Patient has inserted all data	All input has been inserted	The information has been saved!

iii. Appointment

Appointment module is used by dentist and patient where they need to choose date and time to make an appointment.

Table 6.9: Test Case of Appointment

Test Case ID	Description	Action	Expected Output
TC AP /01	User did not select date and time	No input data	Error: Please choose appointment date and time
TC AP/02	User did not select appointment date	Date is left blank	Error: Please choose appointment date and time
TC AP/03	User did not select appointment time	Time is left blank	Error: Please choose appointment date and time
TC AP/04	Dentist can click on 'Approve' or 'Reject' button	Dentist click on the button	Status of appointment change to approve or reject
TC AP/05	Status of appointment change to pending whenever patient make new appointment	Click on submit button	Status of appointment change to pending
TC AP/06	System display list of appointments information according from current to upcoming date to the staff	Staff click on appointment button	display list of appointments information according from

			current to upcoming date is displayed
TC AP/07	System display list of appointments information according to date to the dentist	Dentist click on submit button	display list of appointments information according to date is displayed to the dentist
TC AP/08	System display list of appointments information to the patient	Patient click on submit button	Display list of appointments information according to date
TC AP/09	User choose appointment date on Sunday	All input has been inserted	Error: No appointment can be made on Sunday!
TC AP/10	User choose past date	All input has been inserted	Error: Appointment failed!
TC AP/11	User has chosen date and time	All input has been inserted	Appointment date and time has been saved!

iv. Treatment

Treatment module is used by dentist and they need to enter patient name and choose type of treatment as shown in the table below.

Table 6.10: Test Case of Treatment

Test Case ID	Description	Action	Expected Output
TC T/01	Dentist left patient name, treatment and tooth blank	No input data	Error: Please choose patient name and treatment
TC T/02	Dentist left patient name blank	Patient name is left blank	Error: Please choose patient name
TC T/03	Dentist left treatment and tooth blank	Treatment is left blank	Error: Please choose treatment
TC T/04	Data has inserted all data	Data is inserted	Treatment information has been saved!

v. Medicine

Medicine module is used by the dentist. Table 6.11 shows the requirement that need to be
fulfilled by the dentist.

Table 6.11: Test Case of Medicine

Test Case ID	Description	Action	Expected Output
TC M /01	Dentist left description field blank	Description is left blank	Error: Please fill-in the text field!
TC M /02	Dentist left total field blank	Total is left blank	Error: Please fill-in the text field!
TC M /03	System display current quantity of the medicine	Click on medicine button	Display current quantity of the medicine
TC M /04	Dentist has inserted all data	All input has been inserted	The information has been saved!

vi. Visit

Medicine module is used by the dentist. Table 6.12 shows the requirement that need to be fulfilled by the dentist.

Table 6.12: Test Case of Visit

Test Case ID	Description	Action	Expected Output
TC V /01	System display list of visiting patients when they click patient name during patient registration	Click patient name	Display list of names of the visiting patient

TC V /02	System can display payment page when staff click on patient registration button on visit page	Click on patient information button	Navigate to payment page of the patient

vii. Payment

Payment module is used by the staff. Table 6.13 shows the requirement that need to be fulfilled by the dentist.

Table 6.13: Test Case of Payment

Test Case ID	Description	Action	Expected Output
TC PY/01	System display total amount on payment page	Click on visit button of the patient name	Display and total amount
TC PY/02	System can display payment receipt	Click on pay button	Display payment receipt

viii. Logout

Medicine module is used by the dentist. Table 6.14 shows the requirement that need to be
fulfilled by the dentist.

Table 6.14: Test Case of Logout

Test Case ID	Description	Action	Expected Output
TC L /01	Patient can log out from the system	Click logout	Navigate to login page
TC L/02	Dentist can log out from the system	Click logout	Navigate to login page
TC L/03	Staff can log out from the system	Click logout	Navigate to login page

6.4.2 Test Data

Test data is the input given to a software program and it will be used to ensure the system effectiveness.

i. Login

Login module involved patient, dentist and staff. This module allows authorized user with correct email address and matching password to log into the system. Table 6.15 shows the details of login test data.

Table 6.15: Details of Login Test Data for Patient

Test No.	Test Case ID	Input Test/ Data	Result
T01	TC P/01	Email: wnoraqilah@gmail.com Password: blank	Error: Please fill-out this field

T02	TC P/02	Email: blank Password: ********	Error: Please fill-out this field
T03	TC P/04	Email: wnraqla@gmail.com Password: ********	Error: Incorrect email or password!
T04	TC P/05	Email: wnoraqilah@gmail.com Password: ********	Successfully login!

Table 6.16: Details of Login Test Data for Dentist

Test No.	Test Case ID	Input Test/ Data	Result
T05	TC D/02	Email: ahmad@gmail.com Password: blank	Error: Please fill-out this field
T06	TC D/03	Email: blank Password: ********	Error: Please fill-out this field
T07	TC D/04	Email: ahmd@gmail.com Password: ********	Error: Incorrect email or password!
T08	TC D/05	Email: ahmad@gmail.com Password: ********	Successfully login!

Table 6.17: Details of Login Test Data for Staff

Test No.	Test Case ID	Input Test/ Data	Result
T09	TC S/02	Email: siti@gmail.com Password: blank	Error: Please fill-out this field
T10	TC S/03	Email: blank Password: ********	Error: Please fill-out this field
T11	TC S/04	Email: st@gmail.com Password: ********	Error: Incorrect email or password!
T12	TC S/05	Email: siti@gmail.com Password: ********	Successfully login!

ii. Patient Registration

Table 6.18 Details of Patient Registration Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T13	TC PR/01	None	Able to enter data in all input fields

T14	TC PR/02	Name*: Wan Noraqilah Age*: 22 Gender*: Female Address*:2, Jalan Bukit Beruang Postcode*: 73500 Street*: Taman Bukit Beruang City*: Bukit Beruang State*: Melaka Email*: wnoraqilah@gmail.com Contact Number*: 0123344567 Password*: ********	Error: Please enter value in the required field”
T15	TC PR/03	Check the sequence navigation of all text fields, radio buttons and button	Able to navigate in sequence
T16	TC PR/04	Name: Amira Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City:	Error: “Please enter valid email”

		Bukit Beruang State: Melaka Email: mira_at_gmail Password:	
T17	TC PR/05	Name: Amira Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail Password:	Error: “Please enter valid email”
T18	TC PR/06	Name: Amira Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470	Show message “Data has been saved!”

		City: Bukit Beruang State: Melaka Email: mira_at_gmail Password: ********	
T19	TC PR/07	Name: Amira123 Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password: **********	Show message “Please enter a valid name”
T20	TC PR/08	Name: Amira Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4	Show message “Password should be at least 8!”

		Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password: **	
T21	TC PR/09	Name: Amira Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password: **********	Show message "Password is too long!"
T22	TC PR/10	Name: Amira Age: 40 Gender: Female ContactNumber: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street:	Show message "Data has been saved!"

		Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password: ********	

iii. Appointment

Table 6.19: Details of Appointment Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T18	TC AP /01	Date: blank Time: blank	Error:" Please choose date and time"
T19	TC AP /02	Date: blank Time: 10:00 am	Error:" Please choose date and time"
T20	TC AP /03	Date: 20/08/2017 Time: blank	Error:" Please choose date and time"
T21	TC AP /04	Click on 'Approve' or 'Reject' button	Status of appointment change to approve or reject

T22	TC AP /05	Patient click on Submit button and appointment status change to ‘pending’	Status of appointment change to pending
T23	TC AP /06	List of appointment is displayed in the table in staff module according from current to upcoming date	List of appointment is displayed in the table
T24	TC AP /07	List of appointment is displayed in the table in dentist module according to date	List of appointment is displayed in the table
T25	TC AP /08	Date: 21/08/2017 Time: 11:00 a.m.	Appointment that has been requested on 21st August at 11:00 a.m. is displayed in the table
T26	TC AP /09	Date: 20/08/2017 Time: 10:00 a.m.	Error:” No appointment can be made on Sunday
T27	TC AP /10	Date: 1/08/2017 Time: 10:00 a.m.	Error:” No Appointment failed!
T28	TC AP /11	Date: 15/08/2017 Time: 10:00 a.m.	Success!

iv. Treatment

Table 6.20: Details of Treatment Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T24	TC T/01	Patient Name: blank Treatment: blank Tooth: blank	Error: Please choose patient name, treatment and tooth
T25	TC T/02	Patient Name: blank Treatment: Bridged Tooth: 4-2 incisor	Error: Please choose patient name
T26	TC T/03	Patient Name: Wan Noraqilah Treatment: blank Tooth: blank	Error: Please choose treatment and tooth
T27	TC T/04	Patient Name: Wan Noraqilah Treatment: Bridged Tooth: 4-2 incisor	Treatment information has been saved!

v. Medicine

Table 6.21: Details of Medicine Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T26	TC M /01	Total: 1 Description: blank	Error: “Please fill out this field.”

T27	TC M /02	Total: blank Description: Twice a day	Error: “Please fill out this field.”
T28	TC M /03	None	Clarithromycin- 3
T29	TC M /04	Total: 1 Description: Twice a day	The information has been saved!

vi. Visit

Table 6.22: Details of Visit Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T20	TC V /01	None	Display list of visiting patients based on that day
T21	TC V /02	None	Display payment information of the patient

vii. Payment

Table 6.23: Details of Visit Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T22	TC PY/01	None	Display total amount of the payment

T23	TC PY/02	None	Display payment receipt

i. Logout

Table 6.24: Details of Logout Test Data

Test No.	Test Case ID	Input Test/ Data	Result
T34	TC L/01	None	Patient log out from the system
T35	TC L/02	None	Dentist log out from the system
T36	TC L/03	None	Staff log out from the system

6.5 Test Results and Analysis

Test result and analysis is used to validate and to see whether the result in the test design is the same with test result and analysis. The details of the result and analysis has been shown in the table below.

Table 6.25 Login Test Result and Analysis

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status (Pass/Fail)

TC P/01	Validate that patient did not enter password	Check the required fields by not fill-in data on password field		1. Patient enter email	Email: wnoraqilah@gmail.com Password: blank	Patient can log into the system	Patient click on submit button	Error: Please-fill out this field	Pass
TC P/02	Validate that patient did not enter email	Check the required fields by not fill-in any data on email			Email: blank Password: ********	Patient can log into the system	Patient click on submit button	Error: Please-fill out this field	Pass
TC P/03	Validate that patient enter incorrect email or password	Check the required fields by entering incorrect email or password			Email: wnraqla@gmail.com Password: blank	Patient cannot log into the system	Patient click on submit button	Error: Incorrect email or password	Pass
TC P/04	Validate that the system uses hidden	Check the hidden password of the patient		1. Patient enter email and password	Email: wnoraqilah@gmail.com Password: ********	Patient can log into the system	Patient click on Submit button	"Successfully login!"	Pass

	passw ord								
TC P/05	Valid ate that patien t enter correc t email and passw ord	Check the fields by enteri ng correc t email and passw ord		1. Patient enter email 2. Patient enter passwo rd	Email: wnoraqila h@gmail.c om Password: ********	Patient can log into the system	Patient click on submit button	Succe ssfull y login!	Pass
TC D/01	Valid ate that dentis t did not enter passw ord	Check the requir ed fields by not fill-in data on passw ord field		1. Dentist enter email	Email: ahmad@ gmail.com Password: blank	Dentist can log into the system	Dentist click on submit button	Error: Please -fill out this field	Pass
TC D/02	Valid ate that dentis t did not enter email	Check the requir ed fields by not fill-in any data on email field			Email: blank Password: ********	Dentist can log into the system	Dentist click on submit button	Error: Please -fill out this field	Pass
TC D/03	Valid ate that dentis t enter incorr	Check the requir ed fields by			Email: ahmd@gm ail.com Password: blank	Dentist can log into the system	Dentist click on submit button	Error: Incorr ect email or	Pass

	ect email or password	entering incorrect email or password						password	
TC D/04	Validate that the system uses hidden password	Check the hidden password of the dentist		1. Dentist enter email and password	Email: ahmad@gmail.com Password: ********	Dentist can log into the system	Dentist click on Submit button	"Successfully login! "	Pass
TC D/05	Validate that dentist enter correct email and password	Check the fields by entering correct email and password		1. Dentist enter email 2. Dentist enter password	Email: ahmad@gmail.com Password: ********	Dentist can log into the system	Dentist click on submit button	Successfully login!	Pass
TC S/01	Validate that staff did not enter password	Check the required fields by not fill-in data on password field		1. Staff enter email	Email: siti@gmail.com Password: blank	Staff can log into the system	Staff click on submit button	Error: Please-fill out this field	Pass

TC S/02	Validate that staff did not enter email	Check the required fields by not fill-in any data on email field			Email: blank Password: ********	Staff can log into the system	Staff click on submit button	Error: Please-fill out this field	Pass
TC S/03	Validate that staff enter incorrect email or password	Check the required fields by entering incorrect email or password			Email: st@gmail.com Password: blank	Staff can log into the system	Staff click on submit button	Error: Incorrect email or password	Pass
TC S/04	Validate that the system uses hidden password	Check the hidden password of the staff		1. Staff enter email and password	Email: siti@gmail.com Password: ********	Dentist can log into the system	Dentist click on Submit button	"Successfully login!"	Pass
TC S/05	Validate that staff enter correct	Check the required fields by not fill-in		1. Staff enter email 2. Staff enter	Email: siti@gmail.com Password: ********	Staff can log into the system	Staff click on submit button	Successfully login!	Pass

	email and password	any data on email field		password					

Table 6.26 Patient Registration Test Result and Analysis

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status(Pass/ Fail)
TC PR/ 01	Verify all input field are active	Check all the text fields, radio buttons and button	New registration page is successfully displayed	Click on text fields, radio buttons and button	None	Able to enter data in all input fields			Pass
TC PR/ 02	Verify that the required/ mandatory fields are marke	Check the required fields by not filling any data		1. Do not enter any value in the marked field 2.	Leave blank: Name: Age: Gender: Address: Postcode: Street:	Show message ” Please enter value in the required field”			Fail

	d with asterisk(*)			Click on Submit button	City: State: Email: Contact Number: Password:				
TC PR/ 03	Verify interface	Check the sequence navigation of all text fields, radio buttons and button		Use tab to navigate from 1. Name 2. Age 3. Gender 4. Contact Number 5. Address 6. Street 7. Postcode 8. City 9. State 10. Email 11. Password	None	Able to navigate in sequence			Pass
TC PR/ 04	Validate email field	Check email text field that has @ symbol written in words		1. Enter invalid email 2. Click password text field/sub	Name: Amira Age: 40 Gender: Female Contact Number: 0123344564 Address:	Show message ” Please enter valid email”			Pass

				mit button	17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: Mira_at_gmail Password:				
TC PR/ 05	Validate email field	Check email text field that has missing dot in the email address		1. Enter invalid email 2. Click password text field/submit button	Name: Amira Age: 40 Gender: Female Contact Number: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email:	Show message ” Please enter valid email”			Pass

					mira@gmail Password:				
TC PR/ 06	Validate that the system uses hidden password	Check the hidden password of the patient		1. Enter data into the password field 2. Click on submit button		Show message “Data has been saved!”		Show message “Data has been saved!”	Pass
TC PR/ 07	Validate full name	Check the name text field that has number		1. Enter alphanumeric name 2. Click on submit button/other text field	Name: Amira123 Age: 40 Gender: Female Contact Number: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password:	Show message “Please enter a valid name!”		Show message “Data has been saved!”	Fail

TC PR/ 08	Validate password	Check the minimum value of characters		Enter password less than 8	Name: Amira Age: 40 Gender: Female Contact Number: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street: Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password: **	Show message "Password should be at least 8"		Show message "Data has been saved! "	Fail
TC PR/ 09	Validate password	Check the minimum value of characters		Enter password more than 8	Name: Amira Age: 40 Gender: Female Contact Number: 0123344564 Address: 17, Jalan Bestari 8/4, taman Bestari Street:	Show message "Password is too long"		Show message "Pass word is too long"	Pass

					Jalan Bestari 8/4 Postcode: 75470 City: Bukit Beruang State: Melaka Email: mira@gmail.com Password: **********				
TC PR/ 10	Verify all the data inserted	Check the data of the patient		1. Enter data into the fields require 2.Click on Submit button		Show message "Data has been saved!"		Show message "Data has been saved!"	Pass

Table 6.27 Appointment Test Result and Analysis

TC AP/02	Validate that user did not select date	Check the required fields by not choosing appointment date	1. User login to the system	1. User did not choose date 2. Click on Submit button	Leave Blank: Date: Time: 10:00 a.m.	Show message " Please choose date and time"		"Success"	Fail
TC AP/03	Validate that user did not choo	Check the required fields by not filling any data	1. User login to the system	1. User did not choose appointment time	Leave Blank: Date: 14/08/2017 Time:	Show message " Please choose time"		Success	Fail

	se appointment time			2. Click on Submit button					
TC AP/04	Validate that dentist can click on 'Approve' or 'Reject' button	Check the required fields by clicking on the button	1. Dentist login to the system 2. Dentist choose patient name 3. Dentist choose date and time	Click on 'Approve' or 'Reject' button	Date: 15/08/2017 Time: 11:00 a.m.	Status of appointment change to approve or reject		Status of appointment change to approve or reject	Pass
TC AP/05	Validate that status appointment on patient page	Check the required fields by clicking on the button	1. Patient login to the system 2. Patient	Click on Submit button	Date: 15/08/2017 Time: 11:00 a.m.	Status of appointment change to pending		Status of appointment change to pending	Pass

	change to ‘pending’ whenever they request new appointment		choose date and time						
TC AP/06	Validate that system can display appointment date, time and status to the staff according from current date upcoming date	Check the required fields by clicking on appointment button on staff module	1. Staff login to the system	1. Staff click on appointment		List of appointment is displayed in the table		List of appointment is displayed in the table	Pass

TC AP/07	Validate that system can display appointment date, time and status to the dentist according to date	Check the required fields by clicking on appointment button on dentist module	1. Dentist login to the system	1. Dentist click on appointment		List of appointment is displayed in the table		List of appointment is displayed in the table	Pass
TC AP/08	Validate that system can display appointment information to the patient whenever they mak	Check the required fields by choosing date, time and click on Submit button	1. Patient login to the system	1. Patient choose appointment date on Sunday 2. Click Submit button	Date: 21/08/2017 Time: 11:00 a.m.	Appointment that has been requested on 21st August at 11:00 a.m. is displayed in the table		Appointment that has been requested on 21st August at 11:00 a.m. is displayed in the table	Pass

	e a new appointment								
TC AP/09	Validate that user choose appointment date on Sunday	Check the required fields by choosing date on Sunday	1. User login to the system	1. User choose appointment date on Sunday	Date: 20/08/2017 Time: 10:00 a.m.	Show message ” No appointment can be made on Sunday		Show message” No appointment can be made on Sunday	Pass
TC AP/10	Validate that user choose past date	Check the required fields by choosing past date	1. User login to the system	1. User choose past date	Date: 1/08/2017 Time: 10:00 a.m.	Show message ” Appointment failed!		Show message” Appointment failed! Sunday	Pass
TC AP/11	Validate that choose date and time	Check the required fields by choosing the correct date and time	1. User login to the system	1. User choose date and time 2. User click on Submit button	Date: 15/08/2017 Time: 10:00 a.m.	Show message ” Success”		Show message” Success”	Pass

Table 6.8 Treatment Test Result and Analysis

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status(Pass/ Fail)
TC T/01	Verify that the required fields are all filled	Check the required fields by not filling any data	1. Dentist log into the system 2. Click on treatment button	1. Dentist did not enter any value in the field	Leave Blank: Patient Name: Treatment: Tooth:	Show message “Please enter data”		Data has been saved	Fail
TC T/02	Verify that the patient name field is left blank	Check the patient name fields by not choosing patient name	1. Dentist log into the system 2. Click on treatment button	1. Dentist did not choose patient name	Leave Blank: Patient Name: Treatment: Bridged Tooth: 4-2 incisor	Show message “Please choose patient name”		Data has been saved	Fail
TC T/03	Verify that the treatment and tooth fields	Check the treatment and tooth fields by not	1. Dentist log into the system	1. Dentist did not enter treatment and tooth	Leave Blank: Patient Name: Amira Treatment: Tooth	Show message “Please choose treatment and tooth”		Data has been saved	Fail

	are left blank	enterin g data	2.Cli ck on treat ment butt on						
TC T/04	Verify that the requir e field by enteri ng data in all fields	Check the require d fields by filling the data	1. Dent ist log into the syste m 2. Clic k on treat ment butt on	Enter patient name, treatme nt and tooth	Patient Name: Amira Treatment: Bridged Tooth: 4-2 incisor	Show message “Data has been saved”		“Data has been saved! ”	Pass

Table 6.9 Medicine Test Result and Analysis

Test Case ID	Test Scena rio	Test Case	Pre-cond ition	Test Steps	Test Data	Expected Results	Post Conditi on	Actua l Result	Stat us(P ass/ Fail)
TC M/ 01	Valid ate that descri ption field is left blank	Check the descript ion text field	1. Dent ist click on Save n butt on on treat ment page	1. Enter total quantit y	Leave Blank: Total: 1 Descriptio n:	Show message ” Please fill-out this field”	1. Click Give Medica tion button	“Pleas e fill-out this field”	Pass

TCM/02	Validate that total field is left blank	Check the total text field	1. Dentist click on Save n button on on treatment page		Leave Blank: Total: Description: Twice a day	Show message ” Please fill-out this field”	1. Click Give Medication button	“Please fill-out this field”	Pass
TCM/03	Validate that system can display current quantity of medicine to the staff	Click on medicine button	1. Staff log in to the system		None	Display current quantity of the medicine		Display current quantity of the medicine	Pass
TCM/04	Validate that all data has been inserted	Check the required field	1. Dentist click on Save n button on on treatment page		Total: 1 Description: Twice a day	Show message ” New Medicine Assign”		“New Medicine Assign”	Pass

Table 6.9 Visit Test Result and Analysis

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status (Pass/ Fail)
TC V/01	Validate that system can display list f visiting patient name when they click patient name during registration	Click on patient name	Staff login to the system	1. Staff click on visit button	None	Display list of visiting patients based on that day		Display list of visiting patients based on that day	Pass
TC V/02	Validate that system can display payment page when staff click on Patient Information button on visit page	Click on Patient Information button	Staff login to the system	1. Staff click on Patient Information button	None	Display payment information of the patient		Display payment information of the patient	Pass

Table 6.10 Payment Test Result and Analysis

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status (Pass/Fail)
TCP/01	Validate that system can display total amount of the payment of for a particular patient	Click on patient name	Staff login to the system	1. Staff click on visit button	None	Display total amount of the payment		Display total amount of the payment	Pass
TCP/02	Validate that system can display payment receipt	Click on pay button	Staff login to the system	1. Staff click on pay button	None	Display payment receipt		Display payment receipt	Pass

Table 6.11 Logout Test Result and Analysis

Test Case ID	Test Scenario	Test Case	Pre-condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Status (Pass/Fail)
TC L/01	Validate that patient can log out from the system	Click on log out button	Patient click on log out button	1. Patient click on logout button	None	Patient log out from the system	System navigate to login page	Patient log out from the system	Pass
TC L/02	Validate that dentist can log out from the system	Click on log out button	Patient click on log out button	1. Dentist click on logout button	None	Dentist log out from the system	System navigate to login page	Dentist log out from the system	Pass
TC L/03	Validate that staff can log out from the system	Click on log out button	Staff click on log out button	1. Staff click on logout button	None	Staff log out from the system	System navigate to login page	Staff log out from the system	Pass

6.6 Conclusion

In this chapter, all activities involved in testing phase has been explained and there are three personnel that involved in testing the system and test strategy that has been used is dynamic testing which consists of two different methods which are black-box and white-box. In the next chapter, all the strengths, weaknesses, improvement and contribution of this system will be explained.

CHAPTER VII

CONCLUSION

7.1 Introduction

The last chapter will describe the weaknesses, strengths, proposition for improvement and the contribution of DCMS system. DCMS has successfully completed within the time given but it has several strengths and weaknesses. All the strengths and weaknesses will be explained and there are some suggestions to improve the system.

7.2 Observation on Weaknesses and Strengths

DCMS were introduced to solve the existing dental clinic system. The computerized system, DCMS has many strengths compared to the existing system but there are several weaknesses.

7.2.1 Strengths

i. This system simplifies appointment process as it can be made online. Appointment request is available all the time.

ii. Patient can know their appointment status that has been requested either it is approved or rejected.

iii. DCMS allows patient to update their personal information online

iv. Dentist can view lists of appointment and follow-up appointment accordingly

v. Staff can view list of upcoming appointments and quantity of medicine

vi. This system is classified into three types of user which are staff, patient and dentist. Each user has different access level to the system. Hence, it enhances security of patient information as it requires authentication.

7.2.2 Weaknesses

i. There is no interactive tooth chart as the dentist can easily tick on the chart of defect tooth

ii. This system does not provide notification system that reminds user of their appointment

iii. DCMS does not provide backup and recovery

7.3 Propositions for Improvement

DCMS system can be improved better by having an interactive tooth chart as the dentist can automatically tick on the defect tooth. Other than that, notification system should be implemented by sending notification via messaging or email as it can remind patient of their appointments. Backup and recovery should be implemented to avoid data lost.

7.4 Contribution

DCMS is a system that is mainly develop for dental clinic. This system helps dental clinic to have smooth and better management. Firstly, patient information is more secured as it requires user authentication. There is a lot of dental clinic that still using paper-based system and it can cause data lost.

Besides, appointment can be made online and patient can also view lists of upcoming appointment without having to go to the dental clinic. Dentist can also record patient treatment easily as they can tick on the tooth checkbox and medicine can be given out to the patient by click on the medicine name and quantity without having to write as it can cause confusion to the staff.

Other than that, staff can print receipt and the receipt include patient treatment and prescribed medicine from dentist

7.5 Conclusion

DCMS not only provides an opportunity to the dental clinic to improve their operation but it can increase the profitability to an organization. DCMS does meet the objectives as it includes variety of functions such as manage appointment,

manage visit of the patient, treatment chart and payment. It also includes concepts that facilitate the transition from a manual dental office to a computerized system.

Furthermore, DCMS has meet non-functional requirements such as security and performance requirements and the objectives and it indicates the system has reached the goal.

.

REFERENCES

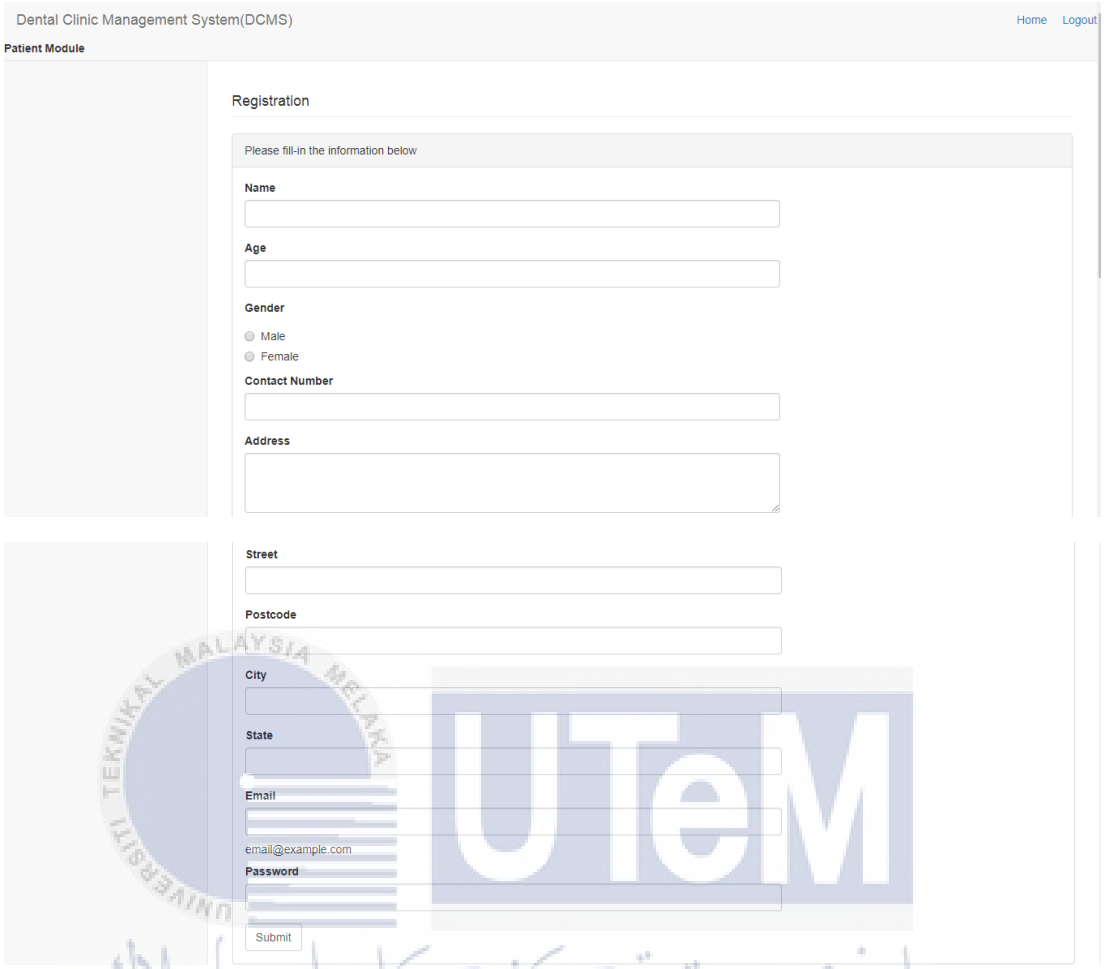
Calonge, M. F. (2011). *Oral Health Program Report*. Accessed on May 16 from http://www.doh.gov.ph/content/oral-health-program.html

Kahn, E. and Benjamin, S. DDS (2017)Computerized Charting and the Electronic Health Record in Dentistry. Retrieved from http://www.dentaleconomics.com/articles/print/volume-92/issue-4/features/computerized-charting-and-the-electronic-health-record-in-dentistry.html

Song, M., Spallek, H., Polk, D., Schleyer, T. & Wali, T. (2009). How information systems should support the information needs of general dentists in clinical settings: Suggestions from a qualitative study. *BMC Medical Informatics and Decision Making*. doi:10.1186/1472-6947-10-7.

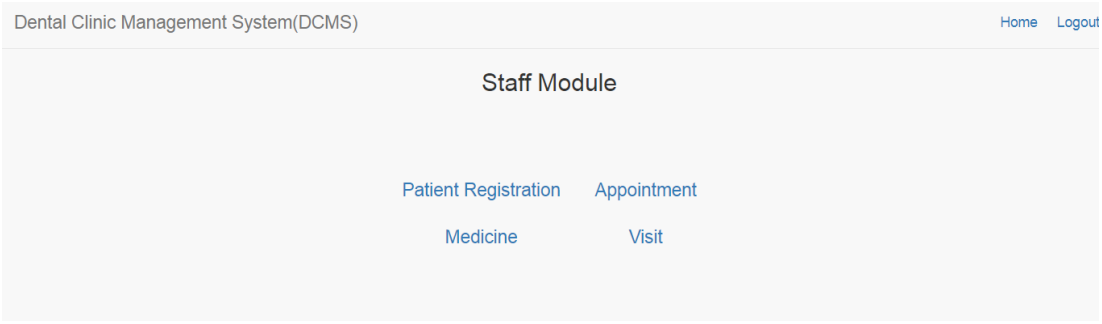
USER MANUAL

This is a homepage of DCMS and there are about, operating hours, contact, sign-up and logout button that enable user to get information regarding the dental clinic. When the user click on the button, it will navigate to the section.

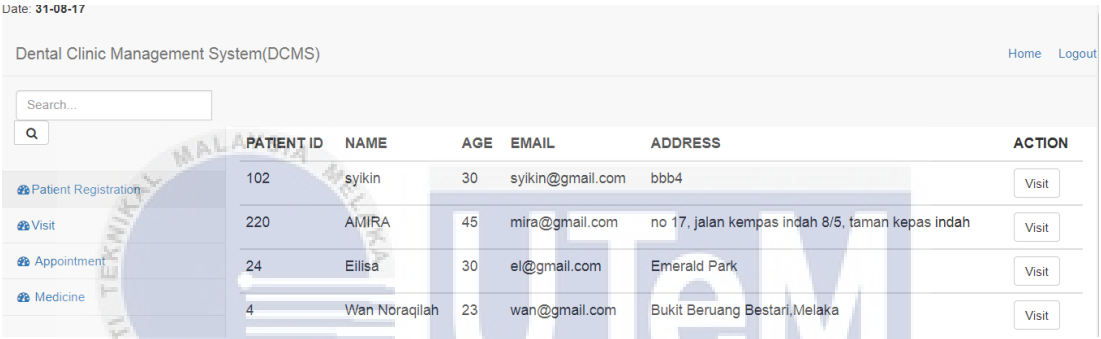


Sign-up page is the registration of new patient and patient need to fill-in all the data.

This is a login page for three different users which are patient, staff and dentist. They need to enter their email address and password. Then, click on login button.

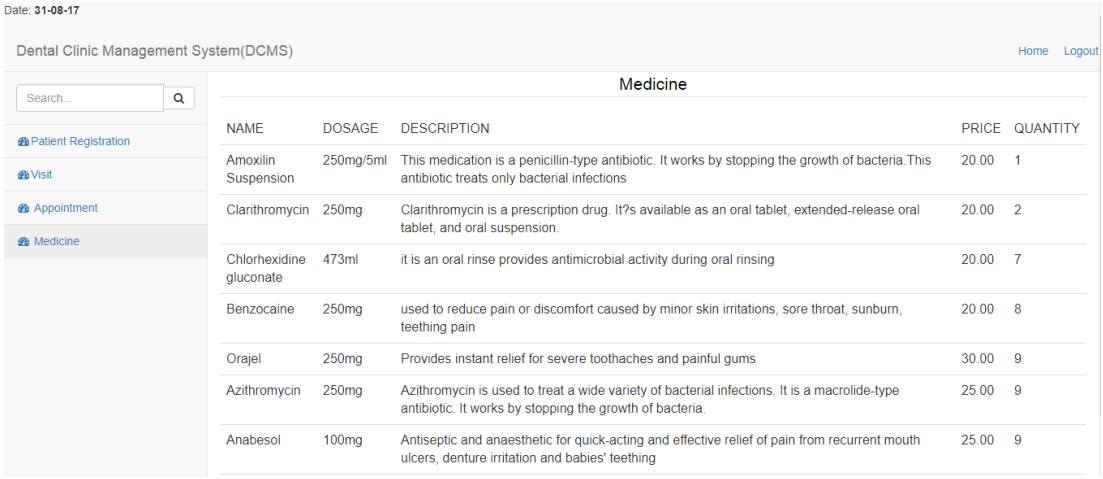


There are four functions in staff module which are patient registration, appointment, medicine and visit.

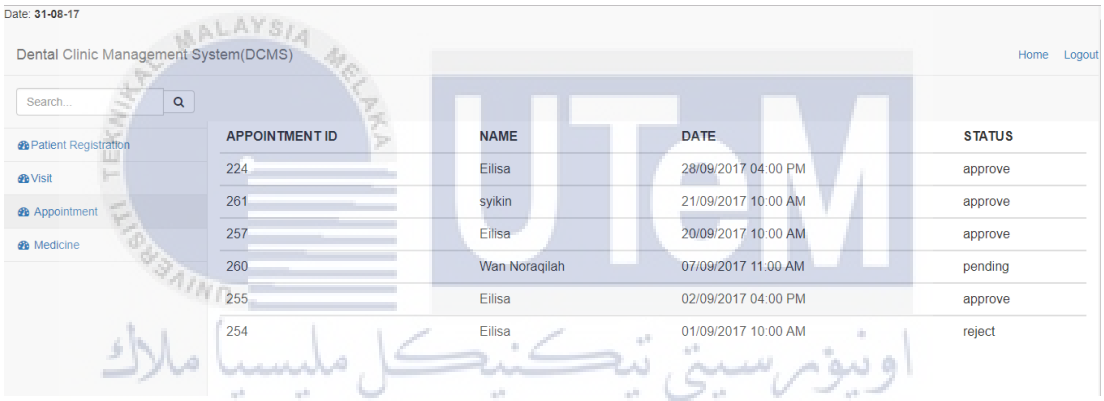


This is a patient registration page on staff module. Those are list of patient name from the database. Staff need to click on visit button when the patient visit the dental clinic.

When the staff click on the visit button of the visiting patient, the page will navigate to visit page for further use such as payment.

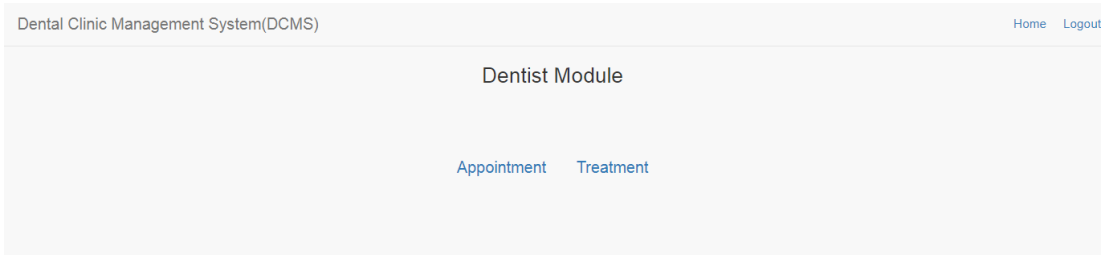


Staff can view quantity of medicine by click on medicine button

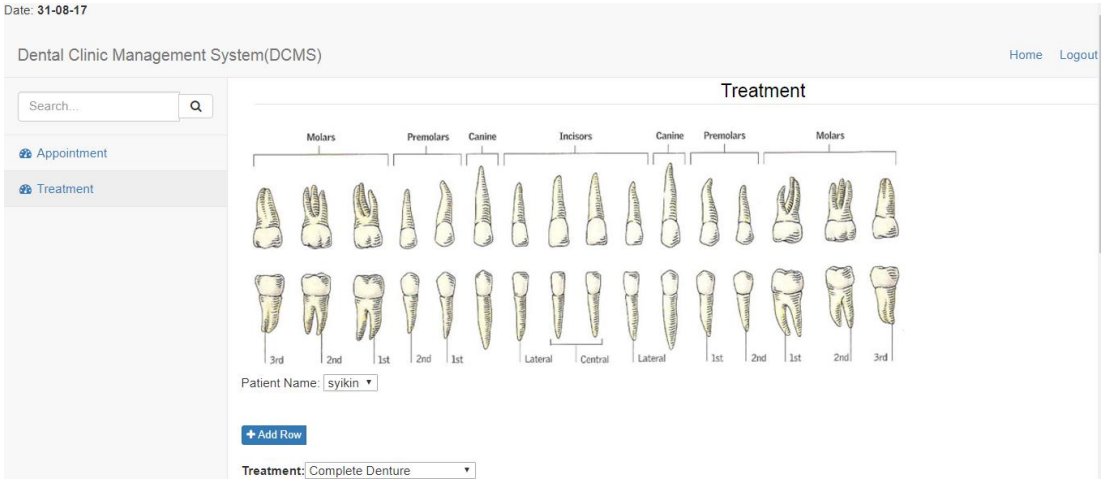


This is a list of appointments that can be viewed by the staff.

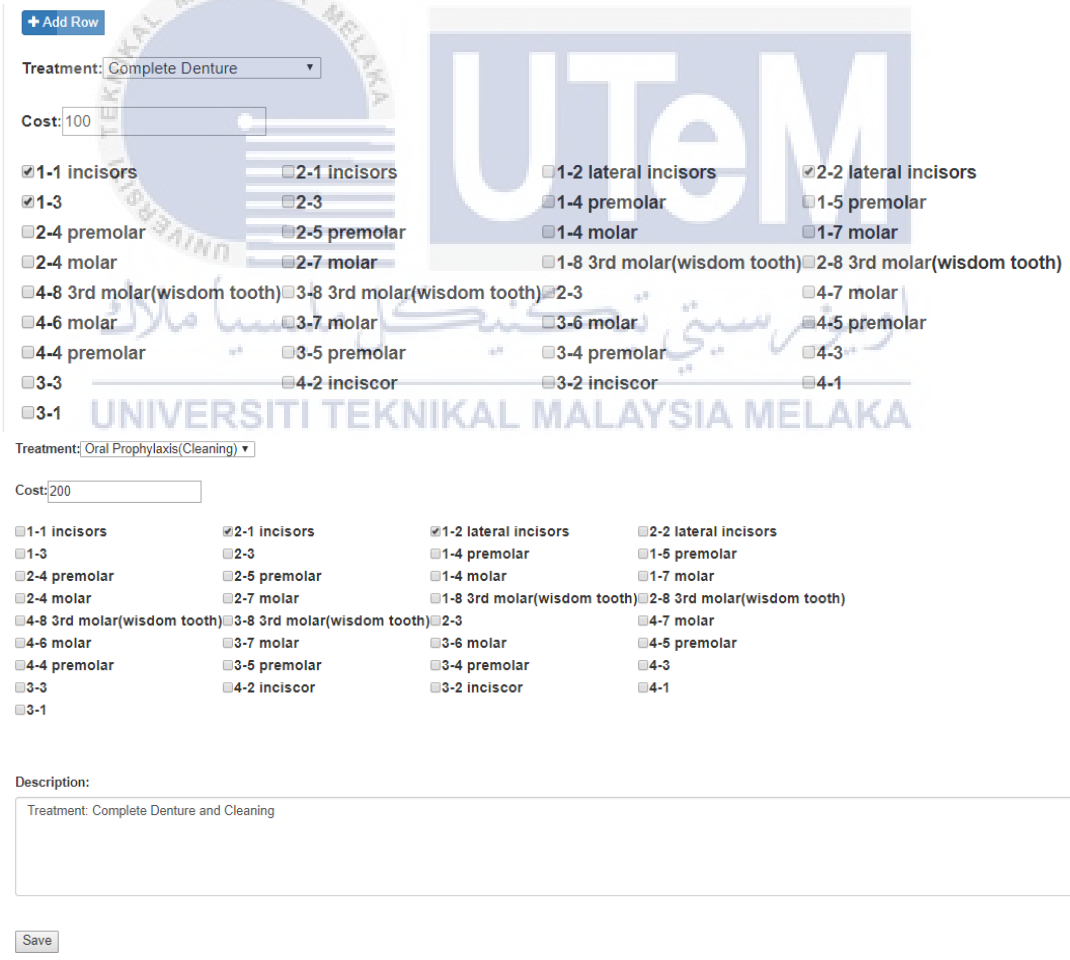
Treatment:



This is a dentist module and there are two functions on dentist module which are appointment and treatment.

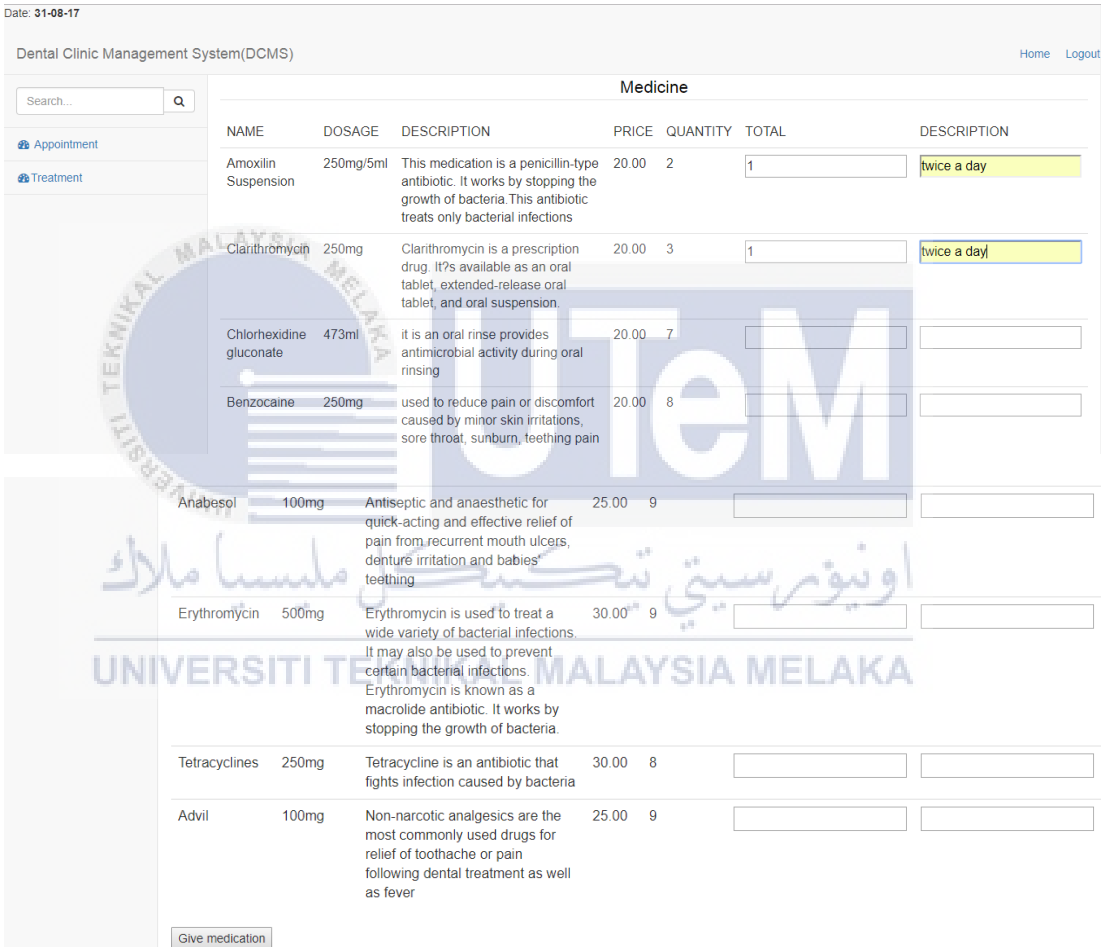


This is a treatment page. The dentist need to select patient name as it will display list of visiting patients.

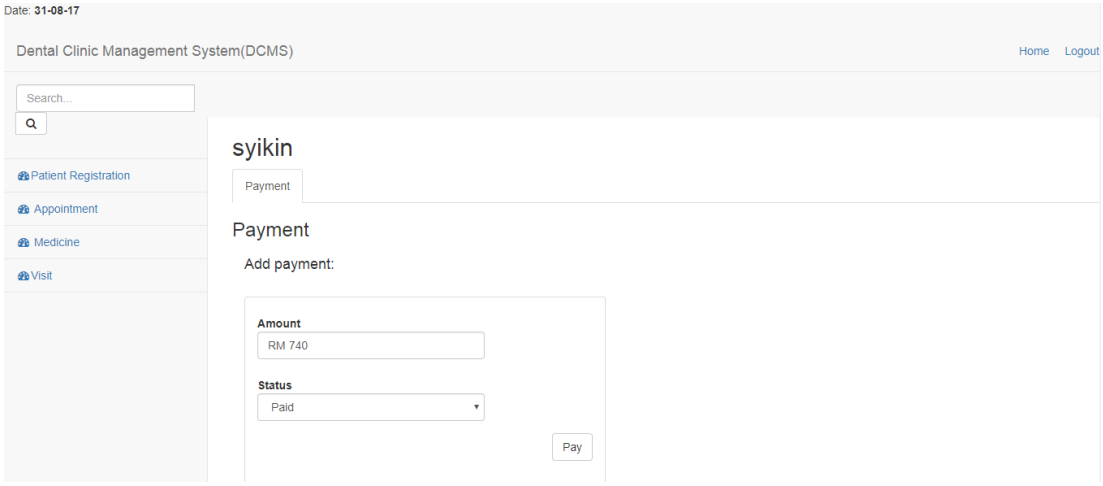


The dentist need to choose treatment and they can click on the checkbox. Dentist can

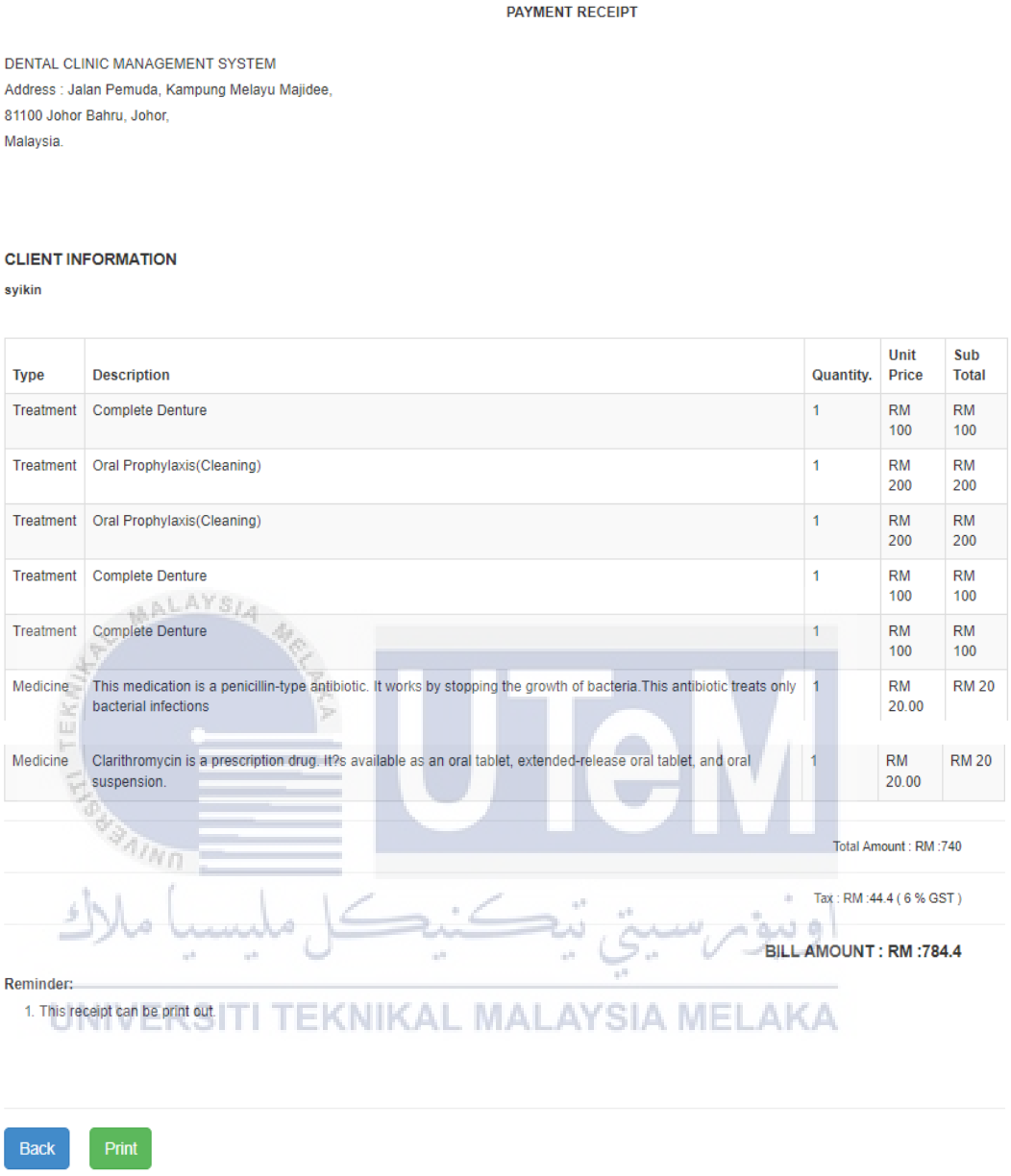
also add new treatment on the same patient and they can write comment on the description box below.



it will proceed to medicine page and the dentist need to fill-in number of medicine to be given out and the description.



This is a payment page. This page is from the visit page where the staff need to click on patient information button and it will display total amount of the treatment. Staff need to click on pay button to view payment receipt.



PAYMENT RECEIPT

DENTAL CLINIC MANAGEMENT SYSTEM
Address : Jalan Pemuda, Kampung Melayu Majidee,
81100 Johor Bahru, Johor,
Malaysia.

CLIENT INFORMATION

syikin

Type	Description	Quantity.	Unit Price	Sub Total
Treatment	Complete Denture	1	RM 100	RM 100
Treatment	Oral Prophylaxis(Cleaning)	1	RM 200	RM 200
Treatment	Oral Prophylaxis(Cleaning)	1	RM 200	RM 200
Treatment	Complete Denture	1	RM 100	RM 100
Treatment	Complete Denture	1	RM 100	RM 100
Medicine	This medication is a penicillin-type antibiotic. It works by stopping the growth of bacteria.This antibiotic treats only bacterial infections	1	RM 20.00	RM 20
Medicine	Clarithromycin is a prescription drug. It?s available as an oral tablet, extended-release oral tablet, and oral suspension.	1	RM 20.00	RM 20

Total Amount : RM :740

Tax : RM :44.4 (6 % GST)

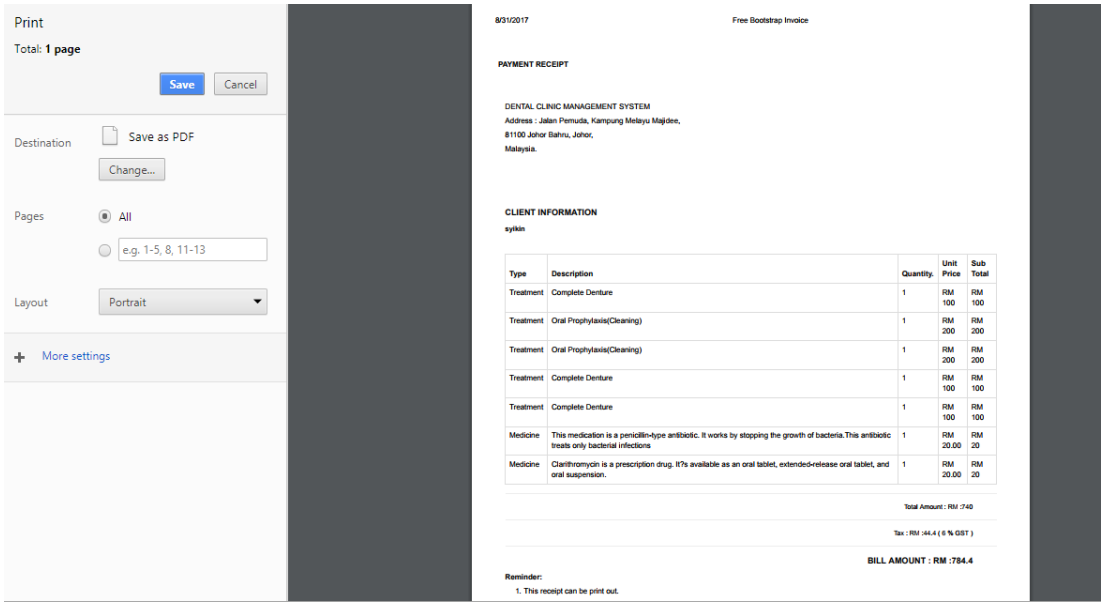
BILL AMOUNT : RM :784.4

Reminder:

1. This receipt can be print out.

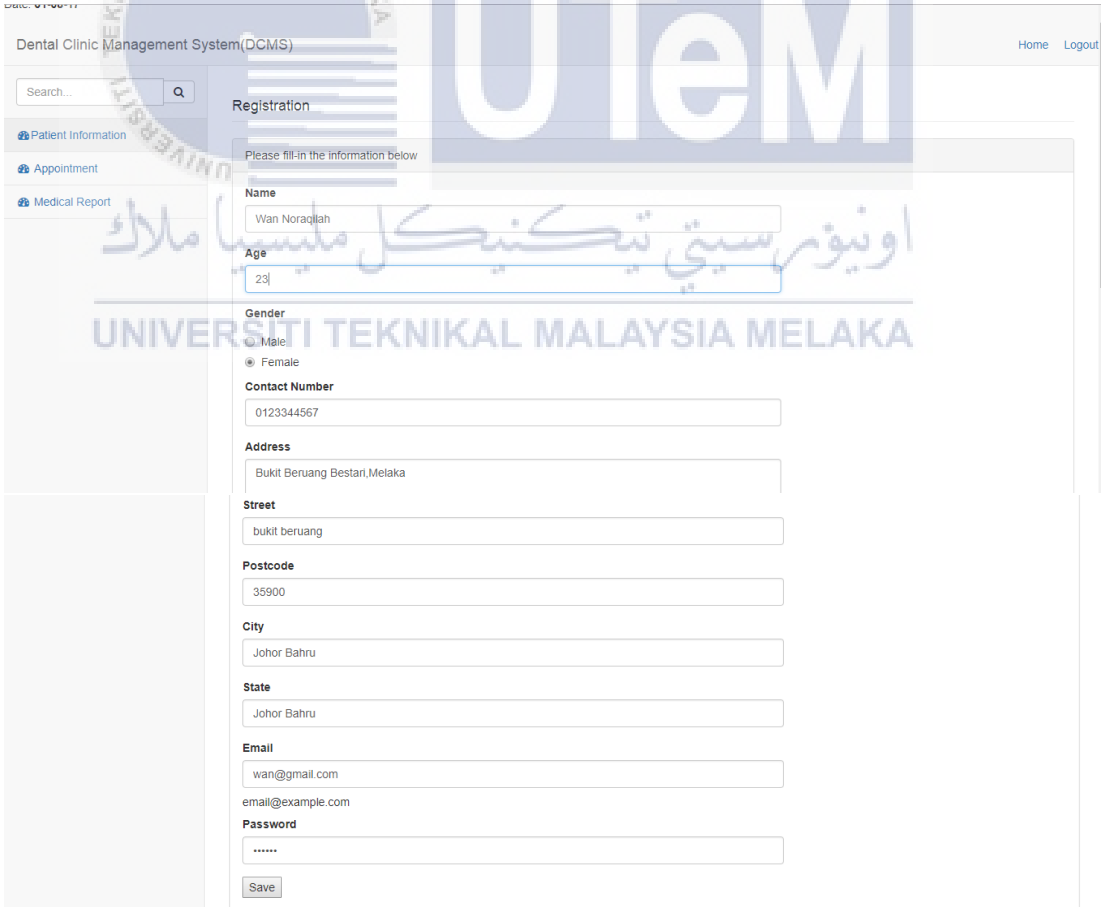
Back Print

This is a payment receipt page and it will display list of treatment, and total amount including GST. Staff can print the receipt by click on print button.



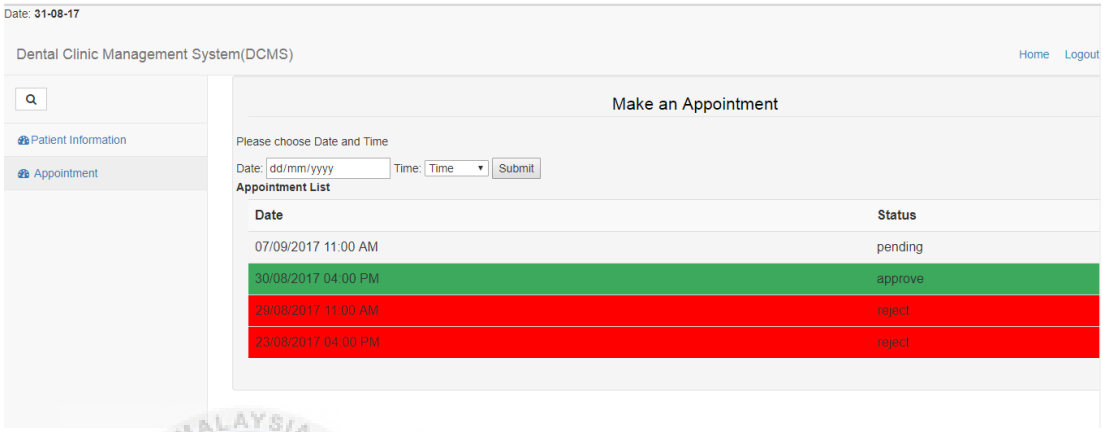
This is a receipt that can be printed.

Patient Module:

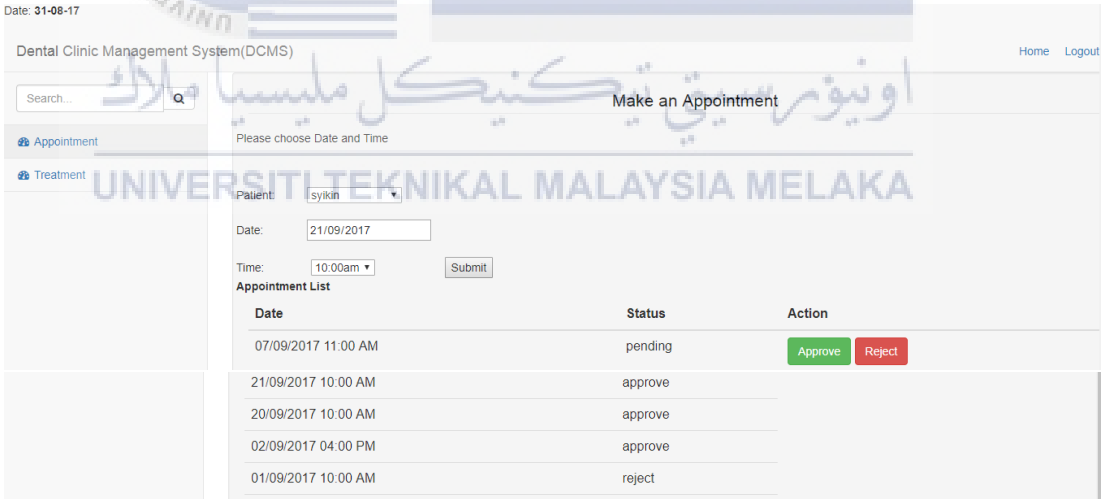


This is a registration page on patient module. This page enable patient to update or edit their personal information.

Appointment:



This is appointment page on patient module. Patient need to choose date and time and the chosen date and time will be displayed in the table with appointment status. Patient can also view their appointment status that has been approved or rejected by the dentist.



This is appointment page on dentist module. Dentist need to choose patient name, date and time. Requested appointment will be displayed in the table including appointment that has been requested by the patient and the dentist need to approve or reject the appointment.