



**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

# **W O R K S H O P 1**

## **R E P O R T**

<b>Name</b>	: Shazana binti Mohammed Affandi
<b>Course</b>	: BITI
<b>Project Title</b>	: Dental Clinic Management System
<b>Supervisor Name</b>	: Nuzulha Khilwani binti Ibrahim
<b>Evaluator Name</b>	: Dr. Zeratul Izzah binti Mohd Yusoh

## TABLE OF CONTENT

CONTENTS	PAGES
CHAPTER 1 1) Introduction 2) Problem Statement 3) Objective 4) Scope	3
CHAPTER 2 1) Analysis of Problem 2) Structure Chart	4 - 5
CHAPTER 3 1) Flowchart 2) ERD 3) Data Dictionary 4) User Interface	6 - 16
CHAPTER 4 1) Introduction 2) Coding	12 - 27
CHAPTER 5 1) Conclusion	28
REFERENCES	29

# **CHAPTER 1**

## **Introduction**

Dental Clinic Management System is a system developed to facilitate the employees who is working in dental clinics. This system helps to reduce the problem of losing patients' data and clinical information that occurs when they use system manual. This system provides the facility to record and store patient data more systematically.

## **Problem Statement**

In this era, number of community that stays in Malaysia is increasing day by day. If the clinic still using manual system, this giving difficulties for the employees and patients because the process is unsystematic. Apart from that, it finds hard for employees to search earlier patients' data in fact, they were in risk if the data lost. The manual system also giving difficulties for dentists to know about their appointment.

## **Objective**

The objective of Dental Clinic Management System is designed to assist employees in dental clinics to record, store and update data of patients more systematically. Moreover, this system makes it easier for employees to search for patients when needed rapidly. The information stored is well maintained, as this system provides security such as the admin, nurse and dentist must enter the username and password before opening the application. This project also builds and designs a more systematic dental clinic system and to assess the consistency the use of more advanced system.

## **Scope**

Module to be developed in this system is log in, registration, add, update, and delete staffs' and patients' data. Target user for this system is admin, nurses and dentists.

## CHAPTER 2

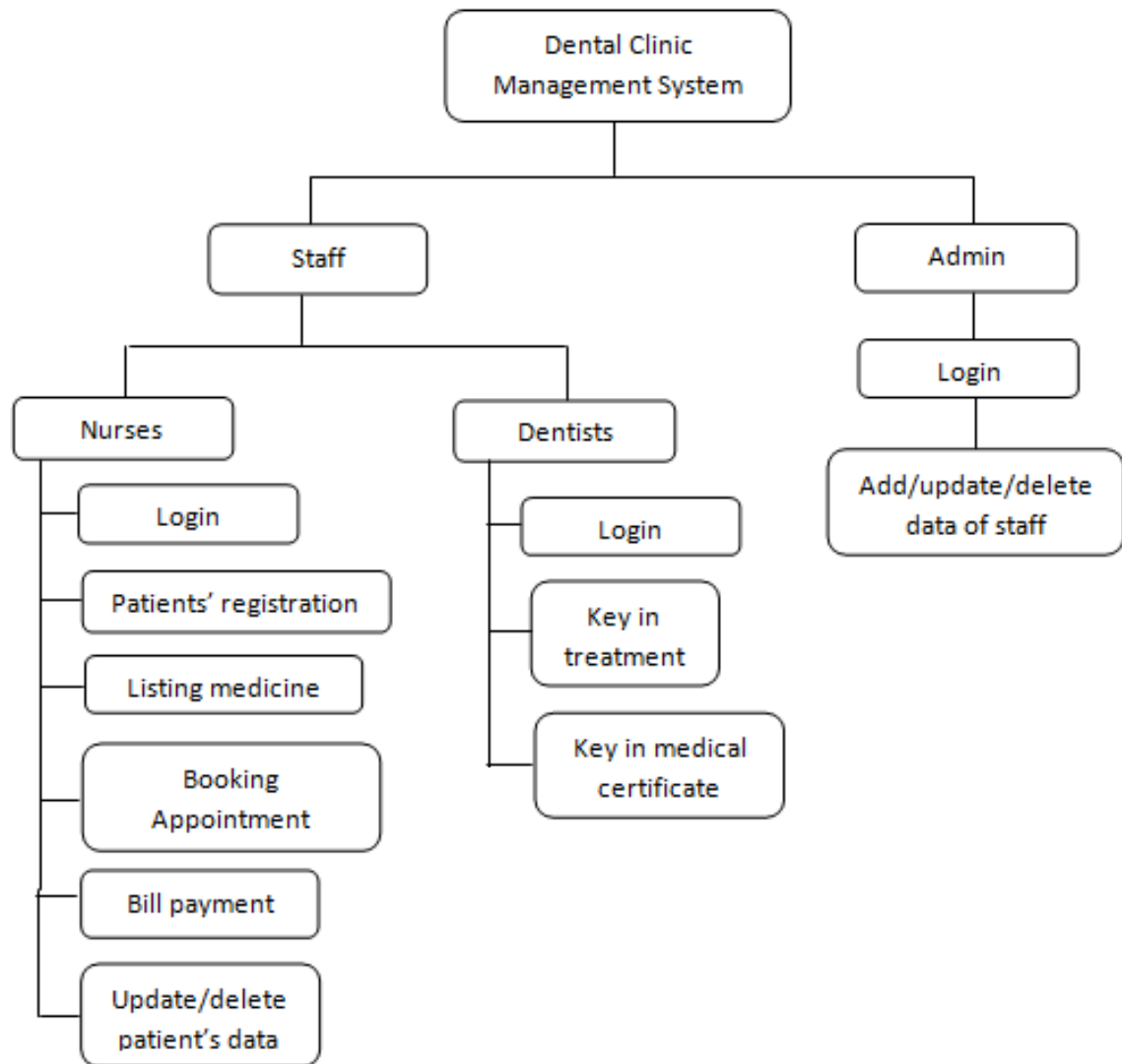
### **Analysis of Problem**

The first problem is difficulties for nurse and dentist to search earlier patients' data. This is because if they still using manual system, there is a lot of file and paper they need to keep, so it found difficult when they want to search for patients' data since they need to one by one. But with this system, they can only search patients' ID, name or identity card number as it also saves more time and make works more systematic.

Other than that, the percentage to losing patients' data quite high since they will forgot where they put the data and the data might be drop out. This will make harder to for the nurses to key in new data and make dentists more problems when they forgot what treatment and medicines they give to the patients. This problem can be solved with this system which it helps them to stored patients' data in one place.

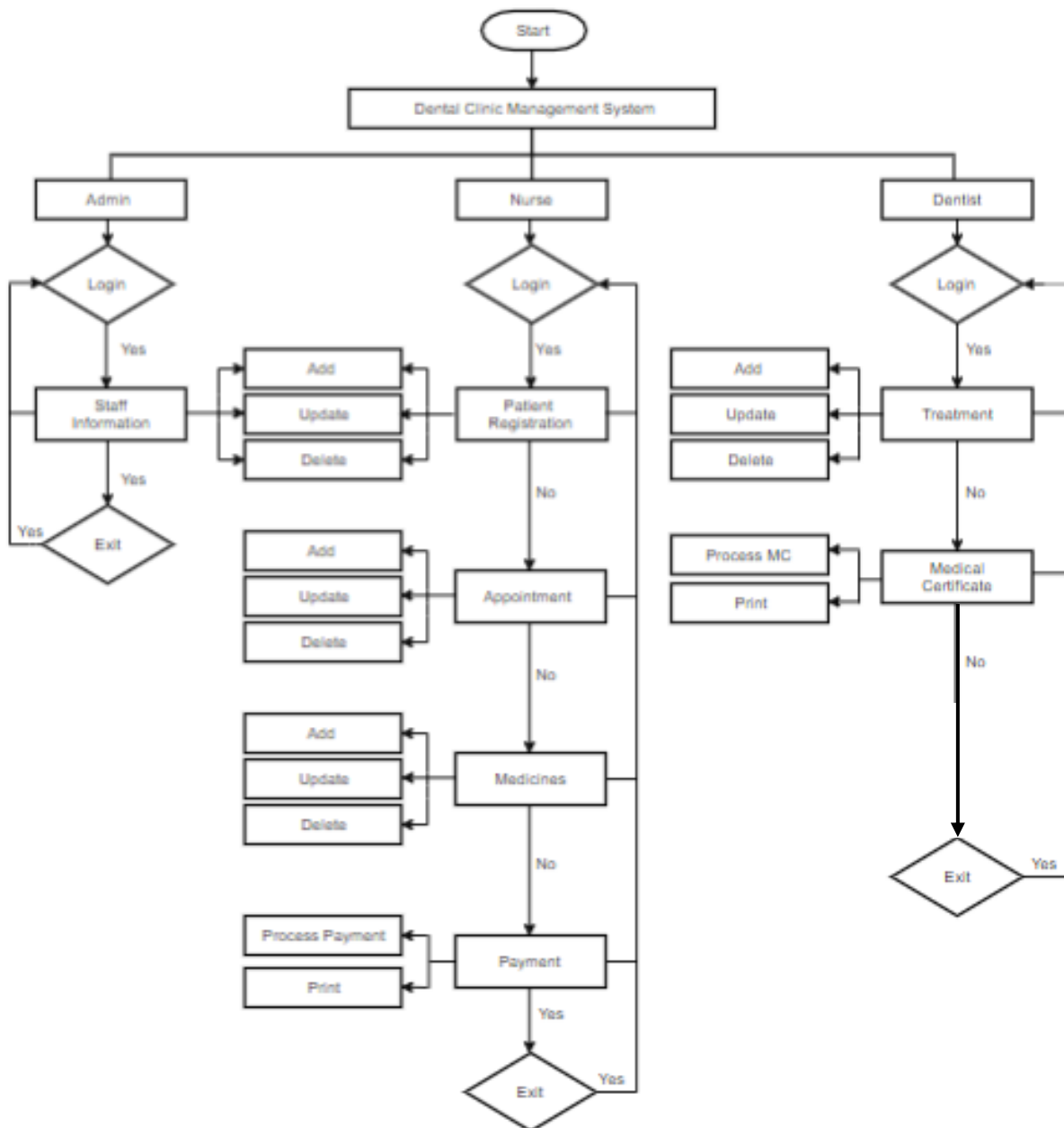
Lastly, it found difficult for dentist for dentists to know about their appointment with patients since nurse should remind them about the appointment. If the nurse also forgot about the appointment, it will be problem to patient that comes on that time. So, with this system, dentists can see their appointment schedule without being reminded by nurse and they can contact their patients if they can't meet on that time.

## Structure Chart

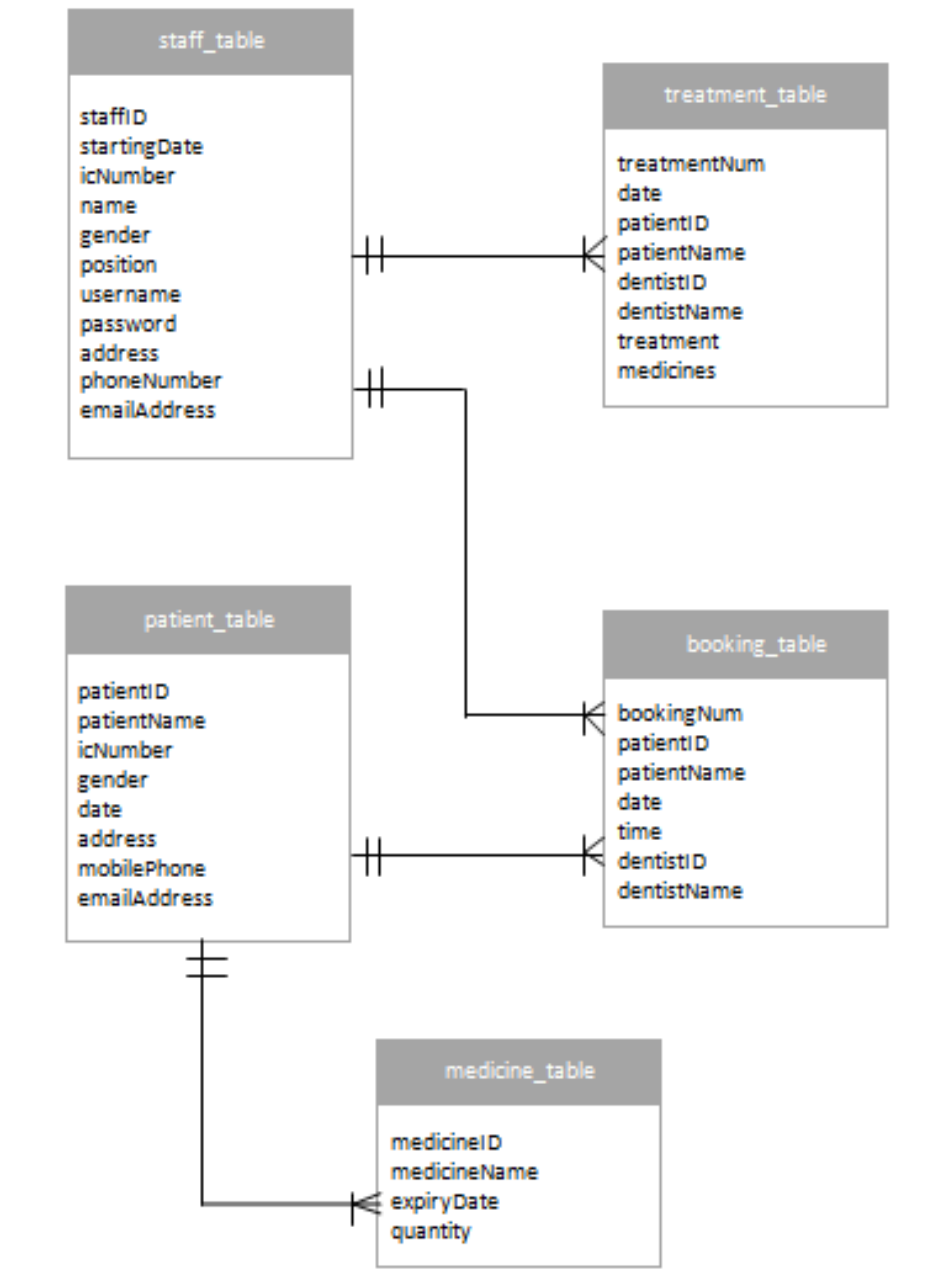


## CHAPTER 3

### Flowchart



## ERD



## Data Dictionary

### 1. staff\_table

Data Name	Data Type	Data Description
staffID	Varchar(10)	Staff id
startingDate	Date	Date of starting work
icNumber	Varchar(20)	Staff's identity card number
name	Varchar(50)	Staff's name
gender	Varchar(10)	Staff's gender
position	Varchar(10)	Staff's position
username	Varchar(50)	Staff's username
password	Varchar(50)	Staff's password
address	Varchar(50)	Staff's address
phoneNumber	Varchar(11)	Staff's phone number
emailAddress	Varchar(50)	Staff's email address

### 2. treatment\_table

Data Name	Data Type	Data Description
treatmentNum	Int(11)	Treatment number
date	Date	Date of treatment
patientID	Varchar(10)	Patient's ID
patientName	Varchar(30)	Patient's name
dentistID	Varchar(10)	Dentist's ID
dentistName	Varchar(30)	Dentist's name
treatment	Varchar(10)	Treatment of patients
medicines	Varchar(50)	Medicines for patients



3. patient\_table

Data Name	Data Type	Data Description
patientID	Varchar(10)	Patient's ID
patientName	Varchar(50)	Patient's name
icNumber	Varchar(20)	Patient's identity card
gender	Varchar(10)	Patient's gender
date	date	Date registration
address	Varchar(50)	Patient's address
mobilePhone	Varchar(11)	Patient's mobile phone
emailAddress	Varchar(50)	Patient's email address

4. booking\_table

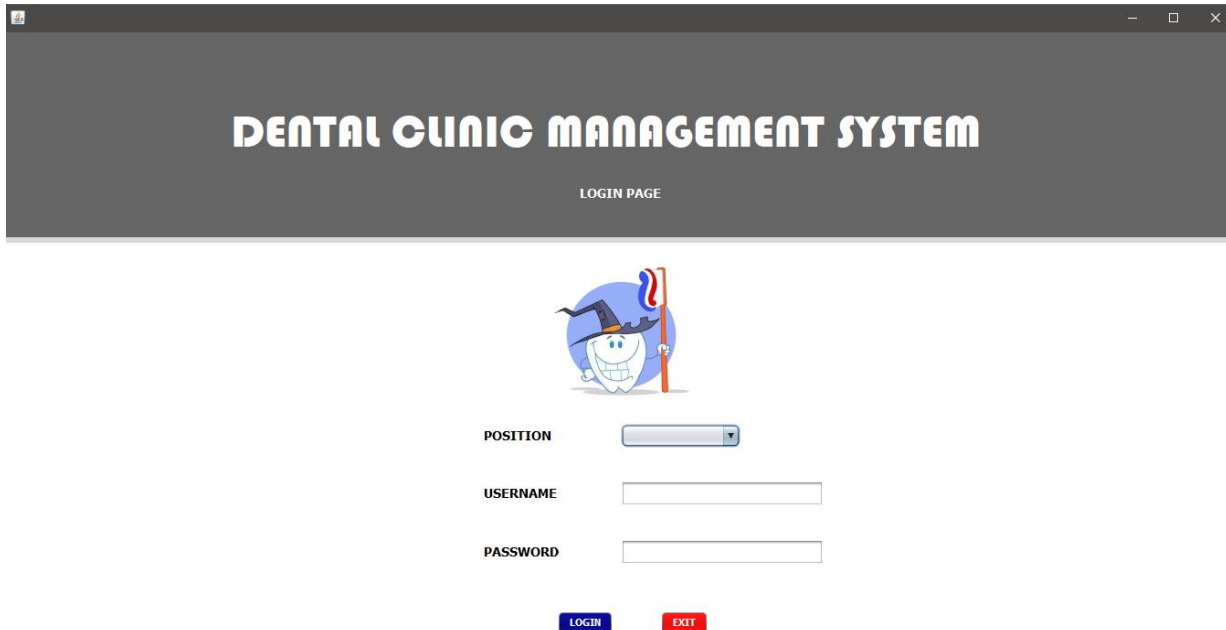
Data Name	Data Type	Data Description
bookingNum	Int(11)	Booking number
patientID	Varchar(10)	Patient's ID
patientName	Varchar(30)	Patient's name
date	Date	Date for appointment
time	Time	Time for appointment
dentistID	Varchar(10)	Dentist's ID
dentistName	Varchar(30)	Dentist's name

5. medicine\_table

Data Name	Data Type	Data Description
medicineID	Varchar(10)	Medicine's ID
medicineName	Varchar(50)	Medicine's name
expiryDate	Date	Expired date for medicine
quantity	Int(10)	Quantity of medicine

## User Interface

1. User should choose which position such as admin, dentist or nurse and enter correct username and password before login.



**DENTAL CLINIC MANAGEMENT SYSTEM**

LOGIN PAGE

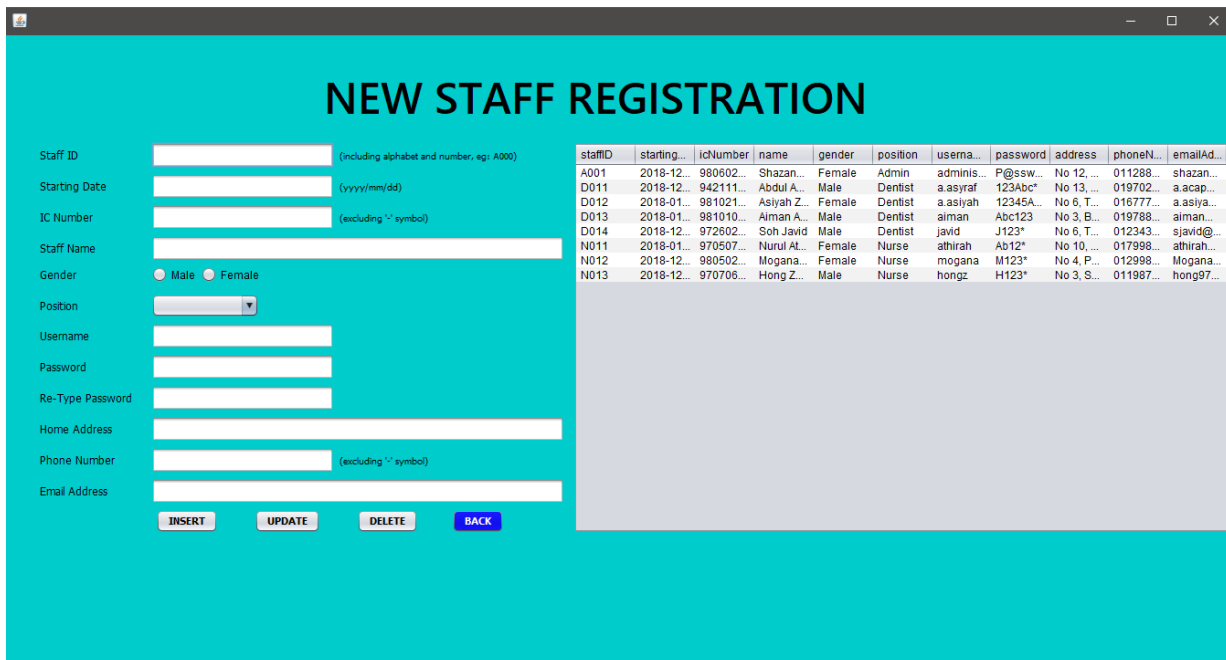


POSITION

USERNAME

PASSWORD

2. This is staffs' information table. Only admin can update, delete and insert data of staffs to the table.



## NEW STAFF REGISTRATION

Staff ID  (including alphabet and number, egi: A000)

Starting Date  (yyyy/mm/dd)

IC Number  (excluding '-' symbol)

Staff Name

Gender ☐ Male ☐ Female

Position

Username

Password

Re-Type Password

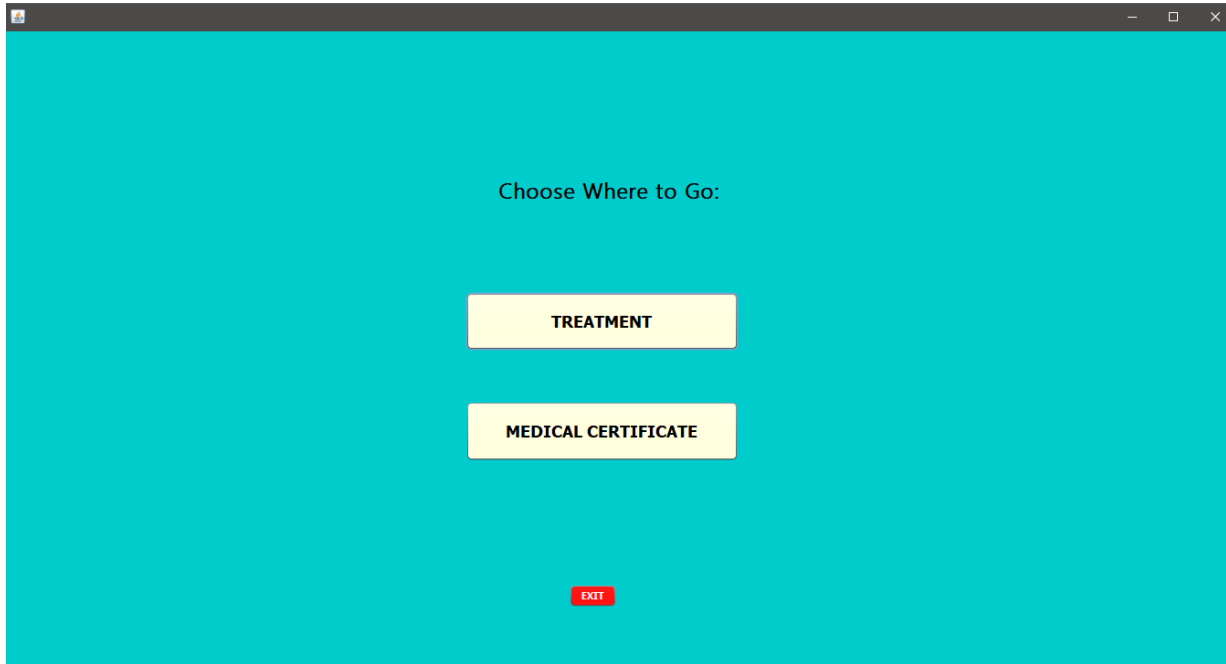
Home Address

Phone Number  (excluding '-' symbol)

Email Address

staffID	starting...	icNumber	name	gender	position	userna...	password	address	phoneN...	emailAd...
A001	2018-12...	980602...	Shazan...	Female	Admin	adminis...	P@ssw...	No 12, ...	011288...	shazan...
D011	2018-12...	942111...	Abdul A...	Male	Dentist	a.asyraf	123Abc*	No 13, ...	019702...	a.acap...
D012	2018-01...	981021...	Asiyah Z...	Female	Dentist	a.asiyah	12345A...	No 6, T...	016777...	a.asiya...
D013	2018-01...	981010...	Alman A...	Male	Dentist	alman	Abc123	No 3, B...	019788...	alman...
D014	2018-12...	972602...	Soh Javid	Male	Dentist	javid	J123*	No 6, T...	012343...	sjauid@...
N011	2018-01...	970507...	Nurul Al...	Female	Nurse	athirah	Ab12*	No 10, ...	017998...	athirah...
N012	2018-12...	980502...	Mogana...	Female	Nurse	mogana	M123*	No 4, P...	012998...	Mogana...
N013	2018-12...	970706...	Hong Z...	Male	Nurse	hongz	H123*	No 3, S...	011987...	hong97...

- When dentist login their account, it will bring them to dentist page which the dentist need to choose which page they want to go. When they click “EXIT”, the dentist will go to login page.



Choose Where to Go:

TREATMENT

MEDICAL CERTIFICATE

EXIT

- When dentist click “TREATMENT” button, they will go to this page which the dentist can insert, update and delete data. The dentist also can see the list of patients’ treatment.



## PATIENT'S TREATMENT

Treatment Number  (eg: 1)

Date  (yyyy/mm/dd)

Patient ID

Patient Name

Treatment

Medicines

treatmentNum	date	patientID	patientName	treatment	medicines
1	2018-12-21	P101	Fatihah Zamri	Scalling	Antibiotic
2	2018-12-21	P102	Safiahani Ishak	Braces	Ibuprofen
3	2018-12-28	P103	Syakirah Rashid	Braces	Ibuprofen
4	2018-01-10	P104	Hani Suraya	Scalling	Antibiotics

INSERT UPDATE DELETE BACK

5. .

**SIJIL CUTI SAKIT**

Dengan ini saya mengesahkan bahawa saya telah memeriksa Encik/Cik/Puan

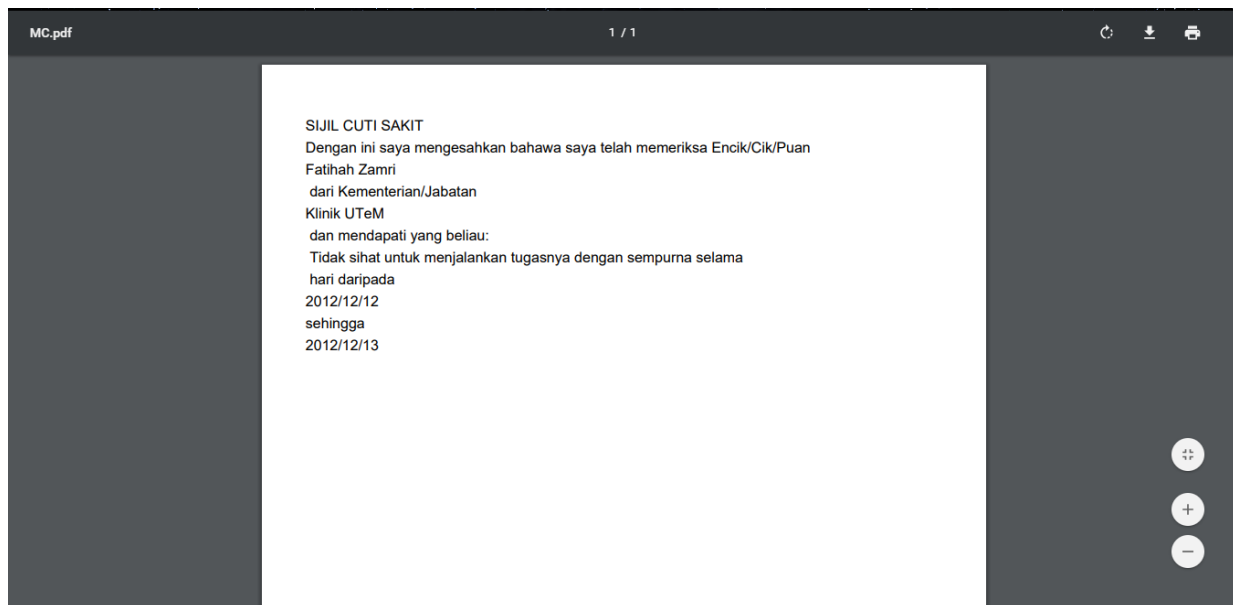
dari Kementerian/Jabatan

dan mendapati yang beliau:

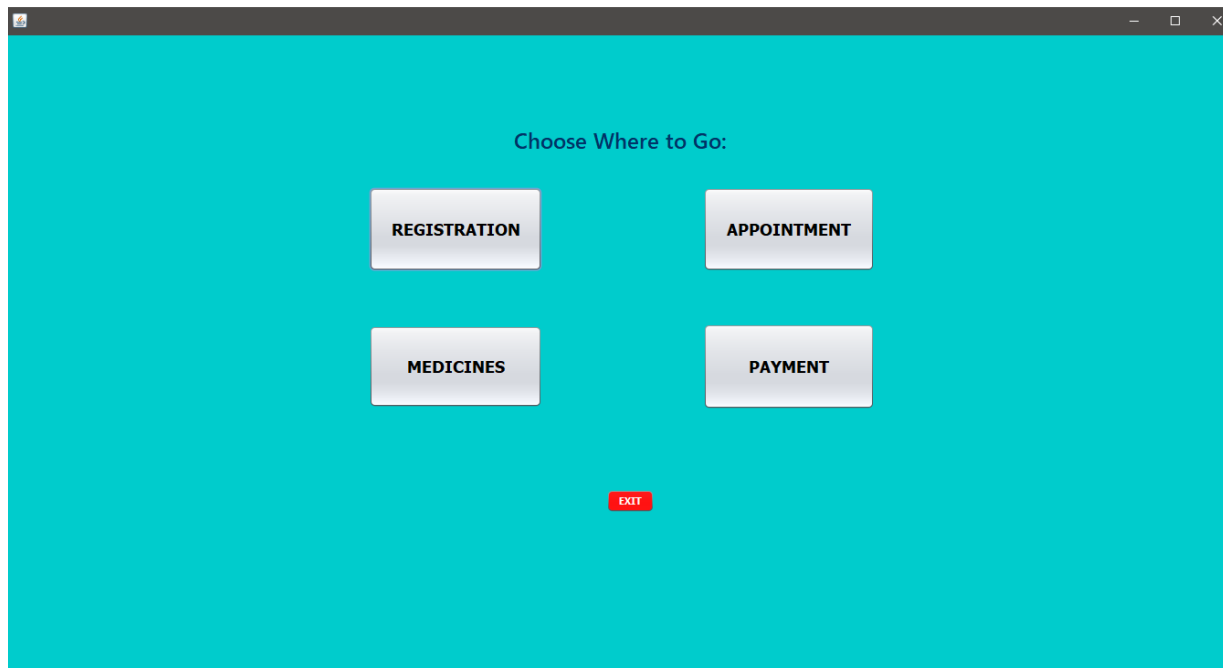
Tidak sihat untuk menjalankan tugasnya dengan sempurna selama

hari daripada  sehingga

6. When dentist click “SAVE” button, the mc will save in .pdf file.



- When nurse login their account, it will bring them to nurse page which the nurse need to choose which page they want to go. When they click “EXIT”, they will go to login page.



- When the nurse clicks on “REGISTRATION” button, they will go to this page. Patient’s registration handles by nurses. They will insert, update and delete all information of patients. After clicking save button, the data will appear on patients’ information table

### PATIENT REGISTRATION

Patient ID:  (eg: P001)

Patient Name:

IC No:  (excluding '-' symbol)

Gender: ☒ Male ☐ Female

Date:  (yyyy-mm-dd)

Address:

Mobile Phone:  (excluding '-' symbol)

Email Address:

INSERT UPDATE DELETE BACK

patientID	patientName	icNumber	gender	date	address	mobilePhone	emailAddress
P101	Fatihah Zamri	9804210356...	Female	2018-11-11	No 20, Tama...	0177329911	fatihahz@gm...
P102	Safiahanis Is...	9809150155...	Female	2018-11-11	No 3, Adda H...	0123347811	safiahanis@g...
P103	Syakirah Ra...	9804030876...	Female	2018-12-17	No 5, Bandar...	0168971223	syakirah@g...
P104	Hani Suraya	9711210388...	Female	2019-01-03	No 5, Chera...	0117887562	hani@gmail...

9. When the nurse clicks on “APPOINTMENT” button, they will go to this page. They will insert, update and delete all booking information for patients who want to make an appointment with dentist.

The screenshot shows a web application titled "Booking Appointment" with a teal background. On the left, there is a form with the following fields: "Booking Number" (text input with placeholder "(eg: 001)"), "Patient's ID" (dropdown menu), "Patient's Name" (dropdown menu), "Date" (text input with placeholder "(yyyy/mm/dd)"), "Time" (text input with placeholder "(eg: HH:MM:SS)"), "Dentist's ID" (dropdown menu), and "Dentist Name" (dropdown menu). Below the form are four buttons: "INSERT", "UPDATE", "DELETE", and "BACK". On the right, there is a table with the following data:

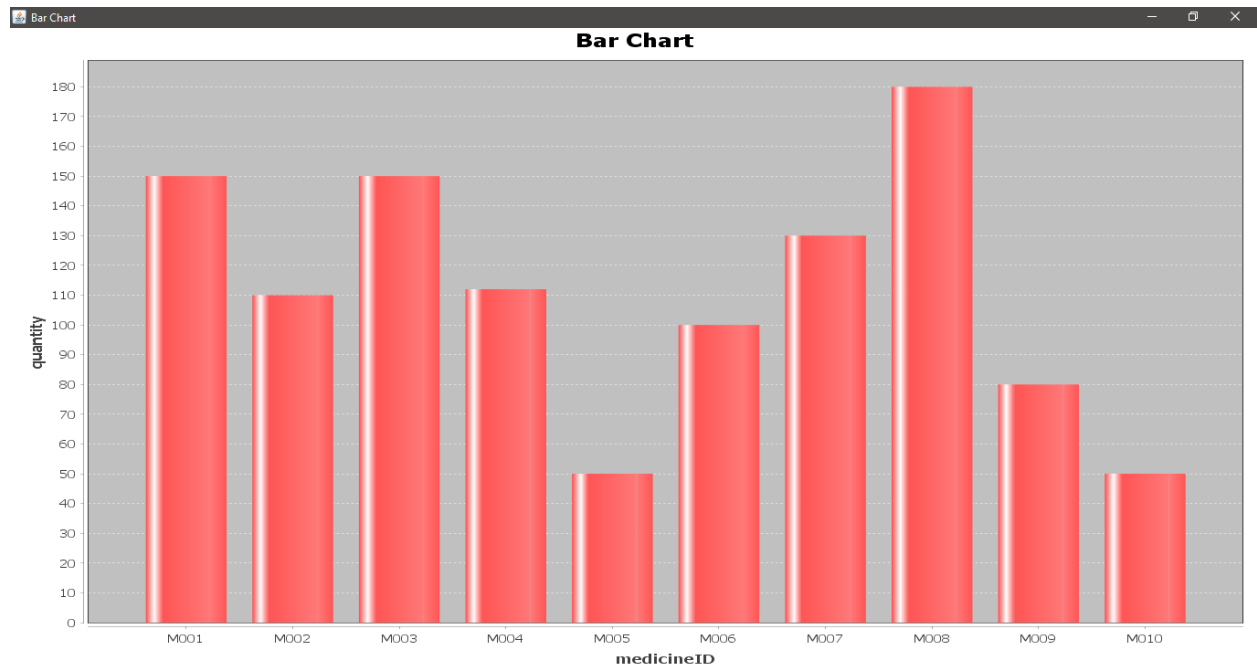
bookingNum	patientID	patientName	date	time	dentistID	dentistName
1	P101	Fatihah Zamri	2019-01-11	10:00:00	D011	Abdul Asyraf
2	P102	Safiahani Ishak	2019-01-08	12:30:00	D012	Asiyah Zuklily
3	P103	Syakirah Rashid	2019-01-08	15:00:00	D013	Alman Azman
4	P104	Hani Suraya	2019-02-04	09:30:00	D014	Soh Javid

10. When the nurse clicks on “MEDICINE” button, they will go to this page. They will insert, update and delete all information about medicines.

The screenshot shows a web application titled "MEDICINE" with a teal background. On the left, there is a form with the following fields: "Medicine ID" (text input with placeholder "(eg: M001)"), "Medicine Name" (text input), "Expiry Date" (text input with placeholder "(yyyy-mm-dd)"), and "Quantity" (text input). Below the form are five buttons: "INSERT", "UPDATE", "DELETE", "BAR CHART", and "BACK". On the right, there is a table with the following data:

medicineID	medicineName	expiryDate	quantity
M001	Antibiotics	2021-12-12	150
M002	Amoxycillin	2021-12-12	110
M003	Ibuprofen	2021-12-12	150
M004	Prednisolone	2012-12-12	112
M005	Dyfenamic	2012-12-12	50
M006	Antibiotics	2025-12-12	100
M007	Amoxycillin	2025-12-12	130
M008	Ibuprofen	2025-12-12	180
M009	Prednisolone	2025-12-12	80
M010	Dyfenamic	2025-12-12	50

11. When the nurse click on “BAR CHAT”, it will pop up this page.

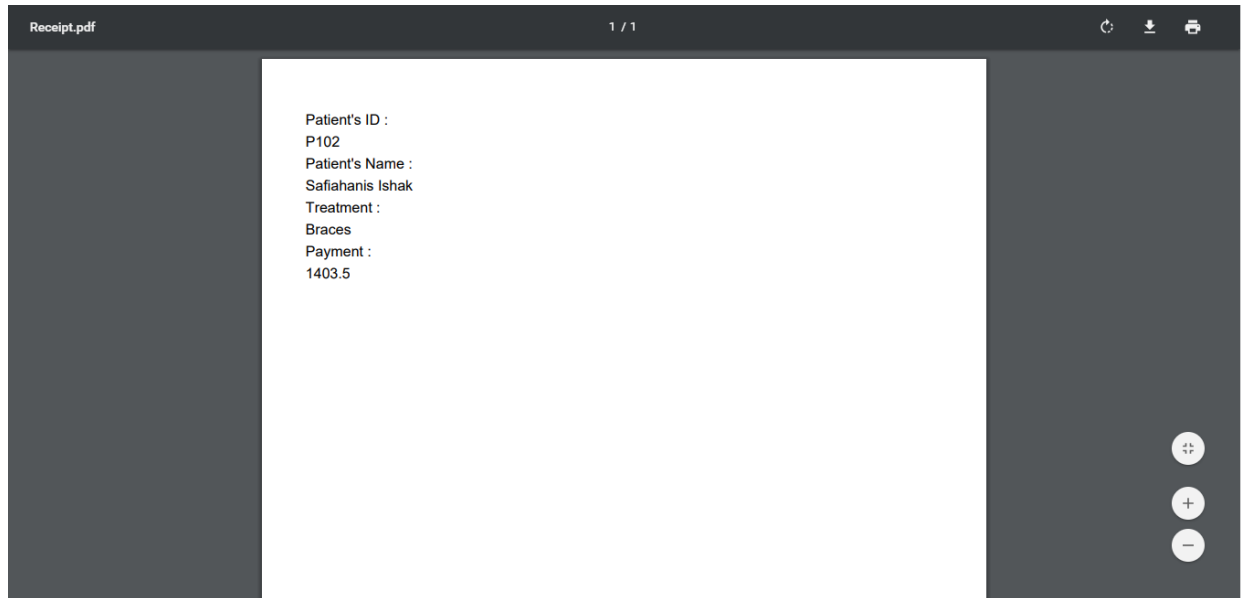


12. This page is where the nurse click on “PAYMENT” button which it will key in by nurse and print it to give to patients

A form titled "PAYMENT RECEIPT" with a teal background. It contains several input fields and buttons. The fields are: Patient's ID (dropdown), Patient's Name (dropdown), Treatment (dropdown), Price (text), Service Charged (text), and Payment (text). At the bottom, there are three buttons: CALCULATE, SAVE, and BACK.

Field	Type
Patient's ID	Dropdown
Patient's Name	Dropdown
Treatment	Dropdown
Price	Text
Service Charged	Text
Payment	Text
CALCULATE	Button
SAVE	Button
BACK	Button

13. When nurse click “SAVE” button, the mc will save in .pdf file.





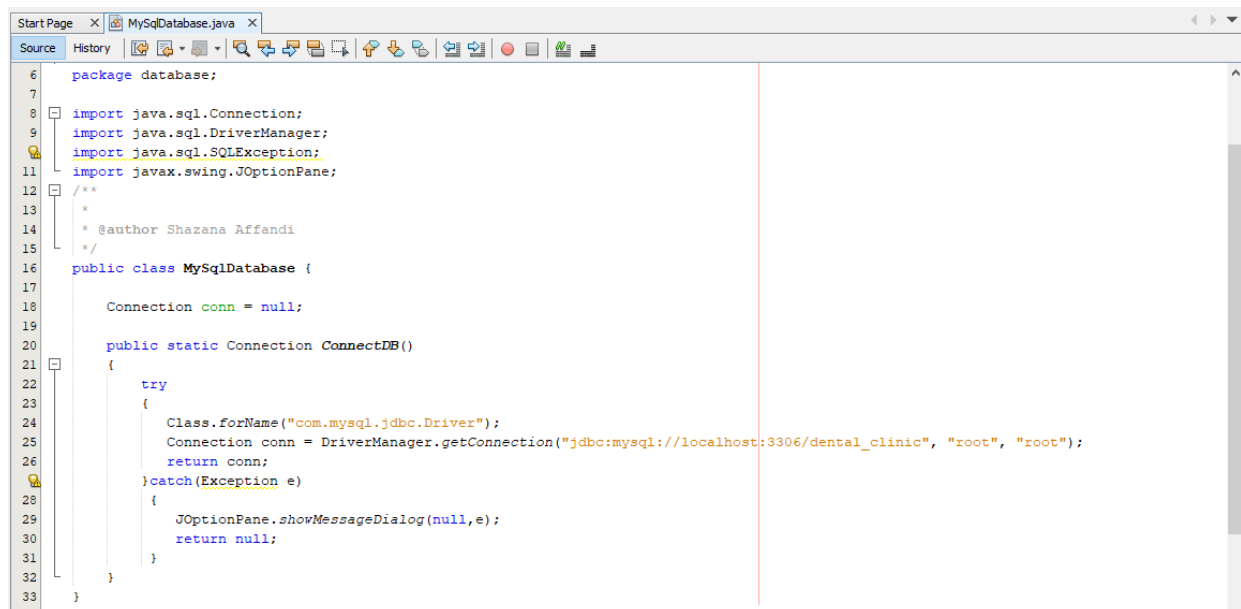
## CHAPTER 4

### Introduction

This Dental Clinic Management System is using Java SE and NetBeans IDE 8.2 in order to build the interface and system. The database used is MySQL.

### Coding

#### 1. Database connection



```
1 package database;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6 import javax.swing.JOptionPane;
7
8 /**
9  *
10  * @author Shazana Affandi
11  */
12 public class MySQLDatabase {
13
14     Connection conn = null;
15
16     public static Connection ConnectDB()
17     {
18         try
19         {
20             Class.forName("com.mysql.jdbc.Driver");
21             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/dental_clinic", "root", "root");
22             return conn;
23         } catch (Exception e)
24         {
25             JOptionPane.showMessageDialog(null, e);
26             return null;
27         }
28     }
29 }
30
31
32
33
```

## 2. Java Class

### a) NewStaff.java

```
Start Page x MySQLDatabase.java x NewStaff.java x
Source History

12 public class NewStaff {
13
14     private String staffID;
15     private String startingDate;
16     private String icNumber;
17     private String name;
18     private String Gender;
19     private String position;
20     private String username;
21     private String password;
22     private String address;
23     private String phoneNumber;
24     private String emailAddress;
25
26     public NewStaff(String staffID, String startingDate, String icNumber, String name, String Gender, String position, String username, String password, St:
27     {
28         this.staffID = staffID;
29         this.startingDate = startingDate;
30         this.icNumber = icNumber;
31         this.name = name;
32         this.Gender = Gender;
33         this.position = position;
34         this.username = username;
35         this.password = password;
36         this.address = address;
37         this.phoneNumber = phoneNumber;
38         this.emailAddress = emailAddress;
39     }
40
41     public String getStaffID()
42     {
43         return staffID;
44     }
45
46     public void setStaffID(String staffID)
47     {
48         this.staffID = staffID;
49     }
50
51     public String getStartingDate()
52     {
53         return startingDate;
54     }
55
56     public void setStartingDate(String startingDate)
57     {
58         this.startingDate = startingDate;
59     }
60
61     public String getIcNumber()
62     {
63         return icNumber;
64     }
65
66     public void setIcNumber(String icNumber)
67     {
68         this.icNumber = icNumber;
69     }
70
71     public String getName()
72     {
73         return name;
74     }
75
76     public void setName(String name)
77     {
78         this.name = name;
79     }
80
81     public String getGender()
82     {
83         return Gender;
84     }
85
86     public void setGender(String Gender)
87     {
88         this.Gender = Gender;
89     }
90
91     public String getPosition()
92     {
93         return position;
94     }
95
96     public void setPosition(String position)
97     {
98         this.position = position;
99     }
}
```

```

101 public String getUsername()
102 {
103     return username;
104 }
105
106 public void setUsername(String username)
107 {
108     this.username = username;
109 }
110
111 public String getPassword()
112 {
113     return password;
114 }
115
116 public void setPassword(String password)
117 {
118     this.password = password;
119 }
120
121 public String getAddress()
122 {
123     return address;
124 }
125
126 public void setAddress(String address)
127 {
128     this.address = address;
129 }
130
131 public String getPhoneNumber()
132 {
133     return phoneNumber;
134 }
135
136 public void setPhoneNumber(String phoneNumber)
137 {
138     this.phoneNumber = phoneNumber;
139 }
140
141 public String getEmailAddress()
142 {
143     return emailAddress;
144 }
145
146 public void setEmailAddress(String emailAddress)
147 {
148     this.emailAddress = emailAddress;
149 }
150 }

```

## b) Treatment.java

```

12 public class Treatment {
13
14     private String treatmentNum;
15     private String date;
16     private String patientID;
17     private String patientName;
18     private String treatment;
19     private String medicines;
20
21     public Treatment(String treatmentNum, String date, String patientID, String patientName, String treatment, String medicines)
22     {
23         this.treatmentNum = treatmentNum;
24         this.date = date;
25         this.patientID = patientID;
26         this.patientName = patientName;
27         this.treatment = treatment;
28         this.medicines = medicines;
29     }
30
31     public String getTreatmentNum()
32     {
33         return treatmentNum;
34     }
35
36     public void setTreatmentNum(String treatmentNum)
37     {
38         this.treatmentNum = treatmentNum;
39     }

```

```

41 public String getDate()
42 {
43     return date;
44 }
45
46 public void setDate(String date)
47 {
48     this.date = date;
49 }
50
51 public String getPatientID()
52 {
53     return patientID;
54 }
55
56 public void setPatientID(String patientID)
57 {
58     this.patientID = patientID;
59 }
60
61 public String getPatientName()
62 {
63     return patientName;
64 }
65
66 public void setPatientName(String patientName)
67 {
68     this.patientName = patientName;
69 }
70
71 public String getTreatment()
72 {
73     return treatment;
74 }
75
76 public void setTreatment(String treatment)
77 {
78     this.treatment = treatment;
79 }
80
81 public String getMedicines()
82 {
83     return medicines;
84 }
85
86 public void setMedicines(String medicines)
87 {
88     this.medicines = medicines;
89 }
90 }

```

### c) Registration.java

```

12 public class Registration {
13
14     private String patientID;
15     private String patientName;
16     private String icNumber;
17     private String gender;
18     private String date;
19     private String address;
20     private String mobilePhone;
21     private String emailAddress;
22
23     public Registration(String patientID, String patientName, String icNumber, String gender, String date, String address, String mobilePhone, String email)
24     {
25         this.patientID = patientID;
26         this.patientName = patientName;
27         this.icNumber = icNumber;
28         this.gender = gender;
29         this.date = date;
30         this.address = address;
31         this.mobilePhone = mobilePhone;
32         this.emailAddress = emailAddress;
33     }
34
35     public String getPatientID()
36     {
37         return patientID;
38     }
39 }

```

```

40 public void setPatientID(String patientID)
41 {
42     this.patientID = patientID;
43 }
44
45 public String getPatientName()
46 {
47     return patientName;
48 }
49
50 public void setPatientName(String patientName)
51 {
52     this.patientName = patientName;
53 }
54
55 public String getICNumber()
56 {
57     return icNumber;
58 }
59
60 public void setICNumber(String icNumber)
61 {
62     this.icNumber = icNumber;
63 }
64
65 public String getGender()
66 {
67     return gender;
68 }
69
70 public void setGender(String gender)
71 {
72     this.gender = gender;
73 }
74
75 public String getDate()
76 {
77     return date;
78 }
79
80 public void setDate(String date)
81 {
82     this.date = date;
83 }
84
85 public String getAddress()
86 {
87     return address;
88 }
89
90 public void setAddress(String address)
91 {
92     this.address = address;
93 }
94
95 public String getMobilePhone()
96 {
97     return mobilePhone;
98 }
99
100 public void setMobilePhone(String mobilePhone)
101 {
102     this.mobilePhone = mobilePhone;
103 }
104
105 public String getEmailAddress()
106 {
107     return emailAddress;
108 }
109
110 public void setEmailAddress(String emailAddress)
111 {
112     this.emailAddress = emailAddress;
113 }
114 }

```

#### d) BookingAppointment.java

```
Start Page x BookingAppointment.java x
Source History

12 public class BookingAppointment {
13
14     private String bookingNum;
15     private String patientID;
16     private String patientName;
17     private String date;
18     private String time;
19     private String dentistID;
20     private String dentistName;
21
22     public BookingAppointment(String bookingNum, String patientID, String patientName, String date, String time, String dentistID, String dentistName) {
23         this.bookingNum = bookingNum;
24         this.patientID = patientID;
25         this.patientName = patientName;
26         this.date = date;
27         this.time = time;
28         this.dentistID = dentistID;
29         this.dentistName = dentistName;
30     }
31
32     public String getBookingNum()
33     {
34         return bookingNum;
35     }
36
37     public void setBookingNum(String bookingNum)
38     {
39         this.bookingNum = bookingNum;
40     }
41
42     public String getPatientID()
43     {
44         return patientID;
45     }
46
47     public void setPatientID(String patientID)
48     {
49         this.patientID = patientID;
50     }
51
52     public String getPatientName()
53     {
54         return patientName;
55     }
56
57     public void setPatientName(String patientName)
58     {
59         this.patientName = patientName;
60     }
61
62     public String getDate()
63     {
64         return date;
65     }
66
67     public void setDate(String date)
68     {
69         this.date = date;
70     }
71
72     public String getTime()
73     {
74         return time;
75     }
76
77     public void setTime(String time)
78     {
79         this.time = time;
80     }
81
82     public String getDentistID()
83     {
84         return dentistID;
85     }
86
87     public void setDentistID(String dentistID)
88     {
89         this.dentistID = dentistID;
90     }
91
92     public String getDentistName()
93     {
94         return dentistName;
95     }
96
97     public void setDentistName(String dentistName)
98     {
99         this.dentistName = dentistName;
100    }
```

### e) Medicines.java

```
Start Page x Medicines.java x
Source History
12 public class Medicines {
13
14     private String medicineID;
15     private String medicineName;
16     private String expiryDate;
17     private String quantity;
18
19     public Medicines(String medicineID, String medicineName, String expiryDate, String quantity)
20     {
21         this.medicineID = medicineID;
22         this.medicineName = medicineName;
23         this.expiryDate = expiryDate;
24         this.quantity = quantity;
25     }
26
27     public String getMedicinesID()
28     {
29         return medicineID;
30     }
31
32     public void setMedicinesID(String medicineID)
33     {
34         this.medicineID = medicineID;
35     }
36
37     public String getExpiryDate()
38     {
39         return expiryDate;
40     }
41
42     public void setExpiryDate(String expiryDate)
43     {
44         this.expiryDate = expiryDate;
45     }
46
47     public String getMedicineName()
48     {
49         return medicineName;
50     }
51
52     public void setMedicineName(String medicineName)
53     {
54         this.medicineName = medicineName;
55     }
56
57     public String getQuantity()
58     {
59         return quantity;
60     }
61
62     public void setQuantity(String quantity)
63     {
64         this.quantity = quantity;
65     }
66 }
```

### 3. Login

```
Start Page x LoginPage.java x
Source Design History
216 private void jButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {
217
218     Connection conn = MySQLDatabase.ConnectDB();
219     String query = "SELECT * from staff_table WHERE position = ? AND username = ? AND password = ?";
220
221     try{
222         PreparedStatement ps = conn.prepareStatement(query);
223         ps.setString(1, (String)jComboBox1.getSelectedItem());
224         ps.setString(2, jtxtUsername.getText());
225         ps.setString(3, jPassword.getText());
226         ResultSet rs = ps.executeQuery();
227
228         if(rs.next())
229         {
230             if(jComboBox1.getSelectedIndex()==1)
231             {
232                 AdminPage adminPage = new AdminPage();
233                 adminPage.setVisible(true);
234                 dispose();
235             }
236             else if(jComboBox1.getSelectedIndex()==2)
237             {
238                 DentistPage dentistPage = new DentistPage();
239                 dentistPage.setVisible(true);
240                 dispose();
241             }
242         }
243     }
```

```

242         else if(jComboBox1.getSelectedIndex()==3)
243         {
244             NursePage nursePage = new NursePage();
245             nursePage.setVisible(true);
246             dispose();
247         }
248         else
249         {
250             JOptionPane.showMessageDialog(null, "Please choose your position");
251         }
252     }
253     else
254     {
255         JOptionPane.showMessageDialog(null, "Username and Password does not match", "Access Denied..", JOptionPane.ERROR_MESSAGE);
256     }
257 }
258 }catch(Exception e)
259 {
260     JOptionPane.showMessageDialog(null, e);
261 }
262 }

```

#### 4. Insert, Update, Delete.

```

Start Page  x  AdminPage.java  x
Source  Design  History
527 private void jBtnAddActionPerformed(java.awt.event.ActionEvent evt) {
528     Connection conn = MySQLDatabase.ConnectDB();
529     String query = "INSERT INTO staff_table(staffID, startingDate, icNumber, name, gender, position, username, password, address, phoneNumber, emailAdd:
530     String Password = new String(jPasswordField.getPassword());
531     String Confirm = new String(jRetypePasswordField.getPassword());
532
533     try{
534         PreparedStatement ps = conn.prepareStatement(query);
535         ps.setString(1, jTxtStaffID.getText());
536         ps.setString(2, jTxtDate.getText());
537         ps.setString(3, jTxtICNum.getText());
538         ps.setString(4, jTxtName.getText());
539
540         if(jRadBtnMale.isSelected())
541         {
542             Gender = "Male";
543         }
544         if(jRadBtnFemale.isSelected())
545         {
546             Gender = "Female";
547         }
548         ps.setString(5, Gender);
549
550         ps.setString(6, (String) jComboBox1.getSelectedItem());
551         ps.setString(7, jTxtUsername.getText());
552
553         ps.setString(8, Password);
554         ps.setString(8, Confirm);
555         ps.setString(9, jTxtAddress.getText());
556         ps.setString(10, jTxtPhone.getText());
557         ps.setString(11, jTxtEmail.getText());
558
559         if(Password.equals(Confirm))
560         {
561             ps.execute();
562             ps.close();
563
564             JOptionPane.showMessageDialog(null, "Your username and password already saved!");
565             JOptionPane.showMessageDialog(null, "Successfully adding new staff's data!");
566
567             AdminPage adminPage = new AdminPage();
568             adminPage.setVisible(true);
569             dispose();
570         }
571         else
572         {
573             JOptionPane.showMessageDialog(this, "Your Password does not match.");
574         }
575     }catch(Exception e)
576     {
577         JOptionPane.showMessageDialog(null, e);
578     }
579 }
580 }

```



```

Start Page x AdminPage.java x
Source Design History
private void jBtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    if(jTxtStaffID.getText() != null)
    {
        String UpdateQuery = null;
        PreparedStatement ps = null;
        Connection conn = MySQLDatabase.ConnectDB();
        String Password = new String(jPassword.getPassword());
        String Confirm = new String(jRetypePassword.getPassword());

        try {
            UpdateQuery = "UPDATE staff_table SET startingDate = ?, icNumber = ?, name = ?, gender = ?, position = ?, username = ?, password = ?, address = ?";
            ps = conn.prepareStatement(UpdateQuery);
            ps.setString(1, jTxtDate.getText());
            ps.setString(2, jTxtICNum.getText());
            ps.setString(3, jTxtName.getText());

            if(jRadBtnMale.isSelected())
            {
                Gender = "Male";
            }
            if(jRadBtnFemale.isSelected())
            {
                Gender = "Female";
            }
            ps.setString(4, Gender);

            ps.setString(5, (String) jComboBox1.getSelectedItem());
            ps.setString(6, jTxtUsername.getText());
            ps.setString(7, Password);
            ps.setString(8, Confirm);

            ps.setString(9, jTxtAddress.getText());
            ps.setString(10, jTxtPhone.getText());
            ps.setString(11, jTxtEmail.getText());

            ps.setString(12, jTxtStaffID.getText());

            ps.executeUpdate();

            JOptionPane.showMessageDialog(null, "Staff's data updated");

            AdminPage adminPage = new AdminPage();
            adminPage.setVisible(true);
            dispose();
        } catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

```

```

private void jBtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    Connection conn = MySQLDatabase.ConnectDB();
    String query = "DELETE FROM newstaffs_table WHERE staffID = ?";
    try{
        PreparedStatement ps = conn.prepareStatement(query);
        ps.setString(1, jTxtStaffID.getText());
        ps.executeUpdate();

        JOptionPane.showMessageDialog(null, "Successfully deleted the data.");

        AdminPage adminPage = new AdminPage();
        adminPage.setVisible(true);
        dispose();
    }catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

## 5. Show Table

```
35 public ArrayList<NewStaff> newStaffList()
36 {
37     ArrayList<NewStaff> staffList = new ArrayList<>();
38     Connection conn = MySQLDatabase.ConnectDB();
39     String query = "SELECT * FROM staff_table";
40     try {
41         PreparedStatement ps = conn.prepareStatement(query);
42         ResultSet rs = ps.executeQuery(query);
43
44         NewStaff newStaff;
45         while(rs.next())
46         {
47             newStaff = new NewStaff(rs.getString("staffID"), rs.getString("startingDate"), rs.getString("icNumber"), rs.getString("name"), rs.getString("gender"));
48             staffList.add(newStaff);
49         }
50     } catch (Exception e)
51     {
52         JOptionPane.showMessageDialog(null, e);
53     }
54     return staffList;
55 }
```

```
57 public void show_NewStaff()
58 {
59     ArrayList<NewStaff> list = newStaffList();
60     DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
61     Object[] row = new Object[11];
62     for(int i=0; i<list.size(); i++)
63     {
64         row[0] = list.get(i).getStaffID();
65         row[1] = list.get(i).getStartingDate();
66         row[2] = list.get(i).getIcNumber();
67         row[3] = list.get(i).getName();
68         row[4] = list.get(i).getGender();
69         row[5] = list.get(i).getPosition();
70         row[6] = list.get(i).getUsername();
71         row[7] = list.get(i).getPassword();
72         row[8] = list.get(i).getAddress();
73         row[9] = list.get(i).getPhoneNumber();
74         row[10] = list.get(i).getEmailAddress();
75         model.addRow(row);
76     }
77 }
```

## 6. Searching

```
461 private void updateCombo() {
462     Connection conn = MySQLDatabase.ConnectDB();
463     String query = "SELECT * FROM patient_table";
464     try{
465         PreparedStatement ps = conn.prepareStatement(query);
466         ResultSet rs = ps.executeQuery();
467
468         while(rs.next())
469         {
470             JComboBoxPatID.addItem(rs.getString("patientID"));
471         }
472     } catch (Exception e)
473     {
474         JOptionPane.showMessageDialog(null, e);
475     }
476
477     String query2 = "SELECT * FROM staff_table";
478     try{
479         PreparedStatement ps = conn.prepareStatement(query2);
480         ResultSet rs = ps.executeQuery();
481
482         while(rs.next())
483         {
484             JComboBoxDenID.addItem(rs.getString("staffID"));
485         }
486     } catch (Exception e)
487     {
488         JOptionPane.showMessageDialog(null, e);
489     }
490 }
```

## 7. Mouse Clicked

```
444 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
445     int i = jTable1.getSelectedRow();  
446     TableModel model = jTable1.getModel();  
447  
448     txtBookingNum.setText(model.getValueAt(i,0).toString());  
449     cmbBoxPatID.setSelectedItem(model.getValueAt(i,1).toString());  
450     cmbBoxPatName.setSelectedItem(model.getValueAt(i,2).toString());  
451     txtDate.setText(model.getValueAt(i,3).toString());  
452     txtTime.setText(model.getValueAt(i,4).toString());  
453     cmbBoxDenID.setSelectedItem(model.getValueAt(i,5).toString());  
454     cmbBoxDenName.setSelectedItem(model.getValueAt(i,6).toString());  
455 }
```

## 8. Bar Chart

```
331 private void jButtonBarChartActionPerformed(java.awt.event.ActionEvent evt) {  
332     try{  
333         String query = "SELECT medicineID, quantity FROM medicine_table";  
334         JDBCCategoryDataset dataset = new JDBCCategoryDataset(MySqlDatabase.ConnectDB(), query);  
335         JFreeChart chart = ChartFactory.createBarChart("Bar Chart", "medicineID", "quantity", (CategoryDataset) dataset, PlotOrientation.VERTICAL, false,  
336             true, false, true);  
337         BarRenderer renderer = null;  
338         CategoryPlot plot = null;  
339         renderer = new BarRenderer();  
340  
341         ChartFrame frame = new ChartFrame("Bar Chart", chart);  
342         frame.setVisible(true);  
343         frame.setSize(840,840);  
344     }catch (Exception e)  
345     {  
346         JOptionPane.showMessageDialog(null, e);  
347     }  
348 }
```

## 9. .pdf File

```
249 private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {  
250     try{  
251         Document doc = new Document();  
252         try{  
253             PdfWriter.getInstance(doc, new FileOutputStream("MC.pdf"));  
254         } catch (FileNotFoundException ex)  
255         {  
256             Logger.getLogger(McForm.class.getName()).log(Level.SEVERE, null, ex);  
257         }  
258         doc.open();  
259         doc.add(new Paragraph(jlblTitle.getText().toString()));  
260         doc.add(new Paragraph(jTextField1.getText().toString()));  
261         doc.add(new Paragraph(jComboBox1.getSelectedItem().toString()));  
262         doc.add(new Paragraph(jTextField3.getText().toString()));  
263         doc.add(new Paragraph(jTextField2.getText().toString()));  
264         doc.add(new Paragraph(jTextField4.getText().toString()));  
265         doc.add(new Paragraph(jTextField5.getText().toString()));  
266         doc.add(new Paragraph(jTextField6.getText().toString()));  
267         doc.add(new Paragraph(jTextField3.getText().toString()));  
268         doc.add(new Paragraph(jTextField2.getText().toString()));  
269         doc.add(new Paragraph(jTextField5.getText().toString()));  
270  
271         doc.close();  
272  
273         JOptionPane.showMessageDialog(null, "Your File Already Save!");  
274     }catch (DocumentException ex)  
275     {  
276         Logger.getLogger(McForm.class.getName()).log(Level.SEVERE, null, ex);  
277     }  
278 }
```

## 10. Calculation

```
285 private void jButtonCalcActionPerformed(java.awt.event.ActionEvent evt) {  
286     double a = Double.parseDouble(jTextFieldPrice.getText());  
287     double b = Double.parseDouble(jTextFieldService.getText());  
288  
289     double total = b+a;  
290     jTextFieldPayment.setText(""+total);  
291 }
```

## **CHAPTER 5**

### **Conclusion**

Dental Clinic Management System helping on making the system in clinic especially dental clinic more efficient. This project giving opportunity to create a useable system. Moreover, this project motivates to learn how to connect programming with database. I was hoping that this system giving good effect towards society especially to the users of this system. I also hope that this project can help producing new Malaysia that advanced in technologies.

## REFERENCES

1. Writing coding in java

<https://www.youtube.com/user/ProgrammingKnowledge>

<https://stackoverflow.com>