



UUM
Universiti Utara Malaysia

**UUM COLLEGE OF ARTS AND SCIENCES
SKIP1103 INTERMEDIATE PROGRAMMING
GROUP PROJECT: REPORT
TOPIC: DENTAL CLINIC APPOINTMENT MANAGEMENT
SYSTEM**

**PREPARED FOR:
PROF. MADYA DR. ROHAIDA BINTI ROMLI**

PREPARED BY:

NAME	MATRIC NO
FARIDATUL NAJIAH BINTI MOHD FUAD	289923
NUR AIN NAJWA BINTI MUHAMMAD ZAKWAN	293289

1. INTRODUCTION/PROJECT BACKGROUND

The topic we have chosen for this project is a dental clinic appointment management system. For context, it is a GUI on an appointment management system for a dental clinic which we named “Seri Maju Clinic”. In the GUI made and provided by us, the client, where in this case a clinic staff, can browse the GUI to book an appointment with all details given by the patient. The GUI is simple but detailed to manage an appointment booking. Clinic staff can go through booking an appointment smoothly and direct through the details needed. The GUI offers a straightforward method for staff to manage appointments, displaying patients' booking details after confirmation and allowing for easy appointment cancellations. The GUI reduces the reliance on manual booking systems, providing a more accessible way for staff to enter appointment details. The use of an array in the GUI streamlines the recording of patients' information, making appointment searching and cancellation much easier.

2. PROBLEM STATEMENT

The current manual appointment management system at the dental clinic faces several challenges, including potential errors when inserting details of patients. Mistakes such as choosing the wrong services, and filling in the incorrect patient information when booking an appointment happen several times. Another problem faced with manual booking is difficulty in displaying appointment details. Manual booking is a traditional way where it uses a book to book an appointment, where the possibility of wrong management might happen. Wrong time, date, or service management might disrupt in displaying appointment details. Furthermore, the information shown manually is sometimes unreadable. In certain cases, the ink is not printed well, resulting in bad-quality of print paper. This problem makes it more difficult for the staff to read and fill in the information needed, especially for those who have trouble seeing. Lastly, the problem commonly occurs in a booking appointment is small or limited space to insert patients' information. This is a common problem that occurs when booking an appointment, where the staff has trouble in filling their information, especially those who have long names, accidentally write in the wrong section, or have typographical errors.

There are some solutions that we came up with in order to reduce any problems and errors in manual booking that we improved and implied in the GUI. In the GUI, it has been improved where multiple service choices are available for patients. This way, it is easier for staff to choose from single or multiple types of services the patients want. Other than that, displaying details of the patient's personal information is in order. The GUI is capable of arranging the displayed appointment by using a method that can fix the type of management to use in displaying, whether it be in the order of the time, date, or even patients' order when booking. Furthermore, the error of filling in the patients' personal information is reduced. Hence, it improves the efficiency of booking. This is because the GUI is made to be a simpler, clearer, and more straightforward type of booking management. The GUI enhances the staff's experience when booking an appointment at any time. Staff also can be less worried about the spaces provided when filling in the patients' personal information. Unlimited spaces for filling in the patients' information reduces staff's worry about inserting the details. Automating the whole process of booking reduces data entry time and provides real-time information. The improvement the GUI has made in addressing these challenges overall features in the management system at the dental clinic.

3. OBJECTIVE OF THE PROJECT

The objective of this project is to reduce errors, centralize the appointment database, and streamline appointment management. The objective of reducing errors in the GUI project is to offer a clear and user-friendly input field for capturing patient information. In order to create more reliable and accurate records for patients' information, any errors are minimized in the patients and appointments frame provided by the GUI. Staff can navigate through the GUI more efficiently, select services with ease, and choose appointment dates and times without confusion, leading to a more streamlined appointment management. A centralized appointment database is made so that all of the appointment information is kept in a well-organized database. This improvised version of booking management is making it easier for data access, where managing and searching is effectively done. Streamlining appointment management refers to optimizing and simplifying the various steps in handling appointments. This means that the entire process of

booking, managing, and canceling appointments is smooth, efficient, and straightforward for the staff to handle.

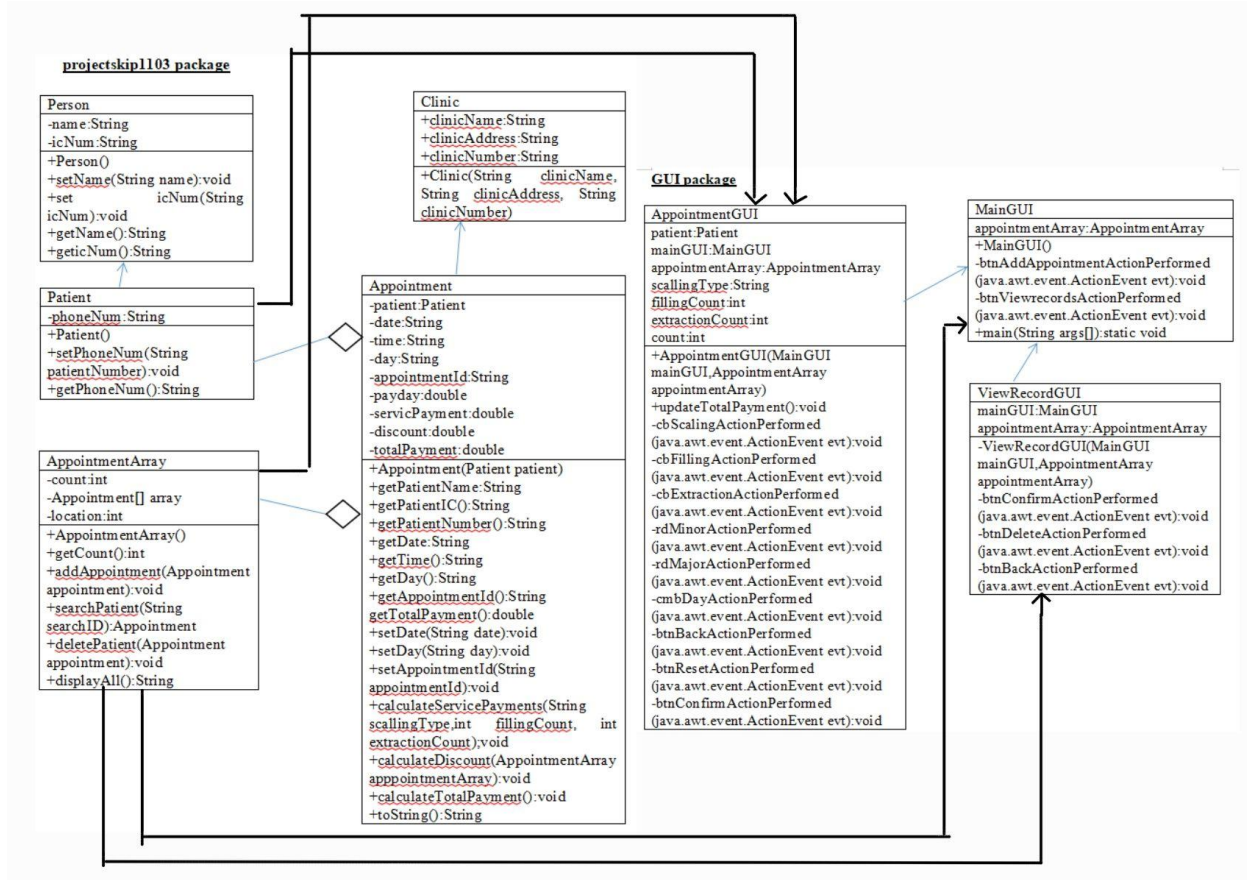
4. SCOPE OF THE PROJECT

The scope of our project has some key features in the GUI and some limitations that we encountered. There are some key features in the GUI that we take into consideration when designing the base ideas which are appointment booking, appointment cancellation, displaying booking list, and storing patients' information. The GUI provides a seamless and intuitive appointment booking process for patients. By inserting their personal details, desired services, preferred time, date, and day, the staff can easily assist patient schedule their appointments according to their convenience. A discount is also given to whom first time booking an appointment at the dental clinic. Other than that, patients also have the flexibility to cancel their appointment if necessary, as it is normal that some circumstances may change. The GUI allows eligible patients to cancel their appointments, ensuring a hassle-free cancellation process. The GUI displays a thorough list of upcoming appointments to help clinic staff manage appointments efficiently. The list's clearly organized format makes it simple for staff members to locate and manage appointments. It is crucial in ensuring the confidentiality and accessibility of patients' data. The GUI helps in storing patients' personal information, including name, Ic number, and contact details. By allowing quick retrieval and references of information needed at the time, this simplifies the administrative process.

Despite these valuable features, it is important to acknowledge certain limitations within the project scope which are online payment systems. The current development of the GUI does not include an online payment method, which means that patients are required to make payments using alternative methods. The current GUI only focuses on handling the appointment booking system thus payment methods are not processed through the GUI.

We aim to address any potential limitations and further refine the GUI to enhance its capabilities as our project develops further. We strive to offer a comprehensive appointment management solution for Seri Maju Clinic by continuously iterating and improving the GUI.

5. THE UML CLASS DIAGRAM



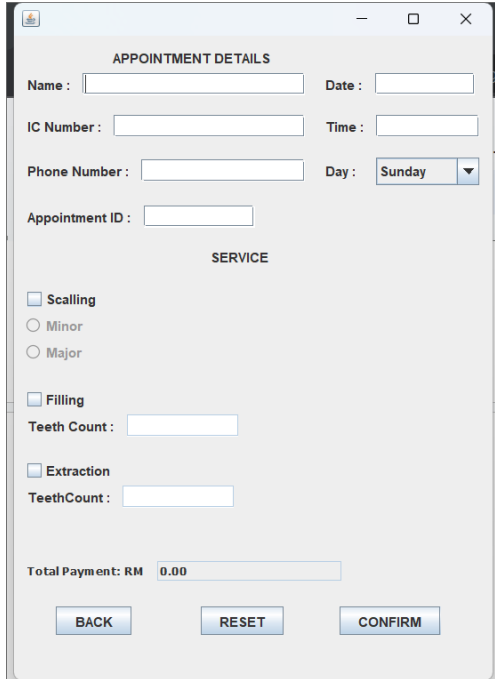
6. USER INTERFACE SCREENSHOTS (INCLUDING WITH USER GUIDELINES)

1. Homepage of our interface' frame



There are two selections. The user will click the Add Appointment button or View Records button.

2. Appointment Details (Add appointment button clicked)



The screenshot shows a software window titled "APPOINTMENT DETAILS". It contains several input fields for patient information: "Name", "Date", "IC Number", "Time", "Phone Number", "Day" (a dropdown menu currently showing "Sunday"), and "Appointment ID". Below these is a section titled "SERVICE" with three radio button options: "Scaling", "Minor", and "Major". Under "Scaling", there is a "Teeth Count" input field. Under "Minor", there is an "Extraction" checkbox and a "TeethCount" input field. At the bottom, there is a "Total Payment: RM" label followed by a text box containing "0.00". Three buttons are at the very bottom: "BACK", "RESET", and "CONFIRM".

After the user clicks Add Appointment button, this frame will be displayed. The user needs to input all the patient's details including the services. The total payment for the service(s) chosen will be shown after patient choose service(s) needed. The reset button is for resetting all to be empty, this will be easy for the user to input the details again. When patient press the confirm button, a frame will shown to information that the patient's appointment has been saved.

3. Display appointment details and search to delete appointment (view records clicked)

The screenshot shows a GUI window with two main sections: 'PATIENTS RECORD' and 'SPECIFIC PATIENT'. The 'PATIENTS RECORD' section on the left contains a list of appointment details for a patient named 'fari'. The 'SPECIFIC PATIENT' section on the right has a 'Search ID:' field, a 'CONFIRM' button, a large empty box for details, a 'DELETE' button, a text area for confirmation, and a 'BACK' button.

PATIENTS RECORD

-----Appointment Details-----
Appointment ID: 288923
Patient Name: fari
Patient IC: 021008040406
Date: 12.2.23
Time: 12.33
Day: Wednesday
Total Payment: RM75.0

SPECIFIC PATIENT

Search ID:

CONFIRM

DELETE

BACK

When the view records button is clicked, displayed details of patients will be shown. If patient desire to delete their apointments from the list, they can insert their ID in the specific patient section to display their details to confirm the right patient details. Then if they click the delete button, a text will be shown to confirm that the appointment has been deleted. The back button will bring back to the main page of the GUI.

The screenshot shows the same GUI window as before, but with the 'Search ID' field filled with '289923'. The 'CONFIRM' button is highlighted. The 'SPECIFIC PATIENT' section now displays details for a patient named 'fari' with appointment ID 289923. The 'DELETE' button is highlighted, and the text area below it shows 'Patient appointment has been deleted'. The 'BACK' button is also visible.

PATIENTS RECORD

-----Appointment Details-----
Appointment ID: 288923
Patient Name: fari
Patient IC: 021008040406
Date: 12.2.23
Time: 12.33
Day: Wednesday
Total Payment: RM75.0

-----Appointment Details-----
Appointment ID: 247999
Patient Name: aina
Patient IC: 021023715054
Date: 12.4.23
Time: 1.07
Day: Friday
Total Payment: RM70.0

SPECIFIC PATIENT

Search ID: 289923

CONFIRM

DELETE

Patient appointment has been deleted

BACK

7. CONCLUSION

In conclusion, the provided codes implement a simple appointment management system for a dental clinic using Java Swing for the Graphical User Interface (GUI). The key features in the code are displaying patients' details, filling in the details of the appointment, cancellation of appointment, and a reset button to assist staff in completing an appointment booking. The GUI offers a user-friendly interface for managing dental appointments. The key features provide a form where staff can insert patients' information and select appointment details needed. After that, the details are displayed, making it easier for both patients and staff to look up any details if needed. The cancellation of the appointment allows staff to enter the patient's name and remove it from the appointment list. The reset button function helps staff to clear all input fields in the appointment booking form, making it convenient for them to enter new appointment details.

The development of the dental appointment management GUI, named “Seri Maju Clinic” marks major advancements in scheduling and managing appointments. Throughout the project, we successfully manage to overcome various challenges faced by manual booking systems and implemented the GUI that gives a range of benefits.

The GUI offers several benefits over the traditional way of the booking system, where it significantly enhances the appointment management process for both clinic staff and patients. As explained before in the problem statement and objective of the project, the benefits such as error reduction, centralized appointment database, and improved appointment display are the key features that help improve the use of this GUI.

The primary goal of this project was to overcome the challenges faced by the manual booking appointment system. Addressing the identified situation helps us to achieve the minimization of errors, enhance readability, and streamlined booking management. This project also brings a positive impact on the clinic's efficiency and productivity. Staff can perform their tasks more effectively with the new user-friendly interface and organized appointment tracking. This led to optimized patient scheduling and reduced administrative burden. This improved efficiency brings better patient care and both patient and staff satisfaction.

We would like to express our gratitude to our lecturer, Prof. Madya Dr. Rohaida Binti Romli, and friends who contributed to the successful completion of this project. Their dedication

to teaching us throughout the semester, and guiding us in making this project helps us to shape the success of this project. We also want to extend our gratitude to our friends who help us in completing the project when we encounter problems along the way. Without the help of our lecturer and friends, this project would have not been possible.

8. APPENDIX

WORKLOAD

Coding

Group Member	Workload
Faridatul Najihah binti Mohd Fuad	GUI
Nur Ain Najwa binti Muhammad Zakwan	Coding

Report

Group Member	Workload
Faridatul Najihah binti Mohd Fuad	Introduction, problem statement, objective of the project, appendix, and conclusion
Nur Ain Najwa binti Muhammad Zakwan	Scope of the project, conclusion, UML class diagram, user interface screenshots, and source code

Slide and Presentation

Group Member	Workload
Faridatul Najihah binti Mohd Fuad	Introduction, problem statement, objective of the project, and conclusion
Nur Ain Najwa binti Muhammad Zakwan	Scope of the project, UML class diagram, and conclusion

MEETING LOGS

Tuesday, 18 July 2023, 2.00 p.m. - 5.00 p.m., Student Lounge Soc

Attendees:

1. Faridatul Najihah binti Mohd Fuad
2. Nur Ain Najwa binti Muhammad Zakwan

Discussed:

1. Group members will meet in any free time in front of the Student Lounge at Soc.
2. Discussed and sketched the UML diagram of the project.
3. Discussed the method to be used in the GUI.

Work division(coding and GUI):

1. Faridatul Najihah
 - a. making the GUI frame
2. Nur Ain Najwa
 - a. coding of the project

Wednesday, 19 July 2023, 11.00 a.m. - 3.30 p.m., Student Lounge Soc

Attendees:

1. Faridatul Najihah binti Mohd Fuad
2. Nur Ain Najwa binti Muhammad Zakwan

Discussed:

1. Continue discussion of the coding due to some errors.
2. Continue making the GUI frame and adding some coding to the source code.
3. Discussed the points to be put in the slide.

Work division (coding and slide):

1. Faridatul Najihah
 - a. GUI - frame and source code
 - b. Slide - introduction, problem statement, and objective of the project
2. Nur Ain Najwa
 - a. Code

b. Slide - scope of the project, UML class diagram, and conclusion

Thursday, 20 July 2023, 2.00 p.m. - 4.00 p.m., Student Lounge Soc

Attendees:

1. Faridatul Najihah binti Mohd Fuad
2. Nur Ain Najwa binti Muhammad Zakwan

Discussed:

1. Continue finishing and updating codes.
2. Continue finishing and updating GUI.
3. Continue finishing the slide.
4. Discussed workload for the report.

Workload(code, GUI, slide, and report):

1. Faridatul Najihah
 - a. GUI - frame and source code
 - b. Slide - introduction, problem statement, objective of the project, and conclusion
 - c. Report - introduction, problem statement, objective of the project, appendix, and conclusion
2. Nur Ain Najwa
 - a. Code
 - b. Slide - scope of the project, UML class diagram, and conclusion
 - c. Report - scope of the project, conclusion, UML class diagram, user interface screenshots, and source code

Monday, 24 July 2023, 9.30 a.m. - 12.00 p.m., Student Lounge Soc

Attendees:

1. Faridatul Najihah binti Mohd Fuad
2. Nur Ain Najwa binti Muhammad Zakwan

Discussed:

1. Continue finishing the coding.

2. Finishing and updating some parts in slides and report.

Workload(code, slide, and report):

1. Faridatul Najihah
 - a. Slide - introduction, problem statement, objective of the project, and conclusion
 - b. Report - introduction, problem statement, objective of the project, appendix, and conclusion
2. Nur Ain Najwa
 - a. Code
 - b. Slide - scope of the project, UML class diagram, and conclusion
 - c. Report - scope of the project, conclusion, UML class diagram, user interface screenshots, and source code

Wednesday, 26 July 2023, 9.30 a.m. - 2.15 p.m., Student Lounge Soc

Attendees:

1. Faridatul Najihah binti Mohd Fuad
2. Nur Ain Najwa binti Muhammad Zakwan

Discussed:

1. Finishing the coding and testing runtime.
2. Finishing some parts in slides and report.

Workload(code, slide, and report):

1. Faridatul Najihah
 - a. Code and GUI
 - b. Slide - introduction, problem statement, objective of the project, and conclusion
 - c. Report - introduction, problem statement, objective of the project, appendix, and conclusion
2. Nur Ain Najwa
 - a. Code and GUI
 - b. Slide - scope of the project, UML class diagram, and conclusion
 - c. Report - scope of the project, conclusion, UML class diagram, user interface

9. SOURCE CODE

Person class

```
package projectskip1103;

public class Person {
    //This class is made up of data fields, first and second names, and relevant methods.

    private String name;
    private String icNum;

    public Person(){
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setIcNum(String icNum) {
        this.icNum = icNum;
    }

    public String getName() {
        return name;
    }

    public String getIcNum() {
        return icNum;
    }
}
```

Patient class

```
package projectskip1103;

public class Patient extends Person {

    private String phoneNum;

    public Patient(){
    }

    public void setPhoneNum(String patientNumber) {
```

```

        this.phoneNum = patientNumber;
    }

    public String getPhoneNum() {
        return phoneNum;
    }
}

```

Clinic class

```
package projectskip1103;
```

```

public class Clinic {
    public String clinicName,clinicAddress,clinicNumber;

    public Clinic(String clinicName, String clinicAddress, String clinicNumber){
        this.clinicName = clinicName;
        this.clinicAddress = clinicAddress;
        this.clinicNumber = clinicNumber;
    }
}

```

Appointment array

```
package projectskip1103;
```

```

public class AppointmentArray {

    private int count = 0;
    private Appointment[] array;
    private int location = 0;

    public AppointmentArray() {
        array = new Appointment[3];
    }

    public int getCount() {
        return count;
    }

    public Appointment[] getArray() {
        return array;
    }

    public void addAppointment(Appointment appointment) {
        array[count] = appointment;
    }
}

```

```

        count++;
    }

    public Appointment searchPatient(String searchID) {
        for (int i = 0; i < array.length; i++) {
            if (array[i] != null && searchID.equals(array[i].getAppointmentId())) {
                System.out.println("Appointment has been found.");
                return array[i];
            }
        }
        return null;
    }

    public void deletePatient(Appointment appointment) {
        for(int i=0; i<array.length; i++){
            if(array[i]==appointment){
                array[i] = null;
                System.out.println("Appointment has been deleted from records.");
            }
        }
    }

    public String displayAll() {
        String display = "";

        for (int i = 0; i < array.length; i++) {
            if (array[i] != null) {
                display += array[i].toString(); //first dia display array yg appointment[0], then update i
                display += "\n\n";
            }
        }
        return display;
    }
}

```

Appointment

```
package projectskip1103;
```

```

public class Appointment {
    private Patient patient;
    private String date, time, day, appointmentId;
    private double payDay = 0.0, servicePayment = 0.0, discount = 0.0, totalPayment = 0.0;

    public Appointment(Patient patient){
        this.patient = patient;
    }
}

```

```
}
```

```
public String getPatientName(){  
    return patient.getName();  
}
```

```
}
```

```
public String getPatientIC(){  
    return patient.getIcNum();  
}
```

```
}
```

```
public String getPatientNumber(){  
    return patient.getPhoneNum();  
}
```

```
}
```

```
public String getDate() {  
    return date;  
}
```

```
}
```

```
public String getTime() {  
    return time;  
}
```

```
}
```

```
public String getDay() {  
    return day;  
}
```

```
}
```

```
public String getAppointmentId() {  
    return appointmentId;  
}
```

```
}
```

```
public double getTotalPayment() {  
    return totalPayment;  
}
```

```
}
```

```
public void setDate(String date) {  
    this.date = date;  
}
```

```
}
```

```
public void setTime(String time) {  
    this.time = time;  
}
```

```
}
```

```
public void setDay(String day) {  
    this.day = day;  
}
```

```
}
```



```

public void setAppointmentId(String appointmentId) {
    this.appointmentId = appointmentId;
}

public void calculateServicePayments(String scallingType,int fillingCount, int extractionCount) {
    servicePayment=0.0;
    if(scallingType!=null){
        if (scallingType.equalsIgnoreCase("Minor")) {
            servicePayment+=40;
        }
        else if (scallingType.equalsIgnoreCase("Major")) {
            servicePayment+=50;
        }
    }
    servicePayment+=(fillingCount*5) + (extractionCount*10);
}

public void calculateDiscount(AppointmentArray appointmentArray) {
    boolean regularPatient = false;

    for (int i = 0; i < appointmentArray.getCount(); i++) {
        if (patient.getName().equalsIgnoreCase(appointmentArray.getArray()[i].getPatientName())) {
            regularPatient = true;
            discount = 0.1;
            break;
        }
        else
            discount = 0.0;
    }
}

public void calculateTotalPayment() {
    if (day.equalsIgnoreCase("Saturday") || day.equalsIgnoreCase("Sunday")) {
        payDay = 20.0;
    }
    else {
        payDay = 10.0;
    }

    totalPayment = (payDay + servicePayment)*(1-discount);
}

public String toString() {

```

```

        return "----Appointment Details----\nAppointment ID:" + appointmentId + "\n Patient Name:" +
patient.getName() + "\nPatient IC:" +patient.getIcNum()+ "\nDate:" + date +
        "\nTime" + time + "\nDay:" + day + "\nTotal Payment:RM"+totalPayment;
    }
}

```

ViewRecordGUI class

```
package GUI;
```

```
import projectskip1103.Appointment;
```

```
import projectskip1103.AppointmentArray;
```

```
public class ViewRecordGUI extends javax.swing.JFrame {
```

```
    AppointmentArray appointmentArray;
```

```
    MainGUI mainGui;
```

```
    public ViewRecordGUI(MainGUI mainGui,AppointmentArray appointmentArray) {
```

```
        initComponents();
```

```
        this.mainGui = mainGui;
```

```
        this.appointmentArray = appointmentArray;
```

```
        txtPatientRecord.setText(appointmentArray.displayAll());
```

```
    }
```

```
@SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```
private void initComponents() {
```

```
    jPanel1 = new javax.swing.JPanel();
```

```
    jScrollPane1 = new javax.swing.JScrollPane();
```

```
    txtPatientRecord = new javax.swing.JTextArea();
```

```
    lblPatientsRecord = new javax.swing.JLabel();
```

```
    jPanel2 = new javax.swing.JPanel();
```

```
    jLabel2 = new javax.swing.JLabel();
```

```
    txtSearchId = new javax.swing.JTextField();
```

```
    jScrollPane2 = new javax.swing.JScrollPane();
```

```
    txtSpecificPatient = new javax.swing.JTextArea();
```

```
    jLabel1 = new javax.swing.JLabel();
```

```
    btnBack = new javax.swing.JButton();
```

```
    btnDelete = new javax.swing.JButton();
```

```
    btnConfirm = new javax.swing.JButton();
```

```
    jScrollPane3 = new javax.swing.JScrollPane();
```

```
    txtOutput = new javax.swing.JTextArea();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

txtPatientRecord.setColumns(20);
txtPatientRecord.setRows(5);
jScrollPane1.setViewportView(txtPatientRecord);

lblPatientsRecord.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
lblPatientsRecord.setText("PATIENTS RECORD");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jScrollPane1)
                .add(lblPatientsRecord)
            )
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .add(jScrollPane1)
        .add(lblPatientsRecord)
);

jLabel2.setText("Search ID:");

txtSpecificPatient.setColumns(20);
txtSpecificPatient.setRows(5);
jScrollPane2.setViewportView(txtSpecificPatient);

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel1.setText("SPECIFIC PATIENT");
```

```
btnBack.setText("BACK");
btnBack.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBackActionPerformed(evt);
    }
});
```

```
btnDelete.setText("DELETE");
btnDelete.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnDeleteActionPerformed(evt);
    }
});
```

```
btnConfirm.setText("CONFIRM");
btnConfirm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnConfirmActionPerformed(evt);
    }
});
```

```
txtOutput.setColumns(20);
txtOutput.setRows(5);
jScrollPane3.setViewportViewView(txtOutput);
```

```
jPanel2Layout = new javax.swing.GroupLayout(jPanel2);  
jPanel2.setLayout(jPanel2Layout);  
jPanel2Layout.setHorizontalGroup(  
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel2Layout.createSequentialGroup()  
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 87,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jPanel2Layout.createSequentialGroup())
    .addGap(116, 116, 116)
    .addComponent(btnBack))
    .addGroup(jPanel2Layout.createSequentialGroup())
    .addGap(85, 85, 85)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel1))
```

```

        .addGroup(jPanel2Layout.createSequentialGroup())
            .addComponent(txtSearchId, javax.swing.GroupLayout.PREFERRED_SIZE, 136,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(btnConfirm))))
        .addGroup(jPanel2Layout.createSequentialGroup())
            .addGap(122, 122, 122)
            .addComponent(btnDelete))
        .addGroup(jPanel2Layout.createSequentialGroup())
            .addGap(46, 46, 46)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 243,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 250,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(26, Short.MAX_VALUE))
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup())
            .addGap(26, 26, 26)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2)
            .addComponent(txtSearchId, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnConfirm))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 265,
Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(btnDelete)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(btnBack)
            .addGap(4, 4, 4))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```

```

getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

pack();
} // </editor-fold>

```

```

private void btnConfirmActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String searchId = txtSearchId.getText();
    Appointment appointment = appointmentArray.searchPatient(searchId);
    if (appointment != null) {
        txtSpecificPatient.setText(appointment.toString());
    }
    else {
        txtSpecificPatient.setText("Patient not found");
    }
}

```

```

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String searchId = txtSearchId.getText();
    Appointment appointment = appointmentArray.searchPatient(searchId);
    appointmentArray.deletePatient(appointment);
    txtOutput.setText("Patient appointment has been deleted");
    txtPatientRecord.setText(appointmentArray.displayAll());
}

```

```

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        // TODO add your handling code here:
        mainGui.setVisible(true);
        this.setVisible(false);

    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnBack;
    private javax.swing.JButton btnConfirm;
    private javax.swing.JButton btnDelete;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JLabel lblPatientsRecord;
    private javax.swing.JTextArea txtOutput;
    private javax.swing.JTextArea txtPatientRecord;
    private javax.swing.JTextField txtSearchId;
    private javax.swing.JTextArea txtSpecificPatient;
    // End of variables declaration
}

MainGUI class
package GUI;

import projectskip1103.AppointmentArray;
public class MainGUI extends javax.swing.JFrame {
    AppointmentArray appointmentArray;

    public MainGUI() {
        initComponents();
        appointmentArray = new AppointmentArray();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        lblCClinicName = new javax.swing.JLabel();
        lblClinicAddress = new javax.swing.JLabel();
        btnAddAppointment = new javax.swing.JButton();
        btnViewRecords = new javax.swing.JButton();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

lblCClinicName.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
lblCClinicName.setText("SERI MAJU DENTAL CLINIC");

lblClinicAddress.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
lblClinicAddress.setText("NO 171, TAMAN LIMAN 33020 KUALA KANGSAR, PERAK");

btnAddAppointment.setText("ADD APPOINTMENT");
btnAddAppointment.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddAppointmentActionPerformed(evt);
    }
});

btnViewRecords.setText("VIEW RECORDS");
btnViewRecords.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnViewRecordsActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(layout.createSequentialGroup()
                            .addGap(66, 66, 66)
                            .addComponent(btnAddAppointment, javax.swing.GroupLayout.PREFERRED_SIZE,
127, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(62, 62, 62)
                            .addComponent(btnViewRecords))
                        .add(layout.createSequentialGroup()
                            .addGap(120, 120, 120)
                            .addComponent(lblCClinicName))
                        .add(layout.createSequentialGroup()
                            .addGap(66, 66, 66)
                            .addComponent(lblClinicAddress)))
                    .addGap(83, 83, 83))
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(layout.createSequentialGroup()
                    .addGap(66, 66, 66)
                    .addComponent(btnAddAppointment, javax.swing.GroupLayout.PREFERRED_SIZE,
127, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(62, 62, 62)
                    .addComponent(btnViewRecords))
                .add(layout.createSequentialGroup()
                    .addGap(120, 120, 120)
                    .addComponent(lblCClinicName))
                .add(layout.createSequentialGroup()
                    .addGap(66, 66, 66)
                    .addComponent(lblClinicAddress)))
            .addGap(83, 83, 83))
            .addContainerGap())
);
layout.setVerticalGroup(

```



```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(72, 72, 72)
            .addComponent(lblCClinicName)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(lblClinicAddress, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(53, 53, 53)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(btnAddAppointment)
                .addComponent(btnViewRecords))
            .addGap(102, 102, 102))
        );

    pack();
} // </editor-fold>

```

```

private void btnAddAppointmentActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
    AppointmentGUI appointmentFrame = new AppointmentGUI(this,appointmentArray);
    //appointmentFrame.setExtendedState(JFrame.MAXIMIZED_BOTH);
    appointmentFrame.setVisible(true);
}

```

```

private void btnViewRecordsActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setVisible(false);
    ViewRecordGUI viewRecordFrame = new ViewRecordGUI(this,appointmentArray);
    //viewRecordFrame.setExtendedState(JFrame.MAXIMIZED_BOTH);
    viewRecordFrame.setVisible(true);
}

```

```

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            MainGUI mainFrame = new MainGUI();
            //mainFrame.setExtendedState(JFrame.MAXIMIZED_BOTH);
            mainFrame.setVisible(true);
        }
    });
}

```

// Variables declaration - do not modify

```

private javax.swing.JButton btnAddAppointment;
private javax.swing.JButton btnViewRecords;
private javax.swing.JLabel lblCClinicName;
private javax.swing.JLabel lblClinicAddress;
// End of variables declaration
}
AppointmentGUI class
package GUI;

import javax.swing.JOptionPane;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import projectskip1103.Appointment;
import projectskip1103.AppointmentArray;
import projectskip1103.Patient;

public class AppointmentGUI extends javax.swing.JFrame {
    Patient patient;
    Appointment appointment;
    MainGUI mainGui;
    AppointmentArray appointmentArray;
    String scallingType;
    int fillingCount, extractionCount;
    int count=0;

    public AppointmentGUI(MainGUI mainGui, AppointmentArray appointmentArray) {
        initComponents();
        this.patient = new Patient();
        this.appointment = new Appointment(patient);
        this.mainGui = mainGui;
        this.appointmentArray = appointmentArray;
        rdMinor.setEnabled(false);
        rdMajor.setEnabled(false);
        txtFillingCount.setEnabled(false);
        txtExtractionCount.setEnabled(false);
    }

    public void updateTotalPayment(){
        appointment.calculateServicePayments(scallingType, fillingCount, extractionCount);
        appointment.calculateDiscount(appointmentArray);
        appointment.calculateTotalPayment();
        txtPayment.setText(Double.toString(appointment.getTotalPayment()));
    }
}

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    buttonGroup1 = new javax.swing.ButtonGroup();
    jPanel1 = new javax.swing.JPanel();
    lblAppointmentDetails = new javax.swing.JLabel();
    txtName = new javax.swing.JTextField();
    lblName = new javax.swing.JLabel();
    lblIcNum = new javax.swing.JLabel();
    txtIcNum = new javax.swing.JTextField();
    lblPhoneNum = new javax.swing.JLabel();
    txtPhoneNum = new javax.swing.JTextField();
    txtAppointmentId = new javax.swing.JTextField();
    lblAppointmentId = new javax.swing.JLabel();
    lblDate = new javax.swing.JLabel();
    txtDate = new javax.swing.JTextField();
    lblTime = new javax.swing.JLabel();
    txtTime = new javax.swing.JTextField();
    cmbDay = new javax.swing.JComboBox();
    lblDay = new javax.swing.JLabel();
    jPanel2 = new javax.swing.JPanel();
    lblService = new javax.swing.JLabel();
    cbScalling = new javax.swing.JCheckBox();
    cbFilling = new javax.swing.JCheckBox();
    cbExtraction = new javax.swing.JCheckBox();
    rdMinor = new javax.swing.JRadioButton();
    rdMajor = new javax.swing.JRadioButton();
    lblFillingCount = new javax.swing.JLabel();
    lblExtractionCount = new javax.swing.JLabel();
    txtExtractionCount = new javax.swing.JTextField();
    txtFillingCount = new javax.swing.JTextField();
    lblPayment = new javax.swing.JLabel();
    txtPayment = new javax.swing.JTextField();
    btnConfirm = new javax.swing.JButton();
    btnReset = new javax.swing.JButton();
    btnBack = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    lblAppointmentDetails.setText("APPOINTMENT DETAILS");

    lblName.setText("Name :");

```

[illegible]

```

        .addComponent(lblIcNum)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtIcNum))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel1Layout.createSequentialGroup()
        .addComponent(lblName)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtName, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(lblDate)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtDate, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(jPanel1Layout.createSequentialGroup()

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblTime)
        .addComponent(lblDay))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(txtTime)
        .addComponent(cmbDay, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))))
        .addContainerGap()
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(89, 89, 89)
        .addComponent(lblAppointmentDetails)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(lblAppointmentDetails)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblName)

```

```

        .addComponent(txtName, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblDate)
        .addComponent(txtDate, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblIcNum)
        .addComponent(txtIcNum, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblTime)
        .addComponent(txtTime, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblPhoneNum)
        .addComponent(txtPhoneNum, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(cmbDay, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblDay))
        .addGap(18, 18, 18)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(lblAppointmentId)
        .addComponent(txtAppointmentId, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    txtName.getDocument().addDocumentListener(new DocumentListener() {
        @Override
        public void insertUpdate(DocumentEvent e) {
            processUserInput();
        }

        @Override
        public void removeUpdate(DocumentEvent e) {
            processUserInput();
        }

        @Override

```

```

    public void changedUpdate(DocumentEvent e) {
        processUserInput();
    }

    private void processUserInput() {
        patient.setName(txtName.getText());
    }
});
txtLcNum.getDocument().addDocumentListener(new DocumentListener() {
    @Override
    public void insertUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void removeUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void changedUpdate(DocumentEvent e) {
        processUserInput();
    }

    private void processUserInput() {
        patient.setLcNum(txtLcNum.getText());
    }
});
txtPhoneNum.getDocument().addDocumentListener(new DocumentListener() {
    @Override
    public void insertUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void removeUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void changedUpdate(DocumentEvent e) {
        processUserInput();
    }
});

```

```

        private void processUserInput() {
            patient.setPhoneNum(txtPhoneNum.getText());
        }
    });
    txtAppointmentId.getDocument().addDocumentListener(new DocumentListener() {
        @Override
        public void insertUpdate(DocumentEvent e) {
            processUserInput();
        }

        @Override
        public void removeUpdate(DocumentEvent e) {
            processUserInput();
        }

        @Override
        public void changedUpdate(DocumentEvent e) {
            processUserInput();
        }
    });

    private void processUserInput() {
        appointment.setAppointmentId(txtAppointmentId.getText());
    }
});
txtDate.getDocument().addDocumentListener(new DocumentListener() {
    @Override
    public void insertUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void removeUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void changedUpdate(DocumentEvent e) {
        processUserInput();
    }
});

    private void processUserInput() {
        appointment.setDate(txtDate.getText());
    }
});

```



```

txtTime.getDocument().addDocumentListener(new DocumentListener() {
    @Override
    public void insertUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void removeUpdate(DocumentEvent e) {
        processUserInput();
    }

    @Override
    public void changedUpdate(DocumentEvent e) {
        processUserInput();
    }

    private void processUserInput() {
        appointment.setTime(txtTime.getText());
    }
});

lblService.setText("SERVICE");

cbScalling.setText("Scalling");
cbScalling.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbScallingActionPerformed(evt);
    }
});

cbFilling.setText("Filling");
cbFilling.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbFillingActionPerformed(evt);
    }
});

cbExtraction.setText("Extraction");
cbExtraction.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cbExtractionActionPerformed(evt);
    }
});

```

```

buttonGroup1.add(rdMinor);
rdMinor.setText("Minor");
rdMinor.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        rdMinorActionPerformed(evt);
    }
});

buttonGroup1.add(rdMajor);
rdMajor.setText("Major");
rdMajor.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        rdMajorActionPerformed(evt);
    }
});

lblFillingCount.setText("Teeth Count :");

lblExtractionCount.setText("TeethCount :");

lblPayment.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
lblPayment.setText("Total Payment: RM");

txtPayment.setEditable(false);
txtPayment.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
txtPayment.setText("0.00");

btnConfirm.setText("CONFIRM");
btnConfirm.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnConfirmActionPerformed(evt);
    }
});

btnReset.setText("RESET");
btnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResetActionPerformed(evt);
    }
});

btnBack.setText("BACK");
btnBack.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        btnBackActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(btnBack)
                    .addGap(10, 10, 10)
                    .addComponent(btnReset)
                    .addGap(10, 10, 10)
                    .addComponent(btnConfirm))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(lblService))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(rdMajor))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(cbFilling))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(lblExtractionCount)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(txtExtractionCount, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(lblFillingCount)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(txtFillingCount, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(cbScalling))
            )
        )
    );

```

```

        .addComponent(rdMinor)))
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(cbExtraction))
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(lblPayment)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtPayment, javax.swing.GroupLayout.PREFERRED_SIZE, 166,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel2Layout.createSequentialGroup()
    .addComponent(lblService)
    .addGap(18, 18, 18)
    .addComponent(cbScrolling)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(rdMinor)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(rdMajor)
    .addGap(18, 18, 18)
    .addComponent(cbFilling)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblFillingCount)
        .addComponent(txtFillingCount, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addComponent(cbExtraction)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblExtractionCount)
        .addComponent(txtExtractionCount, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 48,
Short.MAX_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lblPayment)

```

```
        .addComponent(txtPayment, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(23, 23, 23)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnReset)
    .addComponent(btnBack)
    .addComponent(btnConfirm))
    .addGap(29, 29, 29))
);
```

```
txtExtractionCount.getDocument().addDocumentListener(new DocumentListener() {
    @Override
    public void insertUpdate(DocumentEvent e) {
        processUserInput();
    }
}
```

```
    @Override
    public void removeUpdate(DocumentEvent e) {
        processUserInput();
    }
}
```

```
    @Override
    public void changedUpdate(DocumentEvent e) {
        processUserInput();
    }
}
```

```
private void processUserInput() {
    if(txtExtractionCount.getText().isEmpty())
    {
        extractionCount = 0;
    }
    else
    {
        extractionCount = Integer.parseInt(txtExtractionCount.getText());
    }
    updateTotalPayment();
}
});
txtFillingCount.getDocument().addDocumentListener(new DocumentListener() {
    @Override
    public void insertUpdate(DocumentEvent e) {
        processUserInput();
    }
}
```

```

@Override
public void removeUpdate(DocumentEvent e) {
    processUserInput();
}

@Override
public void changedUpdate(DocumentEvent e) {
    processUserInput();
}

private void processUserInput() {
    if(txtFillingCount.getText().isEmpty())
    {
        fillingCount = 0;
    }
    else
    {
        fillingCount = Integer.parseInt(txtFillingCount.getText());
    }
    updateTotalPayment();
}
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

```

```
    pack();  
} // </editor-fold>
```

```
private void cbScallingActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(cbScalling.isSelected()){  
        rdMinor.setEnabled(true);  
        rdMajor.setEnabled(true);  
    }  
    else{  
        rdMinor.setEnabled(false);  
        rdMajor.setEnabled(false);  
        scallingType = null;  
        updateTotalPayment();  
    }  
}
```

```
private void cbFillingActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(cbFilling.isSelected()){  
        txtFillingCount.setEnabled(true);  
    }  
    else{  
        txtFillingCount.setEnabled(false);  
    }  
}
```

```
private void cbExtractionActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(cbExtraction.isSelected()){  
        txtExtractionCount.setEnabled(true);  
    }  
    else{  
        txtExtractionCount.setEnabled(false);  
    }  
}
```

```
private void rdMinorActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(rdMinor.isSelected()){  
        scallingType = "Minor";  
        updateTotalPayment();  
    }  
}
```

```

private void rdMajorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(rdMajor.isSelected()){
        scallingType = "Major";
        updateTotalPayment();
    }
}

```

```

private void cmbDayActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    appointment.setDay(cmbDay.getItemAt(cmbDay.getSelectedIndex()));
    updateTotalPayment();
}

```

```

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    mainGui.setVisible(true);
    this.setVisible(false);
}

```

```

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtName.setText("");
    txtLcNum.setText("");
    txtPhoneNum.setText("");
    txtAppointmentId.setText("");
    txtDate.setText("");
    txtTime.setText("");
    cmbDay.setSelectedIndex(0);
    cbScalling.setSelected(false);
    rdMajor.setSelected(false);
    rdMajor.setEnabled(false);
    rdMinor.setSelected(false);
    rdMinor.setEnabled(false);
    cbFilling.setSelected(false);
    cbExtraction.setSelected(false);
    txtFillingCount.setText("");
    txtExtractionCount.setText("");
    txtPayment.setText("0.00");
}

```

```

private void btnConfirmActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```



```

        appointmentArray.addAppointment(appointment);
        JOptionPane.showMessageDialog(null,"Your Appointment Has Been
Saved.", "Information",JOptionPane.INFORMATION_MESSAGE);
        mainGui.setVisible(true);
        this.setVisible(false);
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnBack;
    private javax.swing.JButton btnConfirm;
    private javax.swing.JButton btnReset;
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JCheckBox cbExtraction;
    private javax.swing.JCheckBox cbFilling;
    private javax.swing.JCheckBox cbScalling;
    private javax.swing.JComboBox<String> cmbDay;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JLabel lblAppointmentDetails;
    private javax.swing.JLabel lblAppointmentId;
    private javax.swing.JLabel lblDate;
    private javax.swing.JLabel lblDay;
    private javax.swing.JLabel lblExtractionCount;
    private javax.swing.JLabel lblFillingCount;
    private javax.swing.JLabel lblIcNum;
    private javax.swing.JLabel lblName;
    private javax.swing.JLabel lblPayment;
    private javax.swing.JLabel lblPhoneNum;
    private javax.swing.JLabel lblService;
    private javax.swing.JLabel lblTime;
    private javax.swing.JRadioButton rdMajor;
    private javax.swing.JRadioButton rdMinor;
    private javax.swing.JTextField txtAppointmentId;
    private javax.swing.JTextField txtDate;
    private javax.swing.JTextField txtExtractionCount;
    private javax.swing.JTextField txtFillingCount;
    private javax.swing.JTextField txtIcNum;
    private javax.swing.JTextField txtName;
    private javax.swing.JTextField txtPayment;
    private javax.swing.JTextField txtPhoneNum;
    private javax.swing.JTextField txtTime;
    // End of variables declaration
}

```