



**UMS**  
UNIVERSITI MALAYSIA SABAH

**KK14203**

OBJECT-ORIENTED PROGRAMMING

PROJECT 2

Name	Matric No
Mohd Farhan Bin Aliuddin	BI17110005

## 1. Java Code

```
import java.io.*;
import java.util.*;
import java.util.Calendar;
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import java.io.File;
import java.io.FileWriter;
import java.util.Scanner;
import java.io.FileNotFoundException;

class doctor extends JFrame{
    void new_doctor(){
        JFrame doctor = new JFrame();
        doctor.setTitle("Doctor Menu ");
        doctor.setSize(650, 350);
        doctor.setLocationRelativeTo(null);
        doctor.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        doctor.setResizable(false);

        JPanel docPanel = new JPanel();
        docPanel.setLayout(new BorderLayout());
        docPanel.setBackground(Color.lightGray);

        docPanel.setLayout (null);
        JLabel docMenu = new JLabel("Welcome to Doctor Menu");
        JTextField doc_id = new JTextField();
        doc_id.setText("Doctor ID");
        JTextField doc_name = new JTextField();
```

```
doc_name.setText("Doctor Name");
JTextField doc_spec = new JTextField();
doc_spec.setText("Doctor Specialization");
JTextField doc_appt = new JTextField();
doc_appt.setText("Doctor Work Time");
JTextField doc_qual = new JTextField();
doc_qual.setText("Doctor Qualification");
JTextField doc_room = new JTextField();
doc_room.setText("Room No");
JButton addDoc = new JButton("Add New Entry");
JButton viewDoc = new JButton("View Current List of Doctors");
```

```
docPanel.add(docMenu);
docPanel.add(doc_id);
docPanel.add(doc_name);
docPanel.add(doc_spec);
docPanel.add(doc_appt);
docPanel.add(doc_qual);
docPanel.add(doc_room);
docPanel.add(addDoc);
docPanel.add(viewDoc);
```

```
docMenu.setBounds (15, 30, 200, 25);
doc_id.setBounds (15, 70, 100, 25);
doc_name.setBounds (15, 100, 450, 25);
doc_spec.setBounds (15, 130, 200, 25);
doc_appt.setBounds (15, 160, 200, 25);
doc_qual.setBounds (15, 190, 200, 25);
doc_room.setBounds (15, 220, 100, 25);
```

```

addDoc.setBounds (15, 250, 130, 25);

viewDoc.setBounds (150, 250, 200, 25);


addDoc.addActionListener(new ActionListener(){
    String saveDoc = "";
    public void actionPerformed(ActionEvent de){
        String tempSave = "Doctor ID: " + doc_id.getText() + "\nDoctor Name: " +
doc_name.getText()
        + "\nDoctor Specialization: " + doc_spec.getText() + "\nDoctor Work
Time: "
        + doc_appt.getText() + "\nDoctor Qualification: " + doc_qual.getText()
        + "\nDoctor Room No: " + doc_room.getText() + "\n\n";

        saveDoc += tempSave;
        try {
            FileWriter myWriter = new FileWriter("DocFile.txt");
            myWriter.write(saveDoc);
            myWriter.close();
            JFrame f=new JFrame();
            JOptionPane.showMessageDialog(f,"Entry Added.");
        } catch (IOException e) {
            JFrame b=new JFrame();
            JOptionPane.showMessageDialog(b,"Error!.");
            e.printStackTrace();
        }

    }
});


viewDoc.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent dp){

```

```

String tempDoc = "";
try {
    File myObj = new File("DocFile.txt");
    Scanner myReader = new Scanner(myObj);
    while (myReader.hasNextLine()) {
        String data = myReader.nextLine() + "\n";
        tempDoc += data;
    }
    myReader.close();
} catch (FileNotFoundException e) {
    JFrame b=new JFrame();
    JOptionPane.showMessageDialog(b,"Error!.");
    e.printStackTrace();
}

JFrame f= new JFrame();
JTextArea area=new JTextArea(tempDoc);
area.setBounds(10,30, 600,600);
f.add(area);
f.setSize(700,700);
f.setLayout(null);
f.setVisible(true);

}

});

doctor.add(docPanel);
doctor.show();
}

```

```
}
```

```
class patient extends JFrame{
```

```
void new_patient(){
```

```
    JFrame patient = new JFrame();
```

```
    patient.setTitle("Patient Menu ");
```

```
    patient.setSize(650, 350);
```

```
    patient.setLocationRelativeTo(null);
```

```
    patient.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
    patient.setResizable(false);
```

```
    JPanel patPanel = new JPanel();
```

```
    patPanel.setLayout(new BorderLayout());
```

```
    patPanel.setBackground(Color.lightGray);
```

```
    patPanel.setLayout (null);
```

```
    JLabel patMenu = new JLabel("Welcome to Patient Menu");
```

```
    JTextField pat_id = new JTextField();
```

```
    pat_id.setText("Patient ID");
```

```
    JTextField pat_name = new JTextField();
```

```
    pat_name.setText("Patient Name");
```

```
    JTextField pat_dis = new JTextField();
```

```
    pat_dis.setText("Patient Disease");
```

```
    String gender[]={ "Male", "Female"};
```

```
    JComboBox pat_gen = new JComboBox(gender);
```

```
    JTextField pat_status = new JTextField();
```

```
    pat_status.setText("Admission Status");
```

```
    JTextField pat_age = new JTextField();
```

```
    pat_age.setText("Patient Age");
```

```
    JButton addPat = new JButton("Add New Entry");
```

```
    JButton viewPat = new JButton("View Current List of Patients");
```

```
patPanel.add(patMenu);
patPanel.add(pat_id);
patPanel.add(pat_name);
patPanel.add(pat_dis);
patPanel.add(pat_gen);
patPanel.add(pat_status);
patPanel.add(pat_age);
patPanel.add(addPat);
patPanel.add(viewPat);
```

```
patMenu.setBounds (15, 30, 200, 25);
pat_id.setBounds (15, 70, 100, 25);
pat_name.setBounds (15, 100, 450, 25);
pat_dis.setBounds (15, 130, 200, 25);
pat_gen.setBounds (15, 160, 200, 25);
pat_status.setBounds (15, 190, 200, 25);
pat_age.setBounds (15, 220, 100, 25);
addPat.setBounds (15, 250, 130, 25);
viewPat.setBounds (150, 250, 200, 25);
```

```
addPat.addActionListener(new ActionListener(){
    String savePat = "";
    public void actionPerformed(ActionEvent pe){
```

```
        String tempSave = "Patient ID: " + pat_id.getText() + "\nPatient Name: " +
pat_name.getText()
        + "\nPatient Disease: " + pat_dis.getText() + "\nPatient Gender: "
        + pat_gen.getItemAt(pat_gen.getSelectedIndex()) + "\nPatient
Admission Status: " + pat_status.getText()
```

```
        + "\nPatient Age: " + pat_age.getText() + "\n\n";
```

```
savePat += tempSave;
```

```
try {
```

```
    FileWriter myWriter = new FileWriter("PatFile.txt");
```

```
    myWriter.write(savePat);
```

```
    myWriter.close();
```

```
    JFrame f=new JFrame();
```

```
    JOptionPane.showMessageDialog(f,"Entry Added.");
```

```
} catch (IOException e) {
```

```
    JFrame b=new JFrame();
```

```
    JOptionPane.showMessageDialog(b,"Error!.");
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

```
});
```

```
viewPat.addActionListener(new ActionListener(){
```

```
    public void actionPerformed(ActionEvent pp){
```

```
        String tempPat = "";
```

```
        try {
```

```
            File myObj = new File("PatFile.txt");
```

```
            Scanner myReader = new Scanner(myObj);
```

```
            while (myReader.hasNextLine()) {
```

```
                String data = myReader.nextLine() + "\n";
```

```
                tempPat += data;
```

```
            }
```

```
            myReader.close();
```

```
        } catch (FileNotFoundException e) {
```



```
JFrame b=new JFrame();  
OptionPane.showMessageDialog(b,"Error!.");  
e.printStackTrace();  
}
```

```
JFrame g = new JFrame();  
JTextArea area2 =new JTextArea(tempPat);  
area2.setBounds(10,30, 600,600);  
g.add(area2);  
g.setSize(700,700);  
g.setLayout(null);  
g.setVisible(true);  
  
}  
});
```

```
patient.add(patPanel);  
patient.show();  
}  
  
}
```

```
abstract public class HospitalManagement implements ActionListener{  
    public static void main(String args[]){  
        doctor d = new doctor();  
        patient p = new patient();  
        JFrame menu = new JFrame();  
        menu.setTitle("UMS Hospital Management System ");
```

```
menu.setSize(500, 200);
menu.setLocationRelativeTo(null);
menu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
menu.setResizable(false);
```

```
JPanel panel = new JPanel();
panel.setLayout(new BorderLayout());
panel.setBackground(Color.orange);
```

```
panel.setLayout (null);
```

```
String months[] =
{"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"};
Calendar calendar = Calendar.getInstance();
JLabel intro = new JLabel("Welcome to UMS Hospital Management System");
intro.setBounds (120, 5, 350, 30);
panel.add(intro);
```

```
JLabel time = new JLabel("Time: " + calendar.get(Calendar.HOUR) + ":" +
calendar.get(Calendar.MINUTE) + ":" + calendar.get(Calendar.SECOND));
time.setBounds (15, 50, 100, 30);
panel.add(time);
```

```
JLabel date = new JLabel("Date: " + months[calendar.get(Calendar.MONTH)] + " " +
calendar.get(Calendar.DATE) + " " + calendar.get(Calendar.YEAR));
```

```
date.setBounds (100, 50, 100, 30);
panel.add(date);
```

```
JLabel main1 = new JLabel("-----[ MAIN MENU ]-----");
```

```
main1.setBounds (170, 70, 200, 30);
panel.add(main1);

JButton doc1 = new JButton("Doctors");
JButton pat1 = new JButton("Patients");

doc1.setBounds (15, 110, 100, 25);
pat1.setBounds (120, 110, 100, 25);

panel.add(doc1);
panel.add(pat1);

doc1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent x){
        d.new_doctor();

    } });

pat1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent y){
        p.new_patient();

    } });
menu.add(panel);
menu.show();
}
}
```

## 2. Object Oriented Concept Implementation

- 3 Classes were implemented: class HospitalManagement, class patient, class doctor.
- 2 Polymorphism were implemented on 2 Classes: class doctor extends JFrame, class patient extends JFrame.
- 1 Abstract Class was implemented: abstract public class HospitalManagement.
- 4 Exception handler were implemented: line 78-84, line 97-102, line 187-192, line 206-211.

## 3. Read and Write Implementation

- The read implementation can be found in line 89-97 and line 198-206.

```
String tempDoc = "";
try {
    File myObj = new File("DocFile.txt");
    Scanner myReader = new Scanner(myObj);
    while (myReader.hasNextLine()) {
        String data = myReader.nextLine() + "\n";
        tempDoc += data;
    }
    myReader.close();
```

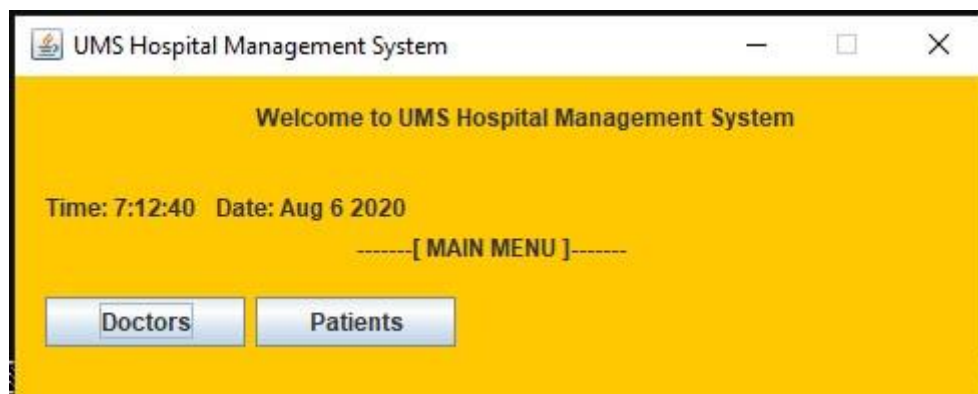
- The write implementation can be found in line 64-75 and line 172-184.

```
String saveDoc = "";
public void actionPerformed(ActionEvent de){
    String tempSave = "Doctor ID: " + doc_id.getText() + "\nDoctor Name: " + doc_name.getText()
        + "\nDoctor Specialization: " + doc_spec.getText() + "\nDoctor Work Time: "
        + doc_appt.getText() + "\nDoctor Qualification: " + doc_qual.getText()
        + "\nDoctor Room No: " + doc_room.getText() + "\n\n";

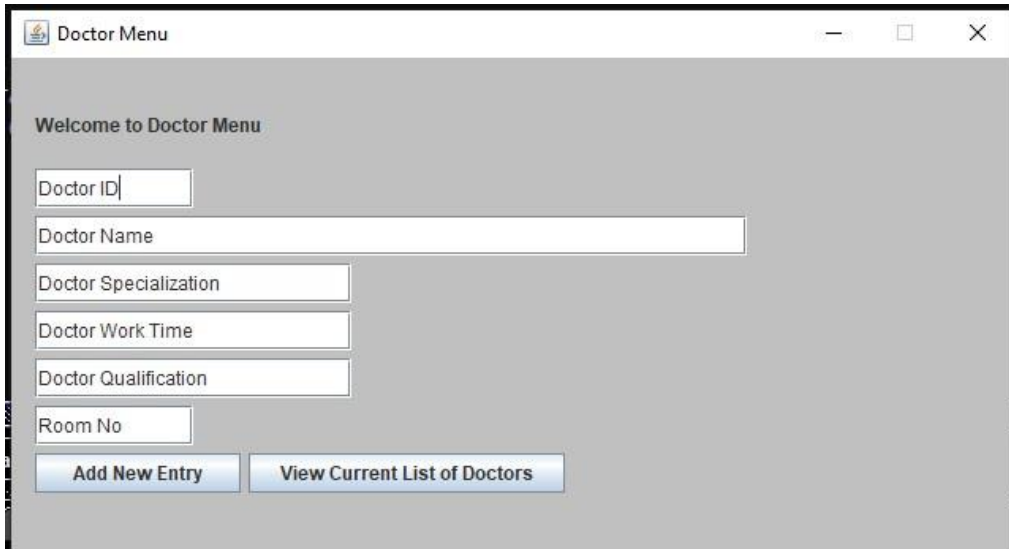
    saveDoc += tempSave;
    try {
        FileWriter myWriter = new FileWriter("DocFile.txt");
        myWriter.write(saveDoc);
        myWriter.close();
```

## 4. Manual

- This is the main menu for the system which allowed for user to begin their work by prompting to user to select on which operation they want to run.

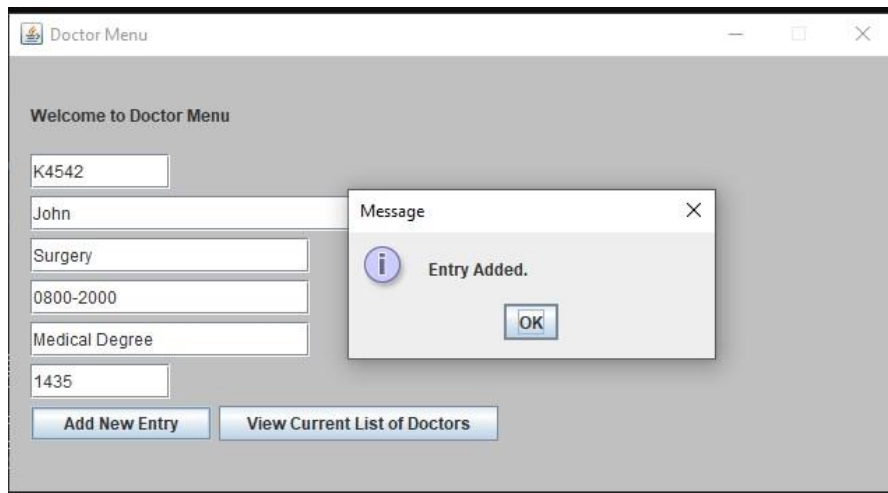


- After picking between Doctors and Patients, user can now choose to add new entry into the system or view existing entry in the system.



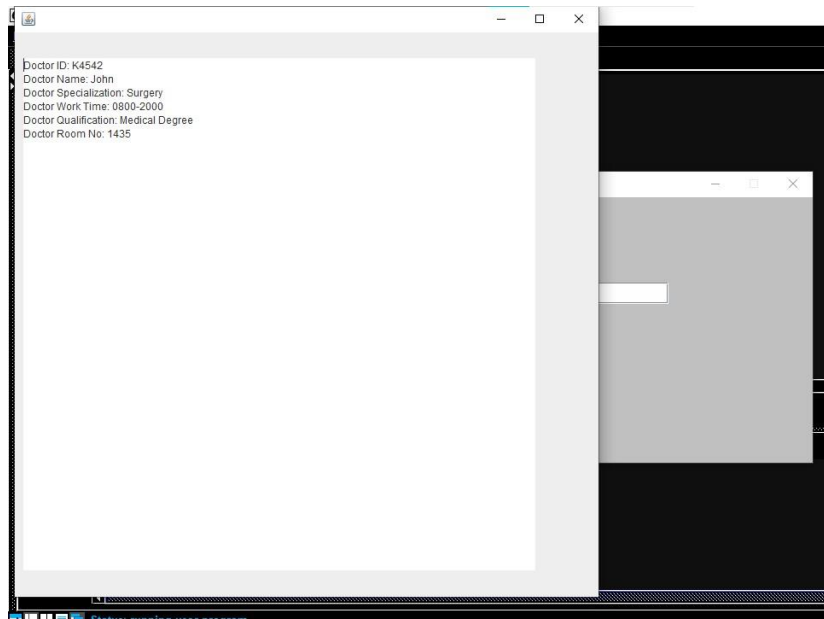
The screenshot shows a window titled "Doctor Menu" with a "Welcome to Doctor Menu" message. Below the message is a form with the following fields: "Doctor ID", "Doctor Name", "Doctor Specialization", "Doctor Work Time", "Doctor Qualification", and "Room No". At the bottom of the form are two buttons: "Add New Entry" and "View Current List of Doctors".

- When user successfully added an entry, they will receive a popup of successful entry.



The screenshot shows the same "Doctor Menu" window as before, but with a "Message" popup dialog box overlaid. The popup contains an information icon, the text "Entry Added.", and an "OK" button. The form fields in the background now contain the following data: "Doctor ID" is "K4542", "Doctor Name" is "John", "Doctor Specialization" is "Surgery", "Doctor Work Time" is "0800-2000", "Doctor Qualification" is "Medical Degree", and "Room No" is "1435".

- User can view existing or newly added entry using the view current button.



- User can close the current window to return to the main menu.

