



CSE 228 Advanced Programming

1/40

Healthcare Management System **Project**

Faculty of Engineering Ain Shams
University

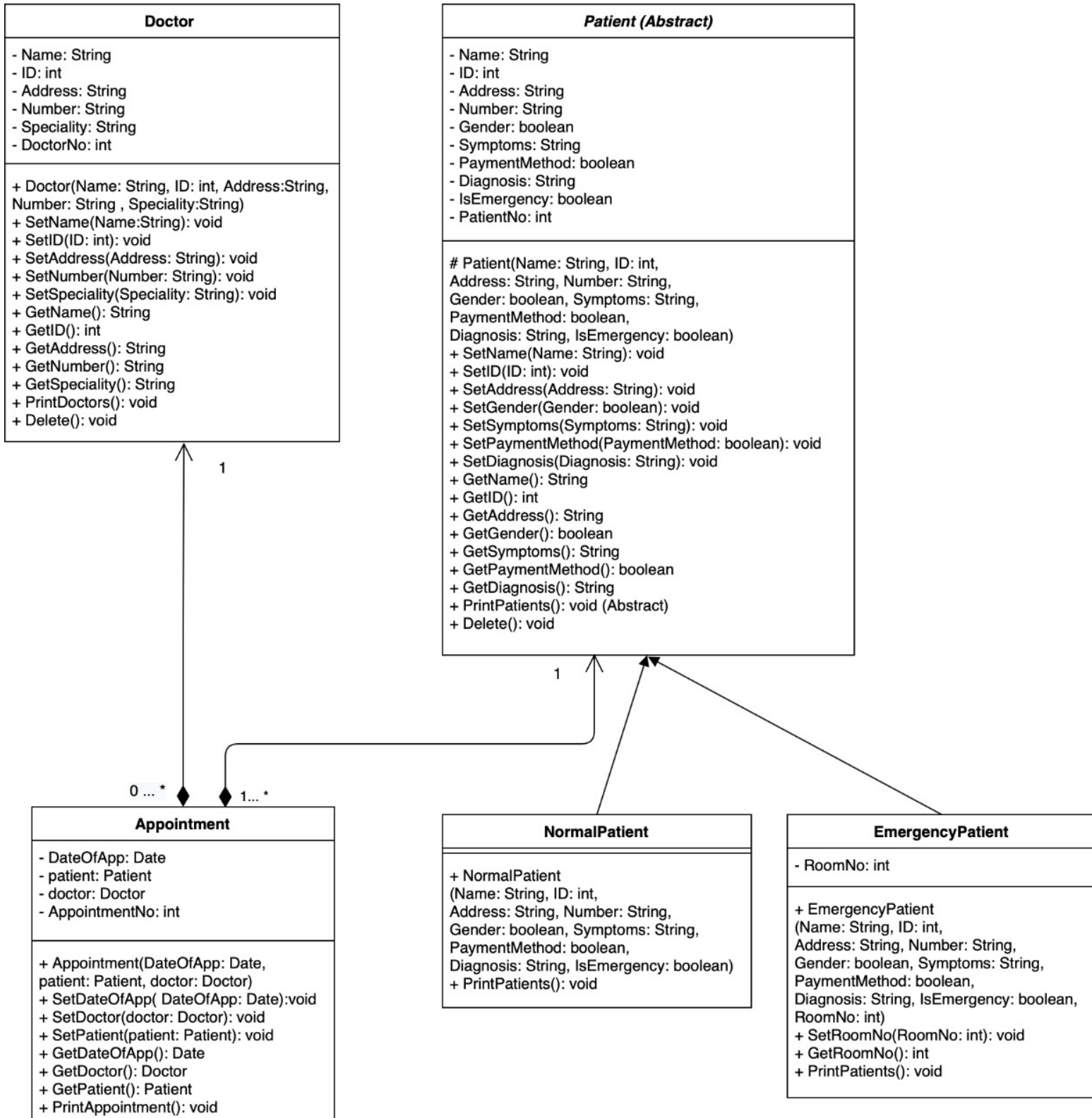
Group 1

<i>Name</i>	<i>ID</i>
<i>Mohamed Abd El Hamed Moustafa</i>	<i>16p8173</i>
<i>George Medhat Halim</i>	<i>16P8197</i>
<i>Omar Ahmed Sayed</i>	<i>16p8199</i>
<i>Omar Moustafa Ali Makin</i>	<i>16p8169</i>
<i>Youssef Mostafa Hammad</i>	<i>16p8179</i>

Video Link:

<https://youtu.be/JH8m60qKIH8>

UML



Code:

GUI:

```
/*  
    * To change this license header, choose License Headers in Project Properties.  
    * To change this template file, choose Tools | Templates  
    * and open the template in the editor.  
    */  
  
package hospital;  
  
  
import java.util.ArrayList;  
import javafx.application.Application;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.GridPane;  
import javafx.geometry.Pos;  
import javafx.geometry.HPos;  
import javafx.scene.control.RadioButton;  
import javafx.scene.control.ToggleGroup;  
import javafx.stage.Stage;  
  
/**  
    *  
    * @author dell  
    */  
  
public class Gui extends Application {  
    private ArrayList<Doctor> Doctors_array = new ArrayList<Doctor>();  
    private ArrayList<Patient> Patients_array = new ArrayList<Patient>();  
    private ArrayList<Appointment> Appointments_array = new ArrayList<Appointment>();  
  
    private Label Select_Field_label= new Label("Select Field :");  
    private Button Doctors_btn = new Button("Doctors");  
    private Button Patients_btn = new Button("Patients");
```

```

private Button Appointments_btn = new Button("Appointments");

private Label Select_DocOption_label= new Label("Select Option :");
private Button DocAdd_btn = new Button("Add Doctor");
private Button DocPrint = new Button("Print all Doctors");
private Button DocEdit = new Button("Edit Doctor");
private Button DocDelete = new Button("Delete Doctor");


private Label DoctorName_label= new Label("Doctor Name");
private Label DoctorID_label= new Label("Doctor ID");
private Label DoctorAddress_label= new Label("Doctor Address");
private Label DoctorSpeciality_label= new Label("Doctor Speciality");
private Label DoctorPhoneNumber_label= new Label("Doctor Phone Number");
private TextField DoctorName_TextField= new TextField();
private TextField DoctorID_TextField= new TextField();
private TextField DoctorAddress_TextField= new TextField();
private TextField DoctorSpeciality_TextField= new TextField();
private TextField DoctorPhoneNumber_TextField= new TextField();
private TextField DoctorDeleteNumber_TextField= new TextField();
private TextField DoctorEditNumber_TextField= new TextField();


private Label EDoctorName_label= new Label("Doctor Name");
private Label EDoctorID_label= new Label("Doctor ID");
private Label EDoctorAddress_label= new Label("Doctor Address");
private Label EDoctorSpeciality_label= new Label("Doctor Speciality");
private Label EDoctorPhoneNumber_label= new Label("Doctor Phone Number");
private TextField EDoctorName_TextField= new TextField();
private TextField EDoctorID_TextField= new TextField();
private TextField EDoctorAddress_TextField= new TextField();
private TextField EDoctorSpeciality_TextField= new TextField();
private TextField EDoctorPhoneNumber_TextField= new TextField();
private TextField EDoctorDeleteNumber_TextField= new TextField();
private TextField EDoctorEditNumber_TextField= new TextField();


private Button DocCreate_btn = new Button("Add the Doctor");
private Button DocEdit_Submit_btn = new Button("Edit Doctor");

```

```
//Patient
```

```

private Label Select_PatOption_label= new Label("Select Option :");
private Button PatAdd_btn = new Button("Add Patient");
private Button PatPrint = new Button("Print all Patients");
private Button PatEdit = new Button("Edit Patient");
private Button PatDelete = new Button("Delete Patient");

private Label PatientName_label= new Label("Patient Name");
private Label PatientID_label= new Label("Patient ID");
private Label PatientAddress_label= new Label("Patient Address");
private Label PatientSymptoms_label= new Label("Patient Symptoms");
private Label PatientPhoneNumber_label= new Label("Patient Phone Number");
private Label PatientGender_label= new Label("Patient Gender");
private Label PatientPaymentMethod_label= new Label("Patient PaymentMethod");
private Label PatientDiagnosis_label= new Label("Patient Diagnosis");
private Label PatientIsEmergency_label= new Label("Patient IsEmergency");
private TextField PatientName_TextField= new TextField();
private TextField PatientID_TextField= new TextField();
private TextField PatientAddress_TextField= new TextField();
private TextField PatientSymptoms_TextField= new TextField();
private TextField PatientPhoneNumber_TextField= new TextField();
private TextField PatientGender_TextField= new TextField();
private TextField PatientPaymentMethod_TextField= new TextField();
private TextField PatientDiagnosis_TextField= new TextField();
private TextField PatientIsEmergency_TextField= new TextField();
private TextField PatientDeleteNumber_TextField= new TextField();
private TextField PatientEditNumber_TextField= new TextField();

private Label EPatientName_label= new Label("Patient Name");
private Label EPatientID_label= new Label("Patient ID");
private Label EPatientAddress_label= new Label("Patient Address");
private Label EPatientSymptoms_label= new Label("Patient Symptoms");
private Label EPatientPhoneNumber_label= new Label("Patient Phone Number");
private Label EPatientGender_label= new Label("Patient Gender");
private Label EPatientPaymentMethod_label= new Label("Patient PaymentMethod");
private Label EPatientDiagnosis_label= new Label("Patient Diagnosis");
private Label EPatientIsEmergency_label= new Label("Patient IsEmergency");
private TextField EPatientName_TextField= new TextField();
private TextField EPatientID_TextField= new TextField();

```

```

private TextField EPatientAddress_TextField= new TextField();
private TextField EPatientSymptoms_TextField= new TextField();
private TextField EPatientPhoneNumber_TextField= new TextField();
private TextField EPatientGender_TextField= new TextField();
private TextField EPatientPaymentMethod_TextField= new TextField();
private TextField EPatientDiagnosis_TextField= new TextField();
private TextField EPatientIsEmergency_TextField= new TextField();
private TextField EPatientDeleteNumber_TextField= new TextField();
private TextField EPatientEditNumber_TextField= new TextField();

private Button PatCreate_btn = new Button("Add the Patient");
private Button PatEdit_Submit_btn = new Button("Edit Patient");

private Label Select_AppOption_label= new Label("Select Option :");
private Button AppAdd_btn = new Button("Add Appointment");
private Button AppPrint = new Button("Print all Appointments");
private Button AppDelete = new Button("Delete Appointment");

private TextField AppointmentDeleteNumber_TextField= new TextField();
private Label AppointmentDoctorNo_label= new Label("Appointment Doctor Number");
private Label AppointmentPatientNo_label= new Label("Appointment Patient Number");
private Label AppointmentDate_label= new Label("Appointment Date");
private TextField AppointmentDoctorNo_TextField= new TextField();
private TextField AppointmentPatientNo_TextField= new TextField();
private TextField AppointmentDate_TextField= new TextField();

private Button AppCreate_btn = new Button("Add the Appointment");

@Override
public void start(Stage primaryStage) {

    GridPane grid_pane = new GridPane();
    grid_pane.setHgap(5);
    grid_pane.setVgap(5);
    grid_pane.setAlignment(Pos.CENTER);

    grid_pane.add(Select_Field_label, 0, 0);

```

```

grid_pane.add(Doctors_btn, 0, 1);
grid_pane.add(Patients_btn, 2, 1);
grid_pane.add(Appointments_btn, 4, 1);

Scene scene = new Scene(grid_pane, 700, 600);

primaryStage.setTitle("Hospital");
primaryStage.setScene(scene);
primaryStage.show();

//Doctors
GridPane Doc_grid_pane = new GridPane();
Doc_grid_pane.setHgap(10);
Doc_grid_pane.setVgap(10);
Doc_grid_pane.setAlignment(Pos.CENTER);

Doc_grid_pane.add(Select_DocOption_label, 0, 0);
Doc_grid_pane.add(DocAdd_btn, 0, 1);
Doc_grid_pane.add(DocPrint, 4, 1);
Doc_grid_pane.add(DocEdit, 6, 1);
Doc_grid_pane.add(DoctorEditNumber_TextField, 6, 2);
Doc_grid_pane.add(DocDelete, 8, 1);
Doc_grid_pane.add(DoctorDeleteNumber_TextField, 8, 2);

Stage Doctors_Stage = new Stage();
Doctors_Stage.setTitle("Doctors");
Scene Doctor_scene = new Scene(Doc_grid_pane, 1000, 600);
Doctors_Stage.setScene(Doctor_scene);

Doctors_btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

```

```

        Doctors_Stage.show();

    }

});

//Doctor Add
GridPane DocAdd_grid_pane = new GridPane();
DocAdd_grid_pane.setHgap(10);
DocAdd_grid_pane.setVgap(10);
DocAdd_grid_pane.setAlignment(Pos.CENTER);

DocAdd_grid_pane.add(DoctorName_label, 0, 0);
DocAdd_grid_pane.add(DoctorName_TextField, 1, 0);
DocAdd_grid_pane.add(DoctorID_label, 0, 1);
DocAdd_grid_pane.add(DoctorID_TextField, 1, 1);
DocAdd_grid_pane.add(DoctorAddress_label, 0, 2);
DocAdd_grid_pane.add(DoctorAddress_TextField, 1, 2);
DocAdd_grid_pane.add(DoctorSpeciality_label, 0, 3);
DocAdd_grid_pane.add(DoctorSpeciality_TextField, 1, 3);
DocAdd_grid_pane.add(DoctorPhoneNumber_label, 0, 4);
DocAdd_grid_pane.add(DoctorPhoneNumber_TextField, 1, 4);
DocAdd_grid_pane.add(DocCreate_btn, 1, 9);

Stage DoctorsAdd_Stage = new Stage();
DoctorsAdd_Stage.setTitle("Add Doctor");
Scene DoctorAdd_scene = new Scene(DocAdd_grid_pane, 500, 700);
DoctorsAdd_Stage.setScene(DoctorAdd_scene);

DocAdd_btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

```



```

DoctorsAdd_Stage.show();

}

});

DocCreate_btn.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

String DoctorName = (DoctorName_TextField.getText());
String DoctorID = (DoctorID_TextField.getText());
String DoctorAddress = (DoctorAddress_TextField.getText());
String DoctorSpeciality = (DoctorSpeciality_TextField.getText());
String DoctorPhoneNumber = (DoctorPhoneNumber_TextField.getText());

Doctors_array.add(new
Doctor(DoctorName,DoctorID,DoctorAddress,DoctorSpeciality,DoctorPhoneNumber));

System.out.println("Doctor Added Successfully" );

}

});

//doctor Print

DocPrint.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent event) {

System.out.println("Total Number of Doctors in Hospital =" + Doctors_array.size()
+" Doctors");

System.out.println(".....");

for (int D = 0; D < Doctors_array.size(); D++){

int n=D+1;

System.out.println("Doctor " +n+" Name is " + (Doctors_array.get(D)).GetName());
System.out.println("Doctor " +n+" ID is " + (Doctors_array.get(D)).GetID());
System.out.println("Doctor " +n+" Address is " + (Doctors_array.get(D)).GetAddress());
System.out.println("Doctor " +n+" Speciality is " + (Doctors_array.get(D)).getSpeciality());
System.out.println("Doctor " +n+" Phone Number is " + (Doctors_array.get(D)).GetNumber());

```

```

        System.out.println(".....");
    }
    System.out.println("Doctors Printed Successfully");
    System.out.println("////////////////////////////////////////");
}
});

//Doctor Delete
DocDelete.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

        try {

            int DoctorDeleteNumber = (Integer.parseInt(DoctorDeleteNumber_TextField.getText()))-1;
            Doctors_array.remove(DoctorDeleteNumber);
            System.out.println("Doctor Deleted Successfully");
            System.out.println("////////////////////////////////////////");
        }
        catch(IndexOutOfBoundsException E1){
            System.out.println("Invalid Doctor Number!!! Check their number from printing");
        }
        catch(Exception E2){
            System.out.println("Invalid Type!!! Enter Doctor number");
        }

    }

});

//Doctor Edit
GridPane DocEdit_grid_pane = new GridPane();
DocEdit_grid_pane.setHgap(10);
DocEdit_grid_pane.setVgap(10);
DocEdit_grid_pane.setAlignment(Pos.CENTER);

DocEdit_grid_pane.add(EDoctorName_label, 0, 0);
DocEdit_grid_pane.add(EDoctorName_TextField, 1, 0);
DocEdit_grid_pane.add(EDoctorID_label, 0, 1);

```

```

DocEdit_grid_pane.add(EDoctorID_TextField, 1, 1);
DocEdit_grid_pane.add(EDoctorAddress_label, 0, 2);
DocEdit_grid_pane.add(EDoctorAddress_TextField, 1, 2);
DocEdit_grid_pane.add(EDoctorSpeciality_label, 0, 3);
DocEdit_grid_pane.add(EDoctorSpeciality_TextField, 1, 3);
DocEdit_grid_pane.add(EDoctorPhoneNumber_label, 0, 4);
DocEdit_grid_pane.add(EDoctorPhoneNumber_TextField, 1, 4);
DocEdit_grid_pane.add(DocEdit_Submit_btn, 1, 9);

```

```

Stage DoctorsEdit_Stage = new Stage();
DoctorsEdit_Stage.setTitle("Edit Doctor");
Scene DoctorEdit_scene = new Scene(DocEdit_grid_pane,500,700);
DoctorsEdit_Stage.setScene(DoctorEdit_scene);

```

```

DocEdit.setOnAction(new EventHandler<ActionEvent>() {

```

```

@Override

```

```

public void handle(ActionEvent event) {

```

```

try{
    int E = (Integer.parseInt(DoctorEditNumber_TextField.getText()))-1;
    EDoctorName_TextField.setText(Doctors_array.get(E).GetName());
    EDoctorID_TextField.setText(Doctors_array.get(E).GetID());
    EDoctorAddress_TextField.setText(Doctors_array.get(E).GetAddress());
    EDoctorSpeciality_TextField.setText(Doctors_array.get(E).getSpeciality());
    EDoctorPhoneNumber_TextField.setText(Doctors_array.get(E).GetNumber());

    DoctorsEdit_Stage.show();
}
catch(IndexOutOfBoundsException E1){
    System.out.println("Invalid Doctor Number!!! Check their number from printing");
}
catch(Exception E2){
    System.out.println("Invalid Type!!! Enter Doctor number");
}
}

```

```
}  
});
```

```
DocEdit_Submit_btn.setOnAction(new EventHandler<ActionEvent>() {
```

```
@Override
```

```
public void handle(ActionEvent event) {
```

```
int E = (Integer.parseInt(DoctorEditNumber_TextField.getText()))-1;
```

```
String EDoctorName = (EDoctorName_TextField.getText());
```

```
String EDoctorID = (EDoctorID_TextField.getText());
```

```
String EDoctorAddress = (EDoctorAddress_TextField.getText());
```

```
String EDoctorSpeciality = (EDoctorSpeciality_TextField.getText());
```

```
String EDoctorPhoneNumber = (EDoctorPhoneNumber_TextField.getText());
```

```
Doctors_array.get(E).SetName(EDoctorName);
```

```
Doctors_array.get(E).SetID(EDoctorID);
```

```
Doctors_array.get(E).SetAddress(EDoctorAddress);
```

```
Doctors_array.get(E).SetSpeciality(EDoctorSpeciality);
```

```
Doctors_array.get(E).SetNumber(EDoctorSpeciality);
```

```
System.out.println("Doctor Edited Successfully" );
```

```
System.out.println("////////////////////////////////////////");
```

```
}  
});
```

```
//Patient
```

```
GridPane Pat_grid_pane = new GridPane();
```

```
Pat_grid_pane.setHgap(10);
```

```
Pat_grid_pane.setVgap(10);
```

```
Pat_grid_pane.setAlignment(Pos.CENTER);
```

```
Pat_grid_pane.add(Select_PatOption_label, 0, 0);
```

```
Pat_grid_pane.add(PatAdd_btn, 0, 1);
```

```
Pat_grid_pane.add(PatPrint, 4, 1);
```

```
Pat_grid_pane.add(PatEdit, 6, 1);
```

```
Pat_grid_pane.add(PatientEditNumber_TextField, 6, 2);
Pat_grid_pane.add(PatDelete, 8, 1);
Pat_grid_pane.add(PatientDeleteNumber_TextField, 8, 2);
```

```
Stage Patients_Stage = new Stage();
Patients_Stage.setTitle("Patients");
Scene Patient_scene = new Scene(Pat_grid_pane, 1000, 600);
Patients_Stage.setScene(Patient_scene);
```

```
Patients_btn.setOnAction(new EventHandler<ActionEvent>() {
```

```
    @Override
```

```
    public void handle(ActionEvent event) {
```

```
        Patients_Stage.show();
```

```
    }
```

```
});
```

```
//Patient Add
```

```
GridPane PatAdd_grid_pane = new GridPane();
```

```
PatAdd_grid_pane.setHgap(10);
```

```
PatAdd_grid_pane.setVgap(10);
```

```
PatAdd_grid_pane.setAlignment(Pos.CENTER);
```

```
    // create a toggle group
```

```
ToggleGroup Gender = new ToggleGroup();
```

```
// create radiobuttons
```

```
RadioButton MaleRB = new RadioButton("male");
```

```
RadioButton FemaleRB = new RadioButton("female");
```

```
// add radiobuttons to toggle group
```

```
MaleRB.setToggleGroup(Gender);
```

```
FemaleRB.setToggleGroup(Gender);
```

```
ToggleGroup Payment = new ToggleGroup();
```

```
// create radiobuttons
```

```
RadioButton CashRB = new RadioButton("Cash Payment");
```

```
RadioButton VisaRB = new RadioButton("Visa Payment");
```

```
// add radiobuttons to toggle group
```

```
CashRB.setToggleGroup(Payment);
```

```
VisaRB.setToggleGroup(Payment);
```

```
ToggleGroup PatientEmergency = new ToggleGroup();
```

```
// create radiobuttons
```

```
RadioButton PatientEmergencyRB = new RadioButton("Emergency Condition");
```

```
RadioButton PatientNormalRB = new RadioButton("Normal Condition");
```

```
// add radiobuttons to toggle group
```

```
PatientEmergencyRB.setToggleGroup(PatientEmergency);
```

```
PatientNormalRB.setToggleGroup(PatientEmergency);
```

```
PatAdd_grid_pane.add(PatientName_label, 0, 0);
```

```
PatAdd_grid_pane.add(PatientName_TextField, 1, 0);
```

```
PatAdd_grid_pane.add(PatientID_label, 0, 1);
```

```
PatAdd_grid_pane.add(PatientID_TextField, 1, 1);
```

```
PatAdd_grid_pane.add(PatientAddress_label, 0, 2);
```

```
PatAdd_grid_pane.add(PatientAddress_TextField, 1, 2);
```

```
PatAdd_grid_pane.add(PatientSymptoms_label, 0, 3);
```

```
PatAdd_grid_pane.add(PatientSymptoms_TextField, 1, 3);
```

```
PatAdd_grid_pane.add(PatientPhoneNumber_label, 0, 4);
```

```
PatAdd_grid_pane.add(PatientPhoneNumber_TextField, 1, 4);
```

```
PatAdd_grid_pane.add(PatientDiagnosis_label, 0, 5);
```

```
PatAdd_grid_pane.add(PatientDiagnosis_TextField, 1, 5);
```

```
PatAdd_grid_pane.add(MaleRB, 1, 6);
```

```
PatAdd_grid_pane.add(FemaleRB, 2, 6);
```

```
PatAdd_grid_pane.add(CashRB, 1, 7);
```

```

PatAdd_grid_pane.add(VisaRB, 2, 7);
PatAdd_grid_pane.add(PatientEmergencyRB, 1, 8);
PatAdd_grid_pane.add(PatientNormalRB, 2, 8);
PatAdd_grid_pane.add(PatCreate_btn, 1, 9);

```

```

Stage PatientsAdd_Stage = new Stage();
PatientsAdd_Stage.setTitle("Add Patient");
Scene PatientAdd_scene = new Scene(PatAdd_grid_pane, 600, 700);
PatientsAdd_Stage.setScene(PatientAdd_scene);

```

```

PatAdd_btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

        PatientsAdd_Stage.show();

    }
});

```

```

PatCreate_btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

        String PatientName = (PatientName_TextField.getText());
        String PatientID = (PatientID_TextField.getText());
        String PatientAddress = (PatientAddress_TextField.getText());
        String PatientPhoneNumber = (PatientPhoneNumber_TextField.getText());
        boolean PatientGender= (MaleRB.isSelected());
        String PatientSymptoms = (PatientSymptoms_TextField.getText());
        boolean PatientPaymentMethod= (CashRB.isSelected());
        String PatientDiagnosis = (PatientDiagnosis_TextField.getText());
        boolean PatientIsEmergency = (PatientEmergencyRB.isSelected());
    }
});

```

```

        if (PatientIsEmergency==true){
            Patients_array.add(new
EmergencyPatient(PatientName,PatientID,PatientAddress,PatientPhoneNumber,PatientGender,PatientSym
ptoms,PatientPaymentMethod,PatientDiagnosis));

        }
        else {

            Patients_array.add(new
NormalPatient(PatientName,PatientID,PatientAddress,PatientPhoneNumber,PatientGender,PatientSym
ptoms,PatientPaymentMethod,PatientDiagnosis));

        }

        System.out.println("Patient Added Successfully" );

    }

});

//Patient Print
PatPrint.setAction(new EventHandler<ActionEvent>() {

@Override
public void handle(ActionEvent event) {

System.out.println("Total Number of Patients in Hospital =" + Patients_array.size() +" Patients");
System.out.println(".....");
    for (int D = 0; D < Patients_array.size(); D++){
        int n=D+1;

        System.out.println("Patient " +n+" Name is "+ (Patients_array.get(D)).GetName());
        System.out.println("Patient " +n+" ID is "+ (Patients_array.get(D)).GetID());
        System.out.println("Patient " +n+" Address is "+(Patients_array.get(D)).GetAddress());
        System.out.println("Patient " +n+" Phone Number is "+ (Patients_array.get(D)).GetNumber());
        if(Patients_array.get(D).GetGender()==true){

            System.out.println("Patient " +n+" Gender is Male");

        }
        else{
            System.out.println("Patient " +n+" Gender is Female");

        }
    }
}
}

```



```

        System.out.println("Patient " +n+" Symptoms is "+ (Patients_array.get(D)).GetSymptoms());

        if(Patients_array.get(D).GetPaymentMethod()==true){

            System.out.println("Patient " +n+" Payment is Cash");
        }
        else{
            System.out.println("Patient " +n+" Payment is Visa");
        }

        System.out.println("Patient " +n+" Diagnosis is "+ (Patients_array.get(D)).GetDiagnosis());

        if(Patients_array.get(D).GetIsEmergency()==true){

            System.out.println("Patient " +n+" Payment is Emergency Patient");
        }
        else{
            System.out.println("Patient " +n+" is Normal Patient");
        }
        System.out.println(".....");
    }
    System.out.println("Patients Printed Successfully" );
    System.out.println("////////////////////////////////////////");
}
});

//Patient Delete
PatDelete.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

        try{
            int PatientDeleteNumber = (Integer.parseInt(PatientDeleteNumber_TextField.getText()))-1;
            Patients_array.remove(PatientDeleteNumber);
            System.out.println("Patient Deleted Successfully" );
            System.out.println("////////////////////////////////////////");
        }
        catch(IndexOutOfBoundsException E1){

```

```

        System.out.println("Invalid Patient Number!!! Check their number from printing");
    }
    catch(Exception E2){
        System.out.println("Invalid Type!!! Enter Patient number");

    }
}
});

```

```

        //Patient Edit
GridPane PatEdit_grid_pane = new GridPane();
PatEdit_grid_pane.setHgap(10);
PatEdit_grid_pane.setVgap(10);
PatEdit_grid_pane.setAlignment(Pos.CENTER);

        // create a toggle group
ToggleGroup EGender = new ToggleGroup();

// create radiobuttons
RadioButton EMaleRB = new RadioButton("male");
RadioButton EFemaleRB = new RadioButton("female");

// add radiobuttons to toggle group
EMaleRB.setToggleGroup(EGender);
EFemaleRB.setToggleGroup(EGender);

ToggleGroup EPayment = new ToggleGroup();

// create radiobuttons
RadioButton ECashRB = new RadioButton("Cash Payment");
RadioButton EVisaRB = new RadioButton("Visa Payment");

// add radiobuttons to toggle group
ECashRB.setToggleGroup(EPayment);
EVisaRB.setToggleGroup(EPayment);

```

```

ToggleGroup EPatientEmergency = new ToggleGroup();

// create radiobuttons
RadioButton EPatientEmergencyRB = new RadioButton("Emergency Condition");
RadioButton EPatientNormalRB = new RadioButton("Normal Condition");

// add radiobuttons to toggle group
EPatientEmergencyRB.setToggleGroup(EPatientEmergency);
EPatientNormalRB.setToggleGroup(EPatientEmergency);

PatEdit_grid_pane.add(EPatientName_label, 0, 0);
PatEdit_grid_pane.add(EPatientName_TextField, 1, 0);
PatEdit_grid_pane.add(EPatientID_label, 0, 1);
PatEdit_grid_pane.add(EPatientID_TextField, 1, 1);
PatEdit_grid_pane.add(EPatientAddress_label, 0, 2);
PatEdit_grid_pane.add(EPatientAddress_TextField, 1, 2);
PatEdit_grid_pane.add(EPatientSymptoms_label, 0, 3);
PatEdit_grid_pane.add(EPatientSymptoms_TextField, 1, 3);
PatEdit_grid_pane.add(EPatientPhoneNumber_label, 0, 4);
PatEdit_grid_pane.add(EPatientPhoneNumber_TextField, 1, 4);
PatEdit_grid_pane.add(EPatientDiagnosis_label, 0, 5);
PatEdit_grid_pane.add(EPatientDiagnosis_TextField, 1, 5);
PatEdit_grid_pane.add(EMaleRB, 1, 6);
PatEdit_grid_pane.add(EFemaleRB, 2, 6);
PatEdit_grid_pane.add(ECashRB, 1, 7);
PatEdit_grid_pane.add(EVisaRB, 2, 7);
PatEdit_grid_pane.add(EPatientEmergencyRB, 1, 8);
PatEdit_grid_pane.add(EPatientNormalRB, 2, 8);
PatEdit_grid_pane.add(PatEdit_Submit_btn, 1, 9);

Stage PatientsEdit_Stage = new Stage();
PatientsEdit_Stage.setTitle("Edit Patient");
Scene PatientEdit_scene = new Scene(PatEdit_grid_pane, 500, 700);
PatientsEdit_Stage.setScene(PatientEdit_scene);

```

```

PatEdit.setOnAction(new EventHandler<ActionEvent>() {

@Override
public void handle(ActionEvent event) {
    try{
        int E = (Integer.parseInt(PatientEditNumber_TextField.getText()))-1;
        EPatientName_TextField.setText(Patients_array.get(E).GetName());
        EPatientID_TextField.setText(Patients_array.get(E).GetID());
        EPatientAddress_TextField.setText(Patients_array.get(E).GetAddress());
        EPatientPhoneNumber_TextField.setText(Patients_array.get(E).GetNumber());

        if(Patients_array.get(E).GetGender()==true){
            EMaleRB.setSelected(true);

        }

        else{
            EFemaleRB.setSelected(true);
        }

        EPatientSymptoms_TextField.setText(Patients_array.get(E).GetSymptoms());

        if(Patients_array.get(E).GetPaymentMethod()==true){

            ECashRB.setSelected(true);
        }
        else{
            EVisaRB.setSelected(true);
        }

        if(Patients_array.get(E).GetIsEmergency()==true){

            EPatientEmergencyRB.setSelected(true);

```

```

    }
else{
    EPatientNormalRB.setSelected(true);
}

EPatientDiagnosis_TextField.setText(Patients_array.get(E).GetDiagnosis());

PatientsEdit_Stage.show();
}

        catch(IndexOutOfBoundsException E1){
            System.out.println("Invalid Patient Number!!! Check their number from printing");
        }
        catch(Exception E2){
            System.out.println("Invalid Type!!! Enter Patient number");

        }

    }

});

PatEdit_Submit_btn.setOnAction(new EventHandler<ActionEvent>() {

@Override
public void handle(ActionEvent event) {
    int E = (Integer.parseInt(PatientEditNumber_TextField.getText()))-1;

    String EPatientName = (EPatientName_TextField.getText());
    String EPatientID = (EPatientID_TextField.getText());
    String EPatientAddress = (EPatientAddress_TextField.getText());
    String EPatientSymptoms = (EPatientSymptoms_TextField.getText());
    String EPatientPhoneNumber = (EPatientPhoneNumber_TextField.getText());
    String EPatientDiagnosis = (EPatientDiagnosis_TextField.getText());
    Patients_array.get(E).SetName(EPatientName);
    Patients_array.get(E).SetID(EPatientID);
    Patients_array.get(E).SetAddress(EPatientAddress);
    Patients_array.get(E).SetSymptoms(EPatientSymptoms);
    Patients_array.get(E).SetNumber(EPatientSymptoms);

```

```

Patients_array.get(E).SetDiagnosis(EPatientDiagnosis);

if(EMaleRB.isSelected()==true){
    Patients_array.get(E).SetGender(true);

}
else{
    Patients_array.get(E).SetGender(false);
}

if( ECashRB.isSelected()==true ){

    Patients_array.get(E).SetPaymentMethod(true);
}
else{
    Patients_array.get(E).SetPaymentMethod(false);
}

if(EPatientEmergencyRB.isSelected()==true ){

    Patients_array.get(E).SetIsEmergency(true);

}
else{
    Patients_array.get(E).SetIsEmergency(false);
}

System.out.println("Patient Edited Successfully" );
System.out.println("////////////////////////////////////////");

}

```

```

});

    //Appointments
    GridPane App_grid_pane = new GridPane();
    App_grid_pane.setHgap(10);
    App_grid_pane.setVgap(10);
    App_grid_pane.setAlignment(Pos.CENTER);

    App_grid_pane.add(Select_AppOption_label, 0, 0);
    App_grid_pane.add(AppAdd_btn, 0, 1);
    App_grid_pane.add(AppPrint, 4, 1);
    App_grid_pane.add(AppDelete, 6, 1);
    App_grid_pane.add(AppointmentDeleteNumber_TextField, 6, 2);

    Stage Appointments_Stage = new Stage();
    Appointments_Stage.setTitle("Appointments");
    Scene Appointment_scene = new Scene(App_grid_pane, 1000, 600);
    Appointments_Stage.setScene(Appointment_scene);

    Appointments_btn.setOnAction(new EventHandler<ActionEvent>() {

        @Override
        public void handle(ActionEvent event) {

            Appointments_Stage.show();

        }
    });

    //Appointment Add
    GridPane AppAdd_grid_pane = new GridPane();
    AppAdd_grid_pane.setHgap(10);
    AppAdd_grid_pane.setVgap(10);
    AppAdd_grid_pane.setAlignment(Pos.CENTER);

    AppAdd_grid_pane.add(AppointmentDoctorNo_label, 0, 0);

```

```

AppAdd_grid_pane.add(AppointmentDoctorNo_TextField, 1, 0);
AppAdd_grid_pane.add(AppointmentPatientNo_label, 0, 1);
AppAdd_grid_pane.add(AppointmentPatientNo_TextField, 1, 1);
AppAdd_grid_pane.add(AppointmentDate_label, 0, 2);
AppAdd_grid_pane.add(AppointmentDate_TextField, 1, 2);

```

```

AppAdd_grid_pane.add(AppCreate_btn, 1, 9);

```

```

Stage AppointmentsAdd_Stage = new Stage();
AppointmentsAdd_Stage.setTitle("Add Appointment");
Scene AppointmentAdd_scene = new Scene(AppAdd_grid_pane,500,700);
AppointmentsAdd_Stage.setScene(AppointmentAdd_scene);

```

```

AppAdd_btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {

        AppointmentsAdd_Stage.show();

    }
});

```

```

AppCreate_btn.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {
        try{
            int AppointmentDoctor = Integer.parseInt(AppointmentDoctorNo_TextField.getText())-1;
            int AppointmentPatient = Integer.parseInt(AppointmentPatientNo_TextField.getText())-1;
            String AppointmentDate = (AppointmentDate_TextField.getText());

```



```

        Appointments_array.add(new
Appointment(AppointmentDate,Patients_array.get(AppointmentPatient),Doctors_array.get(Appointment
Doctor)));

        System.out.println("Appointment Added Successfully" );
    }
    catch(IndexOutOfBoundsException E1){
        System.out.println("Invalid Patient or Doctor Number!!! Check their number from printing");
    }
    catch(Exception E2){
        System.out.println("Invalid Type!!! Enter Patient and Doctor number");

    }

}

});

```

```

        //Appointments Print
AppPrint.setOnAction(new EventHandler<ActionEvent>() {

@Override
public void handle(ActionEvent event) {

        System.out.println("Total Number of Appointments in Hospital =" +
Appointments_array.size() +" Appointments");

        System.out.println(".....");

        for (int D = 0; D < Appointments_array.size(); D++){

            int n=D+1;

            System.out.println("Appointment " +n+" Doctor Name is "+
(Appointments_array.get(D)).GetDoctorOfApp().GetName());

            System.out.println("Appointment " +n+" Patient Name is "+
(Appointments_array.get(D)).GetPatientOfApp().GetName());

            System.out.println("Appointment " +n+" Date is
"+(Appointments_array.get(D)).GetDateOfApp());

            System.out.println(".....");

        }

        System.out.println("Appointments Printed Successfully" );

        System.out.println("////////////////////////////////////////");

    }

});

//Appointment Delete

```

```

AppDelete.setOnAction(new EventHandler<ActionEvent>() {

    @Override
    public void handle(ActionEvent event) {
        try{
            int AppointmentDeleteNumber =
(Integer.parseInt(AppointmentDeleteNumber_TextField.getText()))-1;
            Appointments_array.remove(AppointmentDeleteNumber);
            System.out.println("Appointment Deleted Successfully" );
            System.out.println("////////////////////////////////////////");
        }
        catch(IndexOutOfBoundsException E1){
            System.out.println("Invalid Appointment Number!!! Check their number from printing");
        }
        catch(Exception E2){
            System.out.println("Invalid Type!!! Enter Appointment number");
        }

    }

});

}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    launch(args);
}

}

```

Doctor:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital;

/**
 *
 * @author Mohamed
 */
public class Doctor {
    private String Name;
    private String ID;
    private String Address;
    private String Number;
    private String Speciality;
    private static int DoctorNo=0;
    private static String[] DoctorArray ;

    public Doctor(){

    }
    public Doctor(String Name, String ID,String Address, String Number, String Speciality){
        this.Name = Name;
        this.ID = ID;
        this.Address = Address;
        this.Speciality = Speciality;
        this.Number = Number;

        DoctorArray[DoctorNo]= new String(this.Name);
        DoctorNo=DoctorNo+1;
    }
    public void SetName(String Name){
        this.Name = Name;
    }
    public void SetAddress(String Address){
        this.Address = Address;
    }
    public void SetID(String ID){
        this.ID = ID;
    }
    public void SetNumber(String Number){
        this.Number = Number;
    }
    public void SetSpeciality(String Speciality){
        this.Speciality = Speciality;
    }
    public String GetName(){
        return Name ;
    }
    public String GetID(){
        return ID ;
    }
    public String GetAddress(){
        return Address ;
    }
    public String GetNumber(){
        return Number ;
    }
    public String getSpeciality(){
        return Speciality ;
    }
    public static void PrintDoctors(){
```

```

        System.out.println("Number of Doctors In Hospital : " + DoctorNo);
        for(int i=1; i <= DoctorNo; i++){

            System.out.println("Doctor Name " + (i) + " is " + DoctorArray[i-1]);

        }
        System.out.println("////////////////////////////////////////");
    }
    public static void Delete(int D){

        DoctorArray[D-1]= null;

        for(int i=(D);(i<DoctorNo);i++){

            DoctorArray[i-1]=DoctorArray[i];

        }
        DoctorNo=DoctorNo-1;

    }
}

```

Patient:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital;

/**
 *
 * @author Mohamed
 */
public abstract class Patient {
    private String Name;
    private String ID;
    private String Address;
    private String Number;
    private boolean Gender;
    private String Symptoms;
    private boolean PaymentMethod;
    private String Diagnosis;
    private boolean IsEmergency;
    private static int NormalPatientNo=0;
    private static String[] NormalPatientsArray = new String[100];
    private static int EmergencyPatientNo=0;
    private static String[] EmergencyPatientsArray = new String[100];

    public Patient(){}

    public Patient(String Name, String ID,String Address, String Number,
        boolean Gender, String Symptoms, boolean PaymentMethod,
        String Diagnosis,boolean IsEmergency){
        this.Name = Name;
        this.ID = ID;
        this.Address = Address;
        this.Number = Number;
        this.Diagnosis = Diagnosis;
        this.Gender = Gender;
        this.Symptoms = Symptoms;
        this.IsEmergency = IsEmergency;
        this.PaymentMethod = PaymentMethod;
        if(IsEmergency==false )
        {
            NormalPatientsArray[NormalPatientNo]= this.Name;
            NormalPatientNo=NormalPatientNo+1;
        }
        else if (IsEmergency==true){
            EmergencyPatientsArray[EmergencyPatientNo]= this.Name;
            EmergencyPatientNo=EmergencyPatientNo+1;
        }
    }

}

public void SetName(String Name){
    this.Name = Name;
}

public void SetAddress(String Address){
    this.Address = Address;
}

public void SetID(String ID){
    this.ID = ID;
}

public void SetNumber(String Number){
    this.Number = Number;
}
}
```

```

public void SetGender(boolean Gender){
    this.Gender = Gender;
}
public void SetSymptoms(String Symptoms){
    this.Symptoms = Symptoms;
}
public void SetPaymentMethod(boolean PaymentMethod){
    this.PaymentMethod = PaymentMethod;
}
public void SetDiagnosis(String Diagnosis){
    this.Diagnosis = Diagnosis;
}
public void SetIsEmergency(boolean IsEmergency){
    this.IsEmergency = IsEmergency;
}

public String GetName(){
    return Name ;
}
public String GetID(){
    return ID ;
}
public String GetAddress(){
    return Address ;
}
public String GetNumber(){
    return Number ;
}
public boolean GetGender(){
    return Gender ;
}
public String GetSymptoms(){
    return Symptoms ;
}
public boolean GetPaymentMethod(){
    return PaymentMethod ;
}
public String GetDiagnosis(){
    return Diagnosis ;
}
public boolean GetIsEmergency(){
    return IsEmergency ;
}

public static void PrintPatients(){ // abstract

    System.out.println("Number of Normal Patients In Hospital : " + NormalPatientNo);
    for(int i=1; i <= NormalPatientNo; i++){

        System.out.println("Patient Name " + (i) + " is " + NormalPatientsArray[i-1]);

    }

    System.out.println("////////////////////////////////////////");

    System.out.println("Number of Emergency Patients In Hospital : " + EmergencyPatientNo);
    for(int i=1; i <= EmergencyPatientNo; i++){

        System.out.println("Patient Name " + (i) + " is " + EmergencyPatientsArray[i-1]);

    }

    System.out.println("////////////////////////////////////////");

}
}

```

```

public static void Delete(int D, boolean EmergencyOrNormal){
    if(EmergencyOrNormal ==false)
    {

        NormalPatientsArray[D-1]= null;
        for(int i=(D);(i<NormalPatientNo);i++){
            NormalPatientsArray[i-1]=NormalPatientsArray[i];

        }
        NormalPatientNo=NormalPatientNo-1;
    }

    else if (EmergencyOrNormal ==true) {
        EmergencyPatientsArray[D-1]= null;
        for(int i=(D);(i<EmergencyPatientNo);i++){
            EmergencyPatientsArray[i-1]=EmergencyPatientsArray[i];

        }
        EmergencyPatientNo=EmergencyPatientNo-1;
    }
}
}

```

Emergency Patient:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital;

/**
 *
 * @author dell
 */
public class EmergencyPatient extends Patient {

    public EmergencyPatient(){
        super();
    }

    public EmergencyPatient(String Name, String ID,String Address, String Number, boolean Gender,
String Symptoms, boolean PaymentMethod,String Diagnosis){
        super(Name, ID, Address, Number, Gender, Symptoms, PaymentMethod, Diagnosis, true) ;
    }

}
```

Normal Patient:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital;

/**
 *
 * @author Mohamed
 */
public class NormalPatient extends Patient{

    public NormalPatient(){
        super();
    }

    public NormalPatient(String Name, String ID,String Address, String Number, boolean Gender, String
Symptoms, boolean PaymentMethod,String Diagnosis){
        super(Name, ID, Address, Number, Gender, Symptoms, PaymentMethod, Diagnosis, false) ;
    }

}
```


Appointment:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital;

/**
 *
 * @author dell
 */
public class Appointment {
    private String DateOfApp;
    private Patient PatientOfApp;
    private Doctor DoctorOfApp;
    private static int AppointmentNo= 0;
    private static String[] DateOfAppArray = new String[100];
    private static String[] AppPatientsArray = new String[100];
    private static String[] AppDoctorsArray = new String[100];

    public Appointment(String DateOfApp,Patient PatientOfApp,Doctor DoctorOfApp) {

        this.DateOfApp=DateOfApp;
        this.PatientOfApp=PatientOfApp;
        this.DoctorOfApp=DoctorOfApp;
        DateOfAppArray[AppointmentNo]=this.DateOfApp;
        AppPatientsArray[AppointmentNo]=this.PatientOfApp.GetName();
        AppDoctorsArray[AppointmentNo]=this.DoctorOfApp.GetName();
        AppointmentNo=AppointmentNo+1;
    }

    public void SetDateOfApp(String DateOfApp){
        this.DateOfApp=DateOfApp;
    }

    public void SetPatientOfApp(Patient PatientOfApp){
        this.PatientOfApp=PatientOfApp;
    }

    public void SetDoctorOfApp(Doctor DoctorOfApp){
        this.DoctorOfApp=DoctorOfApp;
    }

    public String GetDateOfApp(){
        return(this.DateOfApp);
    }

    public Patient GetPatientOfApp(){
        return(this.PatientOfApp);
    }

    public Doctor GetDoctorOfApp(){
        return(this.DoctorOfApp);
    }

    public void PrintAppointment(){

        System.out.println("Date is " + DateOfApp);
        System.out.println("Patient Name is " + PatientOfApp.GetName());
    }
}
```

```

        System.out.println("Doctor Name is  " + DoctorOfApp.GetName());
        System.out.println("////////////////////////////////////////");
    }

    public static void AllAppointmentsPrint(){

        System.out.println("Number of Appointments is  " + AppointmentNo);
        System.out.println("////////////////////////////////////////");
        for(int i=1;i<= AppointmentNo; i++){
            System.out.println("Appointment Number:  " + i);
            System.out.println("Date is  " + DateOfAppArray[i-1]);
            System.out.println("Patient Name is  " + AppPatientsArray[i-1]);
            System.out.println("Doctor Name is  " + AppDoctorsArray[i-1]);
            System.out.println("////////////////////////////////////////");
        }
    }

}

}

```

Gui Layout:

The image displays two overlapping windows from a graphical user interface. The top window, titled 'Hospital', contains a 'Select Field :' label and three buttons: 'Doctors' (highlighted with a blue border), 'Patients', and 'Appointments'. The bottom window, titled 'Doctors', contains a 'Select Option :' label and four buttons: 'Add Doctor' (highlighted with a blue border), 'Print all Doctors', 'Edit Doctor', and 'Delete Doctor'. Below the 'Edit Doctor' and 'Delete Doctor' buttons are two empty text input fields.

Hospital Window:

Select Field :

Doctors Patients Appointments

Doctors Window:

Select Option :

Add Doctor Print all Doctors Edit Doctor Delete Doctor

[Empty Text Field] [Empty Text Field]

Doctor Name

Doctor ID

Doctor Address

Doctor Speciality

Doctor Phone Number

Add the Doctor

Select Option :

Add Patient

Print all Patients

Edit Patient

Delete Patient

Patient Name

Patient ID

Patient Address

Patient Symptoms

Patient Phone Number

Patient Diagnosis

☐ male

☐ female

☐ Cash Payment

☐ Visa Payment

☐ Emergency Condition

☐ Normal Condition

Add the Patient

Select Option :

Add Appointment

Print all Appointments

Delete Appointment

Appointment Doctor Number

Appointment Patient Number

Appointment Date

Add the Appointment