Havens Light is Our Guide

Rajshahi University of Engineering & Technology

Course Title

Object Oriented Programming

Course No: ECE 1204

Project-01

Project name:

Building a Hospital Management System using Java Swing.

DATE: 15/01/2024

Submitted To	Submitted by
Md. Nahiduzzaman Assistant Professor. Department of Electrical and Computer Engineering, RUET	Nafisha Nawar Roll: 2110026 Reg: 1080/2021-2022 Department of Electrical and Computer Engineering, RUET

1. Description:

A Hospital Management System (HMS) is a comprehensive software solution designed to streamline and automate various administrative and clinical tasks within a hospital or healthcare facility. Its primary purpose is to improve the efficiency, accuracy, and overall quality of healthcare services. Overall, a Hospital Management System is designed to enhance patient care, optimize resource utilization, reduce administrative burdens, and ensure compliance with healthcare regulations. It plays a critical role in modern healthcare facilities, from small clinics to large hospitals, by improving the overall efficiency and quality of healthcare service.

Designing the user interface (UI) for a hospital management system in Java Swing involves creating various frames, panels, buttons, text fields, and other GUI components. Some of the key components and properties that would be used to build the UI for the project are described below:

- Main Application Window
- Login Screen
- Home page for all features.
- Add new patient record screen.
- Add new diagnosis information screen.
- Full history of patient screen.
- Hospital information screen.
- Dialog Boxes.
- Icons and Images.
- Simple database works to store information.

2. Tools and Technologies

To create this project, usage of following tools and technologies would be:

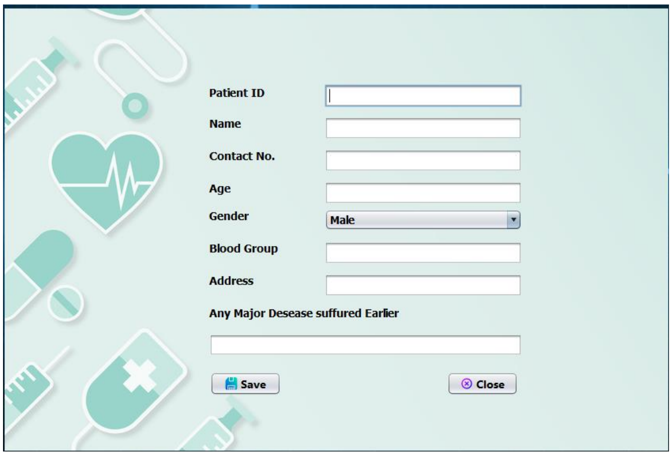
- Java for coding.
- Java Swing for building the GUI.
- An Integrated Development Environment (IDE) such as: NetBeans.
- MySQL database system.

3. Project Building Procedure:

The project has been built with java swing using NetBeans IDE. The drag and drop method have been followed while building the application. Basically, this application helps the administration to organize and maintain patient data and offers them the liberty to review their history and diagnosis and use that information for their benefits.

Firstly, a login page has been built with a dialog box for username and password. The user with the correct username and password can login into the system by simply pressing the login button. But any wrong input for either of those fields will show an error massage of "Incorrect Username or Password". The user can close the application by pressing the close button which will show another confirmation message "Are you sure to close the application?". By pressing the yes button, the user can close the application or by pressing no, user can go back the previous page.

Next, after login brings to the home page which shows all the features of the system in one page. There are buttons and each of them performs specific tasks. Such as: the add new patient record button will let the admin to add a new patient's information into the database and that button will open a form in a new window.



Patient ID

Name

Contact No.

Age

Gender: Male

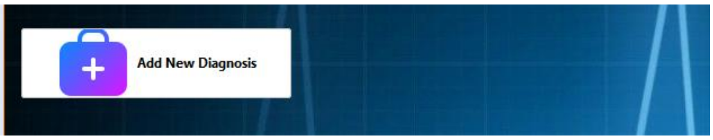
Blood Group

Address

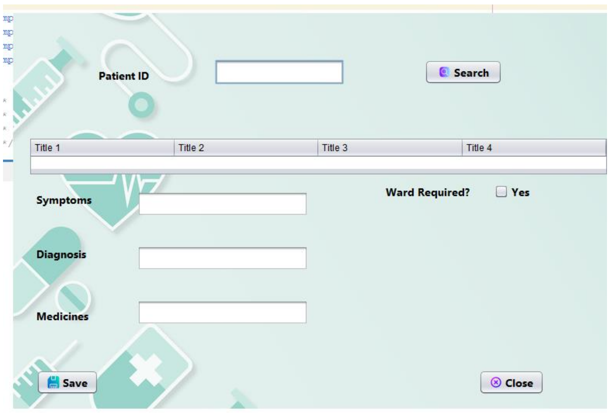
Any Major Desease suffured Earlier

The form contains text fields to acquire basic information about the new patient, such as: name, contact no., age, gender, blood group, address, history of any previous major diseases. jTextField have been used for all the descriptive fields and a combo box has been used for choosing between genders. Again, there's a save button to store all information into the database and in case of any wrong or empty input, it will show a warning message of "invalid input" and will not save anything unless given the right input. Also, there's a close button to close the form and going back to the previous home menu.

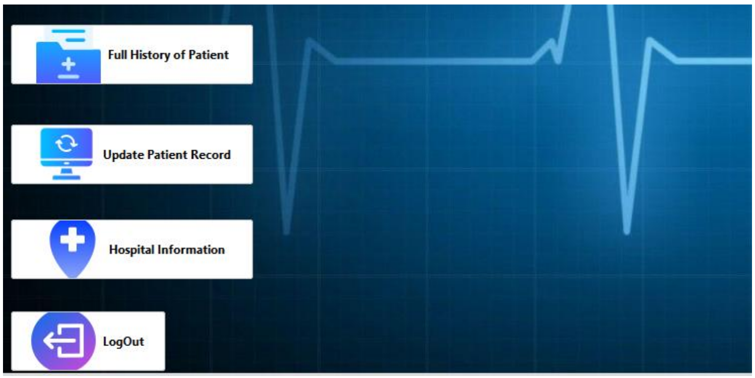
In the next page the new diagnosis information about the patient can be added. As the user progresses with the application, the home page shows another button called “Add New Diagnosis”



This button will take the user to another window which will show a form. By filling the form with necessary information, the user can progress by clicking save and close to exit the page.

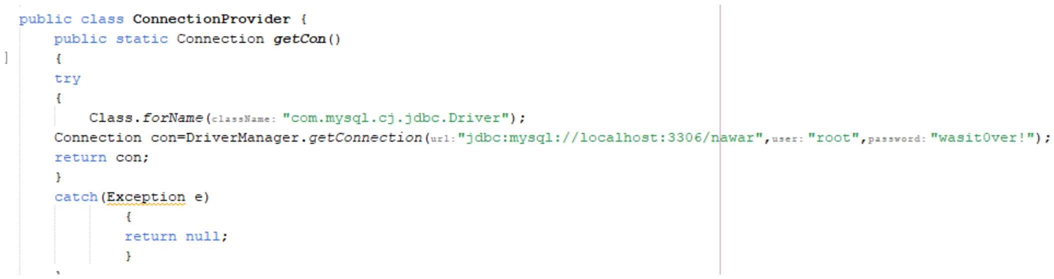


In the same manner, the rest of the buttons perform their consecutive task according to their name. it takes the user to another window where it provides a new form to take necessary input from the patient and saves it in the connected database for future use. In the end, user can logout from the application.



The MySQL database had been connected, to the application to store patient’s information. A jar file of MySQL JDBC driver had been added to the java library and a connection provider

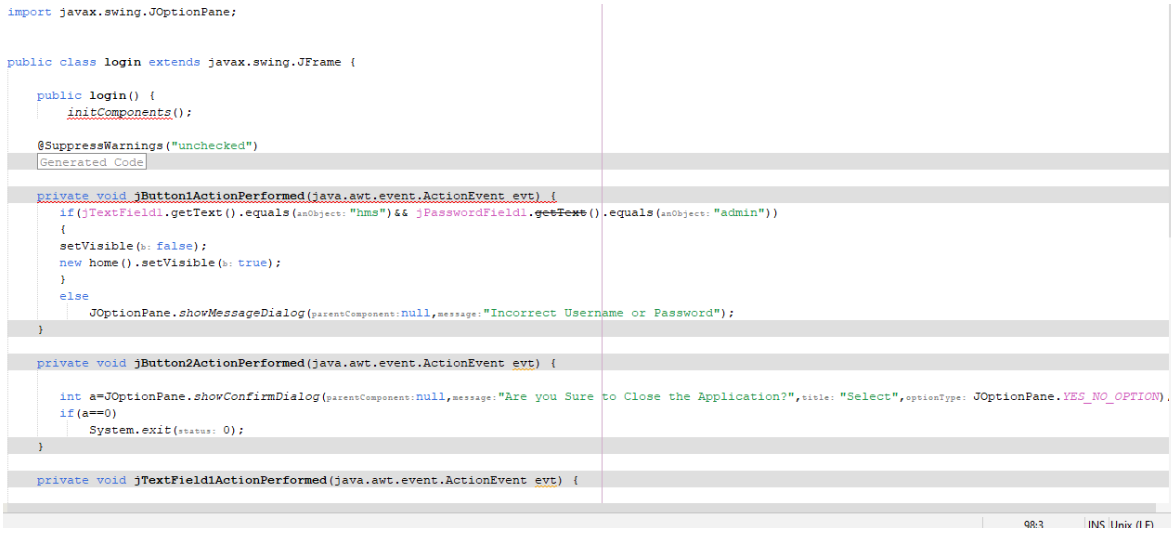
.java class had been created. That file along with the database driver had been imported to every class created that needed database backup.



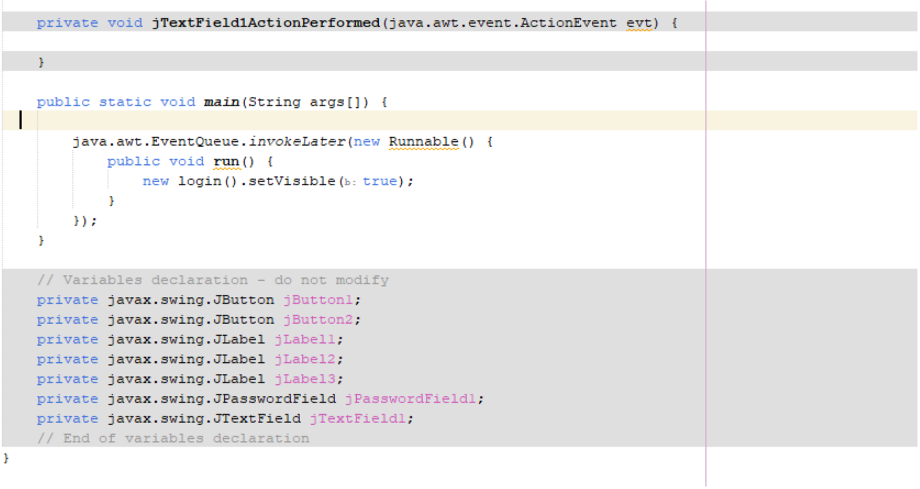
```
public class ConnectionProvider {
    public static Connection getCon()
    {
    try
    {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
    Connection con=DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/nawar",user: "root",password: "wasit0ver!");
    return con;
    }
    catch(Exception e)
            {
            return null;
            }
```

4. Code:

Login page:

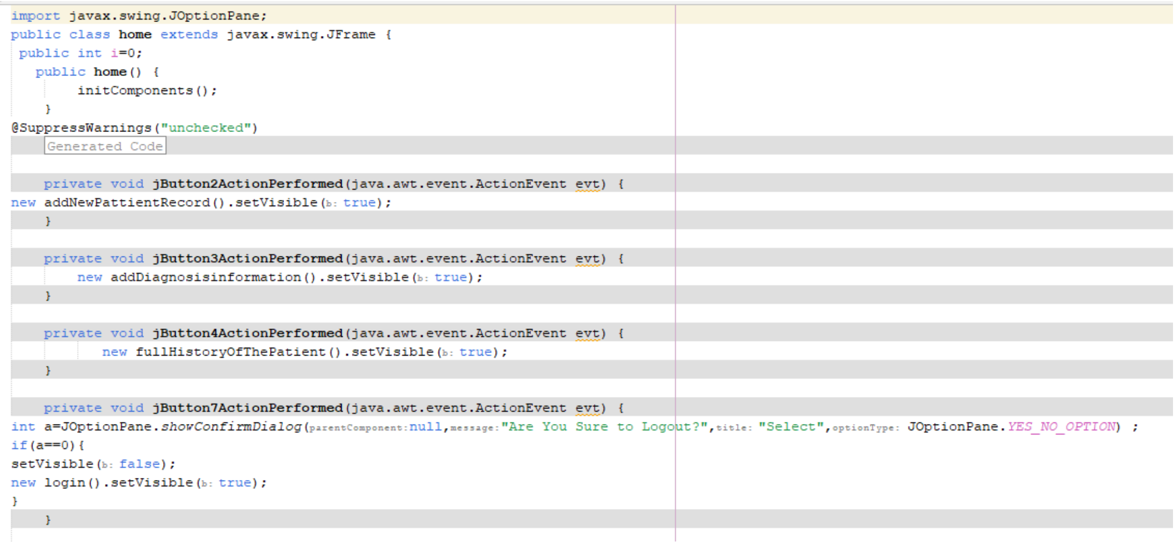


```
import javax.swing.JOptionPane;


public class login extends javax.swing.JFrame {

    public login() {
        initComponents();

    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
       if(jTextField1.getText().equals(anObject: "hms")&& jPasswordField1.~~getText~~().equals(anObject: "admin"))
       {
       setVisible(b: false);
       new home().setVisible(b: true);
       }
       else
           JOptionPane.showMessageDialog(parentComponent:null,message:"Incorrect Username or Password");
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

       int a=JOptionPane.showConfirmDialog(parentComponent:null,message:"Are you Sure to Close the Application?",title: "Select",optionType: JOptionPane.YES_NO_OPTION).
       if(a==0)
           System.exit(status: 0);
    }

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
```

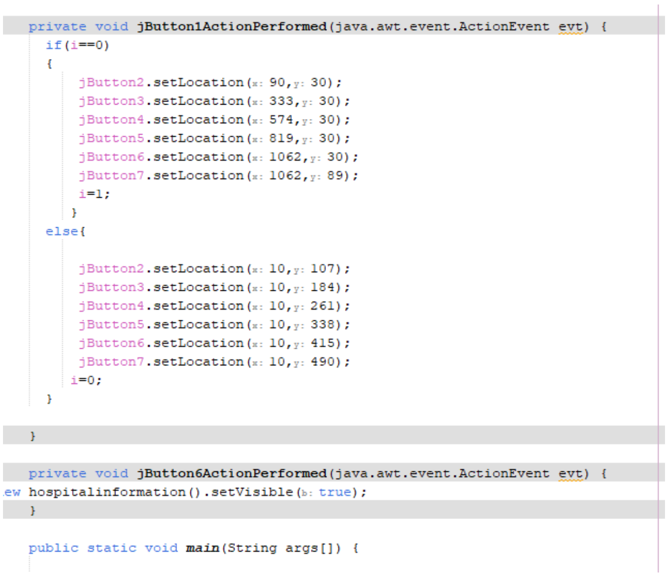


```
    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new login().setVisible(b: true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPasswordField jPasswordField1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

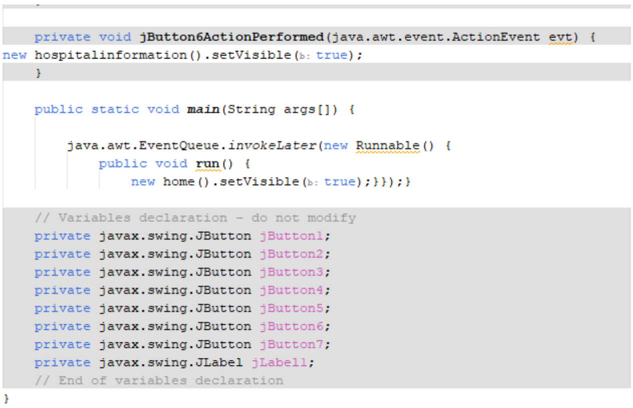
Home Page:



```java
import javax.swing.JOptionPane;
public class home extends javax.swing.JFrame {
 public int i=0;
   public home() {
        initComponents();
    }
@SuppressWarnings("unchecked")
    Generated Code

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
new addNewPattientRecord().setVisible(b: true);
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        new addDiagnosisinformation().setVisible(b: true);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
           new fullHistoryOfThePatient().setVisible(b: true);
    }

    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
int a=JOptionPane.showConfirmDialog(parentComponent:null,message:"Are You Sure to Logout?",title: "Select",optionType: JOptionPane.YES_NO_OPTION) ;
if(a==0){
setVisible(b: false);
new login().setVisible(b: true);
}
    }
```

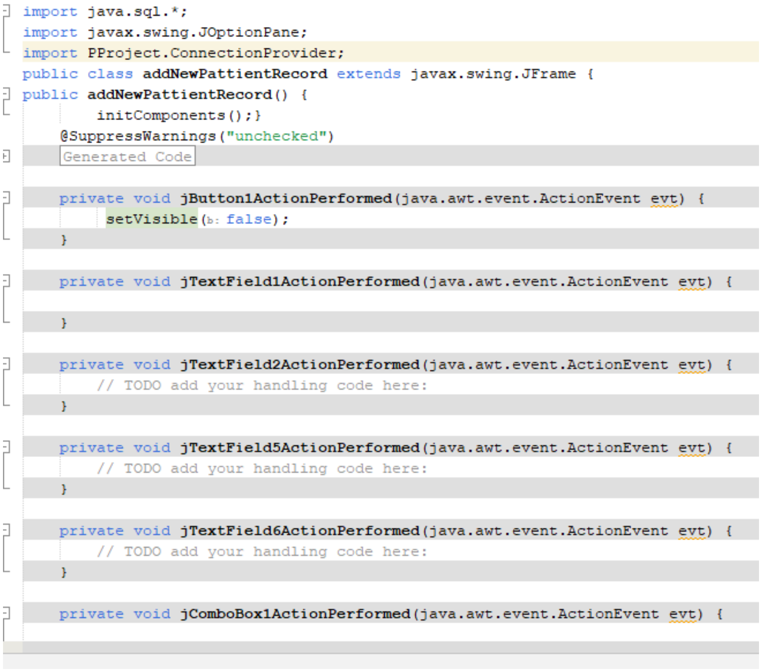


```java
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      if(i==0)
      {
          jButton2.setLocation(x: 90,y: 30);
          jButton3.setLocation(x: 333,y: 30);
          jButton4.setLocation(x: 574,y: 30);
          jButton5.setLocation(x: 819,y: 30);
          jButton6.setLocation(x: 1062,y: 30);
          jButton7.setLocation(x: 1062,y: 89);
          i=1;
         }
      else{

          jButton2.setLocation(x: 10,y: 107);
          jButton3.setLocation(x: 10,y: 184);
          jButton4.setLocation(x: 10,y: 261);
          jButton5.setLocation(x: 10,y: 338);
          jButton6.setLocation(x: 10,y: 415);
          jButton7.setLocation(x: 10,y: 490);
         i=0;
      }

    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
ew hospitalinformation().setVisible(b: true);
    }

    public static void main(String args[]) {
```

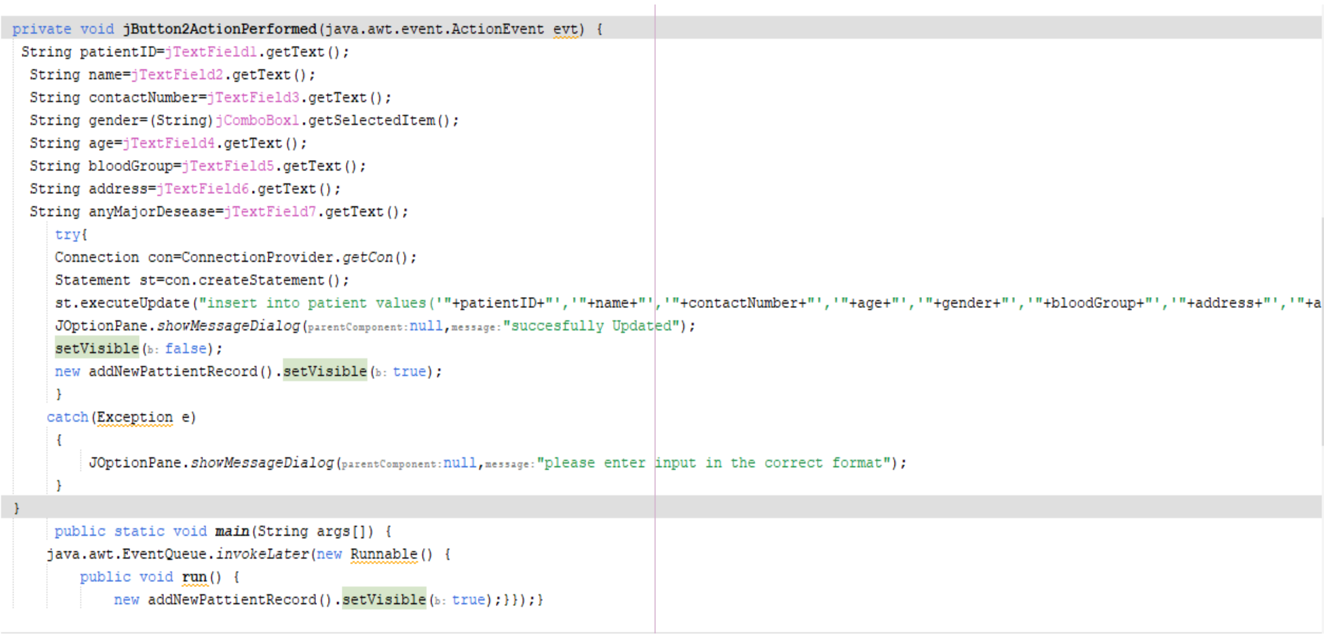


```java
    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
new hospitalinformation().setVisible(b: true);
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new home().setVisible(b: true);}});}

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

Add New Patient Information:

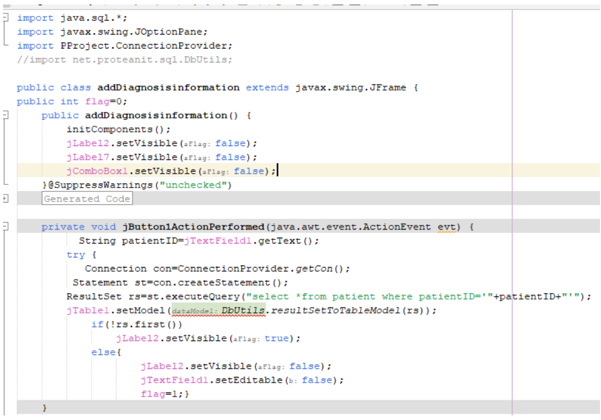


```
import java.sql.*;
import javax.swing.JOptionPane;
import PProject.ConnectionProvider;
public class addNewPattientRecord extends javax.swing.JFrame {
public addNewPattientRecord() {
        initComponents();}
    @SuppressWarnings("unchecked")
Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
          setVisible(b: false);
    }

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jTextField5ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jTextField6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
```

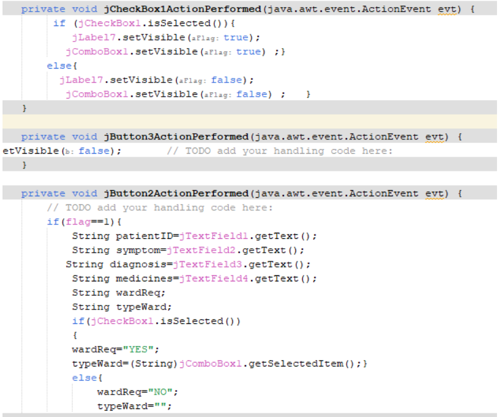


```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
 String patientID=jTextField1.getText();
  String name=jTextField2.getText();
  String contactNumber=jTextField3.getText();
  String gender=(String)jComboBox1.getSelectedItem();
  String age=jTextField4.getText();
  String bloodGroup=jTextField5.getText();
  String address=jTextField6.getText();
  String anyMajorDesease=jTextField7.getText();
     try{
     Connection con=ConnectionProvider.getCon();
     Statement st=con.createStatement();
     st.executeUpdate("insert into patient values('"+patientID+"','"+name+"','"+contactNumber+"','"+age+"','"+gender+"','"+bloodGroup+"','"+address+"','"+a
     JOptionPane.showMessageDialog(parentComponent:null,message:"succesfully Updated");
     setVisible(b: false);
     new addNewPattientRecord().setVisible(b: true);
     }
    catch(Exception e)
     {
         JOptionPane.showMessageDialog(parentComponent:null,message:"please enter input in the correct format");
     }
}
     public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new addNewPattientRecord().setVisible(b: true);}});}
```

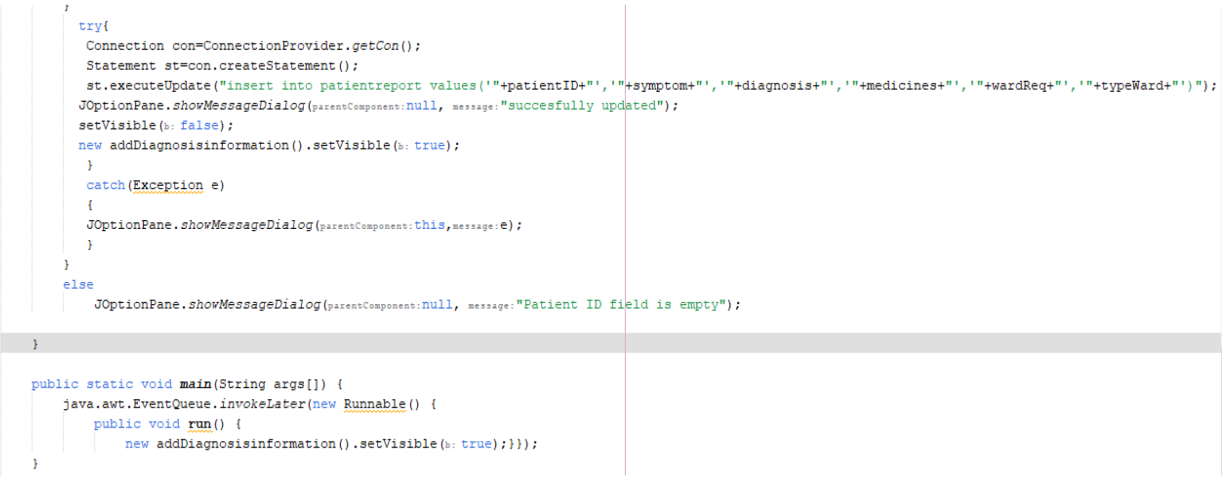
Add new Diagnosis Information:



```
import java.sql.*;
import javax.swing.JOptionPane;
import PProject.ConnectionProvider;
//import net.proteanit.sql.DbUtils;

public class addDiagnosisinformation extends javax.swing.JFrame {
public int flag=0;
    public addDiagnosisinformation() {
        initComponents();
        jLabel2.setVisible(aFlag: false);
        jLabel7.setVisible(aFlag: false);
        jComboBox1.setVisible(aFlag: false);
    }@SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
          String patientID=jTextField1.getText();
        try {
           Connection con=ConnectionProvider.getCon();
         Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from patient where patientID='"+patientID+"'");
        jTable1.setModel(dataModel: DbUtils.resultSetToTableModel(rs));
            if(!rs.first())
                jLabel2.setVisible(aFlag: true);
            else{
                    jLabel2.setVisible(aFlag: false);
                    jTextField1.setEditable(b: false);
                    flag=1;}
    }
```

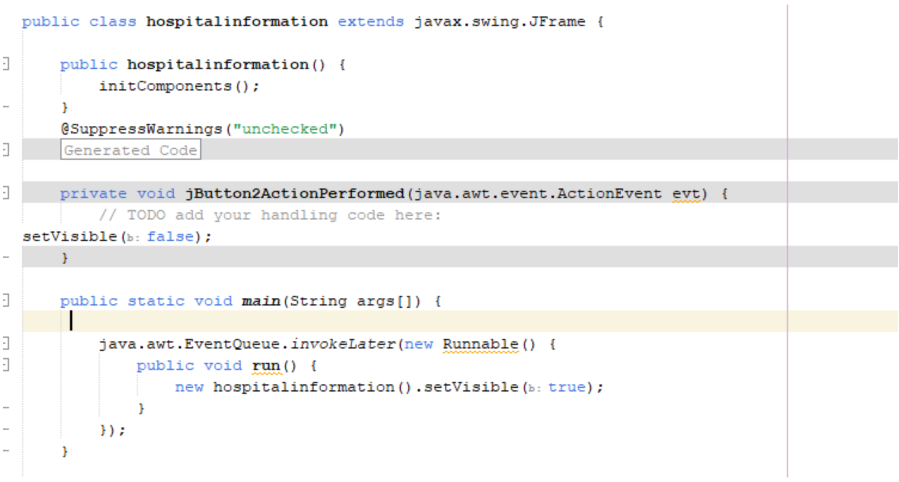


```
    private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) {
         if (jCheckBox1.isSelected()){
            jLabel7.setVisible(aFlag: true);
           jComboBox1.setVisible(aFlag: true) ;}
        else{
          jLabel7.setVisible(aFlag: false);
           jComboBox1.setVisible(aFlag: false) ;    }
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
etVisible(b: false);        // TODO add your handling code here:
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if(flag==1){
            String patientID=jTextField1.getText();
            String symptom=jTextField2.getText();
          String diagnosis=jTextField3.getText();
            String medicines=jTextField4.getText();
            String wardReq;
            String typeWard;
            if(jCheckBox1.isSelected())
            {
            wardReq="YES";
            typeWard=(String)jComboBox1.getSelectedItem();}
            else{
                wardReq="NO";
                typeWard="";
```



```
        ;
          try{
           Connection con=ConnectionProvider.getCon();
           Statement st=con.createStatement();
           st.executeUpdate("insert into patientreport values('"+patientID+"','"+symptom+"','"+diagnosis+"','"+medicines+"','"+wardReq+"','"+typeWard+"')");
          JOptionPane.showMessageDialog(parentComponent:null, message:"succesfully updated");
          setVisible(b: false);
          new addDiagnosisinformation().setVisible(b: true);
           }
           catch(Exception e)
           {
           JOptionPane.showMessageDialog(parentComponent:this,message:e);
           }
        }
        else
            JOptionPane.showMessageDialog(parentComponent:null, message:"Patient ID field is empty");

    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new addDiagnosisinformation().setVisible(b: true);}});
    }
```

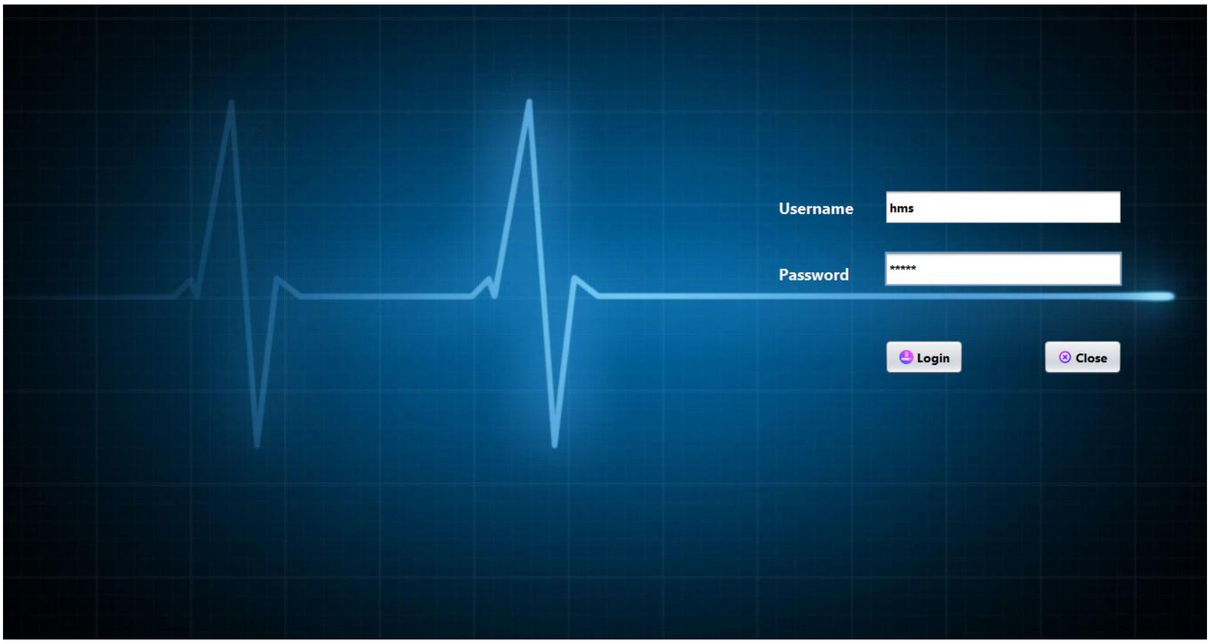
Hospital Information:



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

    public hospitalinformation() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
setVisible(b: false);
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new hospitalinformation().setVisible(b: true);
            }
        });
    }
```

5. Output:

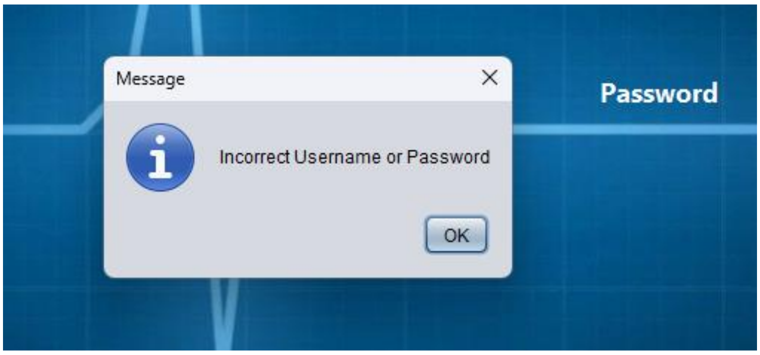
Main application window:



Username hms

Password *****

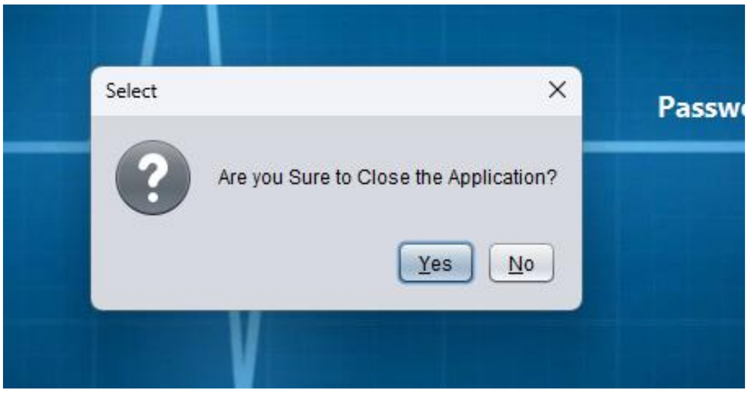
Error message for wrong password input:



Message

Incorrect Username or Password

Warning message for closing the application:

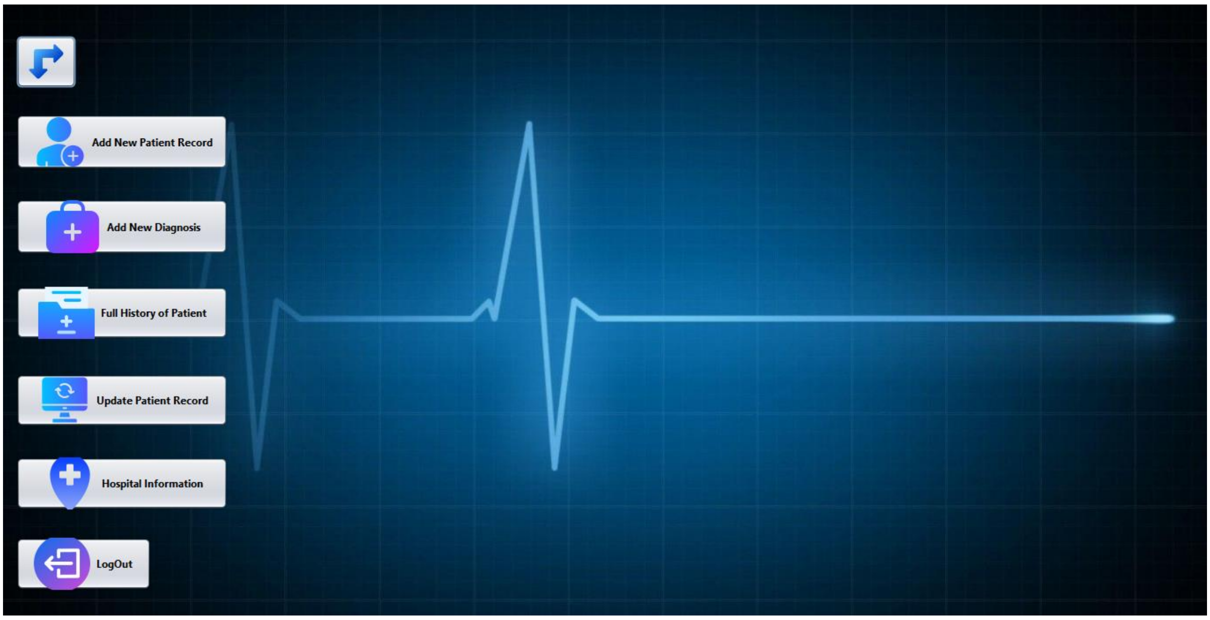


Select

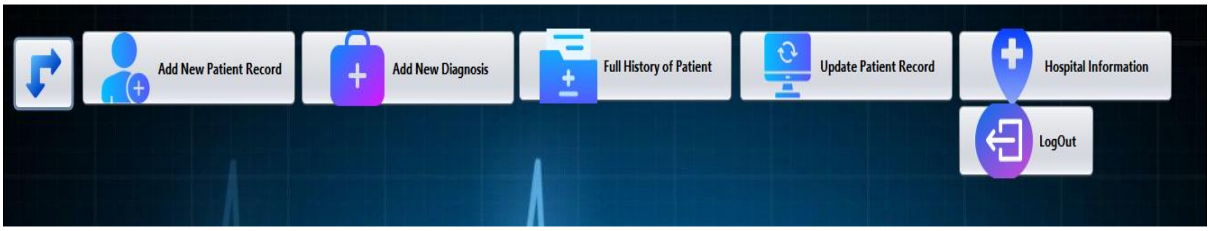
Are you Sure to Close the Application?

Home Page:

Main window for Home Page:

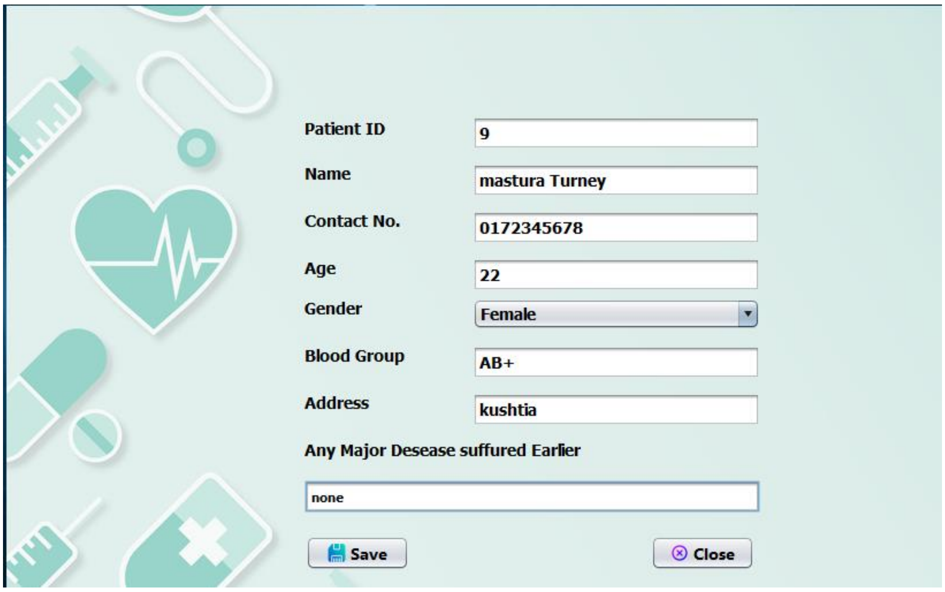


Rearranged positions after pressing the arrow button:



Add new patient record:

Saves the correct input in the database:

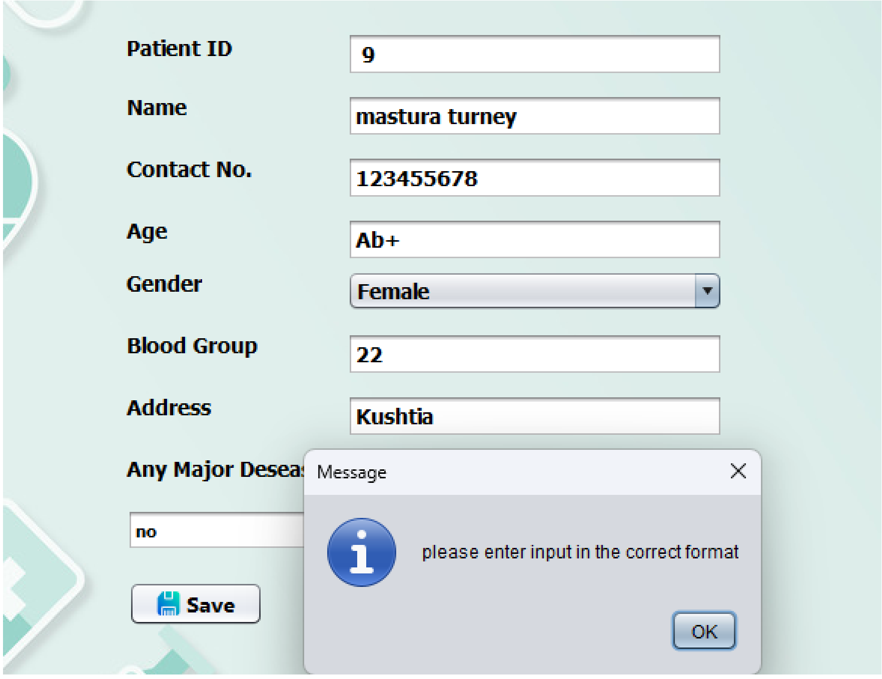


Patient ID	9
Name	mastura Turney
Contact No.	0172345678
Age	22
Gender	Female
Blood Group	AB+
Address	kushtia

Any Major Desease suffured Earlier

none

Shows error message for incorrect input format:



Patient ID	9
Name	mastura turney
Contact No.	123455678
Age	Ab+
Gender	Female
Blood Group	22
Address	Kushtia

Any Major Desea

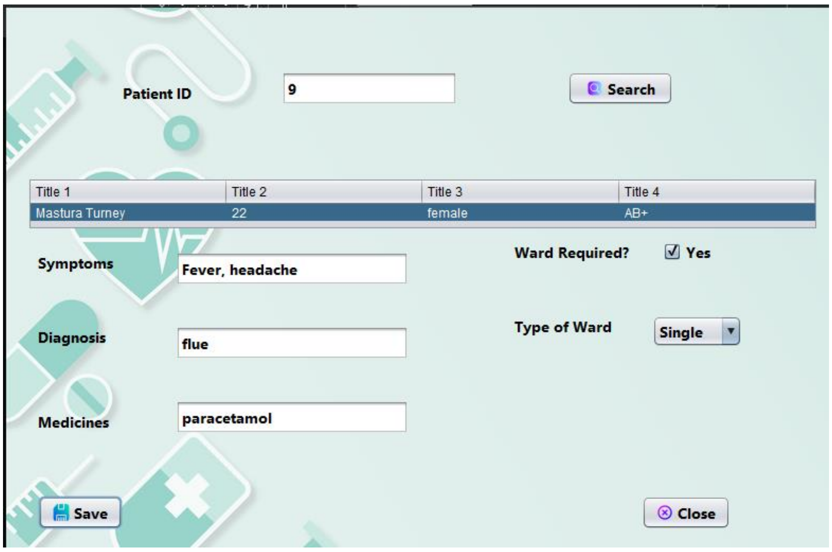
no

Message

please enter input in the correct format

Add new diagnosis information:

Searching with patient ID, name, age, gender, and blood group information showed:



Patient ID: 9

Title 1	Title 2	Title 3	Title 4
Mastura Turney	22	female	AB+

Symptoms: Fever, headache

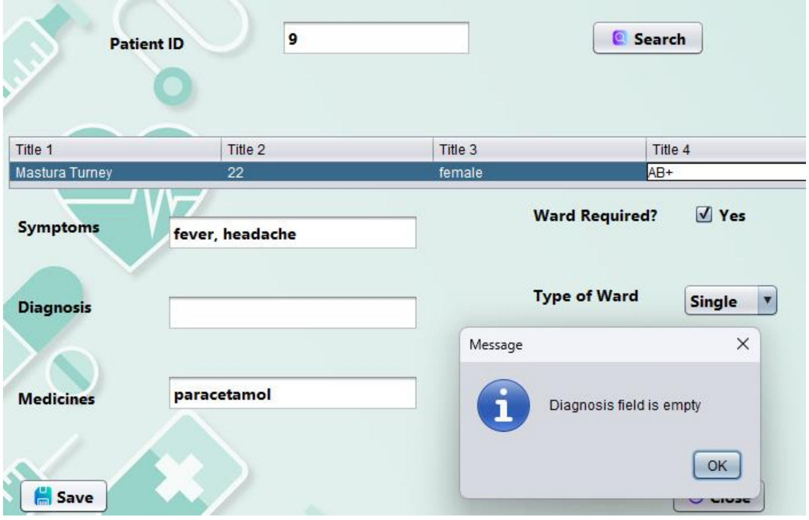
Diagnosis: flue

Medicines: paracetamol

Ward Required? ☑ Yes

Type of Ward: Single

For any empty field input, shows error message:



Patient ID: 9

Title 1	Title 2	Title 3	Title 4
Mastura Turney	22	female	AB+

Symptoms: fever, headache

Diagnosis:

Medicines: paracetamol

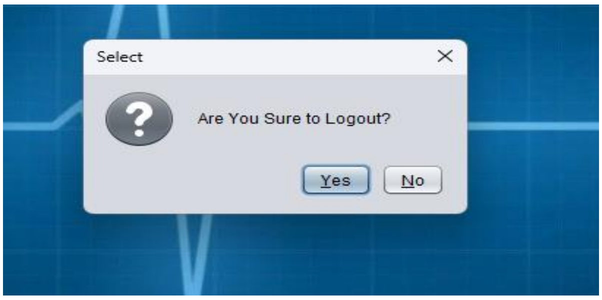
Ward Required? ☑ Yes

Type of Ward: Single

Message

Diagnosis field is empty

In the end user can log out of the system by pressing the logout button in the homepage:



Select

Are You Sure to Logout?

6. Conclusion:

The hospital management system application had been built up upon the purpose of helping the patient user to find their necessary information all in one place and stored properly for future use. It also helps the administration to manage information along with offering a way to stay connected with their patients. Java Swing and GUI application had been used to build this project and the basic swing component’s execution had been reflected in the project. Although the system can run fine with these basic features, there’s still scope for modifying this project with more advanced features in the future.