

# **E-MEDICINE SALES SYSTEM**

*A*

*Report*

*Submitted in partial fulfilment of the  
Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING  
IN  
INFORMATION TECHNOLOGY**

**By**

**MOHAMMED QAWIUDDIN <1602-21-737-033>**

**Under the guidance of Ms B. Leelavathy**



**Department of Information Technology  
Vasavi College of Engineering (Autonomous)  
(Affiliated to Osmania University)  
Ibrahimbagh, Hyderabad-31**

**2022-2023**

## **Abstract:**

The e-medicine sales system is a web-based application that provides a platform for medical suppliers to sell their products to healthcare facilities and individuals. The system allows for easy browsing and searching of available products, online ordering, and secure payment processing. It also features a comprehensive inventory management system that ensures accurate and up-to-date information on product availability, pricing, and delivery. The e-medicine sales system aims to simplify the process of buying and selling medical supplies, making it more convenient for both suppliers and buyers. With its user-friendly interface and efficient functionality, the system helps streamline the healthcare supply chain, ultimately leading to improved patient care.

## **Design requirements:**

### **1.Company**

ATTRIBUTE	DOMAIN	CONSTRAINT
Name	VARCHAR(50)	PRIMARY KEY
Address	VARCHAR(50)	NOT NULL
Phone	INT	NOT NULL

### **2.Users**

ATTRIBUTE	DOMAIN	CONSTRAINT
Name	VARCHAR(50)	NOT NULL
Address	VARCHAR(50)	NOT NULL
Password	VARCHAR(20)	NOT NULL
Id	INT	PRIMARY KEY
Phone	INT	NOT NULL

### **3.Login**

ATTRIBUTE	DOMAIN	CONSTRAINT
Login_id	VARCHAR(50)	NOT NULL

Time	VARCHAR(20)	NOT NULL
Login_date	DATE	NOT NULL
Type	VARCHAR(20)	NOT NULL

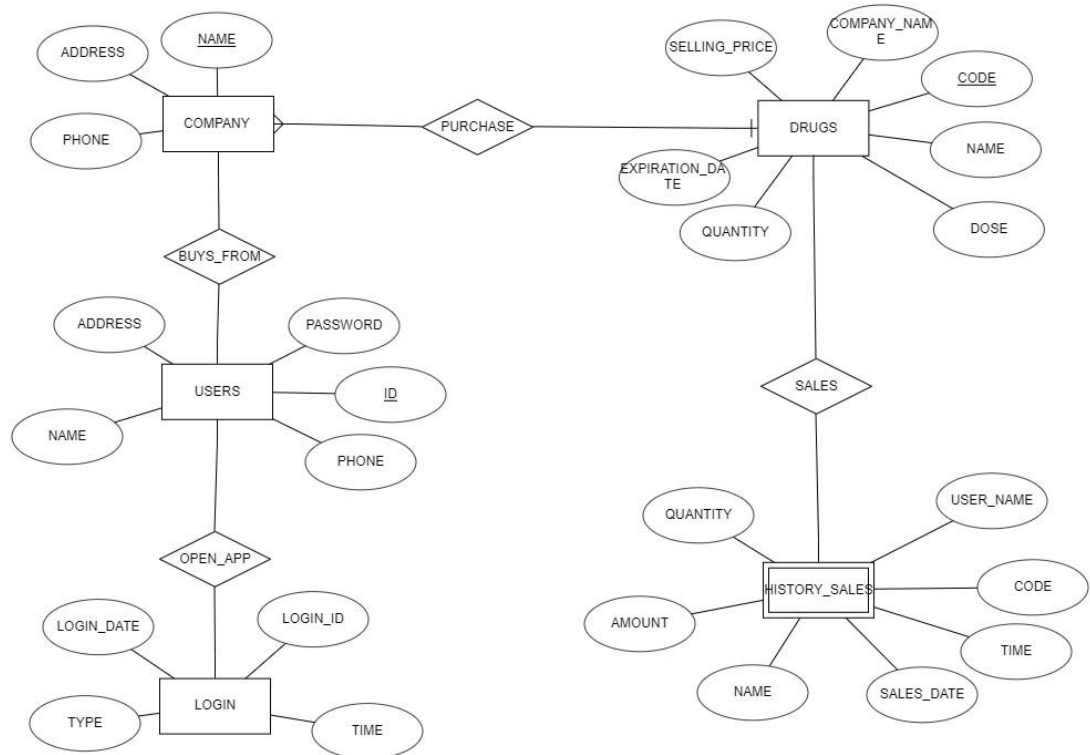
#### 4.Drugs

ATTRIBUTE	DOMAIN	CONSTRAINT
Company_name	VARCHAR(50)	FOREIGN KEY
Code	INT	PRIMARY KEY
Name	VARCHAR(50)	NOT NULL
Dose	VARCHAR(20)	-
Quantity	INT	NOT NULL
Expiration_date	DATE	NOT NULL
Selling_price	INT	NOT NULL

#### 5.History\_sales

ATTRIBUTE	DOMAIN	CONSTRAINT
User_name	VARCHAR(50)	NOT NULL
Code	INT	FOREIGN KEY
Time	VARCHAR(20)	NOT NULL
Sales_date	DATE	NOT NULL
Name	VARCHAR(20)	NOT NULL
Amount	INT	NOT NULL
quantity	INT	NOT NULL

## ER DIAGRAM:



## DDL COMMANDS:

### 1. Creating table for users with constraints:

**Query:** create table users(

2 id INT PRIMARY KEY,

3 password VARCHAR(20) NOT NULL,

4 phone INT NOT NULL,

5 name VARCHAR(50) NOT NULL,

6 address VARCHAR(50) NOT NULL

7 );

```
SQL> create table users(  
2 id INT PRIMARY KEY,  
3 password VARCHAR(20) NOT NULL,  
4 phone INT NOT NULL,  
5 name VARCHAR(50) NOT NULL,  
6 address VARCHAR(50) NOT NULL  
7 );
```

Table created.

## 2.Creating table company with constraints:

Query: CREATE TABLE company(  
2 name VARCHAR(50) PRIMARY KEY,  
3 address VARCHAR(50) NOT NULL,  
4 phone INT NOT NULL);

```
SQL> CREATE TABLE company(  
2 name VARCHAR(50) PRIMARY KEY,  
3 address VARCHAR(50) NOT NULL,  
4 phone INT NOT NULL);
```

Table created.

## 3.Creating table drugs with constraints:

Query: CREATE TABLE drugs(  
2 code INT PRIMARY KEY,  
3 name VARCHAR(50) NOT NULL,  
4 company\_name VARCHAR(50) NOT NULL,  
5 selling\_price INT NOT NULL,  
6 expiration\_date DATE NOT NULL,

7 quantity INT NOT NULL,  
8 dose VARCHAR(20),  
9 FOREIGN KEY (company\_name) REFERENCES company(name));

```
SQL> CREATE TABLE drugs(  
  2 code INT PRIMARY KEY,  
  3 name VARCHAR(50) NOT NULL,  
  4 company_name VARCHAR(50) NOT NULL,  
  5 selling_price INT NOT NULL,  
  6 expiration_date DATE NOT NULL,  
  7 quantity INT NOT NULL,  
  8 dose VARCHAR(20),  
  9 FOREIGN KEY (company_name) REFERENCES company(name));  
  
Table created.
```

#### **4.Creating table login with constraints:**

**Query:** CREATE TABLE login(  
2 login\_id VARCHAR(50) NOT NULL,  
3 time VARCHAR(20) NOT NULL,  
4 login\_date DATE NOT NULL,  
5 type VARCHAR(20) NOT NULL);

```
SQL> CREATE TABLE login(  
  2 login_id VARCHAR(50) NOT NULL,  
  3 time VARCHAR(20) NOT NULL,  
  4 login_date DATE NOT NULL,  
  5 type VARCHAR(20) NOT NULL);  
  
Table created.
```

#### **5.Creating table history\_sales with constraints:**

**Query:** CREATE TABLE history\_sales(  
2 user\_name VARCHAR(50) NOT NULL,

3 code INT NOT NULL,  
4 time VARCHAR(20) NOT NULL,  
5 sales\_date DATE NOT NULL,  
6 name VARCHAR(20) NOT NULL,  
7 amount INT NOT NULL,  
8 quantity INT NOT NULL,  
9 FOREIGN KEY (code) REFERENCES drugs(code));

```
SQL> CREATE TABLE history_sales(  
  2  user_name VARCHAR(50) NOT NULL,  
  3  code INT NOT NULL,  
  4  time VARCHAR(20) NOT NULL,  
  5  sales_date DATE NOT NULL,  
  6  name VARCHAR(20) NOT NULL,  
  7  amount INT NOT NULL,  
  8  quantity INT NOT NULL,  
  9  FOREIGN KEY (code) REFERENCES drugs(code));
```

Table created.

## Tables:

```
SQL> desc users;
```

Name	Null?	Type
ID	NOT NULL	NUMBER(38)
PASSWORD	NOT NULL	VARCHAR2(20)
PHONE	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(50)
ADDRESS	NOT NULL	VARCHAR2(50)

```
SQL> desc company;
```

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(50)
ADDRESS	NOT NULL	VARCHAR2(50)
PHONE	NOT NULL	NUMBER(38)

```
SQL> desc drugs;
```

Name	Null?	Type
CODE	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(50)
COMPANY_NAME	NOT NULL	VARCHAR2(50)
SELLING_PRICE	NOT NULL	NUMBER(38)
EXPIRATION_DATE	NOT NULL	DATE
QUANTITY	NOT NULL	NUMBER(38)
DOSE		VARCHAR2(20)

```
SQL> desc login;
```

Name	Null?	Type
LOGIN_ID	NOT NULL	VARCHAR2(50)
TIME	NOT NULL	VARCHAR2(20)
LOGIN_DATE	NOT NULL	DATE
TYPE	NOT NULL	VARCHAR2(20)

```
SQL> desc history_sales;
```

Name	Null?	Type
USER_NAME	NOT NULL	VARCHAR2(50)
CODE	NOT NULL	NUMBER(38)
TIME	NOT NULL	VARCHAR2(20)
SALES_DATE	NOT NULL	DATE
NAME	NOT NULL	VARCHAR2(20)
AMOUNT	NOT NULL	NUMBER(38)
QUANTITY	NOT NULL	NUMBER(38)



## DML OPERATIONS:

SQL>insert into users

values(&id,'&password;,&phone,'&name','&address');

```
SQL> insert into users values(&id,'&password',&phone,'&name','&address');
Enter value for id: 007100
Enter value for password: user
Enter value for phone: 9010986719
Enter value for name: qawi
Enter value for address: narsingi
old 1: insert into users values(&id,'&password',&phone,'&name','&address')
new 1: insert into users values(007100,'user',9010986719,'qawi','narsingi')

1 row created.

SQL> /
Enter value for id: 007101
Enter value for password: dbms
Enter value for phone: 9398922373
Enter value for name: suchith
Enter value for address: gachibowli
old 1: insert into users values(&id,'&password',&phone,'&name','&address')
new 1: insert into users values(007101,'dbms',9398922373,'suchith','gachibowli')

1 row created.

SQL> /
Enter value for id: 007102
Enter value for password: vasavi
Enter value for phone: 907864565
Enter value for name: raheem
Enter value for address: mehdipatnam
old 1: insert into users values(&id,'&password',&phone,'&name','&address')
new 1: insert into users values(007102,'vasavi',907864565,'raheem','mehdipatnam')

1 row created.

SQL> /
Enter value for id: 007103
Enter value for password: medi
Enter value for phone: 9845637896
Enter value for name: vinay
Enter value for address: begum bazar
old 1: insert into users values(&id,'&password',&phone,'&name','&address')
new 1: insert into users values(007103,'medi',9845637896,'vinay','begum bazar')

1 row created.

SQL> /
Enter value for id: 007104
Enter value for password: king
Enter value for phone: 9745231890
Enter value for name: naga
Enter value for address: gandipet
old 1: insert into users values(&id,'&password',&phone,'&name','&address')
new 1: insert into users values(007104,'king',9745231890,'naga','gandipet')

1 row created.
```

SQL> insert into company values('&name','&address',&phone);

```
SQL> insert into company values('&name','&address',&phone);
Enter value for name: pharma
Enter value for address: abids
Enter value for phone: 9746355890
old 1: insert into company values('&name','&address',&phone)
new 1: insert into company values('pharma','abids',9746355890)

1 row created.

SQL> /
Enter value for name: glasko
Enter value for address: hitex
Enter value for phone: 9645327109
old 1: insert into company values('&name','&address',&phone)
new 1: insert into company values('glasko','hitex',9645327109)

1 row created.

SQL> /
Enter value for name: india medicals
Enter value for address: narsingi
Enter value for phone: 9452617309
old 1: insert into company values('&name','&address',&phone)
new 1: insert into company values('india medicals','narsingi',9452617309)

1 row created.

SQL> /
Enter value for name: pfizer
Enter value for address: koti
Enter value for phone: 9452178965
old 1: insert into company values('&name','&address',&phone)
new 1: insert into company values('pfizer','koti',9452178965)

1 row created.

SQL> /
Enter value for name: bharat
Enter value for address: masab tank
Enter value for phone: 9453627584
old 1: insert into company values('&name','&address',&phone)
new 1: insert into company values('bharat','masab tank',9453627584)

1 row created.
```

SQL> insert into drugs  
values(&code,&'&name','&company\_name',&selling\_price,&'&expir  
ation\_date',&qty,&'&dose');

```
SQL> insert into drugs values(&code, '&name', '&company_name', &selling_price, '&expiration_date', &qty, '&dose');
Enter value for code: 107201
Enter value for name: aspirin
Enter value for company_name: glasko
Enter value for selling_price: 30
Enter value for expiration_date: 03-feb-2024
Enter value for qty: 50
Enter value for dose: 500ml
old 1: insert into drugs values(&code, '&name', '&company_name', &selling_price, '&expiration_date', &qty, '&dose')
new 1: insert into drugs values(107201, 'aspirin', 'glasko', 30, '03-feb-2024', 50, '500ml')

1 row created.

SQL> /
Enter value for code: 107202
Enter value for name: dolo
Enter value for company_name: pharma
Enter value for selling_price: 10
Enter value for expiration_date: 23-mar-2024
Enter value for qty: 100
Enter value for dose: 500mg
old 1: insert into drugs values(&code, '&name', '&company_name', &selling_price, '&expiration_date', &qty, '&dose')
new 1: insert into drugs values(107202, 'dolo', 'pharma', 10, '23-mar-2024', 100, '500mg')

1 row created.

SQL> /
Enter value for code: 107203
Enter value for name: eldoper
Enter value for company_name: pfizer
Enter value for selling_price: 20
Enter value for expiration_date: 01-jun-2024
Enter value for qty: 50
Enter value for dose: 500mg
old 1: insert into drugs values(&code, '&name', '&company_name', &selling_price, '&expiration_date', &qty, '&dose')
new 1: insert into drugs values(107203, 'eldoper', 'pfizer', 20, '01-jun-2024', 50, '500mg')

1 row created.
```

SQL> insert into login  
values('&login\_id','&time','&login\_date','&type');

```
SQL> insert into login values('&login_id','&time','&login_date','&type');
Enter value for login_id: 007100
Enter value for time: 03:20
Enter value for login_date: 01-may-2022
Enter value for type: user
old 1: insert into login values('&login_id','&time','&login_date','&type')
new 1: insert into login values('007100','03:20','01-may-2022','user')

1 row created.

SQL> /
Enter value for login_id: 007100
Enter value for time: 09:15
Enter value for login_date: 01-may-2022
Enter value for type: user
old 1: insert into login values('&login_id','&time','&login_date','&type')
new 1: insert into login values('007100','09:15','01-may-2022','user')

1 row created.

SQL> /
Enter value for login_id: 007100
Enter value for time: 02:23
Enter value for login_date: 01-may-2022
Enter value for type: user
old 1: insert into login values('&login_id','&time','&login_date','&type')
new 1: insert into login values('007100','02:23','01-may-2022','user')

1 row created.

SQL> /
Enter value for login_id: 007100
Enter value for time: 05:54
Enter value for login_date: 01-may-2022
Enter value for type: user
old 1: insert into login values('&login_id','&time','&login_date','&type')
new 1: insert into login values('007100','05:54','01-may-2022','user')

1 row created.
```

## **IMPLEMENTATION:**

### **JAVA-SQL Connectivity using JDBC:**

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases. The connection to the database can be performed using Java programming (JDBC API) as:

```
{ DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver()); //
Connect to Oracle Database Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE"
,"qawi","qawi"); Statement statement = con.createStatement() String
query = "UPDATE COMPANY SET SS1=" +"""+ jTextField3.getText()
+""",SS2=" +"""+ jTextField5.getText() +""", AOI ="+" ""+
jTextField2.getText() +"" WHERE SID =" + jTextField4.getText();
ResultSet rs = statement.executeQuery(query);
JOptionPane.showMessageDialog(new JFrame(), "Updated
Successfully", "INFORMATION",
JOptionPane.INFORMATION_MESSAGE); rs.close(); statement.close();
con.close(); }
```

## **Front-end Programs (User Interfaces) Home Page:**

### **1.Login Page:**

```
package java_app;

import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Login extends JFrame {

    private JTextField signUpUserIdField;
    private JTextField signUpNameField;
    private JPasswordField signUpPasswordField;
    private JTextField signUpPhoneField;
    private JTextField signUpAddressField;
    private JTextField loginUserIdField;
    private JPasswordField loginPasswordField;
    private JTextField loginUserIdField1;
    private JPasswordField loginPasswordField1;

    public Login() {

        // Initialize the database

        setTitle("User Sign Up/Login");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500, 250);
        setLocationRelativeTo(null);

        // Create a panel for padding space
        JPanel paddingPanel = new JPanel();
        paddingPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10)); // Add padding
        paddingPanel.setLayout(new BorderLayout());

        //Create panels for sign up and login components
        JPanel signUpPanel = createSignUpPanel();
        JPanel loginPanel = createLoginPanel();
        JPanel adminPanel = createAdminPanel();

        // Create tabbed pane to switch between sign up and login
        JTabbedPane tabbedPane = new JTabbedPane();
        tabbedPane.addTab("User SignUp", signUpPanel);

```

```

        tabbedPane.addTab("User Login", loginPanel);
        tabbedPane.addTab("Admin Login", adminPanel);
// Add tabbed pane to the padding panel
        paddingPanel.add(tabbedPane, BorderLayout.CENTER);
// Add padding panel to the frame
        getContentPane().add(paddingPanel);
    }
    private JPanel createSignUpPanel() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(6, 2, 10, 10)); // Add gaps between components
// Sign Up components
        JLabel signUpUserIdLabel = new JLabel("User ID:");
        signUpUserIdField = new JTextField(15); // Reduced text field size
        JLabel signUpNameLabel = new JLabel("User Name:");
        signUpNameField = new JTextField(15); // Reduced text field size
        JLabel signUpPasswordLabel = new JLabel("Password:");
        signUpPasswordField = new JPasswordField(15); // Reduced text field size
        JLabel signUpPhoneLabel = new JLabel("Phone");
        signUpPhoneField = new JTextField(15);
        JLabel signUpAddressLabel = new JLabel("Address");
        signUpAddressField = new JTextField(15);
        JButton signUpButton = new JButton("Sign Up");
        signUpButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                handleSignIn();
            }
        });
// Add components to the sign-up panel
        panel.add(signUpUserIdLabel);
        panel.add(signUpUserIdField);
        panel.add(signUpNameLabel);
        panel.add(signUpNameField);
        panel.add(signUpPasswordLabel);

```

```

panel.add(signUpPasswordField);
panel.add(signUpPhoneLabel);
panel.add(signUpPhoneField);
panel.add(signUpAddressLabel);
panel.add(signUpAddressField);
panel.add(new JLabel()); // Empty label for spacing
panel.add(signUpButton);
return panel;
}

private void handleSignIn() {
    String userId = signUpUserIdField.getText();
    String name = signUpNameField.getText();
    String password = new String(signUpPasswordField.getPassword());
    String phone = signUpUserIdField.getText();
    String address = signUpNameField.getText();
    if (userId.isEmpty() || name.isEmpty() || password.isEmpty() ) {
        JOptionPane.showMessageDialog(this, "Please fill in all the fields", "Error",
JOptionPane.ERROR_MESSAGE);
        return;
    }
    else
    {
        String driverClassName = "oracle.jdbc.driver.OracleDriver";
        try {
// Load the JDBC driver
            Class.forName(driverClassName);

// Establish a connection to the database
            Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "qawi", "qawi");
// Perform database operations using the connection
            Statement stmt=con.createStatement();

            int a=stmt.executeUpdate("insert into users
values("+userId+"','"+password+"','"+phone+"','"+name+"','"+address+"')");
            if(a>0)

```



```

                                JOptionPane.showMessageDialog(this, "Sign-up successful",
"Message", JOptionPane.INFORMATION_MESSAGE);

                                else

                                JOptionPane.showMessageDialog(this, "Fail", "Error",
JOptionPane.ERROR_MESSAGE);
// Close the connection

                                con.close();

                                System.out.println("Connection closed successfully.");

                                }

                                catch (ClassNotFoundException e) {

                                System.err.println("Failed to load JDBC driver: " + e.getMessage());

                                }

                                catch (SQLException e) {

                                System.err.println("Failed to connect to the database: " + e.getMessage());

                                }

                                }

                                }

//Clear sign in fields

//Clear the sign up fields

                                signUpUserIdField.setText("");

                                signUpNameField.setText("");

                                signUpPasswordField.setText("");

                                signUpPhoneField.setText("");

                                signUpAddressField.setText("");

                                }

private JPanel createLoginPanel() {

                                JPanel panel = new JPanel();

                                panel.setLayout(new GridLayout(3, 2, 5, 5)); // Add gaps between components

// Login components

                                JLabel loginUserIdLabel = new JLabel("User ID:");

                                loginUserIdField1 = new JTextField(15); // Reduced text field size

                                JLabel loginPasswordLabel = new JLabel("Password:");

                                loginPasswordField1 = new JPasswordField(15); // Reduced text field size

```

```

        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                handleLogin();
            }
        });

// Add components to the login panel
        panel.add(loginUserIdLabel);
        panel.add(loginUserIdField1);
        panel.add(loginPasswordLabel);
        panel.add(loginPasswordField1);
        panel.add(new JLabel()); // Empty label for spacing
        panel.add(loginButton);
        return panel;
    }

    @SuppressWarnings("unused")
    private void displayInfoMessage(String message) {
        Dialog infoDialog = new Dialog(this, "Info Message", true);
        infoDialog.setLayout(new FlowLayout());
        infoDialog.setSize(200, 100);
        Label messageLabel = new Label(message);
        Button closeButton = new Button("Close");
        closeButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                infoDialog.dispose();
            }
        });
        infoDialog.add(messageLabel);
        infoDialog.add(closeButton);
        infoDialog.setVisible(true);
    }

```

```

private void handleLogin() {
    String userId = loginUserIdField1.getText();
    String password = new String(loginPasswordField1.getPassword());

    if (userId.isEmpty() || password.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please enter ID and password",
            "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }
    else {
        String driverClassName = "oracle.jdbc.driver.OracleDriver";
        try {
            // Load the JDBC driver
            Class.forName(driverClassName);

            // Establish a connection to the database
            Connection con =
                DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "qawi", "qawi");
            // Perform database operations using the connection
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("select password from users where
            id="+userId+"and password='"+password+"'");
            if(rs.next())
            {
                JOptionPane.showMessageDialog(this, "Login Successful",
                    "Success", JOptionPane.INFORMATION_MESSAGE);
                setVisible(false);
                new HomePage(Integer.parseInt(userId));
            }
            else
                JOptionPane.showMessageDialog(this, "Invalid credentials",
                    "Error", JOptionPane.ERROR_MESSAGE);
            // Close the connection
            con.close();
            System.out.println("Connection closed successfully.");
        }
    }
}

```

```

        } catch (ClassNotFoundException e) {
            System.err.println("Failed to load JDBC driver: " + e.getMessage());
        } catch (SQLException e) {
            System.err.println("Failed to connect to the database: " +
e.getMessage());
        }
    }

// Clear login fields
    loginUserIdField.setText("");
    loginPasswordField.setText("");
}

private JPanel createAdminPanel() {
    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(3, 2, 5, 5)); // Add gaps between components

// Login components
    JLabel loginUserIdLabel = new JLabel("Admin ID:");
    loginUserIdField = new JTextField(15); // Reduced text field size
    JLabel loginPasswordLabel = new JLabel("Password:");
    loginPasswordField = new JPasswordField(15); // Reduced text field size
    JButton loginButton = new JButton("Login");
    loginButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            handleAdminLogin();
        }
    });

// Add components to the login panel
    panel.add(loginUserIdLabel);
    panel.add(loginUserIdField);
    panel.add(loginPasswordLabel);
    panel.add(loginPasswordField);
    panel.add(new JLabel()); // Empty label for spacing

```

```

        panel.add(loginButton);

        return panel;
    }

    private void handleAdminLogin() {
        String adminId = loginUserIdField.getText();

        String password = new String(loginPasswordField.getPassword());

        if (adminId.isEmpty() || password.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Please enter ID and password", "Error",
JOptionPane.ERROR_MESSAGE);

            return;
        }
        else {
            String driverClassName = "oracle.jdbc.driver.OracleDriver";

            try {
// Load the JDBC driver

                Class.forName(driverClassName);

// Establish a connection to the database

                Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "qawi", "qawi");

// Perform database operations using the connection

                Statement stmt=con.createStatement();

                ResultSet rs=stmt.executeQuery("select password from admin where
adminID="+adminId+" and password='"+password+"'");

                if(rs.next())
                {
                    JOptionPane.showMessageDialog(this, "Login Successful", "Success",
JOptionPane.INFORMATION_MESSAGE);

                    setVisible(false);

                    new UserManager();

                }
                else

                    JOptionPane.showMessageDialog(this, "Invalid credentials", "Error",
JOptionPane.ERROR_MESSAGE);

```

```

// Close the connection

        con.close();

        System.out.println("Connection closed successfully.");
    } catch (ClassNotFoundException e) {
        System.err.println("Failed to load JDBC driver: " + e.getMessage());
    } catch (SQLException e) {
        System.err.println("Failed to connect to the database: " + e.getMessage());
    }
}

// Clear login fields
loginUserIdField.setText("");
loginPasswordField.setText("");
}

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    } catch (Exception e) {
        e.printStackTrace();
    }

    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}
}

```

## **2. UserManage.java**

```
package java_app;
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;
public class UserManage extends JFrame{

    private DefaultTableModel tableModel;
    private JTable studentTable;
    private JTextField rollNumberField=new JTextField(20);
    private JButton searchButton;
        private JLabel code=new JLabel(" code:");
        private JTextField code_txt=new JTextField(20);
        private JLabel drug_name=new JLabel("drug Name:");
        private JTextField drug_name_txt=new JTextField(20);
        private JLabel dose=new JLabel("dose:");
        private JTextField dose_txt=new JTextField(20);
        private JLabel qty=new JLabel("quantity:");
        private JTextField qty_txt=new JTextField(20);
        private JButton submit=new JButton("Submit");
        private JButton modify=new JButton("Modify");
        private JButton delete=new JButton("Delete");
        private JMenuBar menubar=new JMenuBar();
    private JButton close=new JButton("Close");
        private JButton Avdrugs=new JButton("Available Drugs");
        private JPanel p1;
        private JPanel p2;
        private JPanel p3;
        private JPanel p4;
        private JPanel p5;
        private JPanel p6;
        private JPanel p7;
        private JPanel p8;
        private JPanel p9;

    public UserManage()
    {
        setTitle("Dashboard");
        setSize(700,300);
        setVisible(true);
        setLayout(new FlowLayout(0,1, getDefaultCloseOperation()));
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        /*menubar.add(ipm);*/
        menubar.add(Avdrugs);
        menubar.add(close);
        setJMenuBar(menubar);
        p1=new JPanel();
        p2=new JPanel();
        p3=new JPanel();
        p4=new JPanel();
        p5=new JPanel();
        p6=new JPanel();
        p7=new JPanel();
        p8=new JPanel();
    }
}

```

```

        p9=new JPanel();
        p1.add(code);
        p1.add(code_txt);
        p1.add(drug_name);
        p1.add(drug_name_txt);
        p2.add(dose);
        p2.add(dose_txt);
        p2.add(qty);
        p2.add(qty_txt);
        p7.add(submit);
        p7.add(modify);
        p7.add(delete);
        add(p1);
        add(p2);
        add(p3);
    add(p4);
    add(p5);
    add(p6);
    add(p7);

    try{
        Class.forName("oracle.jdbc.OracleDriver");
        Connection con=
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","qawi","qawi");
        Statement stmt=con.createStatement();
    }
    catch(SQLException e){System.out.println(e);}
    catch(Exception ex){System.out.println(ex);}
    submit.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
            try{
                Connection conn=
                DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","qawi","qawi");
                Statement stmt=conn.createStatement();
                String sql = "INSERT INTO drugs(code, name, dose, quantity) " +
                "VALUES (" + code_txt.getText() + ", " + drug_name_txt.getText() + ", " + dose_txt.getText() +
                ", " + qty_txt.getText() + ")";

                int a=stmt.executeUpdate(sql);
                if(a>0)
                    JOptionPane.showMessageDialog(null, "Insertion    successfull",
                    "Message", JOptionPane.INFORMATION_MESSAGE);
                else
                    JOptionPane.showMessageDialog(null, "Failed", "Error",
                    JOptionPane.ERROR_MESSAGE);

                //conn.commit();
            }
            catch(SQLException sqle)
            {
                System.out.println("Could not insert tuple"+sqle);
            }

            catch(Exception ex){System.out.println(ex);}

```



```

    }
    });
    modify.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "qawi", "qawi");

                // Start the transaction
                conn.setAutoCommit(false);

                String updateStudentQuery = "UPDATE drugs SET name=?, dose=?, quantity=?
WHERE code=?";
                try (PreparedStatement updateStudentStmt =
conn.prepareStatement(updateStudentQuery))
                {

                    updateStudentStmt.setString(1, code_txt.getText());
                    updateStudentStmt.setString(2, drug_name_txt.getText());
                    updateStudentStmt.setString(3, dose_txt.getText());
                    updateStudentStmt.setString(4, qty_txt.getText());

                    updateStudentStmt.executeUpdate();
                    conn.commit();
                    JOptionPane.showMessageDialog(null, "Data updated successfully!");
                } catch (SQLException ex) {
                    // Rollback the transaction if any update fails
                    conn.rollback();
                    ex.printStackTrace();
                    JOptionPane.showMessageDialog(null, "Failed to update data!");
                } finally {
                    // Enable auto-commit after the transaction
                    conn.setAutoCommit(true);
                }
            } catch (SQLException ex) {
                ex.printStackTrace();
            }
        }
    });

```

```

        //conn.commit();

        delete.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                try{

```

```

        Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "qawi", "qawi");
        Statement stmt = conn.createStatement();
        String sql = "DELETE FROM drugs WHERE code=" + code_txt.getText()+"";
        int a=stmt.executeUpdate(sql);
        if(a>0)
            JOptionPane.showMessageDialog(null, "Delete successful",
"Message", JOptionPane.INFORMATION_MESSAGE);
        else
            JOptionPane.showMessageDialog(null, "Delete Failed", "Error",
JOptionPane.ERROR_MESSAGE);
        conn.close();
    }
    catch(SQLException sqle)
    {
        System.out.println("Could not delete tuple"+sqle);
    }
    catch(Exception ex){System.out.println(ex);}
    }
});

    Avdrugs.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
    // Show all the questions

    //Set up the JFrame
    new DrugsAv();
}

    });
}

```

```

public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){// Run the GUI in the Event-Dispatching thread for
thread-safety
        public void run(){new UserManage();}
    });
}
}

```

### **3.DrugsAv.java**

```

package java_app;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import javax.swing.table.DefaultTableModel;

public class DrugsAv extends JFrame {
    /**
     *

```

```

        */
        private static final long serialVersionUID = 1L;
        private DefaultTableModel tableModel;
private JTable drugTable;
private JButton searchButton;

public DrugsAv() {
    // Set up the JFrame
    setTitle("Available Drugs");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setPreferredSize(new Dimension(600, 400));

    // Initialize the table model and JTable
    tableModel = new DefaultTableModel();
    tableModel.addColumn("Drug Code");
    tableModel.addColumn("Drug Name");
    tableModel.addColumn("Quantity");
        tableModel.addColumn("Dose");
    drugTable = new JTable(tableModel);

    // Set up the roll number input field and search button
    searchButton = new JButton("Search");

    // Add ActionListener to the search button
    searchButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            searchStudent();
        }
    });

    // Set up the layout and add the components
    JPanel topPanel = new JPanel();
    topPanel.add(searchButton);

    setLayout(new BorderLayout());
    add(topPanel, BorderLayout.NORTH);
    add(new JScrollPane(drugTable), BorderLayout.CENTER);

    // Display the JFrame
    pack();
    setLocationRelativeTo(null);
    setVisible(true);
}

private void searchStudent() {

    // Clear the table before performing a new search
    tableModel.setRowCount(0);

    // Set up the database connection and retrieve student details
    try {
        Connection connection
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","qawi","qawi");
    }
}

```

```

        PreparedStatement preparedStatement = connection.prepareStatement("Select * from drugs");
        //preparedStatement.setString(1, code);
        ResultSet resultSet = preparedStatement.executeQuery();
        boolean found = false;
        while (resultSet.next()) {
            String drugCode = resultSet.getString("code");
            String drugName = resultSet.getString("name");
            String quantity = resultSet.getString("quantity");
            String dose = resultSet.getString("dose");
            tableModel.addRow(new Object[] {drugCode, drugName, quantity, dose});
            found = true;
        }
        if (!found) {
            JOptionPane.showMessageDialog(this, "Not found", "Search Result",
JOptionPane.INFORMATION_MESSAGE);
        }

        resultSet.close();
        preparedStatement.close();
        connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new DrugsAv());
}
}

```

#### **4. HomePage.java**

```

package java_app;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class HomePage extends JFrame {

    private int userId;

    public HomePage(int userId) {

```

```

        this.userId=userId;

        setTitle("User Dashboard");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setSize(500, 400);

        setLocationRelativeTo(null);

        setVisible(true);

// Create menu bar

        JMenuBar menuBar = new JMenuBar();

// Create menus

        JMenu userMenu = new JMenu("User");

        JMenu ordersMenu = new JMenu("Orders");

        JMenu drugsMenu = new JMenu("Available drugs");

// Create menu items for Farmer menu

        JMenuItem viewOrdersItem = new JMenuItem("View Orders");

// Create menu items for Farm menu

        JMenuItem orderInsertItem = new JMenuItem("Insert Order");

        JMenuItem orderUpdateItem = new JMenuItem("Update Order");

        JMenuItem orderDeleteItem = new JMenuItem("Delete Order");

        JMenuItem avDrugItem = new JMenuItem("View");

// Add action listeners to menu items

        viewOrdersItem.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                viewOrdersDialog();

            }

        });

        orderInsertItem.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                showInsertOrderDialog();

            }

        });

        orderUpdateItem.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                showUpdateOrderDialog();

```

```

        }
    });
    orderDeleteItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            showDeleteOrderDialog();
        }
    });
    avDrugItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            new DrugsAv();
        }
    });
// Add menu items to menus
    userMenu.add(viewOrdersItem);
    ordersMenu.add(orderInsertItem);
    ordersMenu.add(orderUpdateItem);
    ordersMenu.add(orderDeleteItem);
    drugsMenu.add(avDrugItem);
// Add menus to menu bar
    menuBar.add(userMenu);
    menuBar.add(ordersMenu);
    menuBar.add(drugsMenu);
// Set the menu bar for the frame
    setJMenuBar(menuBar);
}
private void viewOrdersDialog()
{
    String driverClassName = "oracle.jdbc.driver.OracleDriver";
    String url = "jdbc:oracle:thin:@localhost:1521:xe";
    String username = "qawi";
    String pass = "qawi";
    try {
        // Load the JDBC driver

```

```

        Class.forName(driverClassName);

        // Establish a connection to the database
        Connection con = DriverManager.getConnection(url, username, pass);

        // Perform database operations using the connection
        Statement stmt=con.createStatement();
        ResultSet rs=stmt.executeQuery("select code,name,quantity from
history_sales");

        StringBuilder sb = new StringBuilder();
        while(rs.next())
        {
            int x=rs.getInt(1);
            String m=rs.getString(2);
            int n=rs.getInt(3);

            sb.append(x).append(":
").append(m).append(",").append(n).append(".\n");
        }

        JTextArea textArea = new JTextArea(sb.toString());
        textArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(textArea);
        scrollPane.setPreferredSize(new Dimension(400, 300));
        JOptionPane.showMessageDialog(this, scrollPane, "Orders",
JOptionPane.PLAIN_MESSAGE);

        con.close();

        System.out.println("Connection closed successfully.");
    } catch (ClassNotFoundException e) {
        System.err.println("Failed to load JDBC driver: " + e.getMessage());
    } catch (SQLException e) {
        System.err.println("Failed to connect to the database: " + e.getMessage());
    }
}

private void showInsertOrderDialog() {
    JPanel panel = new JPanel();

```

```

        panel.setLayout(new GridLayout(4, 2));

// Insert Farm components

        JLabel drugCodeLabel = new JLabel("Drug code");
        JTextField drugCodeField = new JTextField(20);
        JLabel drugNameLabel = new JLabel("Drug Name:");
        JTextField drugNameField = new JTextField(20);
        JLabel drugQuantityLabel = new JLabel("Drug quantity");
        JTextField drugQuantityField = new JTextField(20);
        JButton addOrderButton = new JButton("Add Order");
        addOrderButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String drugCode = drugCodeField.getText();
                String drugName = drugNameField.getText();
                String drugQuantity = drugQuantityField.getText();

// Perform insertion logic here

                String driverClassName = "oracle.jdbc.driver.OracleDriver";
                String url = "jdbc:oracle:thin:@localhost:1521:xe";
                String username = "qawi";
                String pass = "qawi";
                try {

// Load the JDBC driver

                    Class.forName(driverClassName);

// Establish a connection to the database

                    Connection con = DriverManager.getConnection(url,
username, pass);

// Perform database operations using the connection

                    Statement stmt=con.createStatement();
                    int a=stmt.executeUpdate("insert into
history_sales(code,name,quantity) values("+drugCode+", "+drugName+", "+drugQuantity+"");
                    if(a>0)

                        JOptionPane.showMessageDialog(null,
"Order added successfully", "Message", JOptionPane.INFORMATION_MESSAGE);
                    else

```



```

JOptionPane.showMessageDialog(null,
"Order details should be unique", "Error", JOptionPane.ERROR_MESSAGE);

// Close the connection

con.close();

System.out.println("Connection closed
successfully.");

} catch (ClassNotFoundException s) {

System.err.println("Failed to load JDBC driver: " +
s.getMessage());

} catch (SQLException s) {

System.err.println("Failed to connect to the database:
" + s.getMessage());

}

// Clear the fields

drugCodeField.setText("");
drugNameField.setText("");
drugQuantityField.setText("");

}

});

// Add components to the panel

panel.add(drugCodeLabel);
panel.add(drugCodeField);
panel.add(drugNameLabel);
panel.add(drugNameField);
panel.add(drugQuantityLabel);
panel.add(drugQuantityField);
panel.add(new JLabel());
panel.add(addOrderButton);

JOptionPane.showOptionDialog(null, panel, "Insert Order",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[] {}, null);

}

private void showUpdateOrderDialog() {

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 2));

// Update Farm components

```

```

JLabel drugCodeLabel = new JLabel("Drug Code");
JTextField drugCodeField = new JTextField(20);
JLabel drugNameLabel = new JLabel("Drug Name:");
JTextField drugNameField = new JTextField(20);
JLabel drugQuantityLabel = new JLabel("Drug Quantity");
JTextField drugQuantityField = new JTextField(20);
JButton updateOrderButton = new JButton("Update Order");
updateOrderButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String drugCode = drugCodeField.getText();
        String drugName = drugNameField.getText();
        String drugQuantity = drugQuantityField.getText();

// Perform update logic here

        String driverClassName = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        String username = "qawi";
        String pass = "qawi";
        try {
            // Load the JDBC driver
            Class.forName(driverClassName);

            // Establish a connection to the database
            Connection con = DriverManager.getConnection(url,
username, pass);

            // Perform database operations using the connection
            Statement stmt=con.createStatement();
            int a = stmt.executeUpdate("UPDATE history_sales
SET name='" + drugName + "',quantity='" + drugQuantity + "'WHERE code=" +drugCode );
            if(a>0)
                JOptionPane.showMessageDialog(null,
"Order updated successfully", "Message", JOptionPane.INFORMATION_MESSAGE);
            else
                JOptionPane.showMessageDialog(null,
"Order not found", "Error", JOptionPane.ERROR_MESSAGE);

            // Close the connection

```

```

con.close();

System.out.println("Connection closed

successfully.");

} catch (ClassNotFoundException s) {

System.err.println("Failed to load JDBC driver: " +

s.getMessage());

} catch (SQLException s) {

System.err.println("Failed to connect to the database:

" + s.getMessage());

}

// Clear the fields

drugCodeField.setText("");

drugNameField.setText("");

drugQuantityField.setText("");

}

});

// Add components to the panel

panel.add(drugCodeLabel);

panel.add(drugCodeField);

panel.add(drugNameLabel);

panel.add(drugNameField);

panel.add(drugQuantityLabel);

panel.add(drugQuantityField);

panel.add(new JLabel());

panel.add(updateOrderButton);

JOptionPane.showOptionDialog(null, panel, "Update Order",

JOptionPane.DEFAULT_OPTION,

JOptionPane.PLAIN_MESSAGE, null, new Object[] {},

null);

}

private void showDeleteOrderDialog() {

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(2, 2));

// Delete Farm components

JLabel drugCodeLabel = new JLabel("Drug Code");

```

```

        JTextField drugCodeField = new JTextField(20);

        JButton deleteOrderButton = new JButton("Delete Order");

        deleteOrderButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                String drugCode = drugCodeField.getText();

                // Perform delete logic here

                String driverClassName = "oracle.jdbc.driver.OracleDriver";

                String url = "jdbc:oracle:thin:@localhost:1521:xe";

                String username = "qawi";

                String pass = "qawi";

                try {

                    // Load the JDBC driver

                    Class.forName(driverClassName);

                    // Establish a connection to the database

                    Connection con = DriverManager.getConnection(url,
username, pass);

                    // Perform database operations using the connection

                    Statement stmt=con.createStatement();

                    int a=stmt.executeUpdate("delete from history_sales
where code="+drugCode+"");

                    if(a>0)

                        JOptionPane.showMessageDialog(null,
"Deleted successfully", "Message", JOptionPane.INFORMATION_MESSAGE);

                    else

                        JOptionPane.showMessageDialog(null,
"Order not found", "Error", JOptionPane.ERROR_MESSAGE);

                    // Close the connection

                    con.close();

                    System.out.println("Connection closed
successfully.");

                } catch (ClassNotFoundException s) {

                    System.err.println("Failed to load JDBC driver: " +
s.getMessage());

                } catch (SQLException s) {

                    System.err.println("Failed to connect to the database:
" + s.getMessage());

```

```

        }

        // Clear the field

        drugCodeField.setText("");

    }

});

// Add components to the panel
panel.add(drugCodeLabel);
panel.add(drugCodeField);
panel.add(new JLabel());
panel.add(deleteOrderButton);

JOptionPane.showOptionDialog(null, panel, "Delete Order",
JOptionPane.DEFAULT_OPTION, JOptionPane.PLAIN_MESSAGE, null, new Object[] {}, null);

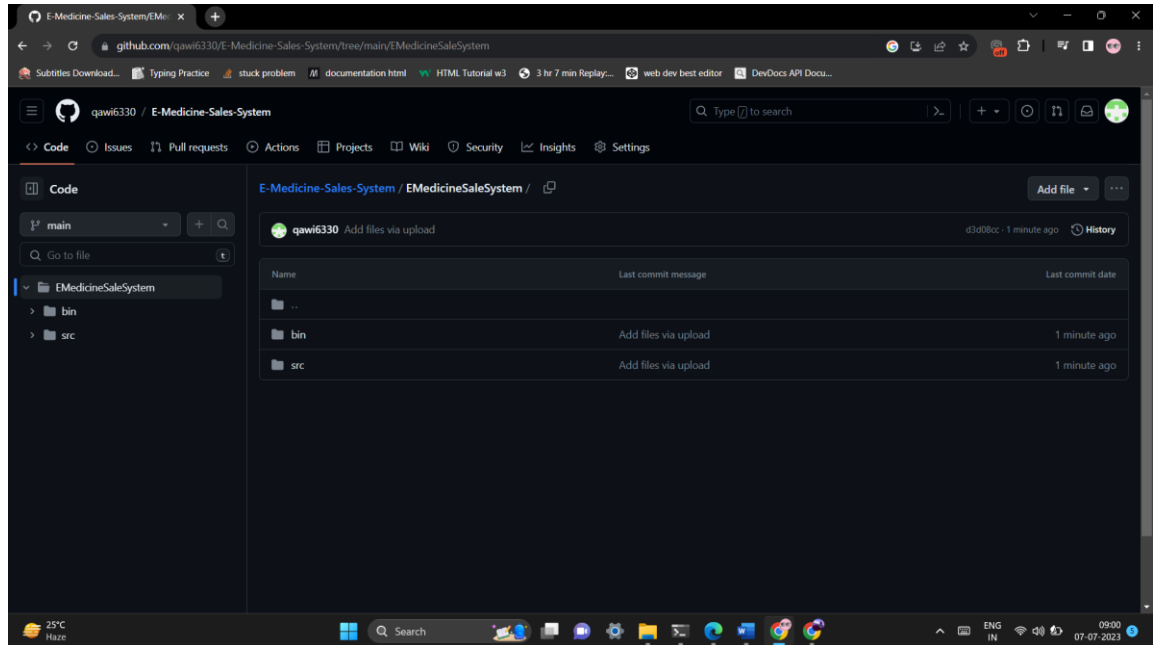
}

/*public static void main(String[] args) {
SwingUtilities.invokeLater(new Runnable() {
public void run() {
new Home().setVisible(true);
}
});
}*/
}

```

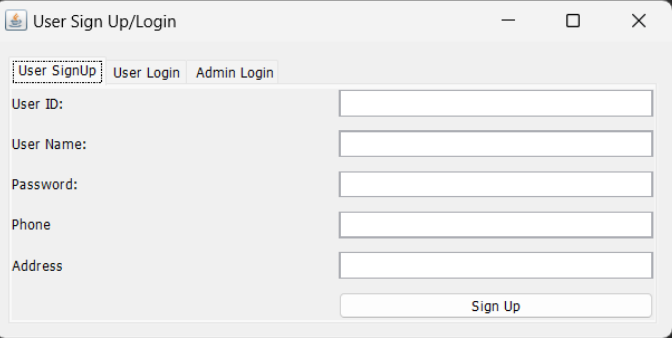
## GITHUB LINKS AND FOLDER STRUCTURE:

<https://github.com/qawi6330/E-Medicine-Sales-System/>

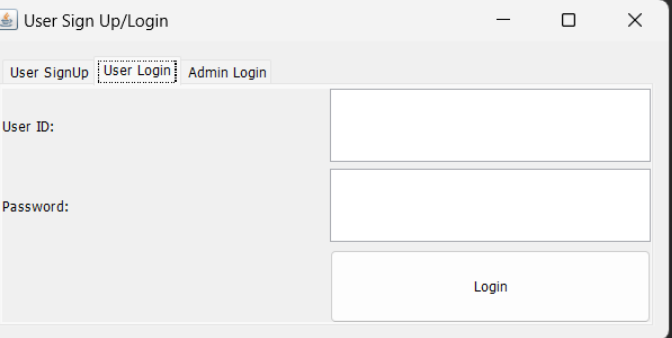


## TESTING:

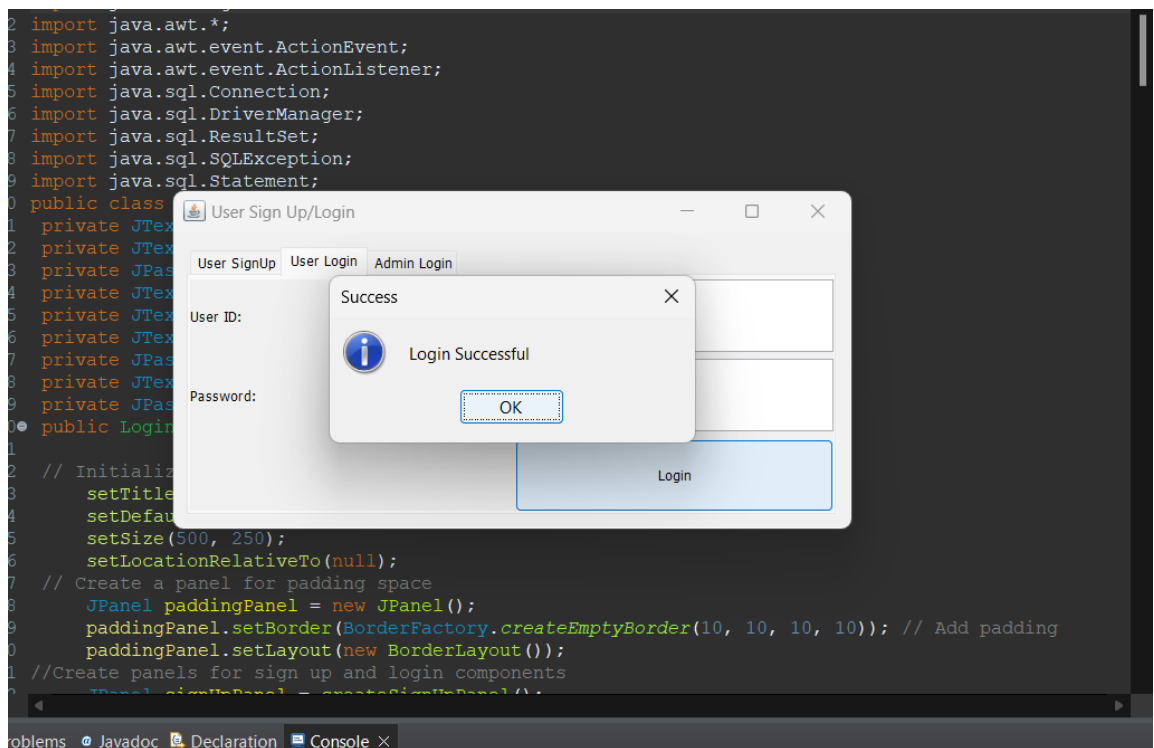
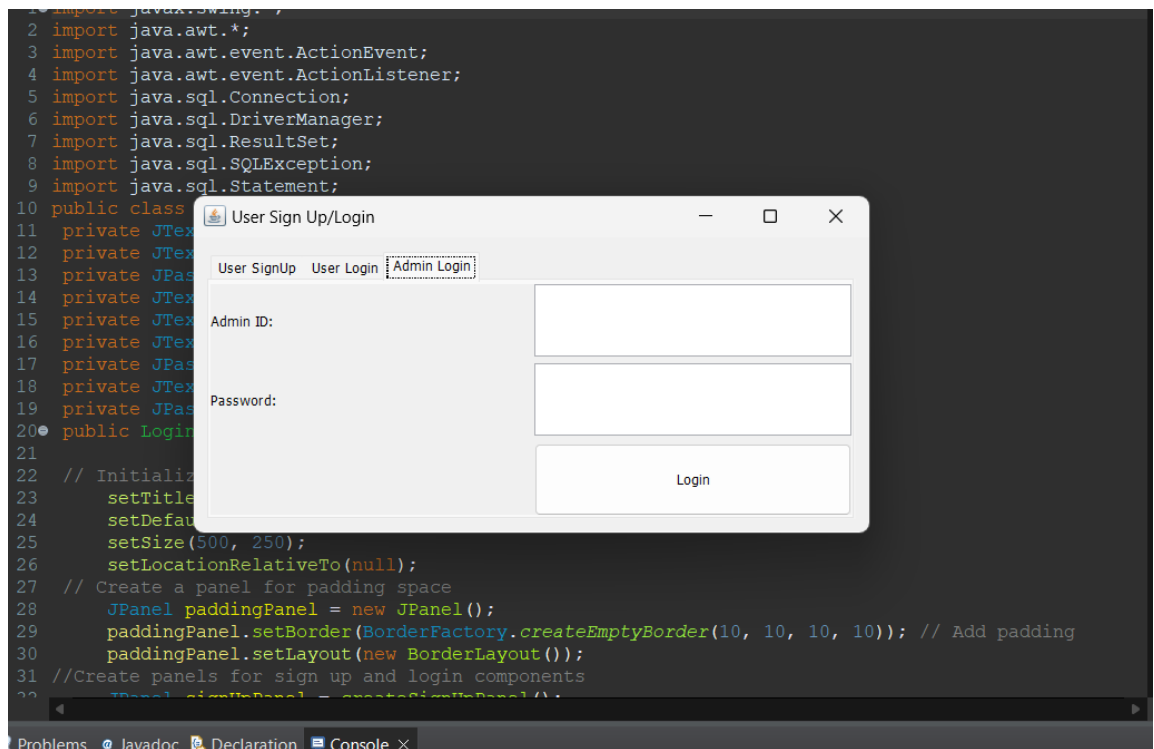
```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.sql.Statement;
10 public class UserSignUp/Login {
11     private JTextArea textArea;
12     private JTextField userLogin;
13     private JPasswordField passwordLogin;
14     private JTextField userSignUp;
15     private JTextField passwordSignUp;
16     private JPasswordField phoneSignUp;
17     private JPasswordField addressSignUp;
18     private JPasswordField loginButton;
19     private JPasswordField signUpButton;
20     public Login() {
21         // Initialize the window
22         setTitle("User Sign Up/Login");
23         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         setSize(500, 250);
25         setLocationRelativeTo(null);
26         // Create a panel for padding space
27         JPanel paddingPanel = new JPanel();
28         paddingPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10)); // Add padding
29         paddingPanel.setLayout(new BorderLayout());
30         // Create panels for sign up and login components
31         JPanel signUpPanel = new JPanel();
32         signUpPanel.setLayout(new BorderLayout());
33         signUpPanel.add(new JLabel("User ID:"));
34         signUpPanel.add(new JTextField());
35         signUpPanel.add(new JLabel("User Name:"));
36         signUpPanel.add(new JTextField());
37         signUpPanel.add(new JLabel("Password:"));
38         signUpPanel.add(new JPasswordField());
39         signUpPanel.add(new JLabel("Phone:"));
40         signUpPanel.add(new JPasswordField());
41         signUpPanel.add(new JLabel("Address:"));
42         signUpPanel.add(new JPasswordField());
43         signUpPanel.add(new JButton("Sign Up"));
44         loginPanel.add(new JLabel("User ID:"));
45         loginPanel.add(new JTextField());
46         loginPanel.add(new JLabel("Password:"));
47         loginPanel.add(new JPasswordField());
48         loginPanel.add(new JButton("Login"));
49         paddingPanel.add(signUpPanel, BorderLayout.CENTER);
50         paddingPanel.add(loginPanel, BorderLayout.SOUTH);
51         getContentPane().add(paddingPanel);
52     }
53 }
```



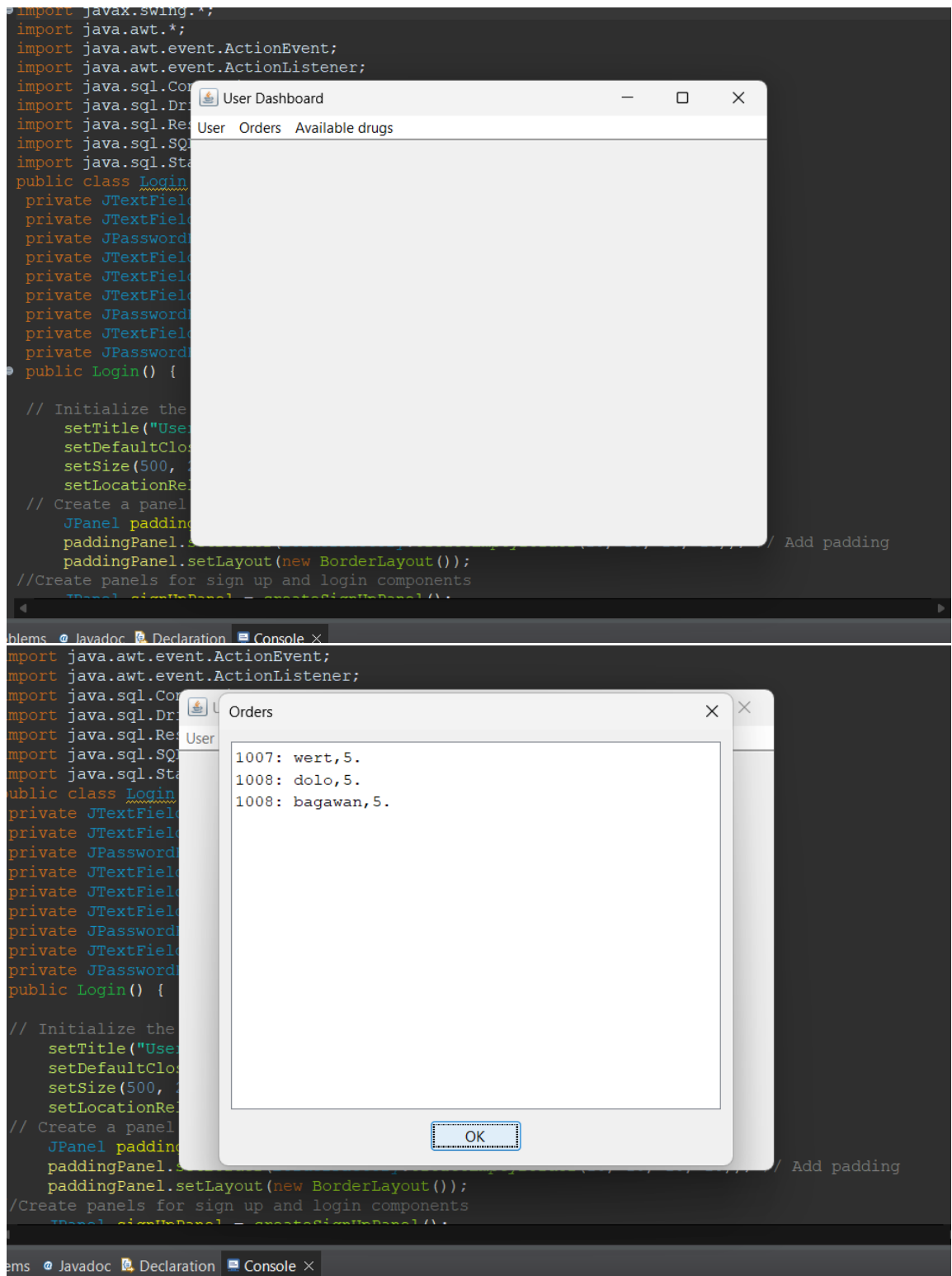
```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.sql.Statement;
10 public class UserSignUp/Login {
11     private JTextArea textArea;
12     private JTextField userLogin;
13     private JPasswordField passwordLogin;
14     private JTextField userSignUp;
15     private JTextField passwordSignUp;
16     private JPasswordField phoneSignUp;
17     private JPasswordField addressSignUp;
18     private JPasswordField loginButton;
19     private JPasswordField signUpButton;
20     public Login() {
21         // Initialize the window
22         setTitle("User Sign Up/Login");
23         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         setSize(500, 250);
25         setLocationRelativeTo(null);
26         // Create a panel for padding space
27         JPanel paddingPanel = new JPanel();
28         paddingPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10)); // Add padding
29         paddingPanel.setLayout(new BorderLayout());
30         // Create panels for sign up and login components
31         JPanel signUpPanel = new JPanel();
32         signUpPanel.setLayout(new BorderLayout());
33         signUpPanel.add(new JLabel("User ID:"));
34         signUpPanel.add(new JTextField());
35         signUpPanel.add(new JLabel("User Name:"));
36         signUpPanel.add(new JTextField());
37         signUpPanel.add(new JLabel("Password:"));
38         signUpPanel.add(new JPasswordField());
39         loginPanel.add(new JLabel("User ID:"));
40         loginPanel.add(new JTextField());
41         loginPanel.add(new JLabel("Password:"));
42         loginPanel.add(new JPasswordField());
43         loginPanel.add(new JButton("Login"));
44         paddingPanel.add(signUpPanel, BorderLayout.CENTER);
45         paddingPanel.add(loginPanel, BorderLayout.SOUTH);
46         getContentPane().add(paddingPanel);
47     }
48 }
```

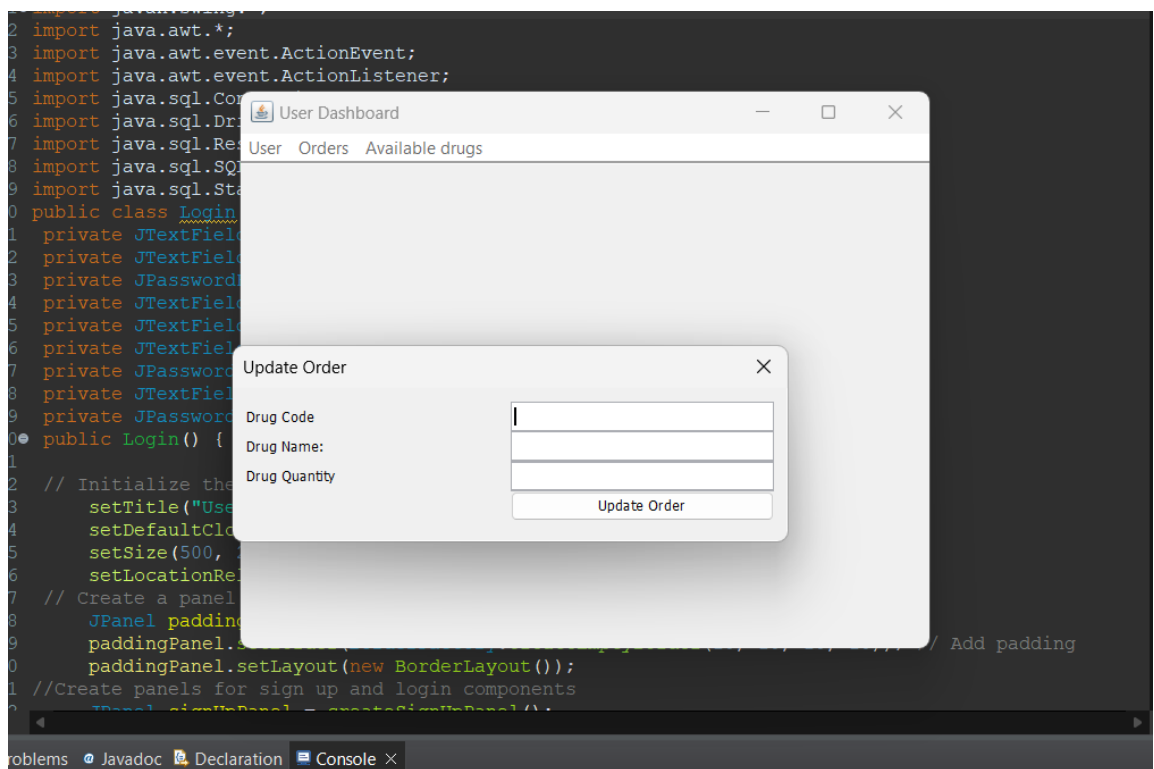
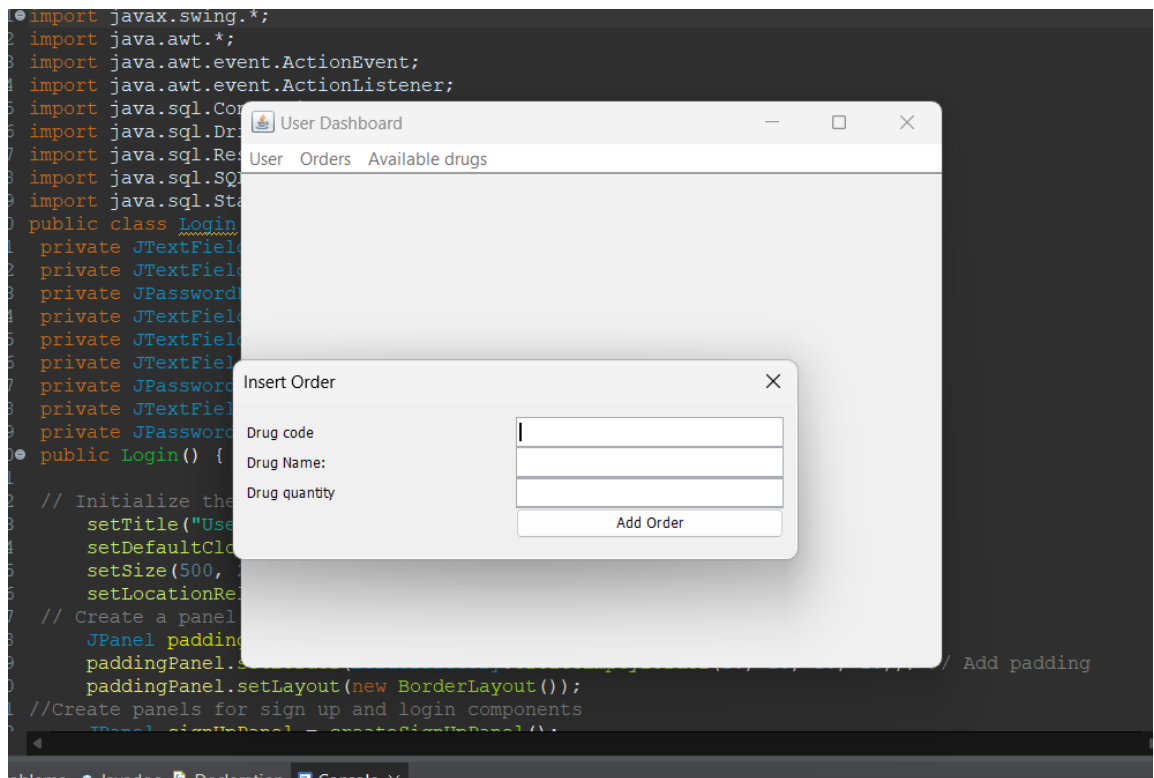


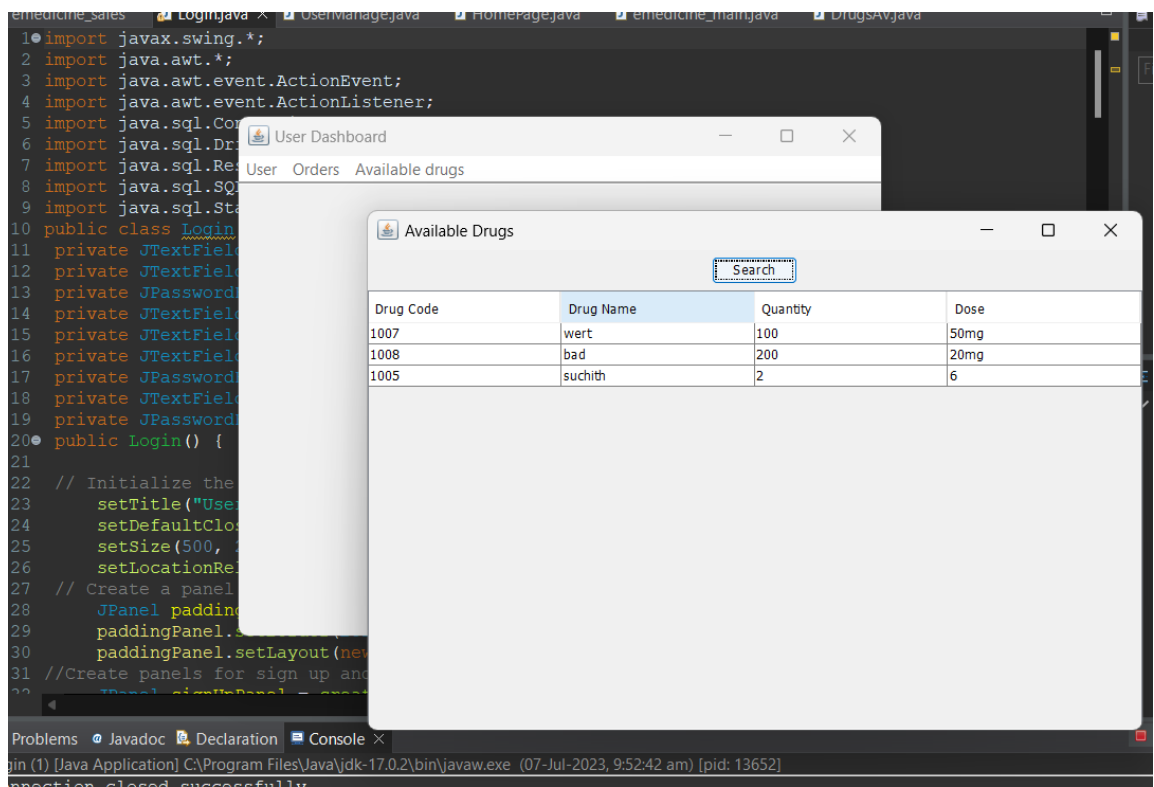
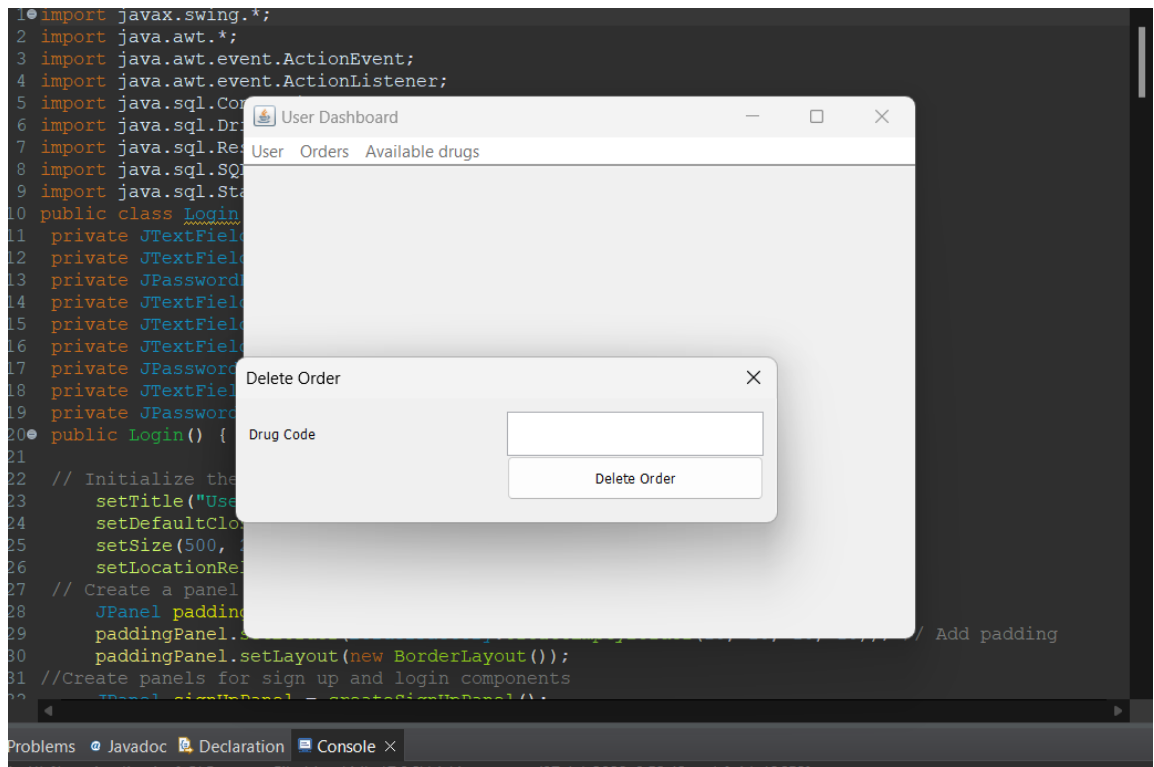
```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.sql.Statement;
10 public class UserSignUp/Login {
11     private JTextArea textArea;
12     private JTextField userLogin;
13     private JPasswordField passwordLogin;
14     private JTextField userSignUp;
15     private JTextField passwordSignUp;
16     private JPasswordField phoneSignUp;
17     private JPasswordField addressSignUp;
18     private JPasswordField loginButton;
19     private JPasswordField signUpButton;
20     public Login() {
21         // Initialize the window
22         setTitle("User Sign Up/Login");
23         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         setSize(500, 250);
25         setLocationRelativeTo(null);
26         // Create a panel for padding space
27         JPanel paddingPanel = new JPanel();
28         paddingPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10)); // Add padding
29         paddingPanel.setLayout(new BorderLayout());
30         // Create panels for sign up and login components
31         JPanel signUpPanel = new JPanel();
32         signUpPanel.setLayout(new BorderLayout());
33         signUpPanel.add(new JLabel("User ID:"));
34         signUpPanel.add(new JTextField());
35         signUpPanel.add(new JLabel("User Name:"));
36         signUpPanel.add(new JTextField());
37         signUpPanel.add(new JLabel("Password:"));
38         signUpPanel.add(new JPasswordField());
39         loginPanel.add(new JLabel("User ID:"));
40         loginPanel.add(new JTextField());
41         loginPanel.add(new JLabel("Password:"));
42         loginPanel.add(new JPasswordField());
43         loginPanel.add(new JButton("Login"));
44         paddingPanel.add(signUpPanel, BorderLayout.CENTER);
45         paddingPanel.add(loginPanel, BorderLayout.SOUTH);
46         getContentPane().add(paddingPanel);
47     }
48 }
```

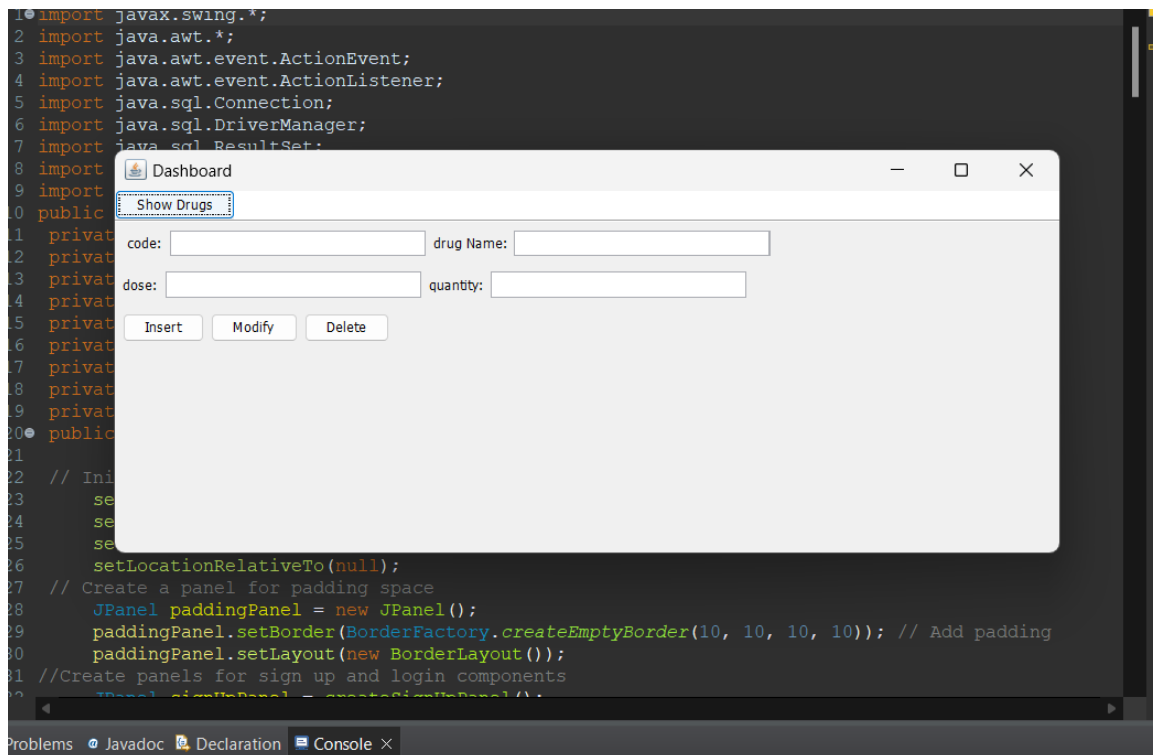
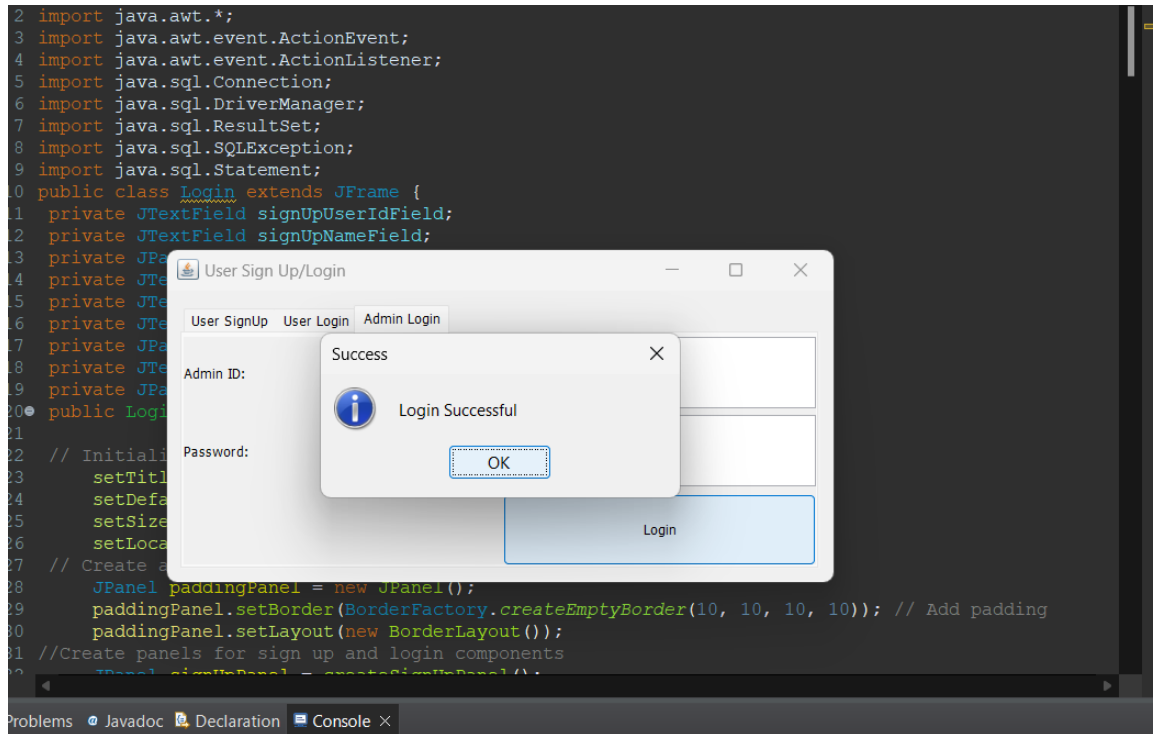


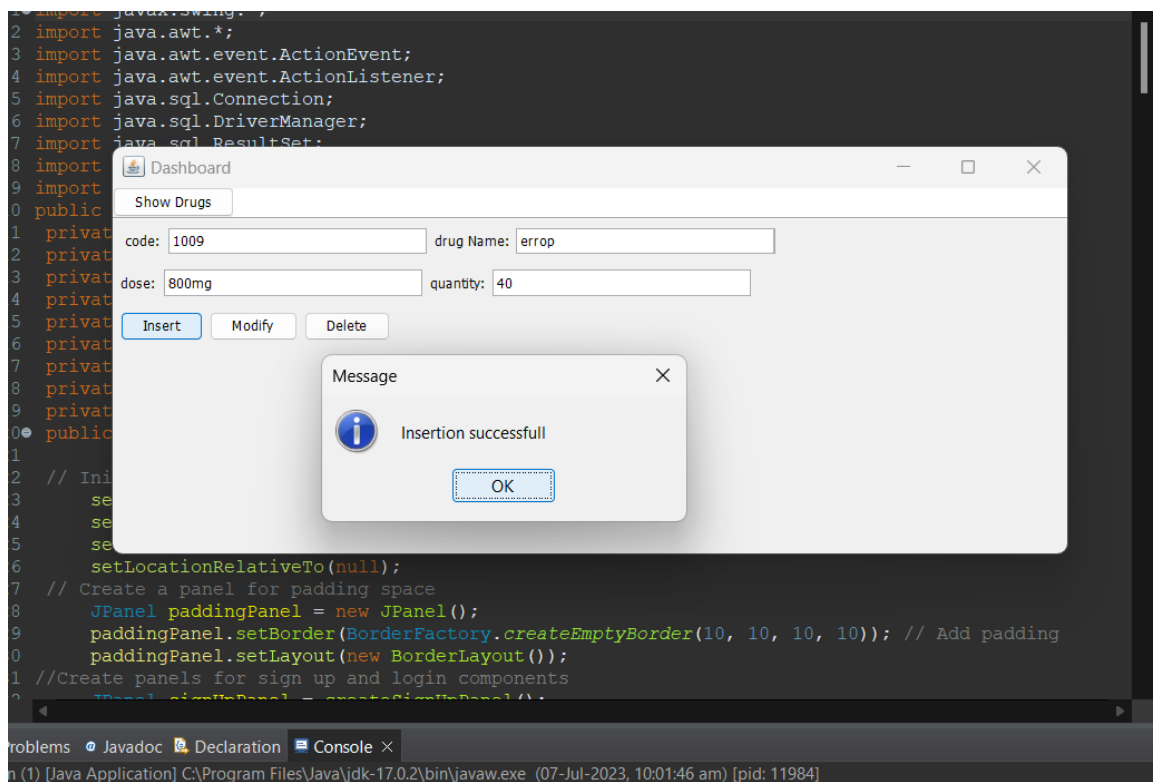
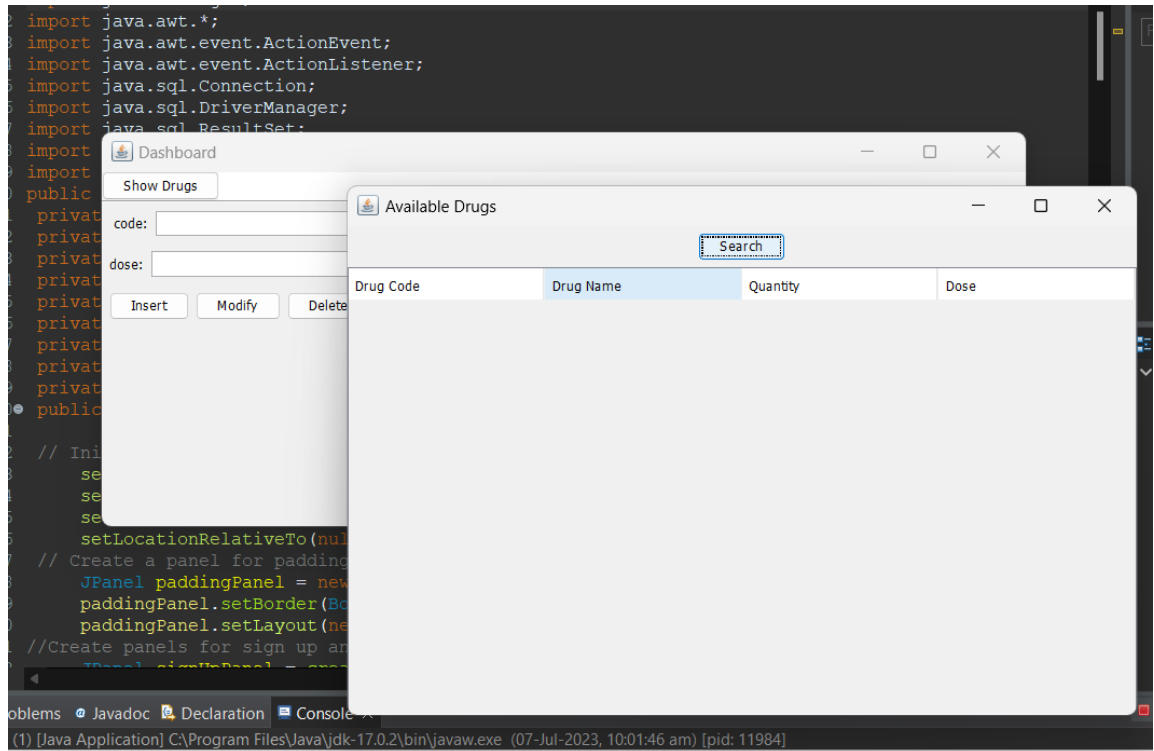


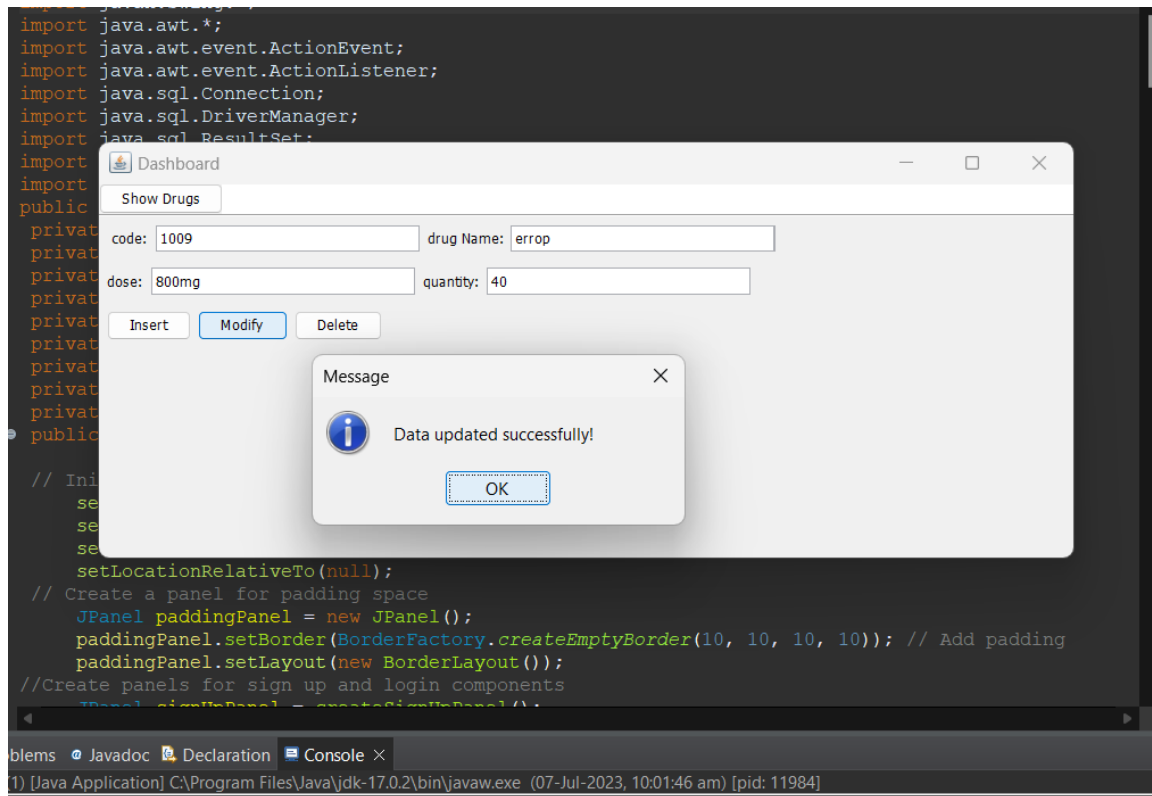


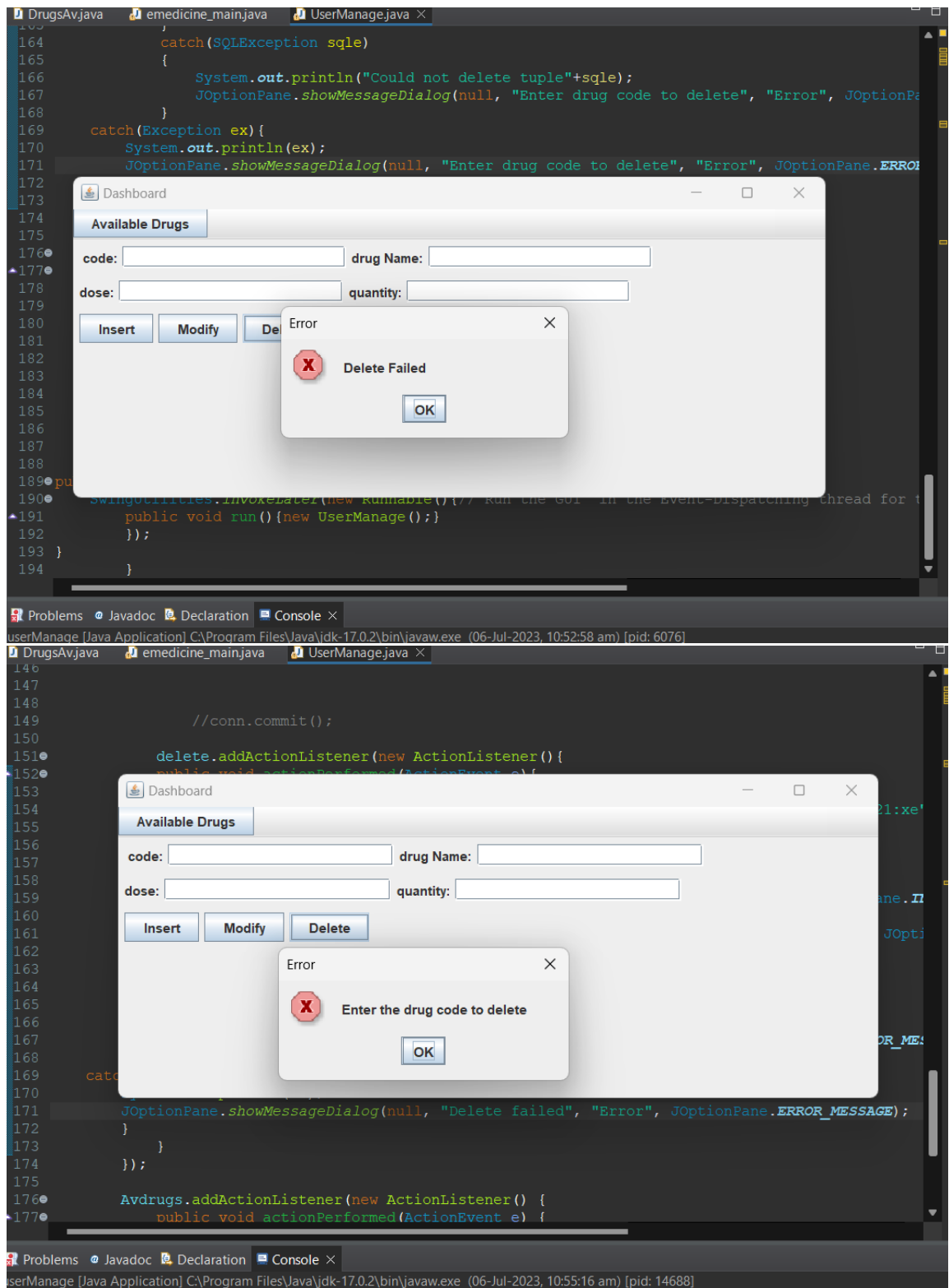












DrugsAV.java emedicine\_main.java UserManage.java

```
76 Connection con= DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","qawi","qawi");
77 Statement stmt=con.createStatement();
78 }
79 catch(SQLException e){System.out.println(e);}
80 ch(Exception ex){System.out.println(ex);}
81 submit.addActionListener(new ActionListener() {
82
83
84
85
86 code:  drug Name: 
87
88 dose:  quantity: 
89
90 Insert Modify Delete
91
92
93
94
95
96
97
98
99
100 JOptionPane.showMessageDialog(null, "Fill all the fields", "Error", JOptionPane.ERROR_MESSAGE);
101 }
102
103 ch(Exception ex){System.out.println(ex);}
104 }
105 });
106 modify.addActionListener(new ActionListener() {
107     public void actionPerformed(ActionEvent e) {
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151 delete.addActionListener(new ActionListener() {
152     public void actionPerformed(ActionEvent e) {
153     try{
154         Connection conn= DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe");
155         Statement stmt = conn.createStatement();
156         String sql = "DELETE FROM drugs WHERE code='"+ code_txt.getText()+"'";
157         int a=stmt.executeUpdate(sql);
158         if(a>0)
159             JOptionPane.showMessageDialog(null, "Delete successfull", "Message", JOptionPane.INFORMATION_MESSAGE);
160         else
161             JOptionPane.showMessageDialog(null, "Enter the drug code to delete", "Error", JOptionPane.ERROR_MESSAGE);
162         conn.close();
163     }
164     catch(Exception ex){
165         System.out.println(ex);
166     }
167     }
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Dashboard

Available Drugs

code:  drug Name:

dose:  quantity:

Insert Modify Delete

Error

Fill all the fields

OK

Problems Javadoc Declaration Console

UserManage [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (06-Jul-2023, 10:58:10 am) [pid: 6448]

DrugsAV.java emedicine\_main.java UserManage.java

```
151 delete.addActionListener(new ActionListener() {
152     public void actionPerformed(ActionEvent e) {
153     try{
154         Connection conn= DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe");
155         Statement stmt = conn.createStatement();
156         String sql = "DELETE FROM drugs WHERE code='"+ code_txt.getText()+"'";
157         int a=stmt.executeUpdate(sql);
158         if(a>0)
159             JOptionPane.showMessageDialog(null, "Delete successfull", "Message", JOptionPane.INFORMATION_MESSAGE);
160         else
161             JOptionPane.showMessageDialog(null, "Enter the drug code to delete", "Error", JOptionPane.ERROR_MESSAGE);
162         conn.close();
163     }
164     catch(Exception ex){
165         System.out.println(ex);
166     }
167     }
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Dashboard

Available Drugs

code: 2222 drug Name:

dose:  quantity:

Insert Modify Delete

Message

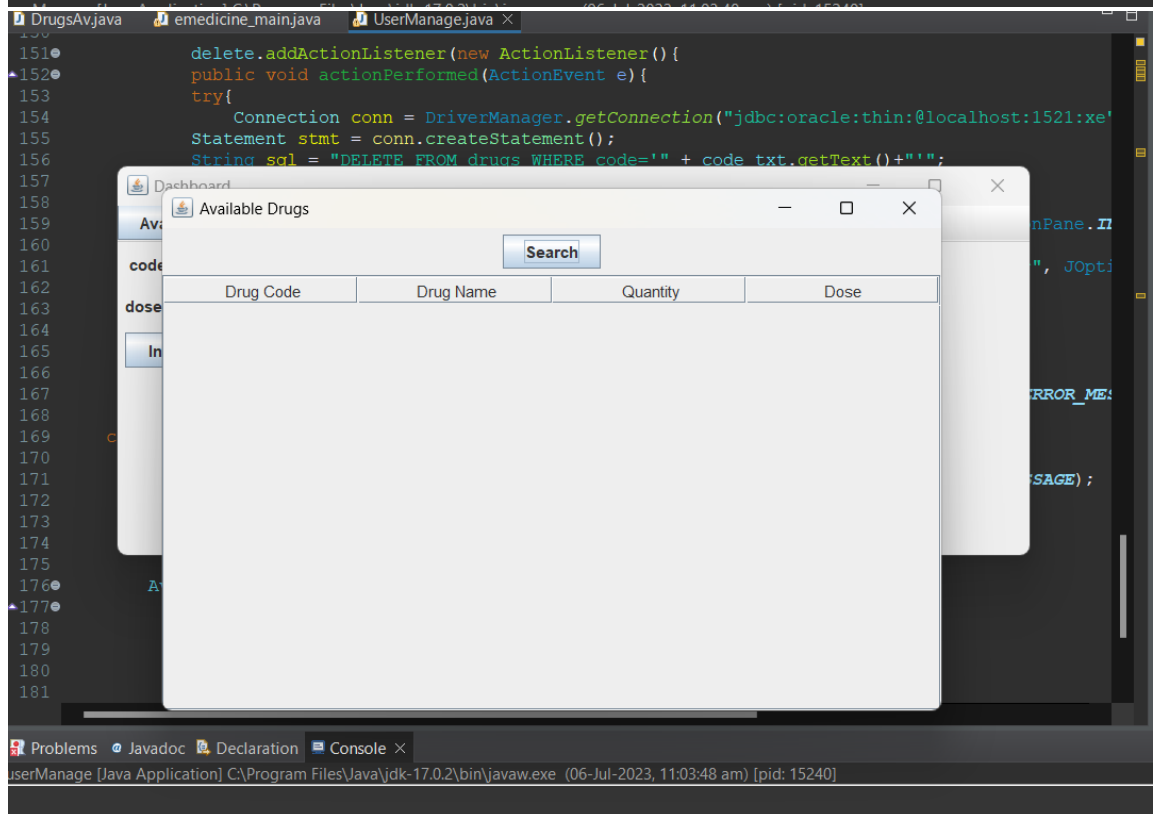
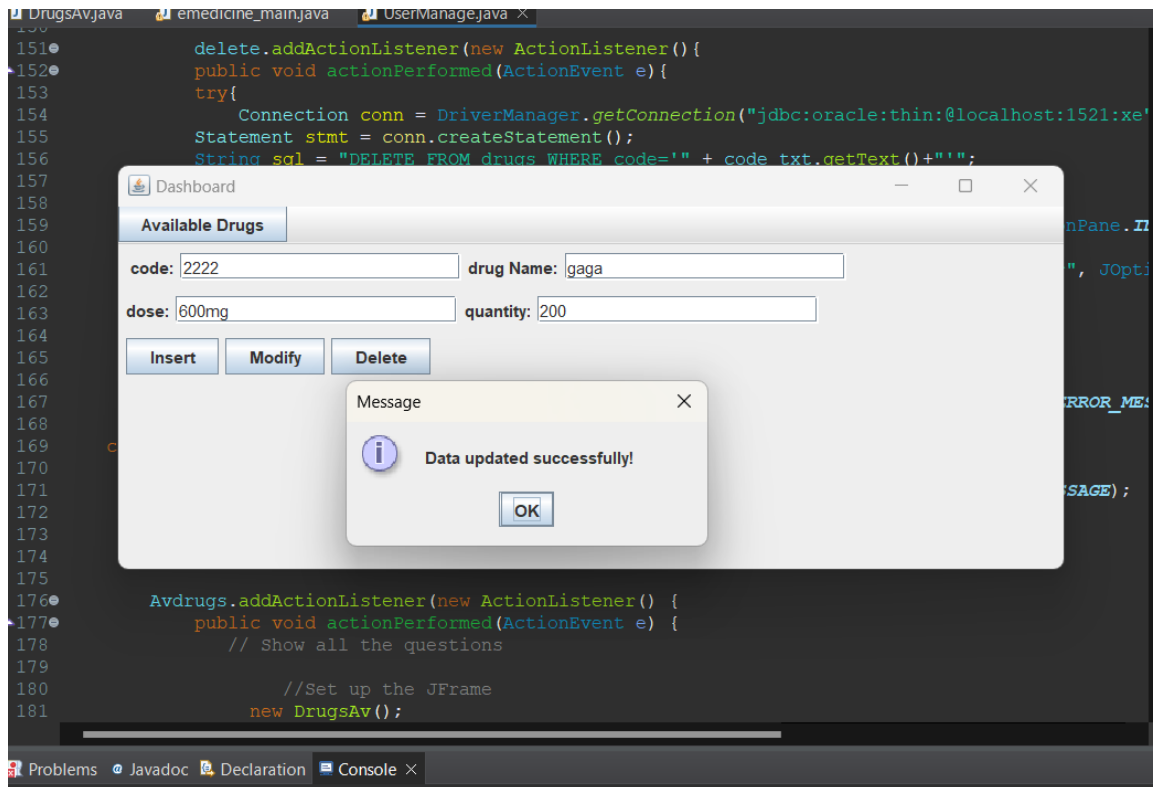
Delete successfull

OK

Problems Javadoc Declaration Console

UserManage [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (06-Jul-2023, 11:03:48 am) [pid: 15240]





DrugsAv.java emedicine\_main.java UserManage.java ×

```
151 delete.addActionListener(new ActionListener() {
152     public void actionPerformed(ActionEvent e) {
153     try{
154         Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe"
155         Statement stmt = conn.createStatement();
156
157
158
159
160
161
162
163
164
165
166
167
168
169 catch (Exception ex) {
170     System.out.println(ex.getMessage());
171     JOptionPane.showMessageDialog(null, "Error", "Error Message",
172     JOptionPane.ERROR_MESSAGE);
173 }
174 });
175
176 Avdr
177
178
179
180 //Set up the JFrame
181 new DrugsAv();
```

Available Drugs

Search

Drug Code	Drug Name	Quantity	Dose
1007	wert	100	50mg
1008	bad	200	20mg
1005	suchith	2	6

Problems Javadoc Declaration Console ×

UserManage [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (06-Jul-2023, 11:03:48 am) [pid: 15240]

## RESULTS

I have successfully completed the mini-project “E-Medicine sales system” .

## DISCUSSION AND FUTURE WORK

This project contains the basic interaction of giving information by students for suggesting the correct career choice. It has a very basic user interface. Future scope would be to make the UI more appealing by using graphics. more feature would be to allow student-users to upload their resumes and official One documents required so that we can suggest more accurate career choices. We can also think of including a feedback system to allow the users to leave their valuable feedback after using this app. Making this feedback to be publicly viewable, would attract many more users to use this app.

## REFERENCES

- <https://docs.oracle.com/javase/7/docs/api/>
- <https://www.javatpoint.com/java-swing>
- <https://stackoverflow.com>