JAVA SWING BASED- RECOMMENDED MEDICINES FOR AGED PEOPLE- SQLCONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

BEJUGAM SATHVIKA <1602-19-737-100>

Under the Guidance of

MS B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020-2021

BONAFIDE CERTIFICATE

This is to certify that this project report titled '**Recommended Medicines for Aged People**' is a project work of Ms. Bejugam Sathvika bearing roll no. 1602-19-737-100 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

external examiner

Signature

internal examiner

ABSTRACT

Whenever we are suffering with some health problems and there are some cases where we cannot consult the doctor or you may not know which medicine should be taken and how much dose to taken. In that scenario this application may help people.

Recommended medicines for aged people is a web application project that connects people looking for medicines with various types of diseases.Every user need to login first.User can go through the application where they can see type of diseases and the correct medicine for that disease.

The user can also see the company which is manufacturing that particular medicine.User can see the price of the medicine and how much he/she can consume that is they can even see the dosage for their respective medicine.

REQUIREMENT ANALYSIS

List of tables:

- LOGIN
- USERS
- DISEASES
- MEDICINES
- MEDICINE-MANUFACTURER

List of attributes with their domain types:

LOGIN:

- Login_id: number
- Password : varchar2()
- User_type: varchar2()

USER

- Login_id:number
- Username: varchar2()
- Age:number
- Gender: varchar2()
- Contact_number:number

DISEASES

- Disease_name: varchar2()
- Type_of_disease: varchar2()

MEDICINES

- Medicine_id:number,
- Medicine_name : varchar2()
- Price:number
- Dosage:number
- Disease_name: varchar2()

MEDICINES_MANUFACTURER

- Company_id:number
- Company_name : varchar2()
- Address: varchar2()
- Medicine_id:number()

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based recommended medicines for aged people like login id, username etc. from the person who uses the application it also stores information of the user and here only admin can insert or update or delete the information from tables. The user can only update his password, contact number and can view other tables. These values are tobe updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY USED:

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL Plus.

Java SWING:

SWING is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

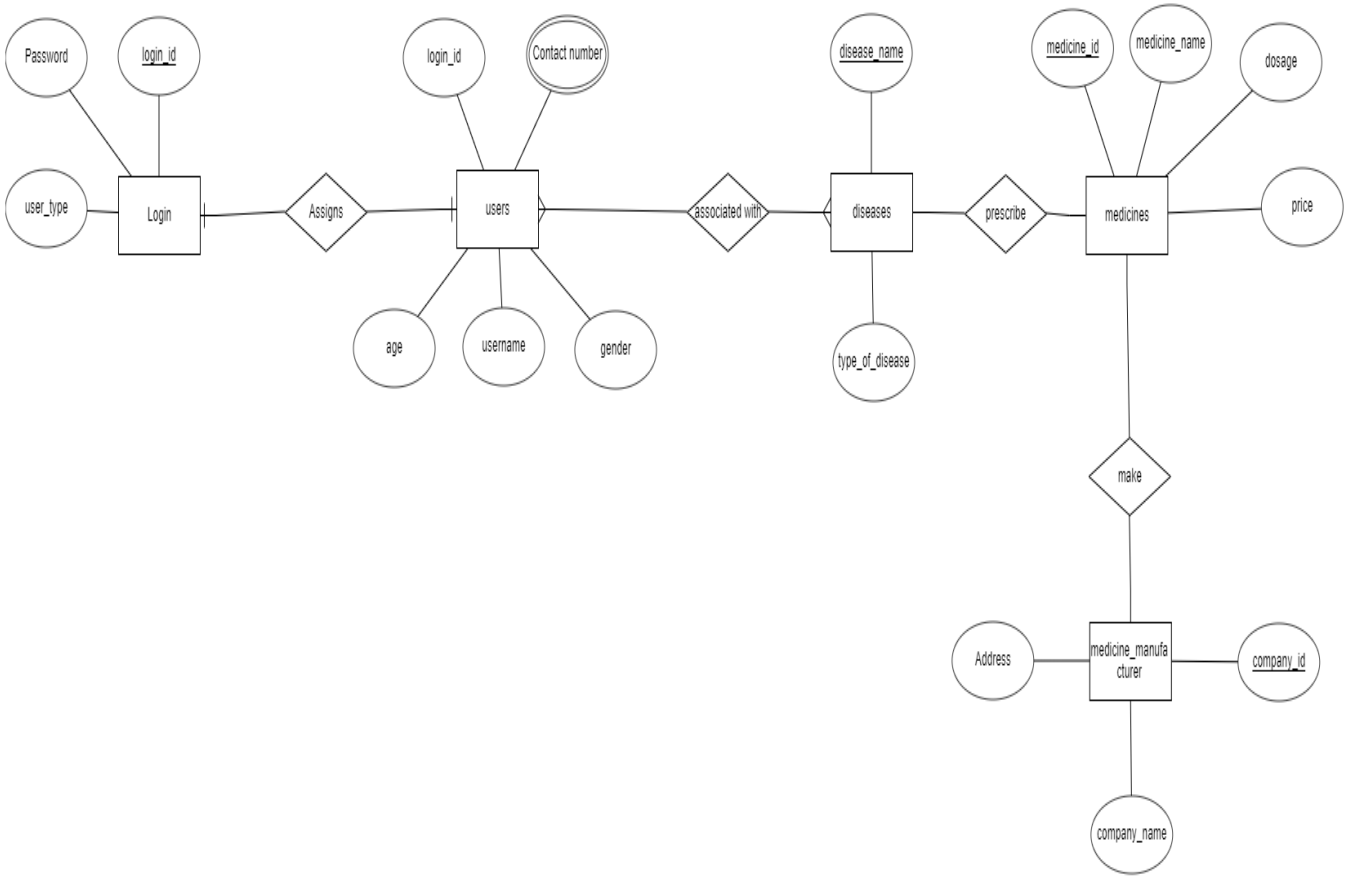
Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

DESIGN:

ER DIAGRAM:



DATA DESIGN:

Mapping Cardinalities and Participation Constraints:

Every user registered need to have one (only one) profile mandatorily. Therefore, this is a total participation and one-to-one relationship.
Every user who login into application need new user id and need to have access to all other parts. So login has full participation to users.
Also, every user have at least one disease, because that is the purpose of this application,therefore it is a total participation from the diseases.
A user can have any number of diseases at the same time one medicine can be applied to many diseases therefore this is a many-to-many relationship.

DDL COMMANDS:

1.LOGIN

```
create table login (
2 login_id number,
3 password varchar (8),
4 user_type varchar2(20));
```

```
SQL> create table login(
  2  login_id number,
  3  password varchar(8),
  4  user_type varchar2(20));

Table created.
```

- alter table login add primary key (login_id);

```
SQL> alter table login add primary key (login_id);

Table altered.
```

```
SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOGIN_ID                                  NOT NULL NUMBER
 PASSWORD                                           VARCHAR2(8)
 USER_TYPE                                          VARCHAR2(20)
```

2.USERS

```
create table users (
2 login_id number,
3 username varchar2(30),
4 age number,
5 gender varchar2(20),
6 contact_number number (10));
```

```
SQL> create table users(
  2  login_id number,
  3  username varchar2(30),
  4  age number,
  5  gender varchar2(20),
  6  contact_number number(10));

Table created.
```

-alter table users add foreign key(login_id) references login;

```
SQL> alter table users add foreign key(login_id) references login;

Table altered.

SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOGIN_ID                                           NUMBER
 USERNAME                                           VARCHAR2(30)
 AGE                                                NUMBER
 GENDER                                             VARCHAR2(20)
 CONTACT_NUMBER                                     NUMBER(10)
```

3.DISEASES

```
create table diseases (
2 disease_name varchar2(20),
3 typeof_disease varchar2 (20));
```

```
SQL> create table diseases(
  2  disease_name varchar2(20),
  3  type_of_disease varchar2(20));

Table created.
```

-alter table diseases add constraint pk_disease_name primary key(disease_name);

```
SQL> alter table diseases add constraint pk_disease_name primary key(disease_name);

Table altered.


SQL> desc diseases;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DISEASE_NAME                              NOT NULL VARCHAR2(20)
 TYPE_OF_DISEASE                                    VARCHAR2(20)
```

4.MEDICINES

create table medicines(
2 medicine_id number,
3 medicine_name varchar2(20) NOT NULL,
4 price number,
5 dosage number,
6 disease_name varchar2(20));

```
SQL> create table medicines(
  2  medicine_id number,
  3  medicine_name varchar2(20) NOT NULL,
  4  price number,
  5  dosage number,
  6  disease_name varchar2(20));

Table created.
```

-alter table medicines add primary key(medicine_id);
-alter table medicines add foreign key(disease_name) references diseases;

```
SQL> alter table medicines add primary key(medicine_id);

Table altered.

SQL> alter table medicines add foreign key(disease_name) references diseases;

Table altered.
```

```
SQL> desc medicines;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEDICINE_ID                               NOT NULL NUMBER
 MEDICINE_NAME                             NOT NULL VARCHAR2(20)
 PRICE                                              NUMBER
 DOSAGE                                             NUMBER
 DISEASE_NAME                                       VARCHAR2(20)
```

5.MEDICINE_MANUFACTURER

```
create table medicine_manufacturer(
2 company_id number,
3 company_name varchar2(30) NOT NULL,
4 address varchar2(30),
5 medicine_id number);
```

```
SQL> create table medicine_manufacturer(
  2  company_id number,
  3  company_name varchar2(30) NOT NULL,
  4  address varchar2(30),
  5  medicine_id number);

Table created.
```

-alter table medicine_manufacturer add primary key(company_id);
-alter table medicine_manufacturer add foreign key(medicine_id) references medicines;

```
SQL> alter table medicine_manufacturer add primary key(company_id);

Table altered.

SQL> alter table medicine_manufacturer add foreign key(medicine_id) references medicines;

Table altered.

QL> desc medicine_manufacturer;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
COMPANY_ID                                NOT NULL NUMBER
COMPANY_NAME                              NOT NULL VARCHAR2(30)
ADDRESS                                            VARCHAR2(30)
MEDICINE_ID                                        NUMBER
```

DML COMMANDS:

1.LOGIN TABLE

```
SQL> insert into login values(&login_id,'&password','&user_type');
Enter value for login_id: 1
Enter value for password: 123456
Enter value for user_type: admin
old   1: insert into login values(&login_id,'&password','&user_type')
new   1: insert into login values(1,'123456','admin')

1 row created.

SQL> /
Enter value for login_id: 2
Enter value for password: 56789
Enter value for user_type: user
old   1: insert into login values(&login_id,'&password','&user_type')
new   1: insert into login values(2,'56789','user')

1 row created.

SQL> select * from login;

  LOGIN_ID PASSWORD USER_TYPE
---------- -------- --------------------
         1 123456   admin
         2 56789    user
```

2.USERS TABLE

```
SQL> insert into users values(&login_id,'&username',&age,'&gender',&contact_number);
Enter value for login_id: 1
Enter value for username: sathvika
Enter value for age: 19
Enter value for gender: female
Enter value for contact_number: 9390205320
old   1: insert into users values(&login_id,'&username',&age,'&gender',&contact_number)
new   1: insert into users values(1,'sathvika',19,'female',9390205320)

1 row created.

SQL> /
Enter value for login_id: 2
Enter value for username: rahul
Enter value for age: 23
Enter value for gender: male
Enter value for contact_number: 9177599923
old   1: insert into users values(&login_id,'&username',&age,'&gender',&contact_number)
new   1: insert into users values(2,'rahul',23,'male',9177599923)

1 row created.

SQL> select * from users;

  LOGIN_ID USERNAME                         AGE GENDER
---------- ------------------------------ ---------- --------------------
CONTACT_NUMBER
--------------
         1 sathvika                        19 female
    9390205320

         2 rahul                           23 male
    9177599923
```

3.DISEASES TABLE

```
SQL> insert into diseases values('&disease_name','&type_of_disease');
Enter value for disease_name: corona
Enter value for type_of_disease: infectious
old   1: insert into diseases values('&disease_name','&type_of_disease')
new   1: insert into diseases values('corona','infectious')

1 row created.


SQL> /
Enter value for disease_name: typhoid
Enter value for type_of_disease: contagious
old   1: insert into diseases values('&disease_name','&type_of_disease')
new   1: insert into diseases values('typhoid','contagious')

1 row created.
```

```
SQL> /
Enter value for disease_name: cold and cough
Enter value for type_of_disease: viral infection
old   1: insert into diseases values('&disease_name','&type_of_disease')
new   1: insert into diseases values('cold and cough','viral infection')

1 row created.


SQL> /
Enter value for disease_name: cancer
Enter value for type_of_disease: genetic change
old   1: insert into diseases values('&disease_name','&type_of_disease')
new   1: insert into diseases values('cancer','genetic change')

1 row created.


SQL> /
Enter value for disease_name: mild fever
Enter value for type_of_disease: inflammatory
old   1: insert into diseases values('&disease_name','&type_of_disease')
new   1: insert into diseases values('mild fever','inflammatory')

1 row created.

SQL> select * from diseases;

DISEASE_NAME          TYPE_OF_DISEASE
--------------------- --------------------
corona                infectious
typhoid               contagious
cold and cough        viral infection
cancer                genetic change
mild fever            inflammatory
```

4.MEDICINES TABLE

```
SQL> insert into medicines values(&medicine_id,'&medicine_name',&price,&dosage,'&disease_name');
Enter value for medicine_id: 1
Enter value for medicine_name: paracetomol
Enter value for price: 5
Enter value for dosage: 1
Enter value for disease_name: mild fever
old   1: insert into medicines values(&medicine_id,'&medicine_name',&price,&dosage,'&disease_name')
new   1: insert into medicines values(1,'paracetomol',5,1,'mild fever')

1 row created.

SQL> /
Enter value for medicine_id: 2
Enter value for medicine_name: dolo65
Enter value for price: 10
Enter value for dosage: 1
Enter value for disease_name: fever
old   1: insert into medicines values(&medicine_id,'&medicine_name',&price,&dosage,'&disease_name')
new   1: insert into medicines values(2,'dolo65',10,1,'fever')

1 row created.

SQL> select * from medicines;

MEDICINE_ID MEDICINE_NAME             PRICE     DOSAGE DISEASE_NAME
----------- -------------------- ---------- ---------- --------------------
          1 paracetomol                   5          1 mild fever
          2 dolo65                       10          1 fever
```

5.MEDICINE_MANUFACTURER

```
SQL> insert into medicine_manufacturer values(&company_id,'&company_name','&address',&medicine_id);
Enter value for company_id: 1
Enter value for company_name: aristro
Enter value for address: 45/hyd
Enter value for medicine_id: 1
old   1: insert into medicine_manufacturer values(&company_id,'&company_name','&address',&medicine_id)
new   1: insert into medicine_manufacturer values(1,'aristro','45/hyd',1)

1 row created.
```

```
SQL> /
Enter value for company_id: 2
Enter value for company_name: dr.reddys
Enter value for address: 654-kukatpally
Enter value for medicine_id: 1
old   1: insert into medicine_manufacturer values(&company_id,'&company_name','&address',&medicine_id)
new   1: insert into medicine_manufacturer values(2,'dr.reddys','654-kukatpally',1)

1 row created.

SQL> select * from medicine_manufacturer;

COMPANY_ID COMPANY_NAME                   ADDRESS
---------- ------------------------------ ------------------------------
MEDICINE_ID
-----------
         1 aristro                        45/hyd
          1

         2 dr.reddys                      654-kukatpally
          1
```

Implementation:

Front end programs and its connectivity

Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how aclient 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 queryand 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:

```
public void connectToDB()
  {
                try
                {
                connection

                =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");
                statement = connection.createStatement();

                }
                catch (SQLException connectException)
                {
                 System.out.println(connectException.getMessage())
                 ;
                 System.out.println(connectException.getSQLState()
                 );
                 System.out.println(connectException.getErrorCode());
                 System.exit(1);
                }
  }
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Program:

LOGININTO:

```
package RM;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import RM.*;
import java.sql.*;
import javax.sql.*;

import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.event.MenuEvent;
import javax.swing.event.MenuListener;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class logininto extends JFrame{

	private JFrame recommendedmedicines;
	private JTextField txtLoginId;
	private JPasswordField txtPassword;
	//private JTextField txtUserName;
	private JPanel contentPane;
	public static String user,user1;
	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
```

```
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					logininto window = new logininto();
					window.recommendedmedicines.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the application.
	 */
	public logininto() {
		initialize();
	}

	/**
	 * Initialize the contents of the frame.
	 */
	JButton btnLogin;
	JButton btnSignup;
	private void initialize() {




		recommendedmedicines = new JFrame();
		recommendedmedicines.setTitle("Recommended Medicines For Aged people");
		recommendedmedicines.setBounds(100, 100, 450, 300);
```

```
recommendedmedicines.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
recommendedmedicines.getContentPane().setLayout(null);



JMenuBar menuBar = new JMenuBar();
recommendedmedicines.setJMenuBar(menuBar);



JMenu mnNewMenu = new JMenu("Log In");
mnNewMenu.addMenuListener(new MenuListener() {
        public void menuCanceled(MenuEvent e) {
        }
        public void menuDeselected(MenuEvent e) {
        }
        public void menuSelected(MenuEvent e) {
                btnLogin.setVisible(true);
                btnSignup.setVisible(false);


        }
});

JMenu mnNewMenu1 = new JMenu("Signup");
mnNewMenu1.addMenuListener(new MenuListener() {
        public void menuCanceled(MenuEvent e) {
        }
        public void menuDeselected(MenuEvent e) {
        }
        public void menuSelected(MenuEvent e) {
                btnLogin.setVisible(false);
                btnSignup.setVisible(true);
        }
```

```
        });
        menuBar.add(mnNewMenu);
        menuBar.add(mnNewMenu1);

        txtLoginId = new JTextField();
        txtLoginId.setBounds(198, 47, 96, 20);
        recommendedmedicines.getContentPane().add(txtLoginId);
        txtLoginId.setColumns(10);



        JLabel lblNewLabel = new JLabel("Login Id");
        lblNewLabel.setBounds(76, 50, 84, 14);
        recommendedmedicines.getContentPane().add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Password");
        lblNewLabel_1.setBounds(76, 88, 84, 14);
        recommendedmedicines.getContentPane().add(lblNewLabel_1);



        btnLogin = new JButton("Login");
        btnLogin.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                boolean isValidUser=false;
                try {
                    connecttoDB connect = new connecttoDB();
                    Connection con = connect.getConnection();
                    Statement stmt=con.createStatement();
                    user1=txtLoginId.getText();
                    //String txt = "select login_id,password,user_type from login where login_id
```

```
= '"+txtLoginId.getText()+"' and password ='"+new String(txtPassword.getPassword())"";
                                        //System.out.println(txt);
                                        String txt = "select login_id,password,user_type from login where login_id =
'"+txtLoginId.getText()+"' and password ='"+new String(txtPassword.getPassword())+"";
                                        ResultSet rs=stmt.executeQuery(txt);
                                        if(rs.next()) {
                                                isValidUser = true;
                                                user=rs.getString(3);


                                        }
                                        rs.close();


                                }
                                catch(Exception ex) {
                                        System.out.println(ex);
                                }
                                if(isValidUser) {
                                        //JOptionPane.showMessageDialog(null,"Valid User");
                                        RMUI h=new RMUI();
                                        //login code here
                                }
                                else {
                                        JOptionPane.showMessageDialog(null,"Invalid User");
                                }


                        }
                });
                btnLogin.setBounds(150, 160, 89, 23);
                recommendedmedicines.getContentPane().add(btnLogin);


                btnSignup = new JButton("Signup");
```

```
		btnSignup.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				try {
					connecttoDB connect = new connecttoDB();
					Connection con = connect.getConnection();
					Statement stmt=con.createStatement();
					String txt1 = "insert into login (login_id,password,user_type) values('"+txtLoginId.getText()+"' ,'"+new String(txtPassword.getPassword())+"','"+"user"+"')";
					//System.out.println(txt);
					ResultSet rs1=stmt.executeQuery(txt1);

					String txt = "select login_id,password,user_type from login where login_id = '"+txtLoginId.getText()+"' and password ='"+new String(txtPassword.getPassword())+"'";
					ResultSet rs=stmt.executeQuery(txt);
					user1=txtLoginId.getText();
					if(rs.next()) {
						JOptionPane.showMessageDialog(null,"Sign in successful");
						user=rs.getString(3);
						RMUI r=new RMUI();
					}
					rs.close();

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

			}
		});
		btnSignup.setBounds(150, 160, 89, 23);
		recommendedmedicines.getContentPane().add(btnSignup);
```

```
		txtPassword = new JPasswordField();
		txtPassword.setBounds(198, 85, 96, 20);
		recommendedmedicines.getContentPane().add(txtPassword);
	}
}
```

CONNECTTODB:

```
package RM;
import java.sql.*;
public class connecttoDB{
	public Connection getConnection() {
		try {
			Class.forName("oracle.jdbc.driver.OracleDriver");
			Connection conn=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");
			return conn;
		}
		catch(Exception e) {
			System.out.println(e);
		}
		return null;
	}
}
```

USER INTERFACE:

```
package RM;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.event.WindowAdapter;
```

```
import java.awt.event.WindowEvent;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;



public class RMUI extends JFrame{
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;

	private JMenuBar mnu;

	private JMenu mnuLogin;
	private JMenu mnuuser;
	private JMenu mnudiseases;
	private JMenu mnuMedicines;
	private JMenu mnuMedicines_manufacturer;

	private JMenuItem insert1,update1,delete1,view1;
	private JMenuItem insert3,update3,delete3,view3;
	private JMenuItem insert4,update4,delete4,view4;
	private JMenuItem insert5,update5,delete5,view5;
	private JMenuItem insert6,update6,delete6,view6;
```

```
    private JLabel labelName;



    private static JPanel p0,p1;



    void initialize() {
            mnu=new JMenuBar();

            mnuLogin= new JMenu("Login");
            mnuuser= new JMenu("User");
            mnudiseases= new JMenu("Diseases");
            mnuMedicines= new JMenu("Medicines");
            mnuMedicines_manufacturer= new JMenu("Medicines_manufacturer");
            labelName=new JLabel("RECOMMENDED MEDICINES FOR AGED PEOPLE");
             p1=new JPanel();
            p0=new JPanel();
            insert1=new JMenuItem("Insert");
            update1=new JMenuItem("Update");
            delete1=new JMenuItem("Delete");
            view1=new JMenuItem("View");
            insert3=new JMenuItem("Insert");
            update3=new JMenuItem("Update");
            delete3=new JMenuItem("Delete");
            view3=new JMenuItem("View");
            insert4=new JMenuItem("Insert");
            update4=new JMenuItem("Update");
            delete4=new JMenuItem("Delete");
            view4=new JMenuItem("View");
            insert5=new JMenuItem("Insert");
            update5=new JMenuItem("Update");
```

```
            delete5=new JMenuItem("Delete");
            view5=new JMenuItem("View");
            insert6=new JMenuItem("Insert");
            update6=new JMenuItem("Update");
            delete6=new JMenuItem("Delete");
            view6=new JMenuItem("View");

    }
    void addComponentsToFrame() {
            if(logininto.user.equals("admin")) {
            mnuLogin.add(insert1);
             mnuLogin.add(delete1);
             mnuLogin.add(update1);
             mnuLogin.add(view1);
             mnuuser.add(insert3);
             mnuuser.add(delete3);
             mnuuser.add(update3);
             mnuuser.add(view3);
             mnudiseases.add(insert4);
             mnudiseases.add(delete4);
             mnudiseases.add(update4);
             mnudiseases.add(view4);
             mnuMedicines.add(insert5);
             mnuMedicines.add(delete5);
             mnuMedicines.add(update5);
             mnuMedicines.add(view5);
             mnuMedicines_manufacturer.add(insert6);
             mnuMedicines_manufacturer.add(delete6);
             mnuMedicines_manufacturer.add(update6);
             mnuMedicines_manufacturer.add(view6);
             mnu.add(mnuLogin);
```

```
                mnu.add(mnuuser);
                mnu.add(mnudiseases);
                mnu.add(mnuMedicines);
                mnu.add(mnuMedicines_manufacturer);
               }

               else if(logininto.user.equals("user"))
               {
                        mnuLogin.add(update1);
                        mnuuser.add(update3);
                        //mnuuser.add(view3);
                        mnudiseases.add(view4);
                        mnuMedicines.add(view5);
                        mnuMedicines_manufacturer.add(view6);
                        mnu.add(mnuLogin);
                         mnu.add(mnuuser);
                         mnu.add(mnudiseases);
                         mnu.add(mnuMedicines);
                         mnu.add(mnuMedicines_manufacturer);
               }

                setJMenuBar(mnu);
                p1.add(labelName);p1.setAlignmentY(CENTER_ALIGNMENT);
                p1.setBounds(500,500,800,100);
               p0.add(p1);
                p0.setBackground(Color.GRAY);
                add(p0);
        }
void closeWindow(){
               try {
                        int a=JOptionPane.showConfirmDialog(this,"Are you sure want to Recommended Medicines
```

```
for Aged People:");
                        if(a==JOptionPane.YES_OPTION){
                                JOptionPane.showMessageDialog(this,
                                        "Thank you!\nRecommended Medicines for Aged People","Quit",
                                        JOptionPane.WARNING_MESSAGE);
                                System.exit(0);
                        }
                        else if (a== JOptionPane.NO_OPTION) {
                                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
                        }
                        else if (a== JOptionPane.CANCEL_OPTION) {
                                setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
                        }
                }
                        catch(Exception e) {
                                System.out.println(e);
                                }
        }
void register() {
                Login log=new Login(p0,RMUI.this,insert1,delete1,update1,view1);
                log.buildGUI();
                users user=new users(p0,RMUI.this,insert3,delete3,update3,view3);
                user.buildGUI();
                diseases dis=new diseases(p0,RMUI.this,insert4,delete4,update4,view4);
                dis.buildGUI();
                Medicines med=new Medicines(p0,RMUI.this,insert5,delete5,update5,view5);
                med.buildGUI();
                medicine_manufacturer                                                   comp=new
medicine_manufacturer(p0,RMUI.this,insert6,delete6,update6,view6);
                comp.buildGUI();
                addWindowListener(new WindowAdapter(){
```

```
                        public void windowClosing(WindowEvent we)
                        {
                                closeWindow();
                        }
                });


}
public RMUI() {
                initialize();
                addComponentsToFrame();
                register();
                pack();
                setTitle("Recommended Medicines for Aged People");
                setSize(800,800);
                setVisible(true);
        }
}
```

LOGIN :

```
package RM;


import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
```

```
import java.sql.*;
import java.util.StringTokenizer;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class Login {
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	private JButton LoginButton,deleteButton,viewButton;
	private JPanel p1,p2,p3,p;
	private JLabel lbllogin_id,lblpwd,lblut;
	private JTextField txtlogin_id,txtpwd,txtut;
	private List users_idList;
	Connection con;ResultSet rs;
	Statement statement;
	private JFrame frame;
	private JMenuItem insert,delete,update,view;
	public Login(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem
```

```
view)
        {

                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connectToDB();
                this.frame=frame;
                this.insert=insert;
                this.delete=delete;
                this.update=update;
                this.view=view;
                lbllogin_id=new JLabel("Login_ID");
                lblpwd=new JLabel("Password");
                lblut=new JLabel("User_Type");

                txtlogin_id=new JTextField(30);
                txtpwd=new JPasswordField(8);
                txtut=new JTextField(20);

                this.p=p;



        }
```

```
	public void connectToDB()
  {
		try {


		Connection con=DriverManager.getConnection(
		"jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");


		statement=con.createStatement();
		statement.executeUpdate("commit");



		}
		catch (SQLException connectException)
		{
		  System.out.println(connectException.getMessage());
		  System.out.println(connectException.getSQLState());
		  System.out.println(connectException.getErrorCode());
		  System.exit(1);
		}
  }
	private void displaySQLErrors(SQLException e)
	{
		JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:      " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");



	}
	public void loadLogins() {
```

```
		try {
			users_idList.removeAll();
			rs=statement.executeQuery("select * from login");
			while(rs.next()) {
				users_idList.add(rs.getString("login_id"));
			}
			}
		catch(SQLException e) {
			displaySQLErrors(e);
		}
	}

	public void buildGUI() {
		if(logininto.user.equals("admin")) {
		LoginButton=new JButton("submit");

		insert.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent arg0) {
				// TODO Auto-generated method stub

				txtlogin_id.setText(null);
				txtpwd.setText(null);
				txtut.setText(null);

				p.removeAll();
				frame.invalidate();
				frame.validate();
				frame.repaint();
```

```
p1=new JPanel();

 p1.setLayout(new GridLayout(4,2));
 p1.add(lbllogin_id);
 p1.add(txtlogin_id);
 p1.add(lblpwd);
 p1.add(txtpwd);
 p1.add(lblut);
 p1.add(txtut);

 p3=new JPanel(new FlowLayout());
 p3.add(LoginButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
 p1.setBackground(Color.WHITE) ;

 p1.setBounds(115,80,300,200);

 p2 = new JPanel(new FlowLayout());
          // p2.add(txtf1);
           users_idList=new List(10);
           loadLogins();
           p2.add(users_idList);p2.setBackground(Color.WHITE) ;
           p2.setBounds(125,320,300,180);



 p. add(p1);
 p.add(p3);
 p. add(p2);
```

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


                        frame.add(p);
                        frame.setSize(800,800);
                        frame.validate();
                LoginButton.addActionListener(new ActionListener() {
                        @Override
                                public void actionPerformed(ActionEvent e) {
                                        // TODO Auto-generated method stub
                try {
                        String query="INSERT INTO Login
VALUES('"+txtlogin_id.getText()+"','"+txtpwd.getText()+"','"+txtut.getText()+"')";


                        int i=statement.executeUpdate(query);
                        JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");
                        loadLogins();


                        System.out.println("done");
                }
                catch(SQLException insertException){
                        displaySQLErrors(insertException);
                }


                }



                        });
        }
```

```
                });

        delete.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub
                        deleteButton=new JButton("delete");

                        txtlogin_id.setText(null);
                        txtpwd.setText(null);
                        txtut.setText(null);

                        p.removeAll();
                        frame.invalidate();
                        frame.validate();
                        frame.repaint();


                        p1=new JPanel();

                         p1.setLayout(new GridLayout(4,2));
                         p1.add(lbllogin_id);
                         p1.add(txtlogin_id);
                         p1.add(lblpwd);
                         p1.add(txtpwd);
                         p1.add(lblut);
                         p1.add(txtut);

                        // p1.add(deleteButton);
                         p3=new JPanel(new FlowLayout());
```

```
p3.add(deleteButton);
//p1.add(txtf1);
p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.WHITE) ;

p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
         users_idList=new List(10);
         loadLogins();
         p2.add(users_idList);p2.setBackground(Color.WHITE) ;
         p2.setBounds(125,320,300,180);



p. add(p1);
p.add(p3);
p. add(p2);



p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
users_idList.addItemListener(new ItemListener()
        {
                public void itemStateChanged(ItemEvent e)
                {
                        try
```

```
                                        {
                                                rs=statement.executeQuery("select * from login");
                                                StringTokenizer                         st=new
StringTokenizer(users_idList.getSelectedItem(),"->");
                                                String p=st.nextToken();
                                                while (rs.next())
                                                {
                                                        if (rs.getString("login_id").equals(p))
                                                        break;
                                                }
                                                if (!rs.isAfterLast())
                                                {
                                                        txtlogin_id.setText(rs.getString("login_id"));
                                                        txtpwd.setText(rs.getString("password"));
                                                        txtut.setText(rs.getString("user_type"));

                                                }
                                        }
                                        catch (SQLException selectException)
                                        {
                                                displaySQLErrors(selectException);
                                        }
                                }
                        });


                deleteButton.addActionListener(new ActionListener() {
                        @Override
                                public void actionPerformed(ActionEvent e) {
                                        // TODO Auto-generated method stub
                try {
```

```
                                        int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                                        if(a==JOptionPane.YES_OPTION){
                                        //String query="DELETE FROM Login WHERE login_id="+txtlogin_id.getText()+"AND PASSWORD="'+txtpwd.getText()+"'";
                                                StringTokenizer st=new StringTokenizer(users_idList.getSelectedItem(),"->");

                                                String query="DELETE FROM login WHERE login_id="+st.nextToken();
                                        int i=statement.executeUpdate(query);
                                        JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");
                                        loadLogins();
                                        }


                                }
                                catch(SQLException deleteException){
                                        displaySQLErrors(deleteException);
                                }


                                 }



                                        });
                        }
                        });
        }

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

```
@Override
public void actionPerformed(ActionEvent arg0) {
	// TODO Auto-generated method stub
	JButton updateButton = new JButton("modify");
	txtlogin_id.setText(null);

	txtpwd.setText(null);
	txtut.setText(null);

	p.removeAll();
	frame.invalidate();
	frame.validate();
	frame.repaint();



	p1=new JPanel();

	 p1.setLayout(new GridLayout(4,2));
	 p1.add(lbllogin_id);
	 p1.add(txtlogin_id);
	 p1.add(lblpwd);
	 p1.add(txtpwd);
	 p1.add(lblut);
	 p1.add(txtut);



	 p3=new JPanel(new FlowLayout());
	 p3.add(updateButton);
	 //p1.add(txtf1);
	 p3.setBackground(Color.BLACK);
	p3.setBounds(200,280,75,35);
```

```
p1.setBackground(Color.WHITE) ;

p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
         users_idList=new List(10);
         //loadLogins();
         if(logininto.user.equals("user")) {
                  users_idList.removeAll();
                  users_idList.add(logininto.user1);
         }
         if(logininto.user.equals("admin")) {
                  loadLogins();
         }
         p2.add(users_idList);
         p2.add(users_idList);p2.setBackground(Color.WHITE) ;
         p2.setBounds(125,320,300,180);


p. add(p1);
p.add(p3);
p. add(p2);


p.setLayout(new BorderLayout());

         frame.add(p);
         frame.setSize(800,800);
         frame.validate();
users_idList.addItemListener(new ItemListener()
```

```
                                        {
                                                public void itemStateChanged(ItemEvent e)
                                                {
                                                        try
                                                        {
                                                                rs=statement.executeQuery("select * from login");
                                                                //StringTokenizer                       st=new
StringTokenizer(users_idList.getSelectedItem(),"->");
                                                                //String p=st.nextToken();
                                                                while (rs.next())
                                                                {
                                                                        if
(rs.getString("login_id").equals(users_idList.getSelectedItem()))
                                                                        break;
                                                                }
                                                                if (!rs.isAfterLast())
                                                                {
                                                                        txtlogin_id.setText(rs.getString("login_id"));
                                                                        txtpwd.setText(rs.getString("password"));
                                                                        txtut.setText(rs.getString("user_type"));



                                                                }
                                                        }
                                                        catch (SQLException selectException)
                                                        {
                                                                displaySQLErrors(selectException);
                                                        }
                                                }
                                        });
```

```
updateButton.addActionListener(new ActionListener() {
    @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
try {
    if(logininto.user.equals("admin")) {
    loadLogins();

        String password=JOptionPane.showInputDialog(p,"Enter the new password");

        txtpwd.setText(password);
     String query=" update login set password='"+password+"' where login_id='"+txtlogin_id.getText()+"'";

     int i=statement.executeUpdate(query);

    JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows succesfully");
    loadLogins();

    }

if(logininto.user.equals("user")) {
    users_idList.removeAll();
    users_idList.add(logininto.user1);
    String password2=JOptionPane.showInputDialog(p,"Enter the new password");

    txtpwd.setText(password2);
     String query=" update Login set password='"+password2+"' where login_id='"+txtlogin_id.getText()+"'";
```

```
                        int i=statement.executeUpdate(query);
                        users_idList.removeAll();
                                users_idList.add(logininto.user1);

                        JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows succesfully");
                        //loadLogins();

                }
                }
                catch(SQLException updateException){

                        displaySQLErrors(updateException);
                }


                 }



                        });
        }
        });

    view.addActionListener(new ActionListener(){

        @Override
        public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub

                p.removeAll();
                frame.invalidate();
                frame.validate();
```

```
                                frame.repaint();


                                Label view1=new Label("Login view");
                                //view1.setAlignment(Label.CENTER);
                                Font myFont = new Font("Serif",Font.BOLD,50);
                                view1.setFont((myFont));
                                viewButton=new JButton("View");
                                p1=new JPanel();
                                p2=new JPanel();
                                p1.add(view1);
                                p2.add(viewButton);p1.setBackground(Color.WHITE)
;p2.setBackground(Color.WHITE) ;
                                p.add(p1);p.add(p2);
                                 p.setLayout(new FlowLayout());



                                p.setBounds(500,800,300,300);
                                frame.add(p);
                                frame.setSize(800,800);
                                frame.validate();
                                 viewButton.addActionListener(new ActionListener() {
                                        @Override
                                                public void actionPerformed(ActionEvent e) {
                                                        // TODO Auto-generated method stub
                                                 JFrame f;


                                                   JTable j;


                                                      f = new JFrame();
```

```
f.setTitle("Login details");


DefaultTableModel model = new DefaultTableModel();
j = new JTable(model);
model.addColumn("login_id");
model.addColumn("password");
model.addColumn("user_type");

try {

rs=statement.executeQuery("select * from
login");

while(rs.next()) {

model.addRow(new
Object[]{rs.getString("login_id"),rs.getString("password"),rs.getString("user_type")});

}

}

catch(SQLException viewException) {

displaySQLErrors(viewException);

}

j.setEnabled(false);
j.setBounds(30, 40, 150, 150);


JScrollPane sp = new JScrollPane(j);
f.add(sp);

f.setSize(400, 400);

f.setVisible(true);
```

```
                                                    }

                                                });

                                    }

                            });

                }

}
```

USERS :

```
package RM;

import java.awt.BorderLayout;
import java.awt.Choice;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import java.util.StringTokenizer;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class users {
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	private JButton LoginButton,deleteButton,viewButton;
	private JPanel p1,p2,p3,p;
	private JLabel lbllogin_id,lblusername,lblage,lblgender,lblcnum;
	private JTextField txtlogin_id,txtusername,txtage,txtgender,txtcnum;
	private Choice login_id;
```

```
        private List usersList;
        Connection con;ResultSet rs;
        Statement statement;
        private JFrame frame;
        private JMenuItem insert,delete,update,view;
        public users(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem
view)
        {

                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connectToDB();
                this.frame=frame;
                this.insert=insert;
                this.delete=delete;
                this.update=update;
                this.view=view;
                lbllogin_id=new JLabel("login_ID");
                lblusername=new JLabel("username");
                lblage=new JLabel("Age");
                lblgender=new JLabel("Gender");
                lblcnum=new JLabel("Contact Number");
                login_id=new Choice();
                txtlogin_id=new JTextField(30);
```

```
                txtusername=new JTextField(30);
                txtage=new JTextField(8);
                txtgender=new JTextField(5);
                txtcnum=new JTextField(10);
                this.p=p;



        }

        public void connectToDB()
    {
                try {


                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");


                statement=con.createStatement();
                statement.executeUpdate("commit");



                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
```

```
   }
        private void displaySQLErrors(SQLException e)
        {
                JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:      " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");




        }
        public void loadusers() {
                try {
                        usersList.removeAll();
                        rs=statement.executeQuery("select * from users");
                        while(rs.next()) {
                                usersList.add(rs.getString("login_id")+"->"+rs.getString("username"));
                        }
                        }
                catch(SQLException e) {
                        displaySQLErrors(e);
                }
        }


        public void loadLogins() {
                try {
                        login_id.removeAll();
                        rs=statement.executeQuery("select * from login");
                        while(rs.next()) {
                                login_id.add(rs.getString("login_id"));
                        }
                        }
                catch(SQLException e) {
                        displaySQLErrors(e);
```

```
            }
        }
        public void buildGUI() {
            if(logininto.user.equals("admin")) {
            LoginButton=new JButton("submit");

            insert.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent arg0) {
                    // TODO Auto-generated method stub
                    loadLogins();
                    //txtlogin_id.setText(null);
                    txtusername.setText(null);
                    txtage.setText(null);
                    txtgender.setText(null);
                    txtcnum.setText(null);

                    p.removeAll();
                    frame.invalidate();
                    frame.validate();
                    frame.repaint();


                    p1=new JPanel();

                     p1.setLayout(new GridLayout(5,2));
                     p1.add(lbllogin_id);
                     p1.add(login_id);
                     p1.add(lblusername);
                     p1.add(txtusername);
```

```
p1.add(lblage);
p1.add(txtage);
p1.add(lblgender);
p1.add(txtgender);
p1.add(lblcnum);
p1.add(txtcnum);
p3=new JPanel(new FlowLayout());
p3.add(LoginButton);
//p1.add(txtf1);
p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.WHITE) ;

p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
         usersList=new List(10);
         loadusers();
         p2.add(usersList);p2.setBackground(Color.WHITE) ;
         p2.setBounds(125,320,300,180);



p. add(p1);
p.add(p3);
p. add(p2);



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

```
                frame.add(p);
                frame.setSize(800,800);
                frame.validate();
            LoginButton.addActionListener(new ActionListener() {
                @Override
                    public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
            try {
                String query="INSERT INTO users VALUES('"+login_id.getSelectedItem()+"','"+txtusername.getText()+"','"+txtage.getText()+"','"+txtgender.getText()+"','"+txtcnum.getText()+"')";

                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");
                loadusers();

                System.out.println("done");
            }
            catch(SQLException insertException){
                displaySQLErrors(insertException);
            }

             }

                });
        }
        });

    delete.addActionListener(new ActionListener() {

        @Override
```

```
public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("delete");

        txtlogin_id.setText(null);
        txtusername.setText(null);
        txtage.setText(null);
        txtgender.setText(null);
        txtcnum.setText(null);

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();


        p1=new JPanel();

         p1.setLayout(new GridLayout(3,2));
         p1.setLayout(new GridLayout(5,2));
         p1.add(lbllogin_id);
         p1.add(txtlogin_id);
         p1.add(lblusername);
         p1.add(txtusername);
         p1.add(lblage);
         p1.add(txtage);
         p1.add(lblgender);
         p1.add(txtgender);
         p1.add(lblcnum);
         p1.add(txtcnum);
```

```
// p1.add(deleteButton);
 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
 p1.setBackground(Color.WHITE) ;

 p1.setBounds(115,80,300,200);

 p2 = new JPanel(new FlowLayout());
          // p2.add(txtf1);
           usersList=new List(10);
           loadusers();
           p2.add(usersList);p2.setBackground(Color.WHITE) ;
           p2.setBounds(125,320,300,180);



 p. add(p1);
 p.add(p3);
 p. add(p2);



 p.setLayout(new BorderLayout());

          frame.add(p);
          frame.setSize(800,800);
          frame.validate();
 usersList.addItemListener(new ItemListener()
          {
                    public void itemStateChanged(ItemEvent e)
```

```
                                        {
                                                try
                                                {
                                                        rs=statement.executeQuery("select * from users");
                                                        StringTokenizer                         st=new
StringTokenizer(usersList.getSelectedItem(),"->");
                                                        String p=st.nextToken();
                                                        while (rs.next())
                                                        {
                                                                if (rs.getString("login_id").equals(p))
                                                                break;
                                                        }
                                                        if (!rs.isAfterLast())
                                                        {
                                                                txtlogin_id.setText(rs.getString("login_id"));
        txtusername.setText(rs.getString("username"));
                                                                txtage.setText(rs.getString("age"));
                                                                txtgender.setText(rs.getString("gender"));
        txtcnum.setText(rs.getString("contact_number"));
                                                        }
                                                }
                                                catch (SQLException selectException)
                                                {
                                                        displaySQLErrors(selectException);
                                                }
                                        }
                                });
```

```
deleteButton.addActionListener(new ActionListener() {
    @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
try {

    //String password=JOptionPane.showInputDialog(p,"Enter the password");

    //txtpwd.setText(password);
    int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
    if(a==JOptionPane.YES_OPTION){
    String query="DELETE FROM users WHERE login_id="+txtlogin_id.getText()+" AND username='"+txtusername.getText()+"' AND age='"+txtage.getText()+"' AND gender='"+txtgender.getText()+"' AND contact_number="+txtcnum.getText()+"";

    int i=statement.executeUpdate(query);
    JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");
    loadusers();
    }

}
catch(SQLException deleteException){
    displaySQLErrors(deleteException);
}

 }

    });
}
});
```

```
}
update.addActionListener(new ActionListener() {

	@Override
	public void actionPerformed(ActionEvent arg0) {
		// TODO Auto-generated method stub
		JButton updateButton = new JButton("modify");
		txtlogin_id.setText(null);
		txtusername.setText(null);
		txtage.setText(null);
		txtgender.setText(null);
		txtcnum.setText(null);

		p.removeAll();
		frame.invalidate();
		frame.validate();
		frame.repaint();


		p1=new JPanel();

		 p1.setLayout(new GridLayout(5,2));

		 p1.add(lbllogin_id);
		 p1.add(txtlogin_id);
		 p1.add(lblusername);
		 p1.add(txtusername);
		 p1.add(lblage);
		 p1.add(txtage);
		 p1.add(lblgender);
		 p1.add(txtgender);
```

```
p1.add(lblcnum);
p1.add(txtcnum);
p3=new JPanel(new FlowLayout());
p3.add(updateButton);
//p1.add(txtf1);
p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.WHITE) ;

p1.setBounds(115,80,300,200);
p2 = new JPanel(new FlowLayout());

        usersList=new List(10);
        //loadusers();
        if(logininto.user.equals("user")) {
                usersList.removeAll();
                usersList.add(logininto.user1);
        }
        if(logininto.user.equals("admin")){
                loadusers();
        }
        p2.add(usersList);
        p2.setBackground(Color.WHITE);
        p2.setBounds(125,320,300,180);

p. add(p1);
p.add(p3);
p. add(p2);

usersList.addItemListener(new ItemListener()
        {
```

```
                                        public void itemStateChanged(ItemEvent e)
                                        {
                                                try
                                                {
                                                        rs=statement.executeQuery("select * from users");
                                                        StringTokenizer                                 st=new
StringTokenizer(usersList.getSelectedItem(),"->");
                                                        String p=st.nextToken();
                                                        //String q=st.nextToken();

                                                        while (rs.next())
                                                        {
                                                                if (rs.getString("login_id").equals(p))
                                                                break;
                                                        }
                                                        if (!rs.isAfterLast())
                                                        {

                                                                txtlogin_id.setText(rs.getString("login_id"));

        txtusername.setText(rs.getString("username"));
                                                                txtage.setText(rs.getString("age"));
                                                                txtgender.setText(rs.getString("gender"));

        txtcnum.setText(rs.getString("contact_number"));
                                                        }
                                                }
                                                catch (SQLException selectException)
                                                {
                                                        displaySQLErrors(selectException);
                                                }
                                        }
```

```
                });
            p.setLayout(new BorderLayout());


                frame.add(p);
                frame.setSize(800,800);
                frame.validate();
            updateButton.addActionListener(new ActionListener() {
                 @Override
                        public void actionPerformed(ActionEvent e) {
                                // TODO Auto-generated method stub
                         try {
                                 if(logininto.user.equals("admin")) {
                                        loadusers();


                                        String query="update users set username='"+txtusername.getText()+"',age='"+txtage.getText()+"',gender='"+txtgender.getText()+"',contact_number="+txtcnum.getText()+" WHERE login_id="+txtlogin_id.getText()+"";


                                        int i=statement.executeUpdate(query);
                                        JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadusers() ;
                                        }
                                if(logininto.user.equals("user")) {
                                                usersList.removeAll();
                                                usersList.add(logininto.user1);
                                                //String contact_number=JOptionPane.showInputDialog(p,"Enter the new contact number");



                                                //txtcnum.setText(contact_number);


                                                String query="update users set username='"+txtusername.getText()+"',age='"+txtage.getText()+"',gender='"+txtgender.getText()+"',contact_number="+txtcnum.getText()+" WHERE login_id="+txtlogin_id.getText()+"";
```

```
                                                                    //String query=" update users set contact_number='"+contact_number+"' where login_id='"+txtlogin_id.getText()+"'";
                                                                    int i=statement.executeUpdate(query);



        JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");
                                                                    }


                                                        }
                                                        catch(SQLException deleteException){
                                                                displaySQLErrors(deleteException);
                                                        }

                                }



                                        });
                        }
                        });

                view.addActionListener(new ActionListener(){

                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub


                                p.removeAll();
                                frame.invalidate();
                                frame.validate();
                                frame.repaint();
```

```
                                Label view1=new Label("User view");
                                //view1.setAlignment(Label.CENTER);
                                Font myFont = new Font("Serif",Font.BOLD,50);
                                view1.setFont((myFont));
                                viewButton=new JButton("View");
                                p1=new JPanel();
                                p2=new JPanel();
                                p1.add(view1);
                                p2.add(viewButton);p1.setBackground(Color.WHITE)
;p2.setBackground(Color.WHITE) ;
                                p.add(p1);p.add(p2);
                                 p.setLayout(new FlowLayout());



                                p.setBounds(500,800,300,300);
                                frame.add(p);
                                frame.setSize(800,800);
                                frame.validate();
                                 viewButton.addActionListener(new ActionListener() {
                                        @Override
                                                public void actionPerformed(ActionEvent e) {
                                                        // TODO Auto-generated method stub
                                                 JFrame f;


                                                   JTable j;


                                                      f = new JFrame();



                                                      f.setTitle("User details");
```

```
                                                DefaultTableModel model = new DefaultTableModel();
                                                j = new JTable(model);
                                                model.addColumn("login_id");
                                                model.addColumn("username");
                                                model.addColumn("age");
                                                model.addColumn("gender");
                                                model.addColumn("contact_number");
                                                try {
                                                        rs=statement.executeQuery("select * from users");
                                                        while(rs.next()) {
                                                                model.addRow(new
Object[]{rs.getString("login_id"),rs.getString("username"),rs.getString("age"),rs.getString("gender"),rs.getString("con
tact_number")});
                                                        }
                                                        }
                                                        catch(SQLException viewException) {
                                                                displaySQLErrors(viewException);
                                                        }
                                                        j.setEnabled(false);
                                                   j.setBounds(30, 40, 150, 150);
                                                JScrollPane sp = new JScrollPane(j);
                                                   f.add(sp);
                                                   f.setSize(400, 400);
                                                     f.setVisible(true);
                                                }
                                        });
                                }
                        });
        }
}
```

DISEASES:

```
package RM;


import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import java.util.StringTokenizer;


import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;


public class diseases{
	/**
	 *
```

```
 */
private static final long serialVersionUID = 1L;
private JButton insertButton,deleteButton,updateButton,viewButton;
private JPanel p1,p2,p3,p;
private JLabel lbldname,lbltod;
private JTextField txtdname,txttod;

private List DISList;
Connection con;ResultSet rs;
Statement statement;
private JFrame frame;
private JMenuItem insert,delete,update,view;
public diseases(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem
view)
{

	try
	{
		Class.forName("oracle.jdbc.driver.OracleDriver");
	}
	catch (Exception e)
	{
		System.err.println("Unable to find and load driver");
		System.exit(1);
	}
	connectToDB();

	this.frame=frame;
	this.insert=insert;
	this.delete=delete;
	this.update=update;
```

```
                this.view=view;

                lbldname=new JLabel("Disease  Name");
                lbltod=new JLabel("Type of disease");




                txtdname=new JTextField(15);
                txttod=new JTextField(15);

                this.p=p;



        }

        public void connectToDB()
    {
                try {


                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");


                statement=con.createStatement();
                statement.executeUpdate("commit");


                }
                catch (SQLException connectException)
```

```
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
   }
        private void displaySQLErrors(SQLException e)
        {
                JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:      " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");



        }
        public void loaddis() {
                try {
                        DISList.removeAll();
                        rs=statement.executeQuery("select * from diseases");
                        while(rs.next()) {
                        DISList.add(rs.getString("disease_name"));
                        }
                        }
                catch(SQLException e) {
                        displaySQLErrors(e);
                }
        }

        public void buildGUI() {

                if(logininto.user.equals("admin")) {
```

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

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        insertButton=new JButton("submit");
        txtdname.setText(null);
        txttod.setText(null);



        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();


        p1=new JPanel();

         p1.setLayout(new GridLayout(2,2));
         p1.add(lbldname);
         p1.add(txtdname);
         p1.add(lbltod);
         p1.add(txttod);

         p3=new JPanel(new FlowLayout());
         p3.add(insertButton);
         //p1.add(txtf1);
         p3.setBackground(Color.BLACK);
         p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
         p1.setBackground(Color.WHITE) ;
```

```
p2 = new JPanel(new FlowLayout());

        DISList=new List(10);
        loaddis();
        p2.add(DISList);p2.setBackground(Color.WHITE) ;

        p2.setBounds(450,150,350,180);


p. add(p1);
p.add(p3);
p. add(p2);


p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();

insertButton.addActionListener(new ActionListener() {
        @Override
                public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
try {
        String          query="INSERT          INTO          diseases
```

```
VALUES('"+txtdname.getText()+"','"+txttod.getText()+"')";

                                        int i=statement.executeUpdate(query);
                                        JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows
succesfully");loaddis();



                                }
                                catch(SQLException insertException){
                                        displaySQLErrors(insertException);
                                }


                                 }



                                        });
                        }
                        });

                delete.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub
                                deleteButton=new JButton("delete");

                                txtdname.setText(null);
                                txttod.setText(null);
```

```
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();


p1=new JPanel();

 p1.setLayout(new GridLayout(2,2));
 p1.add(lbldname);
 p1.add(txtdname);
 p1.add(lbltod);
 p1.add(txttod);

 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
 p3.setBounds(200,350,75,35);
 p1.setBackground(Color.WHITE) ;
 p1.setBounds(115,80,300,250);



// p1.setBounds(100,100,500,300);

 p2 = new JPanel(new FlowLayout());

        DISList=new List(10);
        loaddis();
```

```
                    p2.add(DISList);p2.setBackground(Color.WHITE) ;

                    p2.setBounds(450,150,350,180);


                p. add(p1);
                p.add(p3);
                p. add(p2);

                p.setLayout(new BorderLayout());
                frame.add(p);
                frame.setSize(800,800);
                frame.validate();
                 DISList.addItemListener(new ItemListener()
                        {
                            public void itemStateChanged(ItemEvent e)
                            {
                                try
                                {
                                    rs=statement.executeQuery("select * from
diseases");
                                    while (rs.next())
                                    {
                                        if
(rs.getString("disease_name").equals(DISList.getSelectedItem()))
                                        break;
                                    }
                                    if (!rs.isAfterLast())
                                    {

        txtdname.setText(rs.getString("disease_name"));

        txttod.setText(rs.getString("type_of_disease"));
```

```
                                        }
                                    }
                                    catch (SQLException selectException)
                                    {
                                        displaySQLErrors(selectException);
                                    }
                                }
                            });



                        deleteButton.addActionListener(new ActionListener() {
                            @Override
                                public void actionPerformed(ActionEvent e) {
                                    // TODO Auto-generated method stub
                        try {

                                int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                                if(a==JOptionPane.YES_OPTION){
                                String query="DELETE FROM diseases WHERE disease_name='"+txtdname.getText()+"'";

                                int i=statement.executeUpdate(query);
                                JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loaddis();
                                }


                        }
                        catch(SQLException deleteException){
                                displaySQLErrors(deleteException);
```

```
                    }

                     }


                            });
            }
            });

        update.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                JButton updateButton = new JButton("modify");
                txtdname.setText(null);
                txttod.setText(null);



                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();


                p1=new JPanel();

                 p1.setLayout(new GridLayout(2,2));
                 p1.add(lbldname);
                 p1.add(txtdname);
```

```
p1.add(lbltod);
p1.add(txttod);

p3=new JPanel(new FlowLayout());
p3.add(updateButton);
//p1.add(txtf1);
p3.setBackground(Color.BLACK);
p3.setBounds(200,350,75,35);
p1.setBackground(Color.WHITE) ;
p1.setBounds(115,80,300,250);

p2 = new JPanel(new FlowLayout());

        DISList=new List(10);
        loaddis();
        p2.add(DISList);p2.setBackground(Color.WHITE) ;

        p2.setBounds(450,150,350,180);


p. add(p1);
p.add(p3);
p. add(p2);

p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
 DISList.addItemListener(new ItemListener()
        {
```

```
                                        public void itemStateChanged(ItemEvent e)
                                        {
                                                try
                                                {
                                                        rs=statement.executeQuery("select * from
diseases");
                                                        while (rs.next())
                                                        {
                                                                if
(rs.getString("disease_name").equals(DISList.getSelectedItem()))
                                                                break;
                                                        }
                                                        if (!rs.isAfterLast())
                                                        {

        txtdname.setText(rs.getString("disease_name"));

        txttod.setText(rs.getString("type_of_disease"));


                                                        }
                                                }
                                                catch (SQLException selectException)
                                                {
                                                        displaySQLErrors(selectException);
                                                }
                                        }
                                });




                        updateButton.addActionListener(new ActionListener() {
                                @Override
```

```
                                        public void actionPerformed(ActionEvent e) {
                                            // TODO Auto-generated method stub
                                        try {
                                                loaddis();
                                                String
disease_name=JOptionPane.showInputDialog(p,"Enter the new disease");
                                                txtdname.setText(disease_name);


                                                //int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");
                                                //if(a==JOptionPane.YES_OPTION){
                                                String query="update diseases set
disease_name='"+disease_name+"' where type_of_disease='"+txttod.getText()+"' ";


                                                int i=statement.executeUpdate(query);
                                                JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loaddis();
                                                }




                                            catch(SQLException deleteException){
                                                displaySQLErrors(deleteException);
                                            }


                                    }


                                });
                            }
                            });
                    }
```

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

@Override
public void actionPerformed(ActionEvent arg0) {
// TODO Auto-generated method stub

p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();

Label view1=new Label("Diseases view");
//view1.setAlignment(Label.CENTER);
Font myFont = new Font("Serif",Font.BOLD,50);
view1.setFont((myFont));
viewButton=new JButton("View");
p1=new JPanel();
p2=new JPanel();
p1.add(view1);
p2.add(viewButton);p1.setBackground(Color.WHITE)
;p2.setBackground(Color.WHITE) ;
p.add(p1);p.add(p2); p.setLayout(new FlowLayout());
frame.add(p);
frame.setSize(800,800);
frame.validate();
 viewButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
 JFrame f;
```

```
                                                JTable j;
                                                    f = new JFrame();
                                                    f.setTitle("Diseases details");
                                                    DefaultTableModel model = new DefaultTableModel();
                                                    j = new JTable(model);
                                                   model.addColumn("Disease Name");
                                                   model.addColumn("Type Of Disease");
                                                   try {
                        rs=statement.executeQuery("select * from diseases");
                                                                        while(rs.next()) {
                                                                                model.addRow(new
Object[]{rs.getString("disease_name"), rs.getString("type_of_disease")});
                                                                        }
                                                                        }
                                                                catch(SQLException viewException) {
                                                                        displaySQLErrors(viewException);
                                                                }
                                                                j.setEnabled(false);
                                                   j.setBounds(30, 40, 300, 300);



                                                    JScrollPane sp = new JScrollPane(j);
                                                    f.add(sp);
                                                    f.setSize(800, 400);
                                                    f.setVisible(true);
                                                }
                                        });
                                }
                        });
        }
```

}

MEDICINES:

```
package RM;


import java.awt.BorderLayout;
import java.awt.Choice;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import java.util.StringTokenizer;


import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;


public class Medicines{
```

```
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	private JButton insertButton,deleteButton,updateButton,viewButton;
	private JPanel p1,p2,p3,p;
	private JLabel lblmid,lblmname,lblcost,lbldos,lbldname;
	private JTextField txtmid,txtmname,txtcost,txtdos,txtdname;
	private Choice dname;

	private List MEDList;
	Connection con;ResultSet rs;
	Statement statement;
	private JFrame frame;
	private JMenuItem insert,delete,update,view;
	public Medicines(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem update,JMenuItem
view)
	{

		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();

		this.frame=frame;
```

```
                this.insert=insert;
                this.delete=delete;
                this.update=update;
                this.view=view;

                lblmid=new JLabel("Medicine ID");
                lblmname=new JLabel("Medicine Name");
                lblcost=new JLabel("Price");
                lbldos=new JLabel("Dosage");
                lbldname=new JLabel("Disease Name");


                txtmid=new JTextField(8);
                txtmname=new JTextField(30);
                txtcost=new JTextField(8);
                txtdos=new JTextField(8);
                dname=new Choice();
                txtdname=new JTextField(25);


                this.p=p;



        }

        public void connectToDB()
 {
                try {
```

```
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");



                statement=con.createStatement();
                statement.executeUpdate("commit");



                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
   }
        private void displaySQLErrors(SQLException e)
        {
                JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:      " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");



        }
        public void loadMedIDs() {
                try {
                        MEDList.removeAll();
                        rs=statement.executeQuery("select * from medicines");
                        while(rs.next()) {
                                MEDList.add(rs.getString("medicine_id")+"->"+rs.getString("medicine_name"));
                        }
```

```
                }
        catch(SQLException e) {
                displaySQLErrors(e);
        }
}

public void loaddis() {
try {
        dname.removeAll();
        rs=statement.executeQuery("select * from diseases");
        while(rs.next()) {
                dname.add(rs.getString("disease_name"));
        }
        }
catch(SQLException e) {
        displaySQLErrors(e);
}
}



public void buildGUI() {



        if(logininto.user.equals("admin")) {
        insert.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub
                        insertButton=new JButton("submit");
```

```
loaddis();

txtmid.setText(null);
txtmname.setText(null);
txtcost.setText(null);
txtdos.setText(null);

p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();



p1=new JPanel();

 p1.setLayout(new GridLayout(5,2));

 p1.add(lblmid);
 p1.add(txtmid);
 p1.add(lblmname);
 p1.add(txtmname);
 p1.add(lblcost);
 p1.add(txtcost);
 p1.add(lbldos);
 p1.add(txtdos);
 p1.add(lbldname);
 p1.add(dname);
/* p1.add(insertButton);

        p1.setBackground(Color.orange) ;
 p1.setBounds(100,100,300,400);
```

```
*/
 p3=new JPanel(new FlowLayout());
 p3.add(insertButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
 p1.setBackground(Color.WHITE) ;

 p1.setBounds(115,80,300,200);
 p2 = new JPanel(new FlowLayout());

            MEDList=new List(10);
            loadMedIDs();
            p2.add(MEDList);
            p2.setBackground(Color.WHITE);
            p2.setBounds(125,320,300,180);

 p. add(p1);
 p.add(p3);
 p. add(p2);


 p.setLayout(new BorderLayout());

           frame.add(p);
           frame.setSize(800,800);
           frame.validate();

 insertButton.addActionListener(new ActionListener() {
            @Override
                    public void actionPerformed(ActionEvent e) {
```

```
                                        // TODO Auto-generated method stub
                    try {
                        String query="INSERT INTO medicines VALUES("+txtmid.getText()+",'"+txtmname.getText()+"','"+txtcost.getText()+"','"+txtdos.getText()+"','"+dname.getSelectedItem()+"')";

                        int i=statement.executeUpdate(query);
                        JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

                        loadMedIDs() ;

                    }
                    catch(SQLException insertException){
                        displaySQLErrors(insertException);
                    }

                     }

                        });
                }
                });


        delete.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                deleteButton=new JButton("delete");
                //deleteButton.setSize(1,1);
```

```
txtmid.setText(null);
txtmname.setText(null);
txtcost.setText(null);
txtdos.setText(null);
txtdname.setText(null);

p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();



p1=new JPanel();

 p1.setLayout(new GridLayout(5,2));

 p1.add(lblmid);
 p1.add(txtmid);
 p1.add(lblmname);
 p1.add(txtmname);
 p1.add(lblcost);
 p1.add(txtcost);
 p1.add(lbldos);
 p1.add(txtdos);
 p1.add(lbldname);
 p1.add(txtdname);

 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
```

```
p3.setBounds(200,280,75,35);
p1.setBackground(Color.WHITE) ;

p1.setBounds(115,80,300,200);
p2 = new JPanel(new FlowLayout());

    MEDList=new List(10);
    loadMedIDs();
    p2.add(MEDList);
    p2.setBackground(Color.WHITE);
    p2.setBounds(125,320,300,180);

p. add(p1);
p.add(p3);
p. add(p2);

MEDList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from medicines");
                StringTokenizer st=new StringTokenizer(MEDList.getSelectedItem(),"->");
                String p=st.nextToken();
                //String q=st.nextToken();

                while (rs.next())
                {
                    if (rs.getString("medicine_id").equals(p))
```

```
                                                    break;
                                                }
                                                if (!rs.isAfterLast())
                                                {
                                                    txtmid.setText(rs.getString("medicine_id"));
    txtmname.setText(rs.getString("medicine_name"));
                                                    txtcost.setText(rs.getString("price"));
                                                    txtdos.setText(rs.getString("dosage"));
    txtdname.setText(rs.getString("disease_name"));
                                                }
                                            }
                                            catch (SQLException selectException)
                                            {
                                                displaySQLErrors(selectException);
                                            }
                                        }
                                    });
                                p.setLayout(new BorderLayout());

                                    frame.add(p);
                                    frame.setSize(800,800);
                                    frame.validate();
                                deleteButton.addActionListener(new ActionListener() {
                                     @Override
                                         public void actionPerformed(ActionEvent e) {
                                             // TODO Auto-generated method stub
                                try {

                                    int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                                    if(a==JOptionPane.YES_OPTION){
```

```
                                        StringTokenizer st=new StringTokenizer(MEDList.getSelectedItem(),"->");

                                        String query="DELETE FROM medicines WHERE medicine_ID="+st.nextToken();

                                        int i=statement.executeUpdate(query);
                                        JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows succesfully");loadMedIDs();
                                        }


                                }
                                catch(SQLException deleteException){
                                        displaySQLErrors(deleteException);
                                }


                                 }



                                        });
                        }
                        });

                update.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub
                                JButton updateButton = new JButton("modify");
                                txtmid.setText(null);
                                txtmname.setText(null);
```

```
txtcost.setText(null);
txtdos.setText(null);
txtdname.setText(null);

p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();



p1=new JPanel();

 p1.setLayout(new GridLayout(5,2));
 p1.add(lblmid);
 p1.add(txtmid);
 p1.add(lblmname);
 p1.add(txtmname);
 p1.add(lblcost);
 p1.add(txtcost);
 p1.add(lbldos);
 p1.add(txtdos);
 p1.add(lbldname);
 p1.add(txtdname);


 p3=new JPanel(new FlowLayout());
 p3.add(updateButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
 p1.setBackground(Color.WHITE) ;
```

```
p1.setBounds(115,80,300,200);
p2 = new JPanel(new FlowLayout());

    MEDList=new List(10);
    loadMedIDs();
    p2.add(MEDList);
    p2.setBackground(Color.WHITE);
    p2.setBounds(125,320,300,180);

p. add(p1);
p.add(p3);
p. add(p2);

MEDList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from
medicines");
                StringTokenizer st=new
StringTokenizer(MEDList.getSelectedItem(),"->");
                String p=st.nextToken();
                //String q=st.nextToken();

                while (rs.next())
                {
                    if (rs.getString("medicine_id").equals(p))
                    break;
                }
```

```
                                                    if (!rs.isAfterLast())
                                                    {
                                                            txtmid.setText(rs.getString("medicine_id"));
        txtmname.setText(rs.getString("medicine_name"));
                                                            txtcost.setText(rs.getString("price"));
                                                            txtdos.setText(rs.getString("dosage"));
        txtdname.setText(rs.getString("disease_name"));
                                                    }
                                                }
                                                catch (SQLException selectException)
                                                {
                                                        displaySQLErrors(selectException);
                                                }
                                        }
                                });
                        p.setLayout(new BorderLayout());

                                frame.add(p);
                                frame.setSize(800,800);
                                frame.validate();
                        updateButton.addActionListener(new ActionListener() {
                                @Override
                                        public void actionPerformed(ActionEvent e) {
                                                // TODO Auto-generated method stub
                                         try {

                                                    int  a=JOptionPane.showConfirmDialog(p,"Are  you
sure want to update:");
                                                    if(a==JOptionPane.YES_OPTION){
                                                            StringTokenizer                 st=new
StringTokenizer(MEDList.getSelectedItem(),"->");
```

```
                                                    String query="update medicines set medicine_id="+txtmid.getText()+",medicine_name="'+txtmname.getText()+"',price="'+txtcost.getText()+"',dosage="+txtdos.getText()+",disease_name="'+txtdname.getText()+"' WHERE medicine_id="+st.nextToken();


                                                    int i=statement.executeUpdate(query);
                                                    JOptionPane.showMessageDialog(p,"\nupdated "+i+" rows succesfully");loadMedIDs() ;
                                                    }


                                                }
                                                catch(SQLException deleteException){
                                                    displaySQLErrors(deleteException);
                                                }

                                    }


                                        });
                                }
                                });
                        }

                        view.addActionListener(new ActionListener(){

                                @Override
                                public void actionPerformed(ActionEvent arg0) {
                                        // TODO Auto-generated method stub


                                        p.removeAll();
                                        frame.invalidate();
                                        frame.validate();
```

```
frame.repaint();


Label view1=new Label("Medicines view");
//view1.setAlignment(Label.CENTER);
Font myFont = new Font("Serif",Font.BOLD,50);
view1.setFont((myFont));
viewButton=new JButton("View");
p1=new JPanel();
p2=new JPanel();
p1.add(view1);
p2.add(viewButton);p1.setBackground(Color.WHITE)
;p2.setBackground(Color.WHITE) ;
p.add(p1);p.add(p2); p.setLayout(new FlowLayout());
frame.add(p);
frame.setSize(800,800);
frame.validate();
 viewButton.addActionListener(new ActionListener() {
        @Override
                public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
                 JFrame f;

                   JTable j;

                      f = new JFrame();



                      f.setTitle("Medicine details");



                      DefaultTableModel model = new DefaultTableModel();
```

```
                                                    j = new JTable(model);
                                                    model.addColumn("medicine_id");
                                                    model.addColumn("medicine_name");
                                                    model.addColumn("price");
                                                    model.addColumn("dosage");
                                                    model.addColumn("disease_name");

                                                    try {

                                                            rs=statement.executeQuery("select  *  from
medicines");
                                                            while(rs.next()) {
                                                                    model.addRow(new
Object[]{rs.getString("medicine_id"),
rs.getString("medicine_name"),rs.getString("price"),rs.getString("dosage"),rs.getString("disease_name")});
                                                            }
                                                            }
                                                    catch(SQLException viewException) {
                                                            displaySQLErrors(viewException);
                                                    }
                                                    j.setEnabled(false);
                                                    j.setBounds(30, 40, 150, 150);
                                                    JScrollPane sp = new JScrollPane(j);
                                                    f.add(sp);
                                                    f.setSize(400, 400);
                                                        f.setVisible(true);
                                            }
                                                    });
                                    }
                            });
                    }
            }
```

MEDICINE_MANUFACTURER:

```
package RM;

import java.awt.BorderLayout;
import java.awt.Choice;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import java.util.StringTokenizer;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
```

```
public class medicine_manufacturer{
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	private JButton insertButton,deleteButton,updateButton,viewButton;
	private JPanel p1,p2,p3,p;
	private JLabel lblcid,lblcname,lbladd,lblmid;
	private JTextField txtcid,txtcname,txtadd,txtmid;
	private Choice mid;

	private List MED_MList;
	Connection con;ResultSet rs;
	Statement statement;
	private JFrame frame;
	private JMenuItem insert,delete,update,view;
	public medicine_manufacturer(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
	{

		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
```

```
                this.frame=frame;
                this.insert=insert;
                this.delete=delete;
                this.update=update;
                this.view=view;

                lblcid=new JLabel("Company ID");
                lblcname=new JLabel("Company Name");
                lbladd=new JLabel("Address");
                lblmid=new JLabel("Medicine ID");



                txtcid=new JTextField(15);
                txtcname=new JTextField(15);
                txtadd=new JTextField(8);
                mid=new Choice();
                txtmid=new JTextField(25);


                this.p=p;



        }

        public void connectToDB()
 {
                try {
```

```
                Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737100","vasavi");



                statement=con.createStatement();
                statement.executeUpdate("commit");



                }
                catch (SQLException connectException)
                {
                  System.out.println(connectException.getMessage());
                  System.out.println(connectException.getSQLState());
                  System.out.println(connectException.getErrorCode());
                  System.exit(1);
                }
   }
        private void displaySQLErrors(SQLException e)
        {
                JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() + "\n"+"SQLState:      " +
e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");



        }
        public void loadMedmIDs() {
                try {
                        MED_MList.removeAll();
                        rs=statement.executeQuery("select * from medicine_manufacturer");
                        while(rs.next()) {
                                MED_MList.add(rs.getString("company_id")+"->"+rs.getString("company_name"));
                        }
```

```
                }
        catch(SQLException e) {
                displaySQLErrors(e);
        }
}

public void loadMedIDs() {
try {
        mid.removeAll();
        rs=statement.executeQuery("select * from medicines");
        while(rs.next()) {
                mid.add(rs.getString("medicine_id"));
        }
        }
catch(SQLException e) {
        displaySQLErrors(e);
}
}



public void buildGUI() {

        if(logininto.user.equals("admin")) {

        insert.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub
                        insertButton=new JButton("submit");
```

```
loadMedIDs();

txtcid.setText(null);
txtcname.setText(null);
txtadd.setText(null);

p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();



p1=new JPanel();

 p1.setLayout(new GridLayout(5,2));

 p1.add(lblcid);
 p1.add(txtcid);
 p1.add(lblcname);
 p1.add(txtcname);
 p1.add(lbladd);
 p1.add(txtadd);
 p1.add(lblmid);
 p1.add(mid);
/* p1.add(insertButton);

          p1.setBackground(Color.orange) ;
 p1.setBounds(100,100,300,400);
*/
 p3=new JPanel(new FlowLayout());
 p3.add(insertButton);
```

```
//p1.add(txtf1);
p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.WHITE) ;

p1.setBounds(115,80,300,200);
p2 = new JPanel(new FlowLayout());

        MED_MList=new List(10);
        loadMedmIDs();
        p2.add(MED_MList);
        p2.setBackground(Color.WHITE);
        p2.setBounds(125,320,300,180);

p. add(p1);
p.add(p3);
p. add(p2);

p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();

insertButton.addActionListener(new ActionListener() {
        @Override
                public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
try {
        String query="INSERT INTO medicine_manufacturer
```

```
VALUES("+txtcid.getText()+","'+txtcname.getText()+"','"+txtadd.getText()+"','"+mid.getSelectedItem()+"')";

                                        int i=statement.executeUpdate(query);
                                        JOptionPane.showMessageDialog(p,"\ninserted "+i+" rows succesfully");

                                        loadMedmIDs() ;

                                }
                                catch(SQLException insertException){
                                        displaySQLErrors(insertException);
                                }

                                 }



                                        });
                        }
                        });




                delete.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub
                                deleteButton=new JButton("delete");
                                //deleteButton.setSize(1,1);
                                txtcid.setText(null);
```

```
txtcname.setText(null);
txtadd.setText(null);
txtmid.setText(null);

p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();



p1=new JPanel();

 p1.setLayout(new GridLayout(5,2));

 p1.add(lblcid);
 p1.add(txtcid);
 p1.add(lblcname);
 p1.add(txtcname);
 p1.add(lbladd);
 p1.add(txtadd);
 p1.add(lblmid);
 p1.add(txtmid);

 p3=new JPanel(new FlowLayout());
 p3.add(deleteButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
 p1.setBackground(Color.WHITE) ;

 p1.setBounds(115,80,300,200);
```

```
                    p2 = new JPanel(new FlowLayout());

                        MED_MList=new List(10);
                        loadMedmIDs();
                        p2.add(MED_MList);
                        p2.setBackground(Color.WHITE);
                        p2.setBounds(125,320,300,180);

                    p. add(p1);
                    p.add(p3);
                    p. add(p2);

                    MED_MList.addItemListener(new ItemListener()
                        {
                            public void itemStateChanged(ItemEvent e)
                            {
                                try
                                {
                                    rs=statement.executeQuery("select * from
medicine_manufacturer");
                                    StringTokenizer st=new
StringTokenizer(MED_MList.getSelectedItem(),"->");
                                    String p=st.nextToken();
                                    //String q=st.nextToken();

                                    while (rs.next())
                                    {
                                        if (rs.getString("company_id").equals(p))
                                        break;
                                    }
                                    if (!rs.isAfterLast())
                                    {
```

```
                                                    txtcid.setText(rs.getString("company_id"));

        txtcname.setText(rs.getString("company_name"));
                                                    txtadd.setText(rs.getString("address"));
                                                    txtmid.setText(rs.getString("medicine_id"));
                                                        }
                                                    }
                                                    catch (SQLException selectException)
                                                    {
                                                        displaySQLErrors(selectException);
                                                    }
                                                }
                                            });
                                    p.setLayout(new BorderLayout());

                                            frame.add(p);
                                            frame.setSize(800,800);
                                            frame.validate();
                                    deleteButton.addActionListener(new ActionListener() {
                                            @Override
                                                public void actionPerformed(ActionEvent e) {
                                                    // TODO Auto-generated method stub
                                    try {

                                            int a=JOptionPane.showConfirmDialog(p,"Are you sure want to delete:");
                                            if(a==JOptionPane.YES_OPTION){
                                                StringTokenizer st=new
StringTokenizer(MED_MList.getSelectedItem(),"->");

                                            String query="DELETE FROM medicine_manufacturer WHERE
company_ID="+st.nextToken();
```

```
                                        int i=statement.executeUpdate(query);
                                        JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");loadMedmIDs();
                                        }



                                }
                                catch(SQLException deleteException){
                                        displaySQLErrors(deleteException);
                                }


                                 }



                                        });
                        }
                        });

                update.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub
                                JButton updateButton = new JButton("modify");
                                txtcid.setText(null);
                                txtcname.setText(null);
                                txtadd.setText(null);
                                txtmid.setText(null);

                                p.removeAll();
                                frame.invalidate();
```

```
frame.validate();
frame.repaint();


p1=new JPanel();

 p1.setLayout(new GridLayout(5,2));

 p1.add(lblcid);
 p1.add(txtcid);
 p1.add(lblcname);
 p1.add(txtcname);
 p1.add(lbladd);
 p1.add(txtadd);
 p1.add(lblmid);
 p1.add(txtmid);


 p3=new JPanel(new FlowLayout());
 p3.add(updateButton);
 //p1.add(txtf1);
 p3.setBackground(Color.BLACK);
p3.setBounds(200,280,75,35);
 p1.setBackground(Color.WHITE) ;

 p1.setBounds(115,80,300,200);
 p2 = new JPanel(new FlowLayout());

        MED_MList=new List(10);
        loadMedmIDs();
        p2.add(MED_MList);
```

```
                    p2.setBackground(Color.WHITE);
                    p2.setBounds(125,320,300,180);

                p. add(p1);
                p.add(p3);
                p. add(p2);

                MED_MList.addItemListener(new ItemListener()
                    {
                        public void itemStateChanged(ItemEvent e)
                        {
                            try
                            {
                                rs=statement.executeQuery("select * from
medicine_manufacturer");
                                StringTokenizer st=new
StringTokenizer(MED_MList.getSelectedItem(),"->");
                                String p=st.nextToken();
                                //String q=st.nextToken();

                                while (rs.next())
                                {
                                    if (rs.getString("company_id").equals(p))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {
                                    txtcid.setText(rs.getString("company_id"));
        txtcname.setText(rs.getString("company_name"));
                                    txtadd.setText(rs.getString("address"));
                                    txtmid.setText(rs.getString("medicine_id"));
                                }
```

```
                                        }
                                        catch (SQLException selectException)
                                        {
                                                displaySQLErrors(selectException);
                                        }
                                }
                        });
                p.setLayout(new BorderLayout());

                        frame.add(p);
                        frame.setSize(800,800);
                        frame.validate();
                updateButton.addActionListener(new ActionListener() {
                         @Override
                                public void actionPerformed(ActionEvent e) {
                                        // TODO Auto-generated method stub
                                 try {

                                                int a=JOptionPane.showConfirmDialog(p,"Are you
sure want to update:");
                                                if(a==JOptionPane.YES_OPTION){
                                                        StringTokenizer st=new
StringTokenizer(MED_MList.getSelectedItem(),"->");
                                                String query="update medicine_manufacturer set
company_id="'+txtcid.getText()+",company_name="'+txtcname.getText()+"',address="'+txtadd.getText()+"',medicine_
id="'+txtmid.getText()+" WHERE company_id="'+st.nextToken();

                                                int i=statement.executeUpdate(query);
                                                JOptionPane.showMessageDialog(p,"\nupdated
"+i+" rows succesfully");loadMedmIDs() ;
                                                }
```

```
                                        }
                                        catch(SQLException deleteException){
                                                displaySQLErrors(deleteException);
                                        }

                        }



                                });
                }
                });
        }

        view.addActionListener(new ActionListener(){

                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub


                        p.removeAll();
                        frame.invalidate();
                        frame.validate();
                        frame.repaint();


                        Label view1=new Label("Medicines_Manufacturer view");
                        //view1.setAlignment(Label.CENTER);
                        Font myFont = new Font("Serif",Font.BOLD,50);
                        view1.setFont((myFont));
                        viewButton=new JButton("View");
                        p1=new JPanel();
                        p2=new JPanel();
```

```
                                p1.add(view1);
                                p2.add(viewButton);p1.setBackground(Color.WHITE)
;p2.setBackground(Color.WHITE) ;
                                p.add(p1);p.add(p2); p.setLayout(new FlowLayout());
                                frame.add(p);
                                frame.setSize(800,800);
                                frame.validate();
                                 viewButton.addActionListener(new ActionListener() {
                                        @Override
                                                public void actionPerformed(ActionEvent e) {
                                                        // TODO Auto-generated method stub
                                                 JFrame f;
                                                   JTable j;
                                                     f = new JFrame();
                                                     f.setTitle("Medicine Manufacturer details");
                                                     DefaultTableModel model = new DefaultTableModel();
                                                     j = new JTable(model);
                                                         model.addColumn("Company ID");
                                                               model.addColumn("Company Name");
                                                               model.addColumn("Address");
                                                               model.addColumn("Medicine ID");
                                                          try {
                                                rs=statement.executeQuery("select * from medicine_manufacturer");
                                                                        while(rs.next()) {
                                                                                model.addRow(new
Object[]{rs.getString("company_id"),
rs.getString("company_name"),rs.getString("address"),rs.getString("medicine_id")});
                                                                        }
                                                                        }
                                                                catch(SQLException viewException) {
                                                                        displaySQLErrors(viewException);
                                                                }
```

```
                                                    j.setEnabled(false);
                                                j.setBounds(30, 40, 150, 150);
                                                JScrollPane sp = new JScrollPane(j);
                                                f.add(sp);
                                                f.setSize(400, 400);
                                                f.setVisible(true);
                                            }
                                        });
                                }
                        });
        }
}
```

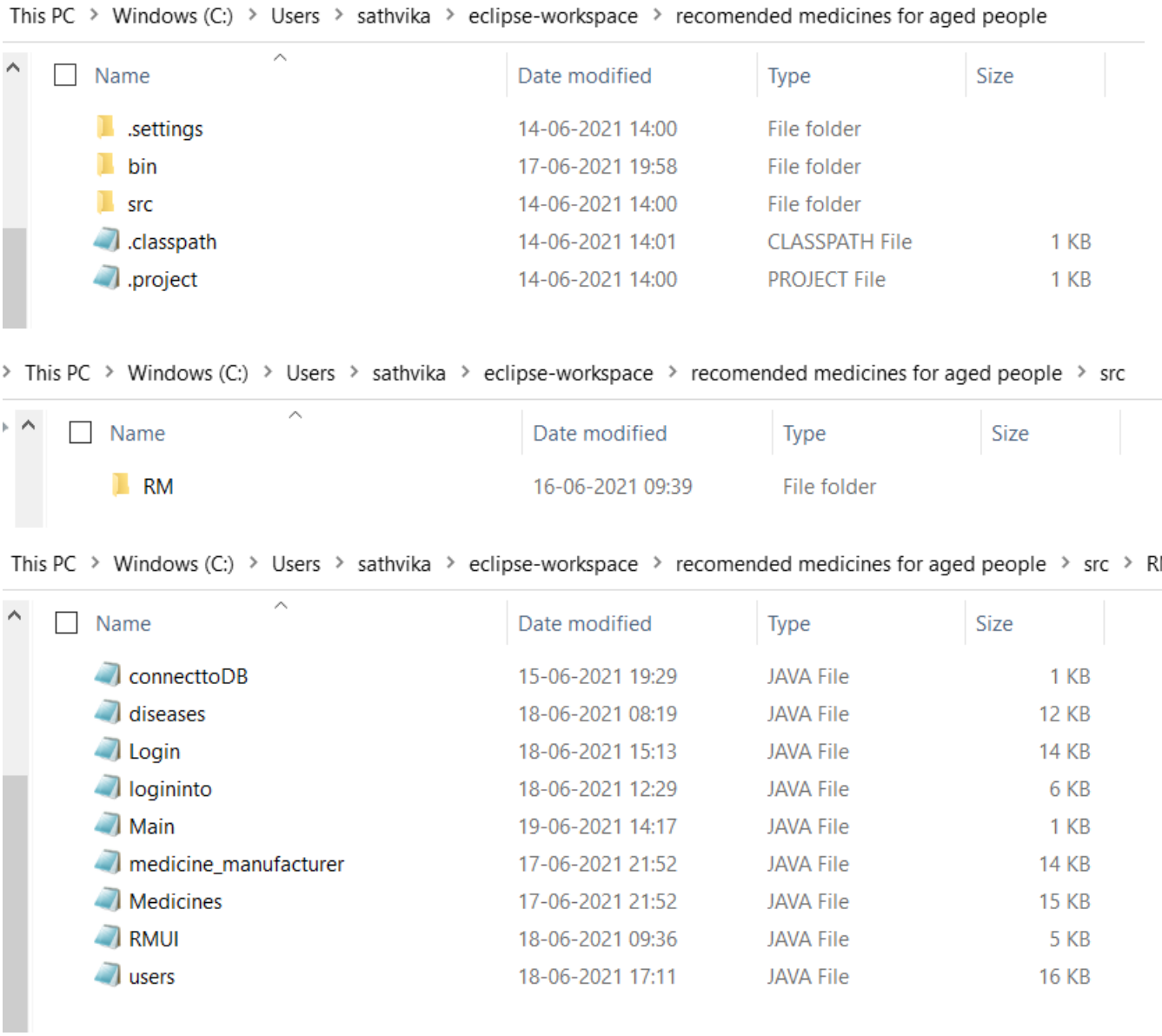
MAIN:

```
package RM;
public class Main {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                new RMUI();
        }

}
```

GITHUB LINK:

https://github.com/BejugamSathvika/Recommended-Medicines-for-Aged-people

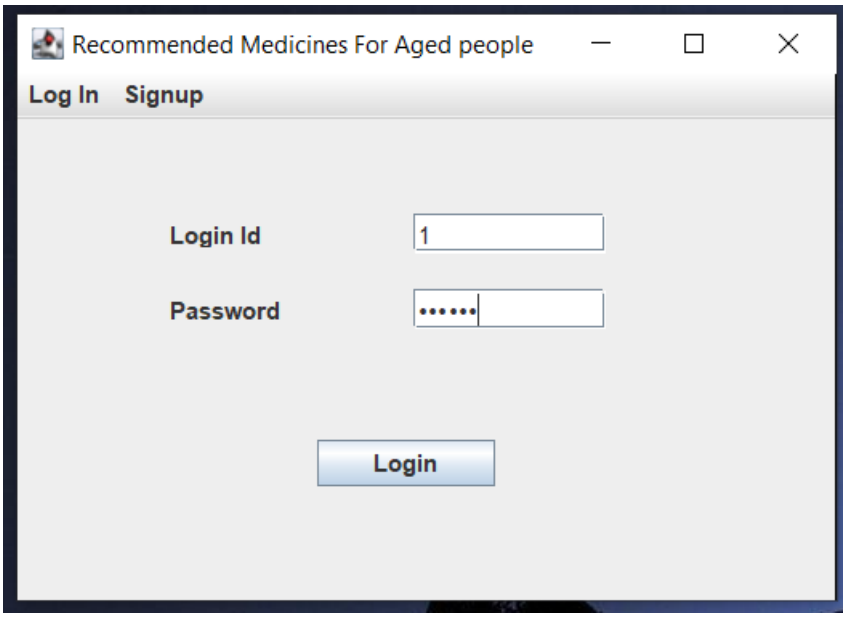
FOLDER STRUCTURE:

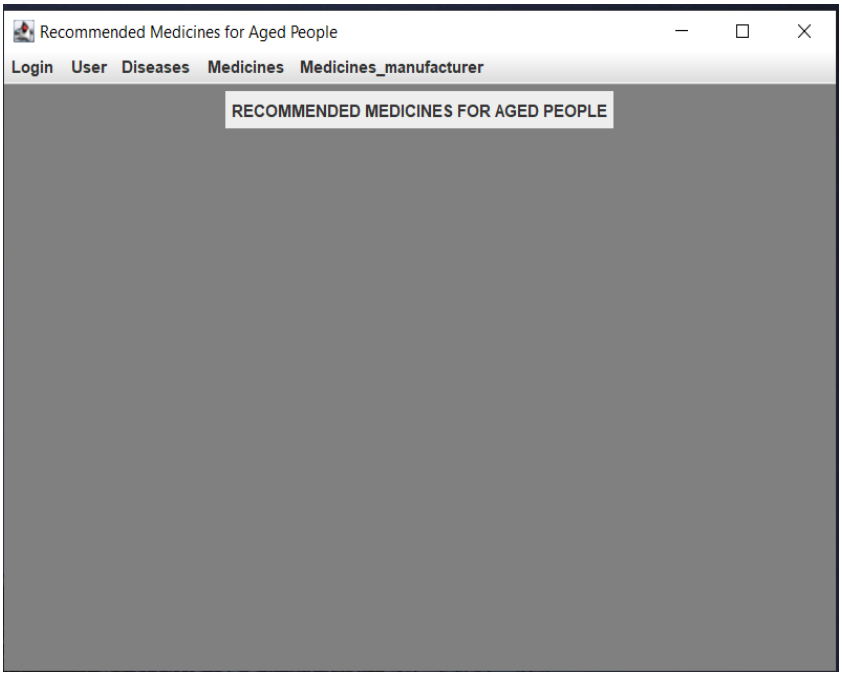


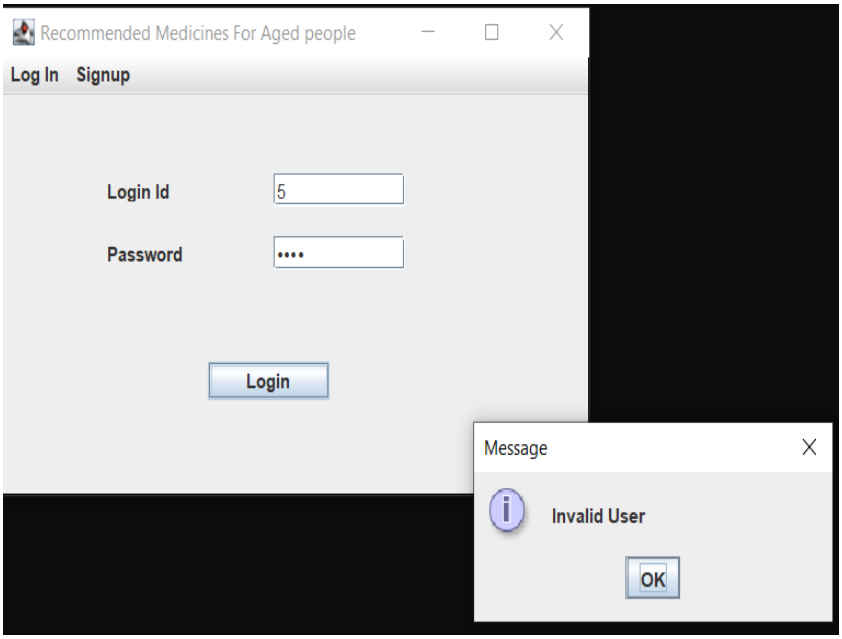
Testing:

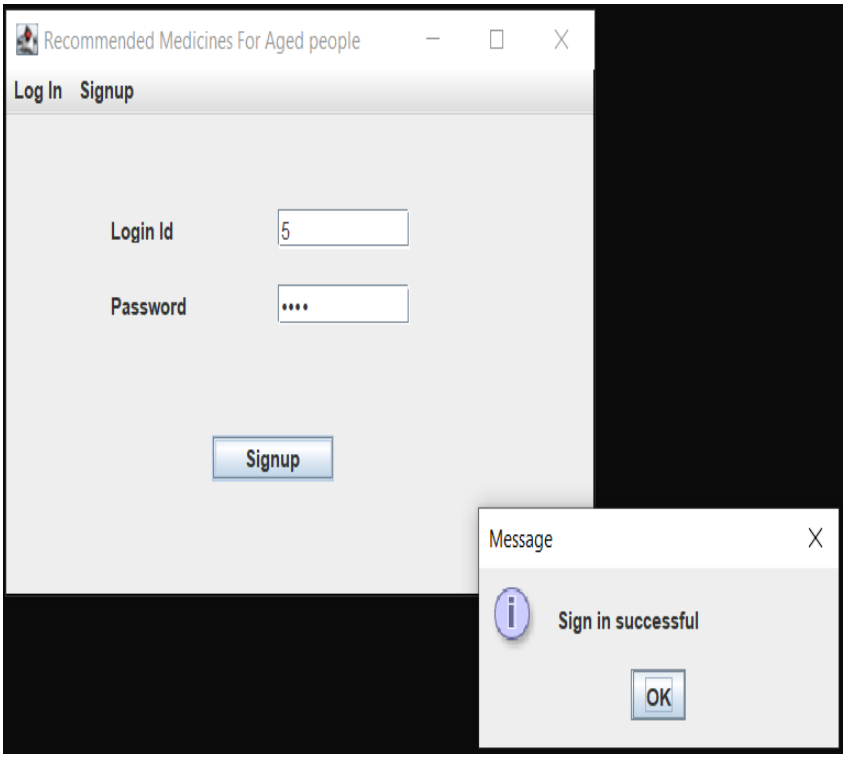
OUTPUT SCREENSHOTS:

If the logined user-type is admin:

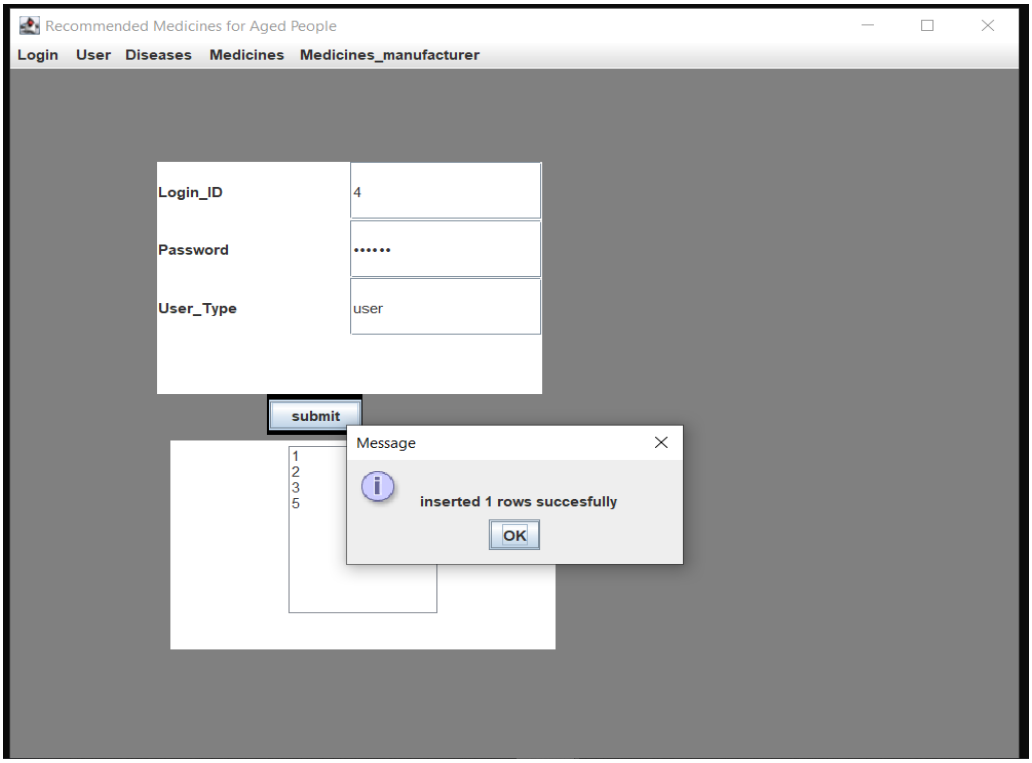


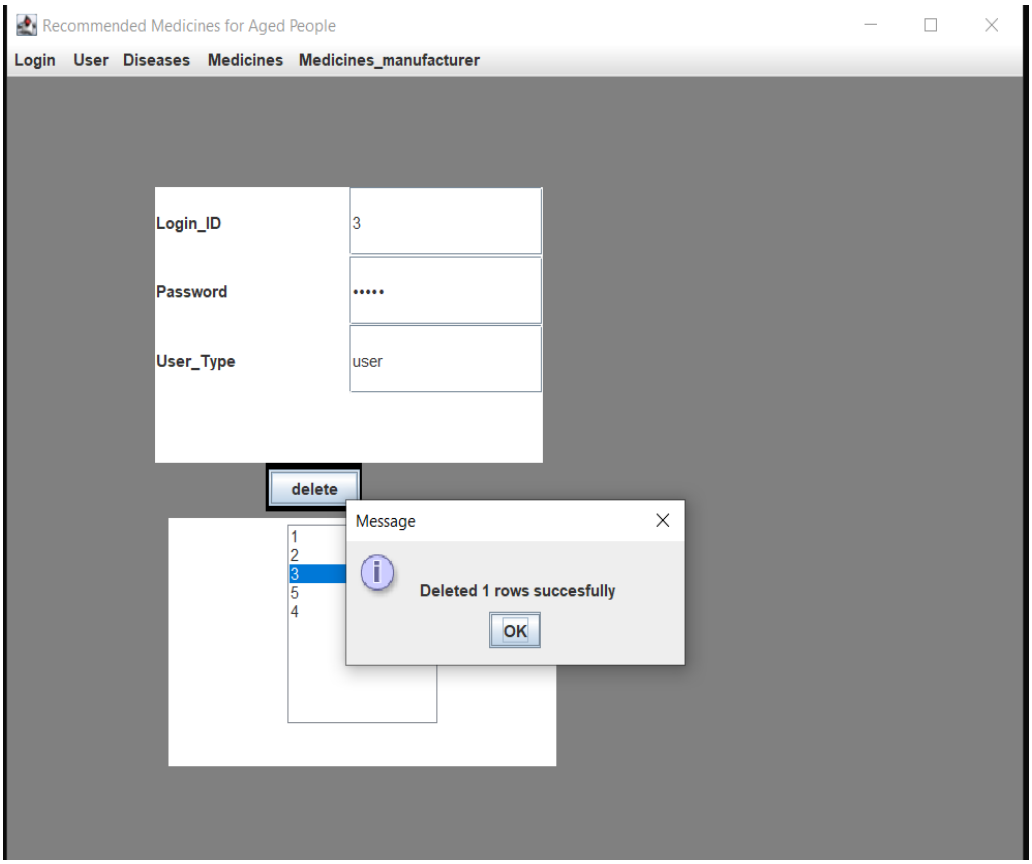



Recommended Medicines For Aged people
Log In
Signup
Login Id
5
Password
Login
Message
Invalid User
OK

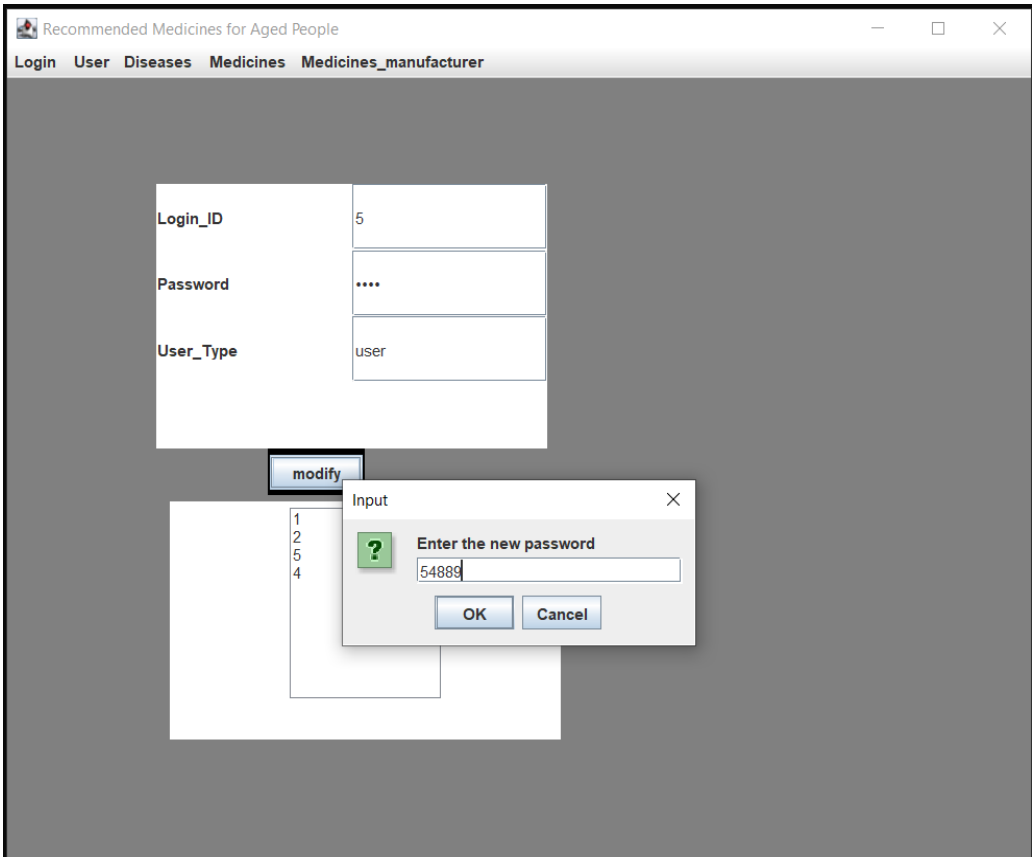

Recommended Medicines For Aged people
Log In
Signup
Login Id
5
Password
Signup
Message
Sign in successful
OK

TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***

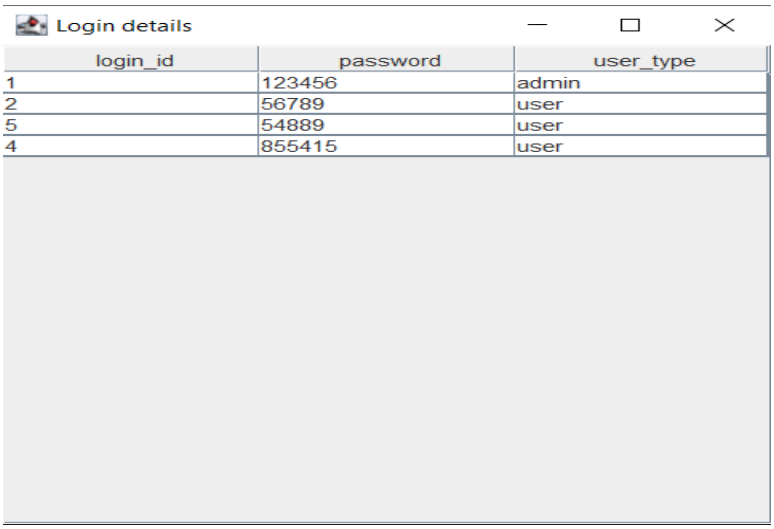


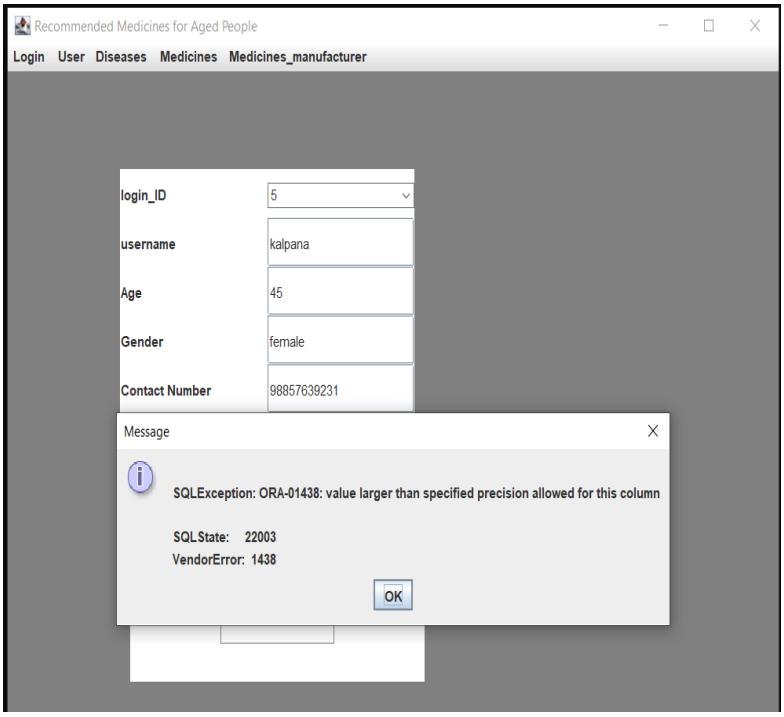


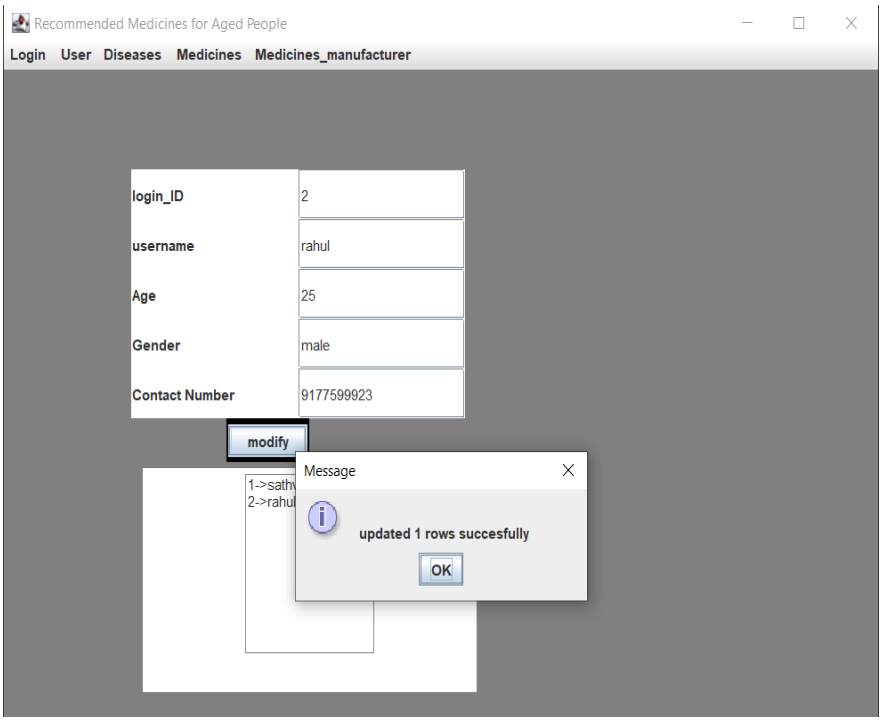
TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***

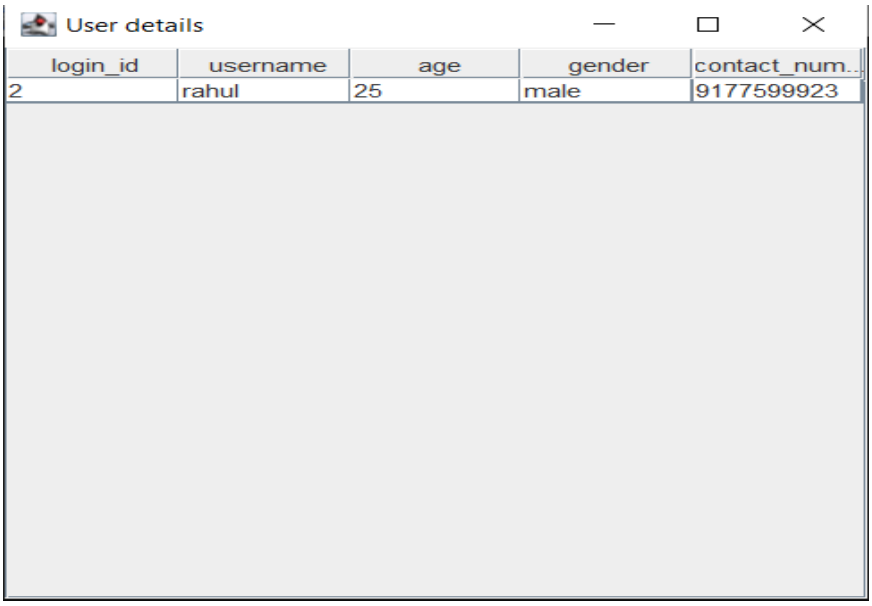


TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***

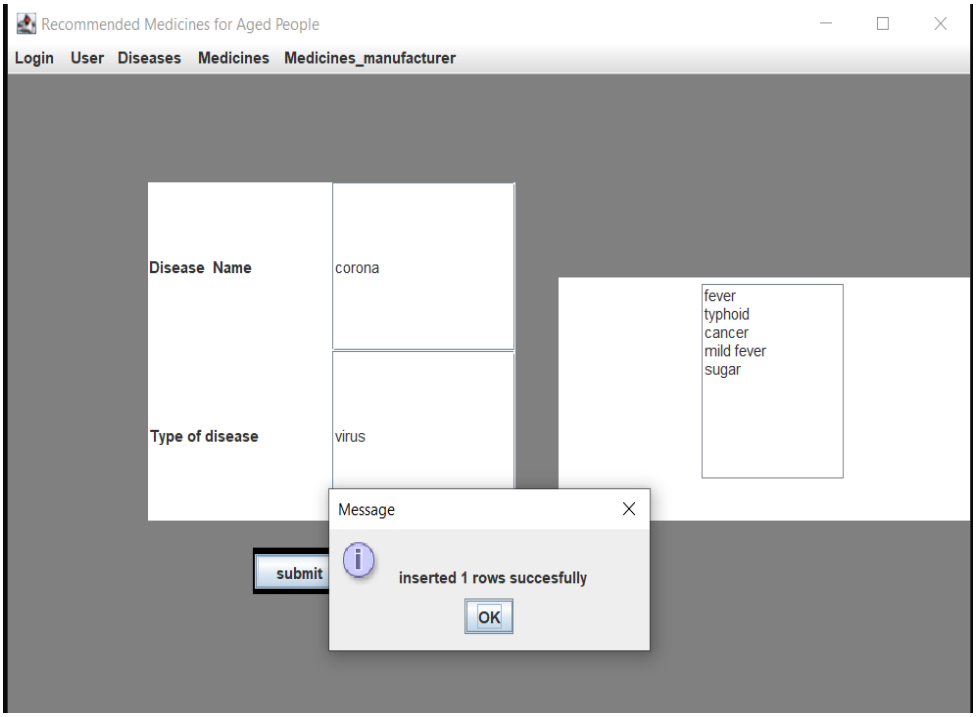


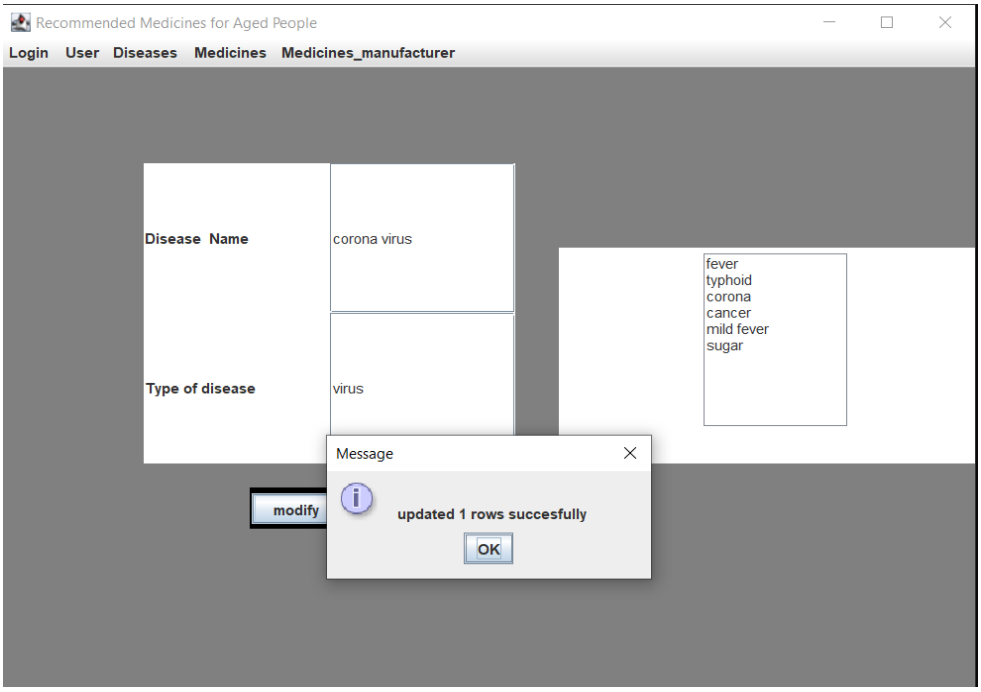



Recommended Medicines for Aged People
Login User Diseases Medicines Medicines_manufacturer
login_ID
2
username
rahul
Age
25
Gender
male
Contact Number
9177599923
modify
Message
updated 1 rows succesfully
OK

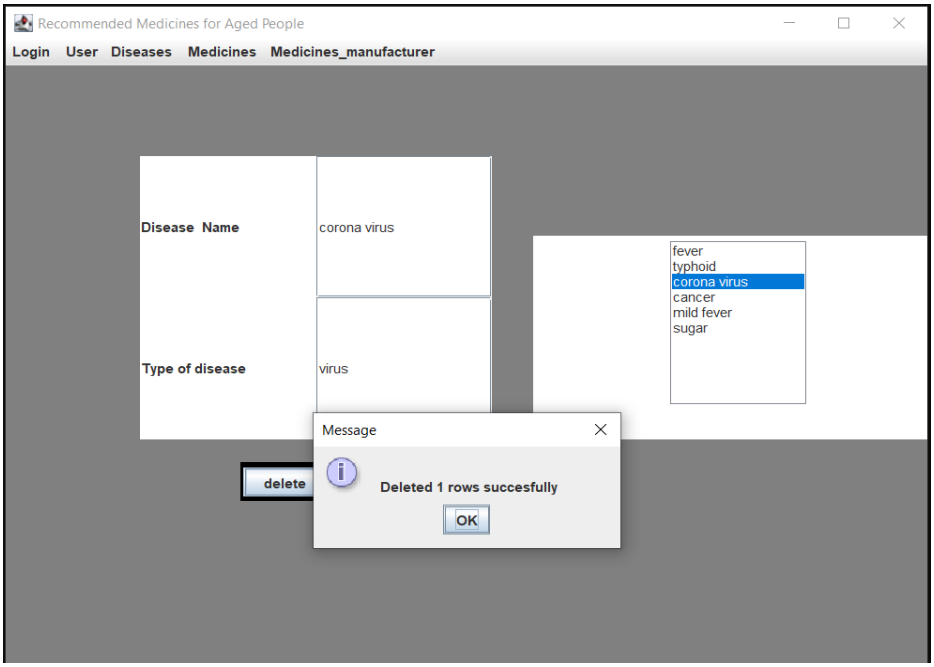

User details
login_id
username
age
gender
contact_num..
2
rahul
25
male
9177599923

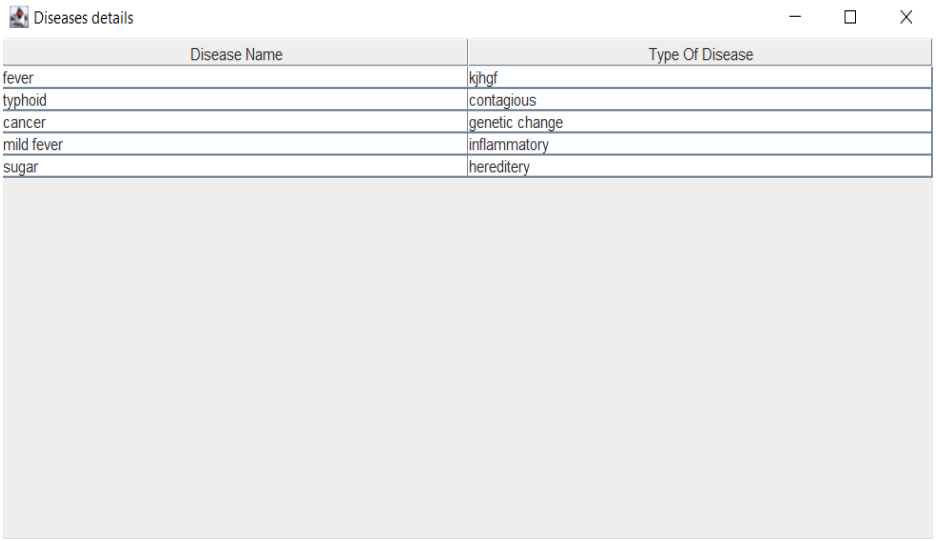
TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***



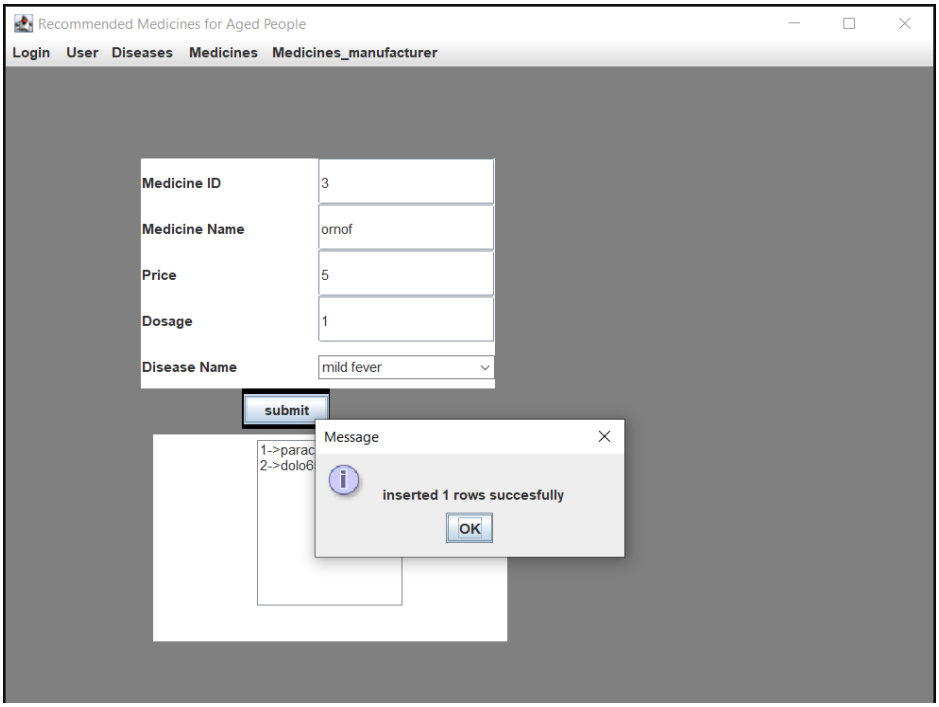


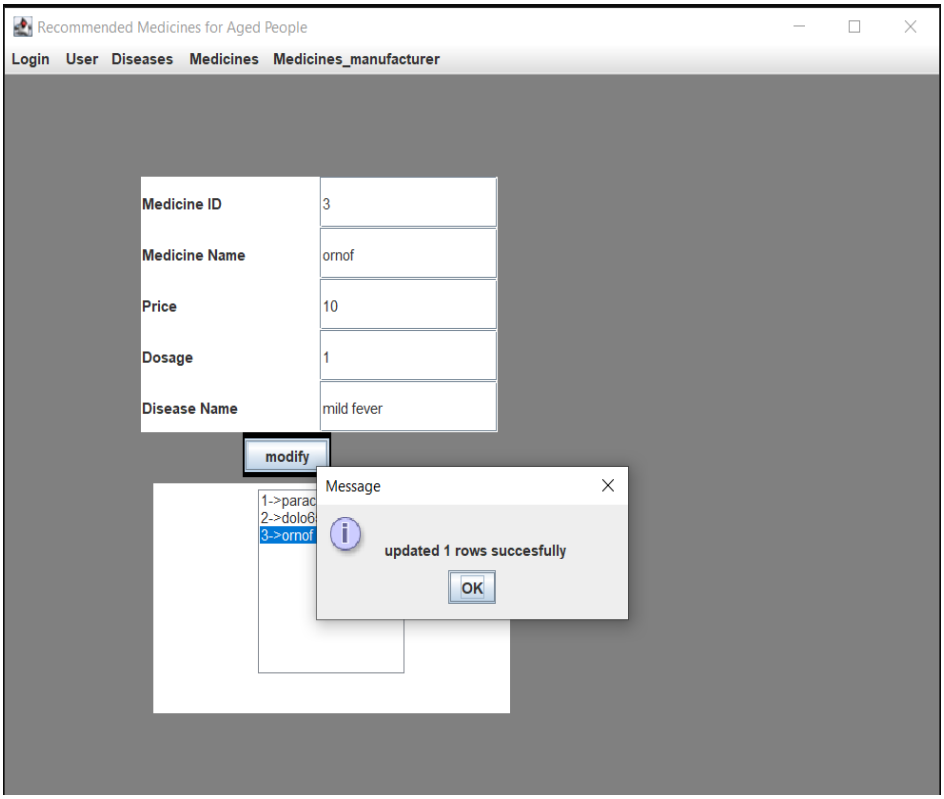
TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***



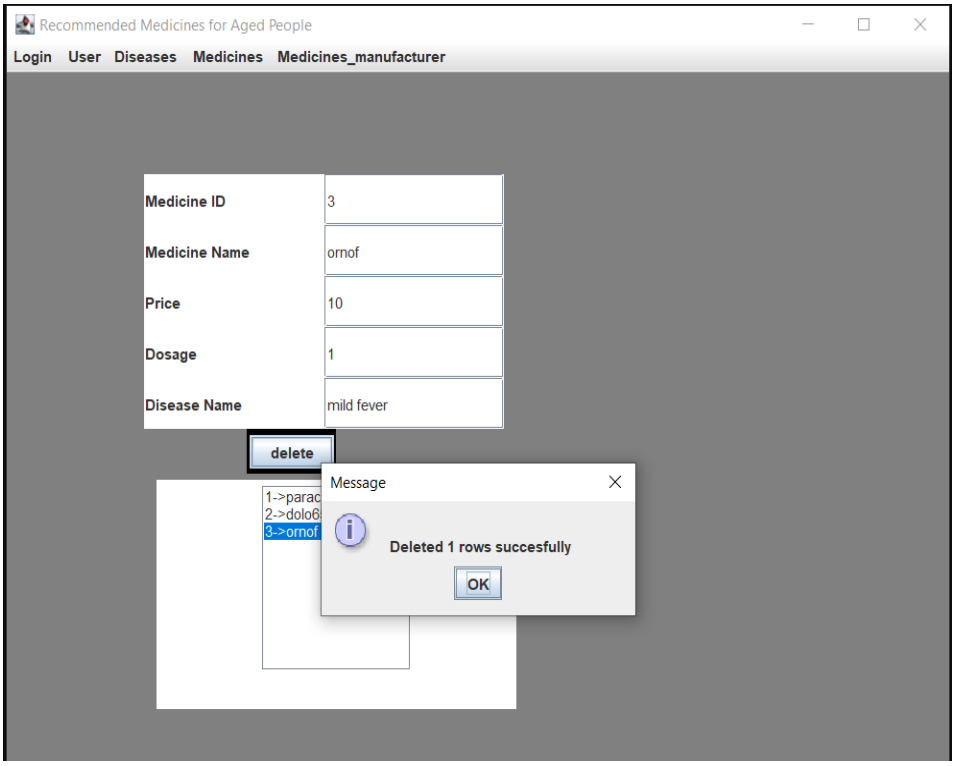


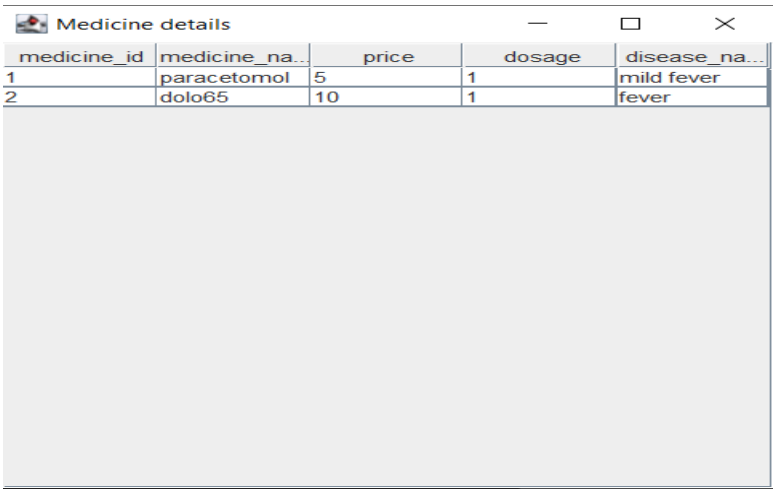
TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***



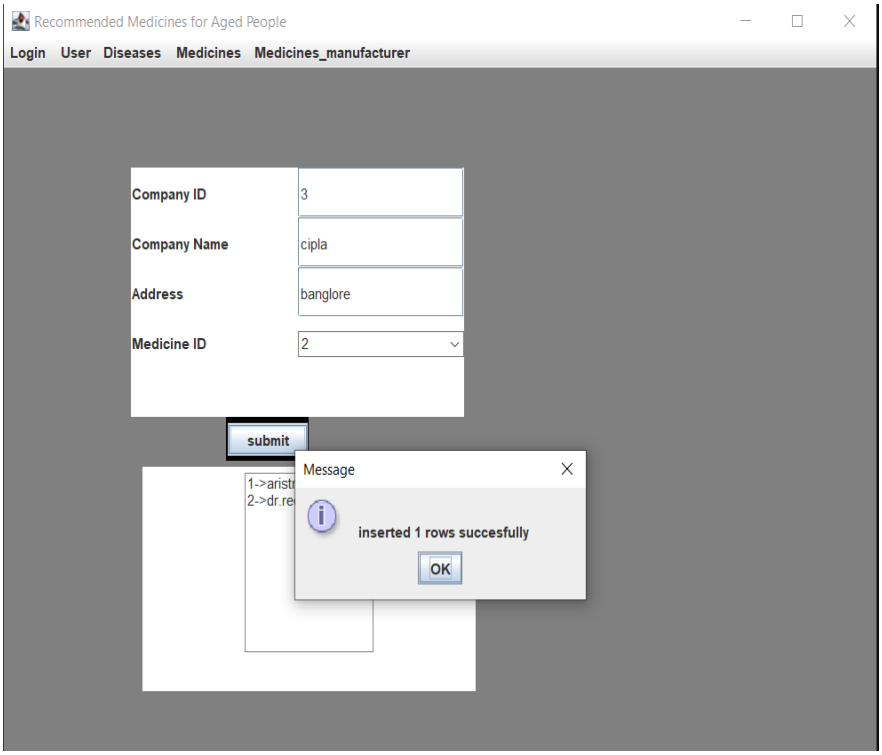


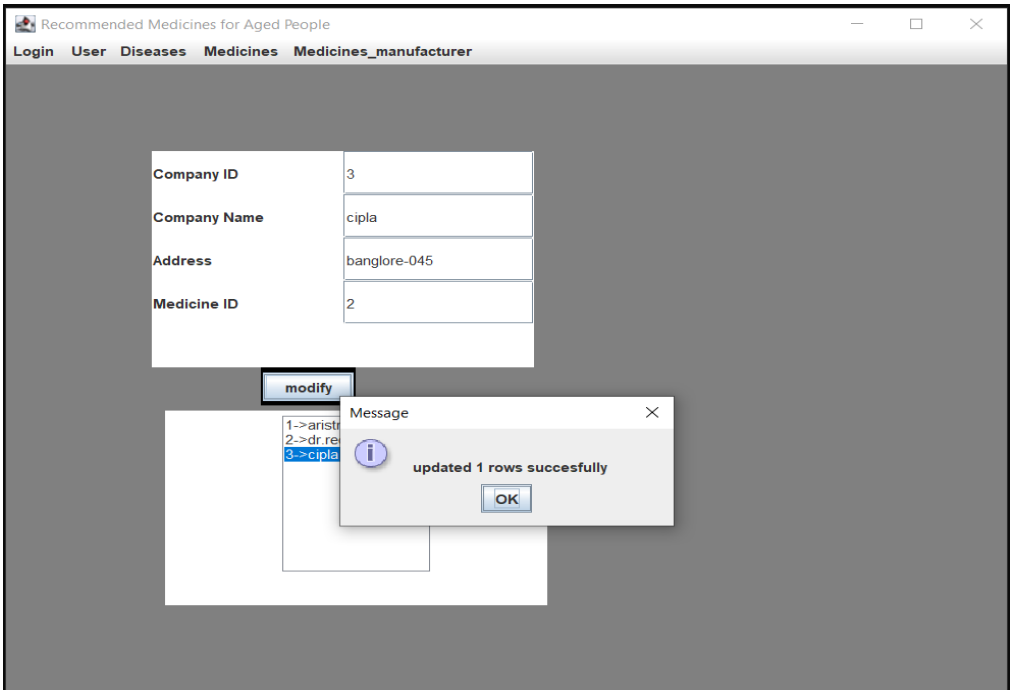
TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***





TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***

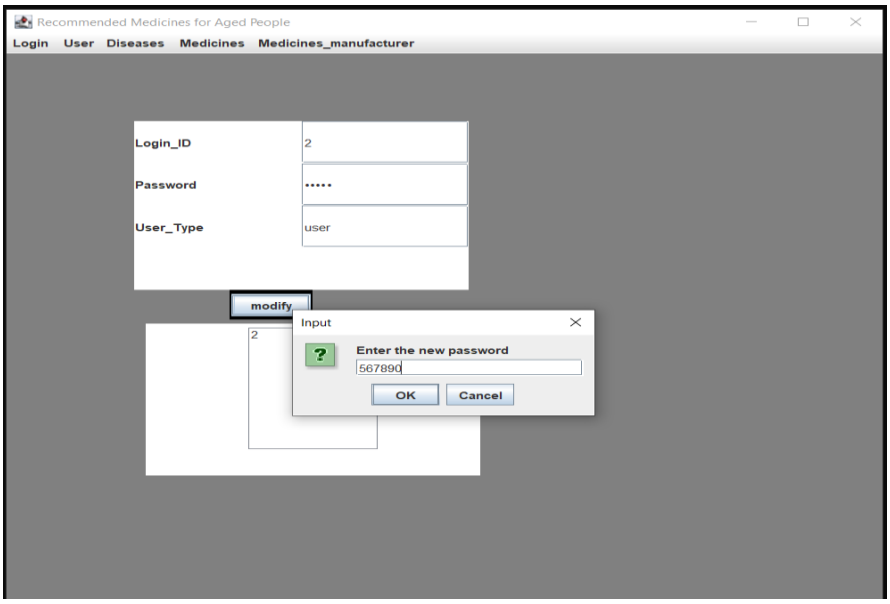


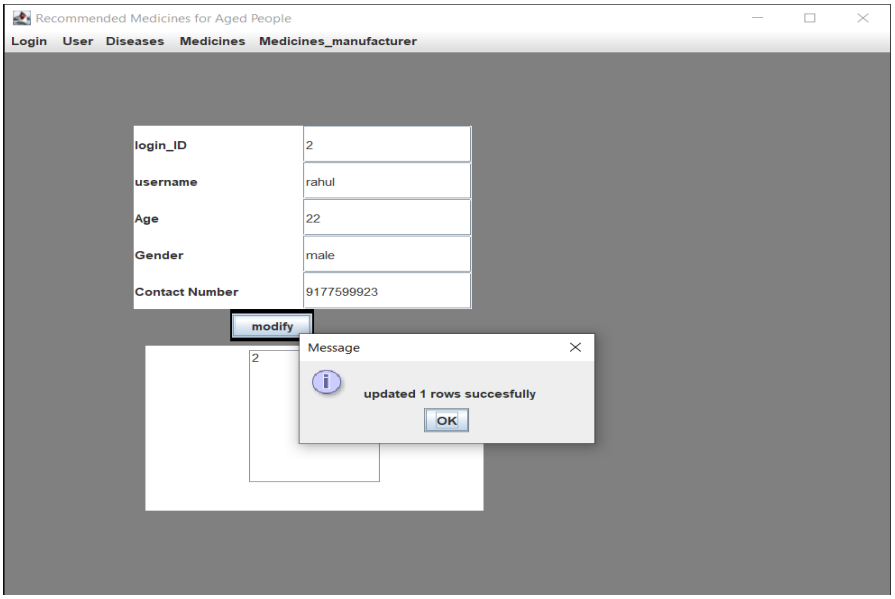


Medicine Manufacturer details

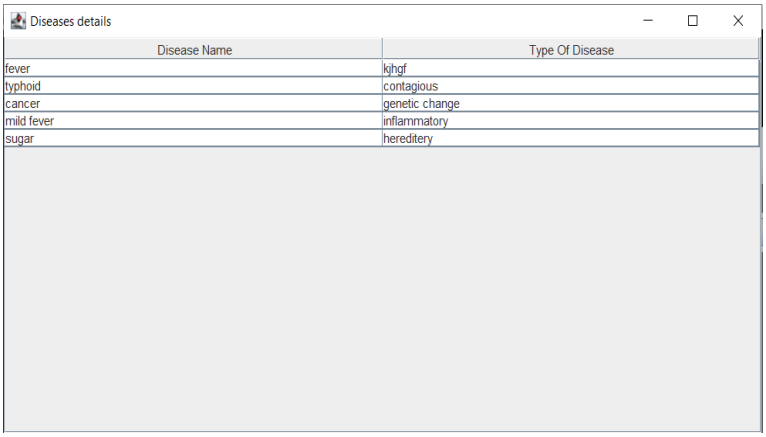
Company ID	Company Name	Address	Medicine ID
1	aristro	45/hyd	1
2	dr.reddys	654-kukatpally	1
3	cipla	banglore-045	2

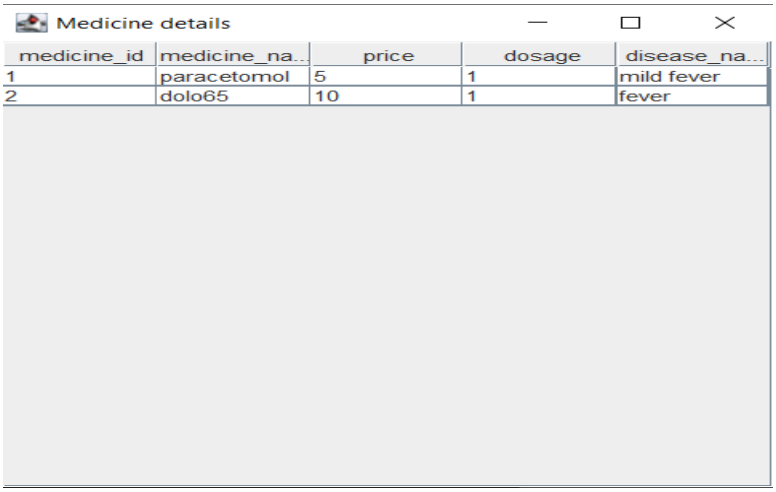
If logined user_type is user:-

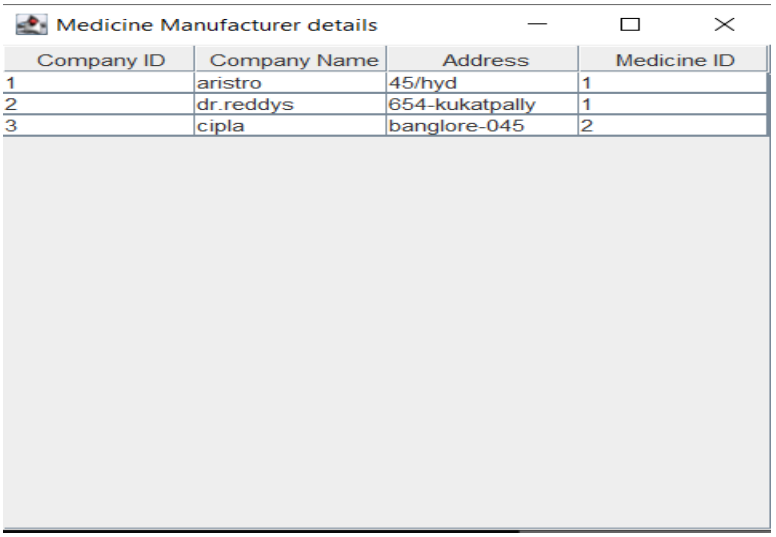




TILTLE: ***RECOMMENDED MEDICINES FOR AGED PEOPLE***







The data entered in the above form is updated in the "login" table of theOracle database 11g as:

```
SQL> select * from login;

  LOGIN_ID PASSWORD USER_TYPE
---------- -------- --------------------
         1 123456   admin
         2 567890   user
         5 54889    user
         4 855415   user
```

The data entered in the above form is updated in the "users" table of theOracle database 11g as:

```
SQL> select * from users;

  LOGIN_ID USERNAME                       AGE GENDER
---------- ---------------------------- ---------- --------------------
CONTACT_NUMBER
-------------
         2 rahul                         22 male
   9177599923

         1 sathvika                      19 female
   9390205320
```

The data entered in the above form is updated in the "diseases" table of the Oracle database 11g as:

```
SQL> select * from diseases;

DISEASE_NAME         TYPE_OF_DISEASE
-------------------- --------------------
fever                kjhgf
typhoid              contagious
cancer               genetic change
mild fever           inflammatory
sugar                hereditery
```

The data entered in the above form is updated in the "medicines" table of the Oracle database 11g as:

```
SQL> select * from medicines;

MEDICINE_ID MEDICINE_NAME            PRICE     DOSAGE DISEASE_NAME
----------- -------------------- ---------- ---------- --------------------
          1 paracetomol                   5          1 mild fever
          2 dolo65                       10          1 fever
```

The data entered in the above form is updated in the "medicine_manufacturer" table of theOracle database 11g as:

```
SQL> select * from medicine_manufacturer;

COMPANY_ID COMPANY_NAME                 ADDRESS
---------- ---------------------------- ------------------------------
MEDICINE_ID
-----------
         1 aristro                      45/hyd
           1

         2 dr.reddys                    654-kukatpally
           1

         3 cipla                        banglore-045
           2


SQL>
```

Results:

I had successfully completed PROJECT on "RECOMMENDED MEDICINES FOR AGED PEOPLE".

Discussion and future Work:

This application provides the customer to select see prescribed medicines for the respective disease. While working on this application I wanted to even make deliver medicines from their nearer medical shops so that they can even track the order. It should store the details and data of users, medicines and orders in appropriate manner.

CONCLUSION:

Thus, a Java SWING based RECOMMENDED MEDICINES FOR AGED PEOPLE is created which is connected to the Oracle 11g database. Therefore, all the entries and details are directly updated on their respective tables created in the database.

REFERENCES:

- https://docs.oracle.com/javase/7/docs/api/
- https://www.javatpoint.com/dbms-tutorial
- https://en.wikipedia.org/wiki/Online_pharmacy