JAVA SWING BASED- E MEDICINE SALES SYSTEM- SQL CONNECTIVITY 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

SUJITHA TADI <1602-18-737-107>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2019-2020

BONAFIDE CERTIFICATE

This to Certify that the project report titled

“***E MEDICINE SALES SYSTEM***”

project work of **SUJITHA TADI** bearing

Roll.no:**1602-18-737-107** who carried out this

project under my supervision in the IV semester for

the academic year 2019-2020.

Signature	Signature
external examiner	internal examiner

ABSTRACT

Digital marketing is also popularly known as online marketing or internet marketing. Online medicine purchase is popular worldwide due to convenience. This is a small-scale project for E medicine sales system. An online E medicine sales system that permits a customer to submit online orders for medicines or services from an online store that serves online customers. The basic idea is that customers can buy medicines online. The administrator can enter name, password and can create an account. User can post requirement for medicine. User can purchase medicine online. Medicine delivery provided by the nearest associate store. User can search for medicine in the online store. This application provides pre-information about the stock of the medicine. Users can order the required medicines online and can get them easily. They can even check the status of ordering. This application provides logins to the users. They can maintain their account. Once the customer decides to submit a purchase order, the customer may print all the contents of medicines in order to obtain a hard copy record of the transaction.

REQUIREMENT ANALYSIS

List of tables:

- LOGIN
- HAS
- USER ACCOUNT
- SEARCH
- MEDICINE
- ORDERS
- ORDER DETAILS

List of attributes with their domain types:

LOGIN:

- Login_id: number
- Username: varchar2
- Password :varchar2

Login_user:

Login_id : number()
User_id: number()
Day:DATE

User_account

User_id : number()
Username :varchar2()
Password : varchar2()
Pno: number()
Email: varchar(20)
Address : varchar()

User_medicine:

User_id: number()
Mid : number()
Quantity:number()
Day:DATE

Medicine:

Mid:number()
Mname:varchar()
Price: number()
In_stock: varchar2()

User_Orderdetails:

ORDER_ID : NUMBER(5)
MID :NUMBER(5)
USER_ID : NUMBER(5)
Day:DATE

Order_details:

ORDER_ID : NUMBER(5)
TOTAL_COST : NUMBER(8)
STATUS :VARCHAR2(20)

THROUGH THIS PROJECT: It develops an online pharmacy application for medical store to get online medical orders from anywhere 24 hours. This project helps to store the data of users and medicines in every shop and track their orders. The data can be stored for future purpose in case the user forgets the order details. User can easily select the medicine he/she want to order and can place it very easily. As there is track of both users, medicines as well as orders this makes this project more efficient. Users can maintain their account. Once the customer decides to submit a purchase order, the customer may print all the contents of medicines in order to obtain a hard copy record of the transaction.

AIM:

To create a **Java GUI based registration form** which takes the values like: student ID, student name, father name, phone number, department, address, city, email id from the user. These values are to be 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.

Java-SQL Connectivity using JDBC:

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

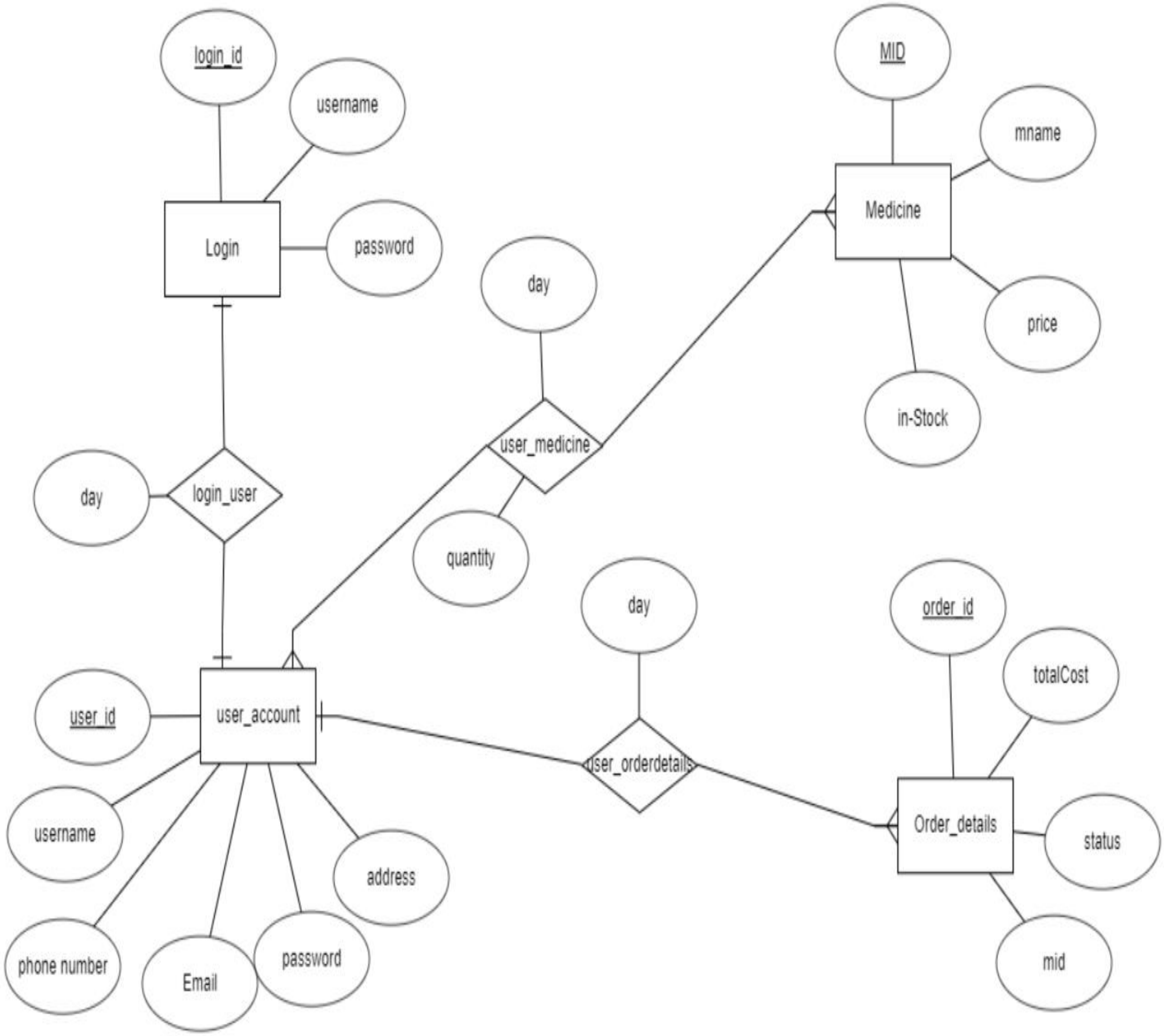
The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()
    {
             try {


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


             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);
             }
```

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

DESIGN:

ER DIAGRAM:



DATA DESIGN:

Mapping Cardinalities and Participation Constraints:

A user can have a single login Id therefore one to one mapping cardinality between login and user_account.

A user can search for many numbers of medicines and one medicine can be searched by many numbers of users so, many to many mapping cardinality between user_account and medicine.

A user can give many numbers of orders but all orders of one user are considered as one therefore many to one mapping cardinality between user_account and order details.

DDL COMMANDS:

Table Created in SQL for above mentioned purpose is as:

```
SQL> create table login( login_id number(5) primary key,username varchar2(15),password
varchar2(15));
        Table created.
```

```
SQL> create table medicine(mid number(5) primary key,mname varchar2(15),price
number(5),in_stock varchar2(7));
        Table created.
```

```
SQL> create table user_account(user_id number(5) primary key,username
varchar2(10),password varchar(10),pno number(10),email varchar(25),address varchar2(30));
Table created.
```

```
SQL> create table order_details(order_id number(5) primary key,total_cost number(8),status
varchar2(20) );
Table created.
```

```
SQL> create table login_user(login_id number(5),user_id number(5),day DATE,primary
key(login_id,user_id),foreign key(login_id) references login(login_id),foreign key(user_id)
references user_account(user_id));
        Table created.
```

```
SQL> create table user_medicine(user_id number(5),mid number(5),quantity number(5),day
DATE,primary key(user_id,mid),foreign key(user_id) references user_account(user_id),foreign
key(mid) references medicine(mid));
        Table created.

SQL> create table user_orderdetails(order_id number(5)  primary key,mid number(5),user_id
number(5),day DATE,foreign key(order_id) references order_details(order_id), foreign
key(mid) references medicine(mid),foreign key(user_id) references user_account(user_id));
Table created.

SQL> select * from tab;

TNAME                        TABTYPE  CLUSTERID
------------------------------ ------- ----------
LOGIN                        TABLE
LOGIN_USER                   TABLE
MEDICINE                     TABLE
ORDER_DETAILS                TABLE
USER_ACCOUNT                 TABLE
USER_MEDICINE                TABLE
USER_ORDERDETAILS            TABLE

7 rows selected.

SQL> desc login;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
LOGIN_ID                                  NOT NULL NUMBER(5)
USERNAME                                           VARCHAR2(15)
PASSWORD                                           VARCHAR2(15)

SQL> desc login_user;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
LOGIN_ID                                  NOT NULL NUMBER(5)
USER_ID                                   NOT NULL NUMBER(5)
DAY                                                DATE

SQL> desc medicine;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
MID                                       NOT NULL NUMBER(5)
MNAME                                              VARCHAR2(15)
PRICE                                              NUMBER(5)
IN_STOCK                                           VARCHAR2(7)
```

```
SQL> desc order_details;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
ORDER_ID                                  NOT NULL NUMBER(5)
TOTAL_COST                                         NUMBER(8)
STATUS                                             VARCHAR2(20)

SQL> desc user_account;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
USER_ID                                   NOT NULL NUMBER(5)
USERNAME                                           VARCHAR2(10)
PASSWORD                                           VARCHAR2(10)
PNO                                                NUMBER(10)
EMAIL                                              VARCHAR2(25)
ADDRESS                                            VARCHAR2(30)

SQL> desc user_medicine;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
USER_ID                                   NOT NULL NUMBER(5)
MID                                       NOT NULL NUMBER(5)
QUANTITY                                           NUMBER(5)
DAY                                                DATE

SQL> desc user_orderdetails;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
ORDER_ID                                  NOT NULL NUMBER(5)
MID                                                NUMBER(5)
USER_ID                                   NOT NULL NUMBER(5)
DAY                                                DATE
```

Implementation:

Program:

User Interface:

```
package EMed;
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 EMedUI extends JFrame{
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;

	private JMenuBar mnu;

	private JMenu mnulogin;
	private JMenu mnulog_user;
	private JMenu mnuuser;
	private JMenu mnumed;
	private JMenu mnuuser_med;
	private JMenu mnuorder_Det;
	private JMenu mnuuser_orderDet;

	private JMenuItem insert1,update1,delete1,view1;
	private JMenuItem insert2,update2,delete2,view2;
	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 JMenuItem insert7,update7,delete7,view7;

	private JLabel labelName;

	private static JPanel p0,p1;
	void initialize() {
		mnu=new JMenuBar();

		mnulogin= new JMenu("Login");
		mnulog_user= new JMenu("log_user");
		mnuuser= new JMenu("User");
		mnumed= new JMenu("Medicines");
		mnuuser_med= new JMenu("User-Medicine");
		mnuorder_Det= new JMenu("Order_Details");
		mnuuser_orderDet= new JMenu("User_orderDet");
		labelName=new JLabel("E-MEDICINE SALES SYSTEM");
		 p1=new JPanel();
		p0=new JPanel();
		insert1=new JMenuItem("Insert");
```

```
            update1=new JMenuItem("Update");
            delete1=new JMenuItem("Delete");
            view1=new JMenuItem("View");
            insert2=new JMenuItem("Insert");
            update2=new JMenuItem("Update");
            delete2=new JMenuItem("Delete");
            view2=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");
            insert7=new JMenuItem("Insert");
            update7=new JMenuItem("Update");
            delete7=new JMenuItem("Delete");
            view7=new JMenuItem("View");
            insert1=new JMenuItem("Insert");
            update1=new JMenuItem("Update");
            delete1=new JMenuItem("Delete");
            view1=new JMenuItem("View");
    }
    void addComponentsToFrame() {
             mnulogin.add(insert1);
             mnulogin.add(delete1);
             mnulogin.add(update1);
             mnulogin.add(view1);
              mnulog_user.add(insert2);
             mnulog_user.add(delete2);
             mnulog_user.add(update2);
             mnulog_user.add(view2);
             mnuuser.add(insert3);
             mnuuser.add(delete3);
             mnuuser.add(update3);
             mnuuser.add(view3);
              mnumed.add(insert4);
             mnumed.add(delete4);
             mnumed.add(update4);
             mnumed.add(view4);
              mnuuser_med.add(insert5);
             mnuuser_med.add(delete5);
```

```
                    mnuuser_med.add(update5);
                    mnuuser_med.add(view5);
                     mnuorder_Det.add(insert6);
                    mnuorder_Det.add(delete6);
                    mnuorder_Det.add(update6);
                    mnuorder_Det.add(view6);
                     mnuuser_orderDet.add(insert7);
                    mnuuser_orderDet.add(delete7);
                    mnuuser_orderDet.add(update7);
                    mnuuser_orderDet.add(view7);
                    mnu.add(mnulogin);
                    mnu.add(mnuuser);
                    mnu.add(mnulog_user);
                    mnu.add(mnumed);
                    mnu.add(mnuuser_med);
                    mnu.add(mnuorder_Det);
                    mnu.add(mnuuser_orderDet);
                    setJMenuBar(mnu);
                    p1.add(labelName);p1.setAlignmentY(CENTER_ALIGNMENT);
                    p1.setBounds(500,500,800,100);
                   p0.add(p1);
                    p0.setBackground(Color.CYAN);
                    add(p0);
          }
void closeWindow(){
                   try {
                             int a=JOptionPane.showConfirmDialog(this,"Are you sure want to Quit EMedicine Sales
System:");
                             if(a==JOptionPane.YES_OPTION){
                                       JOptionPane.showMessageDialog(this,
                                                    "Thank you!\nExiting EMedicine Sales System","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,EMedUI.this,insert1,delete1,update1,view1);
                   log.buildGUI();
                   User_account user=new User_account(p0,EMedUI.this,insert3,delete3,update3,view3);
                   user.buildGUI();
                   login_user log_user=new login_user(p0,EMedUI.this,insert2,delete2,update2,view2);
```

```
        log_user.buildGUI();
        Medicine med=new Medicine(p0,EMedUI.this,insert4,delete4,update4,view4);
        med.buildGUI();
        UserMedicine umed=new UserMedicine(p0,EMedUI.this,insert5,delete5,update5,view5);
        umed.buildGUI();
        orderDetails orders=new orderDetails(p0,EMedUI.this,insert6,delete6,update6,view6);
        orders.buildGUI();
        UserOrderDetails u_orders=new
UserOrderDetails(p0,EMedUI.this,insert7,delete7,update7,view7);
        u_orders.buildGUI();
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent we)
            {
                closeWindow();
            }
        });


}
public EMedUI() {
        initialize();
        addComponentsToFrame();
        register();
        pack();
    //  setBackground(Color.RED);
        setTitle("E-Medicine Sales System");
        setSize(800,800);
        setVisible(true);
    }
}
```

USER_MEDICINE TABLE

```
package EMed;

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 UserMedicine{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JButton insertButton,deleteButton,updateButton,viewButton;
    private JPanel p1,p2,p3,p;
    private JLabel lbluser_id,lblmid,lblquantity,lblday;
    private JTextField txtuser_id,txtmid,txtquantity,txtday;
    private Choice user_id,mid;
    private List UserMIDList;
    Connection con;ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public UserMedicine(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;

        lbluser_id=new JLabel("User ID");
        lblmid=new JLabel("Medicine ID");
        lblquantity=new JLabel("Quantity");
        lblday=new JLabel("Date");

        user_id=new Choice();
```

```
                mid=new Choice();

                txtuser_id=new JTextField(15);
                txtmid=new JTextField(15);
                txtquantity=new JTextField(8);
                txtday=new JTextField(8);

                this.p=p;



        }

        public void connectToDB()
   {
                try {


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


                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 loaduserIDs() {
                try {
                        UserMIDList.removeAll();
                        rs=statement.executeQuery("select * from user_medicine");
                        while(rs.next()) {
                                UserMIDList.add(rs.getString("USER_ID")+"->"+rs.getString("MID"));
                        }
                        }
                catch(SQLException e) {
                        displaySQLErrors(e);
                }
        }
        public void loadusers() {
                try {
                        user_id.removeAll();
                        rs=statement.executeQuery("select * from user_account");
                        while(rs.next()) {
                                user_id.add(rs.getString("User_ID"));
                        }
                        }
                catch(SQLException e) {
                        displaySQLErrors(e);
                }
                }
        public void loadmids() {
                try {
                        mid.removeAll();
                        rs=statement.executeQuery("select * from medicine");
                        while(rs.next()) {
                                mid.add(rs.getString("MID"));
                        }
                        }
                catch(SQLException e) {
                        displaySQLErrors(e);
                }
                }

        public void buildGUI() {



                insert.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent arg0) {
                                // TODO Auto-generated method stub
                                insertButton=new JButton("insert");
                                txtuser_id.setText(null);
                                txtmid.setText(null);
                                txtquantity.setText(null);
                                txtday.setText(null);
                                loadusers();
                                loadmids();
                                p.removeAll();
                                frame.invalidate();
                                frame.validate();
                                frame.repaint();


                                p1=new JPanel();
                        p1.setLayout(new GridLayout(4,1));

                        p1.add(lbluser_id);
                        p1.add(user_id);
                        p1.add(lblmid);
                        p1.add(mid);
                        p1.add(lblquantity);
                        p1.add(txtquantity);
                        p1.add(lblday);
                        p1.add(txtday);
                        p3=new JPanel(new FlowLayout());
                        p3.add(insertButton);
                        //p1.add(txtf1);
                        p3.setBackground(Color.yellow);
                       p3.setBounds(200,350,75,35);
                        p1.setBackground(Color.pink) ;

                        p1.setBounds(115,80,300,250);

                        p2 = new JPanel(new FlowLayout());

                              UserMIDList=new List(10);
                              loaduserIDs();
                              p2.add(UserMIDList);
                              p2.setBounds(450,150,350,180);  p2.setBackground(Color.cyan) ;


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

                       UserMIDList.setEnabled(false);
                        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 USER_MEDICINE
VALUES("+user_id.getSelectedItem()+","+mid.getSelectedItem()+","+txtquantity.getText()+",'"+txtday.getText(
)+"')";

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

```
                        }
                        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);
                        txtuser_id.setText(null);
                        txtmid.setText(null);
                        txtquantity.setText(null);
                        txtday.setText(null);

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


                        p1=new JPanel();

                         p1.setLayout(new GridLayout(4,1));

                         p1.add(lbluser_id);
                         p1.add(txtuser_id);
                         p1.add(lblmid);
                         p1.add(txtmid);
                         p1.add(lblquantity);
                         p1.add(txtquantity);
                         p1.add(lblday);
                         p1.add(txtday);

                         p3=new JPanel(new FlowLayout());
                         p3.add(deleteButton);
                        //p1.add(txtf1);
                         p3.setBackground(Color.yellow);
                        p3.setBounds(200,350,75,35);
                         p1.setBackground(Color.pink) ;

                         p1.setBounds(115,80,300,250);

                         p2 = new JPanel(new FlowLayout());

                             UserMIDList=new List(10);
                             loaduserIDs();
                             p2.add(UserMIDList);
                             p2.setBounds(450,150,350,180);  p2.setBackground(Color.cyan) ;


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

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

                                            while (rs.next())
                                            {
                                                if (rs.getString("user_id").equals(p) &&
            rs.getString("mid").equals(q))
                                                break;
                                            }
                                            if (!rs.isAfterLast())
                                            {
                                                txtmid.setText(rs.getString("mid"));

                txtuser_id.setText(rs.getString("user_id"));

                txtquantity.setText(rs.getString("quantity"));
                                                txtday.setText(rs.getString("day"));
                                            }
                                        }
                                        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(UserMIDList.getSelectedItem(),"->");

                                                String query="DELETE FROM USER_MEDICINE WHERE
USER_ID="+st.nextToken();

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


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

                                         }


                                                });
                                }
                                });

                update.addActionListener(new ActionListener() {

                                @Override
                                public void actionPerformed(ActionEvent arg0) {
                                        // TODO Auto-generated method stub
                                        JButton updateButton = new JButton("update");
                                        txtuser_id.setText(null);
                                        txtmid.setText(null);
                                        txtquantity.setText(null);
                                        txtday.setText(null);

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

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


        p1=new JPanel();

         p1.setLayout(new GridLayout(4,1));

         p1.add(lbluser_id);
         p1.add(txtuser_id);
         p1.add(lblmid);
         p1.add(txtmid);
         p1.add(lblquantity);
         p1.add(txtquantity);
         p1.add(lblday);
         p1.add(txtday);
         p3=new JPanel(new FlowLayout());
         p3.add(updateButton);
         //p1.add(txtf1);
         p3.setBackground(Color.yellow);
        p3.setBounds(200,350,75,35);
         p1.setBackground(Color.pink) ;

         p1.setBounds(115,80,300,250);

         p2 = new JPanel(new FlowLayout());

             UserMIDList=new List(10);
             loaduserIDs();
             p2.add(UserMIDList);
             p2.setBounds(450,150,350,180);  p2.setBackground(Color.cyan) ;


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

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

                         while (rs.next())
                         {
```

```
                                    if (rs.getString("user_id").equals(p) &&
rs.getString("mid").equals(q))
                                    break;
                                }
                                if (!rs.isAfterLast())
                                {
                                    txtmid.setText(rs.getString("mid"));

    txtuser_id.setText(rs.getString("user_id"));

    txtquantity.setText(rs.getString("quantity"));
                                    txtday.setText(rs.getString("day"));
                                }
                            }
                            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(UserMIDList.getSelectedItem(),"->");
                                String query="update user_medicine set
mid="+txtmid.getText()+",quantity="+txtquantity.getText()+",day='"+txtday.getText()+"' WHERE
USER_ID="+st.nextToken()+"and mid="+st.nextToken();

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


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

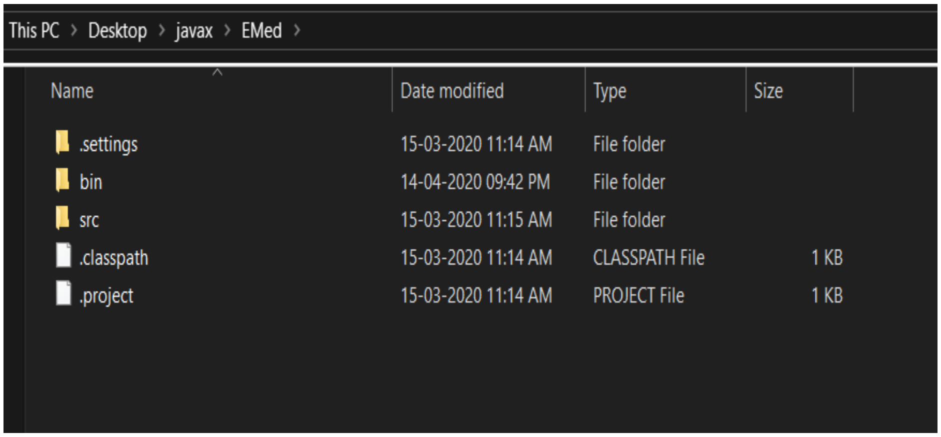
```
                                    }


                                            });
                        }
                        });

                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_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.cyan)
;p2.setBackground(Color.cyan) ;
                                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_Medicine Details ");


                                                      DefaultTableModel model = new DefaultTableModel();
                                                      j = new JTable(model);
                                                     model.addColumn("User ID");
```

```
                                                        model.addColumn("Medicine ID");
                                                        model.addColumn("Quantity");
                                                        model.addColumn("Day");

                                                        try {

                                                                        rs=statement.executeQuery("select * 
from user_medicine");
                                                                        while(rs.next()) {
                                                                                model.addRow(new 
Object[]{rs.getString("user_id"), rs.getString("mid"),rs.getString("quantity"),rs.getString("day")});
                                                                        }
                                                                        }
                                                                catch(SQLException viewException) {
                                                                        displaySQLErrors(viewException);
                                                                }
                                                                j.setEnabled(false);
                                                        j.setBounds(30, 40, 180, 150);


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

                                                        f.setSize(500, 400);

                                                        f.setVisible(true);

                                                      }
                                             });
                              }
                      });
              }
}
```

GITHUB LINK:

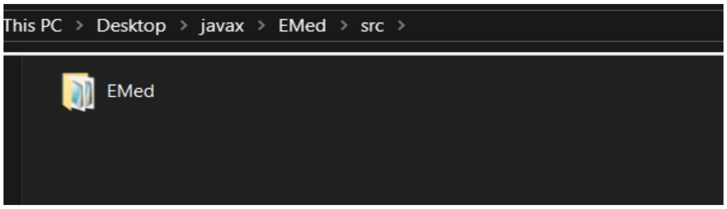
https://github.com/sujithatadi/DBMS-PROJECT

FOLDER STRUCTURE:



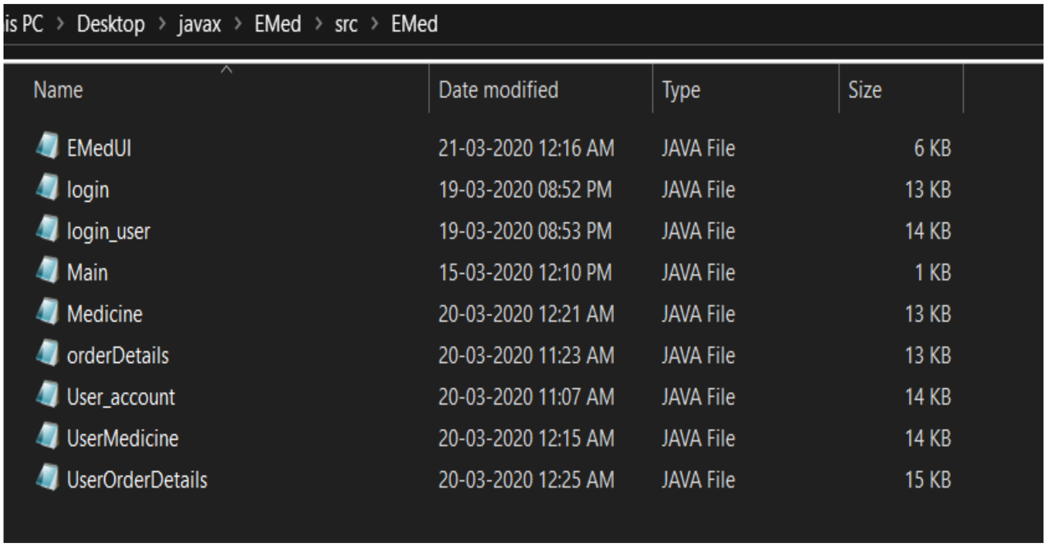
This PC › Desktop › javax › EMed ›

Name	Date modified	Type	Size
.settings	15-03-2020 11:14 AM	File folder	
bin	14-04-2020 09:42 PM	File folder	
src	15-03-2020 11:15 AM	File folder	
.classpath	15-03-2020 11:14 AM	CLASSPATH File	1 KB
.project	15-03-2020 11:14 AM	PROJECT File	1 KB



This PC › Desktop › javax › EMed › src ›

EMed



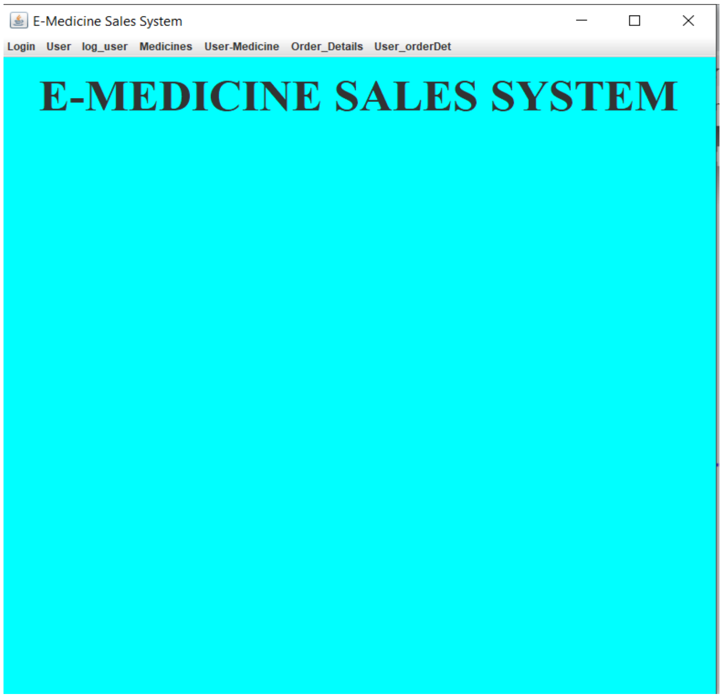
is PC › Desktop › javax › EMed › src › EMed

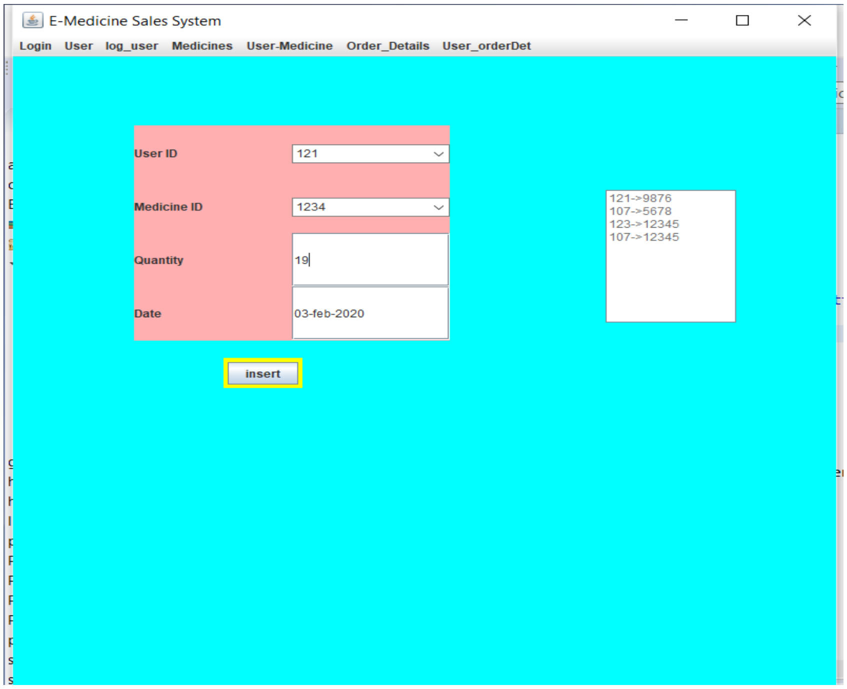
Name	Date modified	Type	Size
EMedUI	21-03-2020 12:16 AM	JAVA File	6 KB
login	19-03-2020 08:52 PM	JAVA File	13 KB
login_user	19-03-2020 08:53 PM	JAVA File	14 KB
Main	15-03-2020 12:10 PM	JAVA File	1 KB
Medicine	20-03-2020 12:21 AM	JAVA File	13 KB
orderDetails	20-03-2020 11:23 AM	JAVA File	13 KB
User_account	20-03-2020 11:07 AM	JAVA File	14 KB
UserMedicine	20-03-2020 12:15 AM	JAVA File	14 KB
UserOrderDetails	20-03-2020 12:25 AM	JAVA File	15 KB

Testing:

OUTPUT SCREENSHOTS:

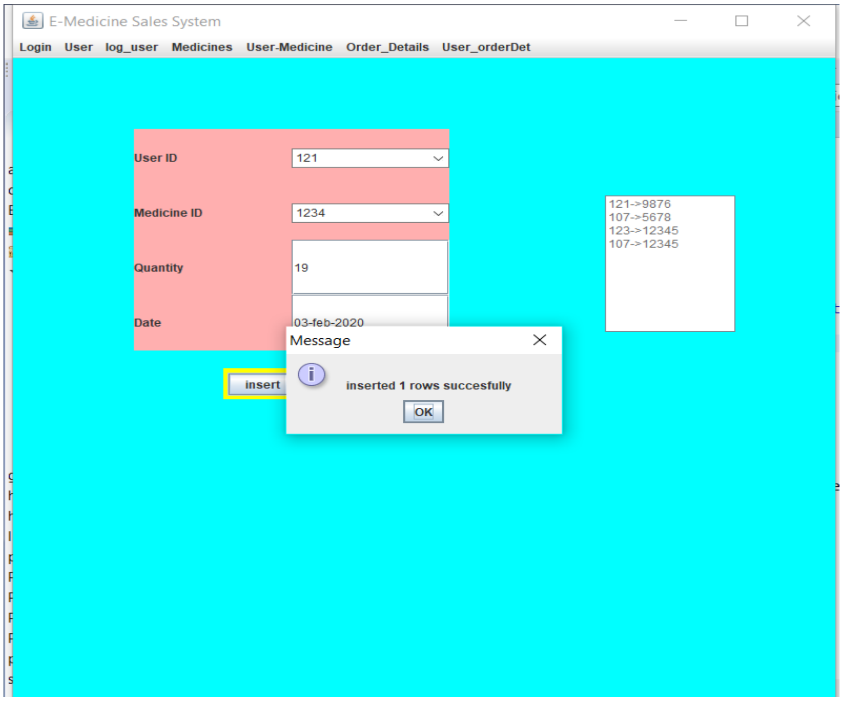
Java GUI Screenshot:





User ID	121
Medicine ID	1234
Quantity	19
Date	03-feb-2020

```
121->9876
107->5678
123->12345
107->12345
```

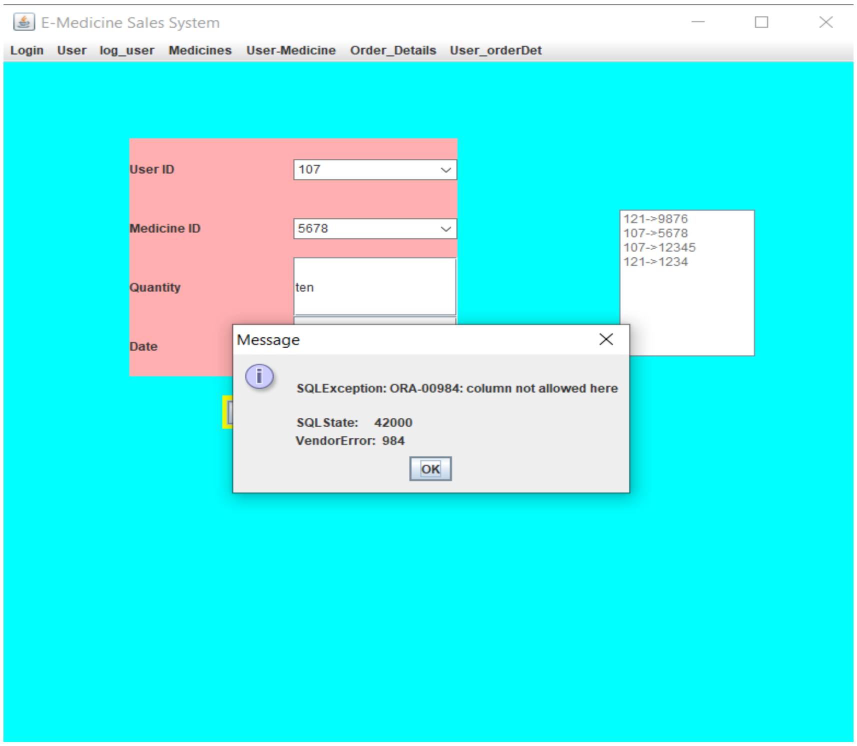

E-Medicine Sales System
Login User log_user Medicines User-Medicine Order_Details User_orderDet
User ID
Medicine ID
Quantity
Date
insert
OK

User ID: 121
Medicine ID: 1234
Quantity: 19
Date: 03-feb-2020

```
121->9876
107->5678
123->12345
107->12345
```

Message

inserted 1 rows succesfully

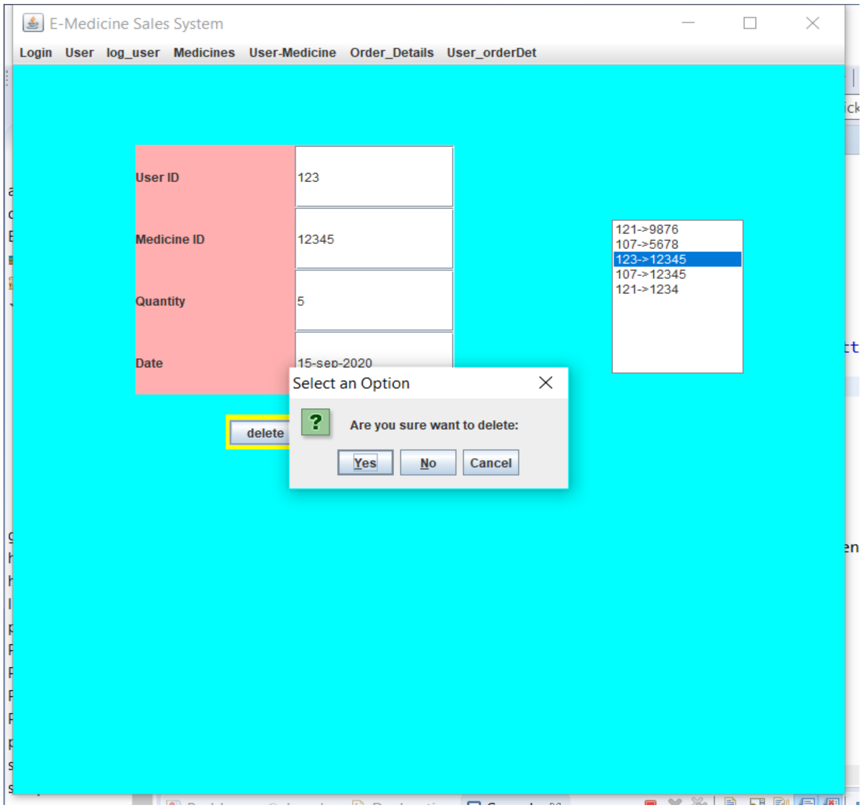

E-Medicine Sales System
Login User log_user Medicines User-Medicine Order_Details User_orderDet
User ID
Medicine ID
Quantity
Date
OK

User ID: 107
Medicine ID: 5678
Quantity: ten

```
121->9876
107->5678
107->12345
121->1234
```

Message

```
SQLException: ORA-00984: column not allowed here

SQLState:    42000
VendorError: 984
```


E-Medicine Sales System
Login User log_user Medicines User-Medicine Order_Details User_orderDet
delete
Yes
No
Cancel

User ID 123

Medicine ID 12345

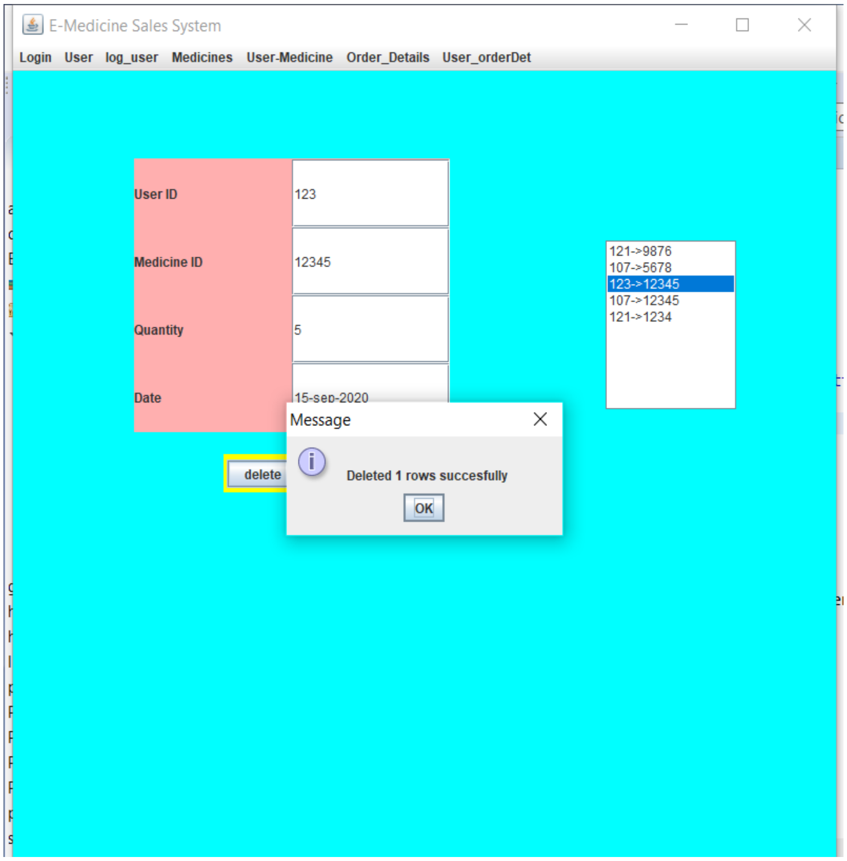
Quantity 5

Date 15-sep-2020

121->9876
107->5678
123->12345
107->12345
121->1234

Select an Option

Are you sure want to delete:


E-Medicine Sales System
Login User log_user Medicines User-Medicine Order_Details User_orderDet
delete
OK

User ID 123

Medicine ID 12345

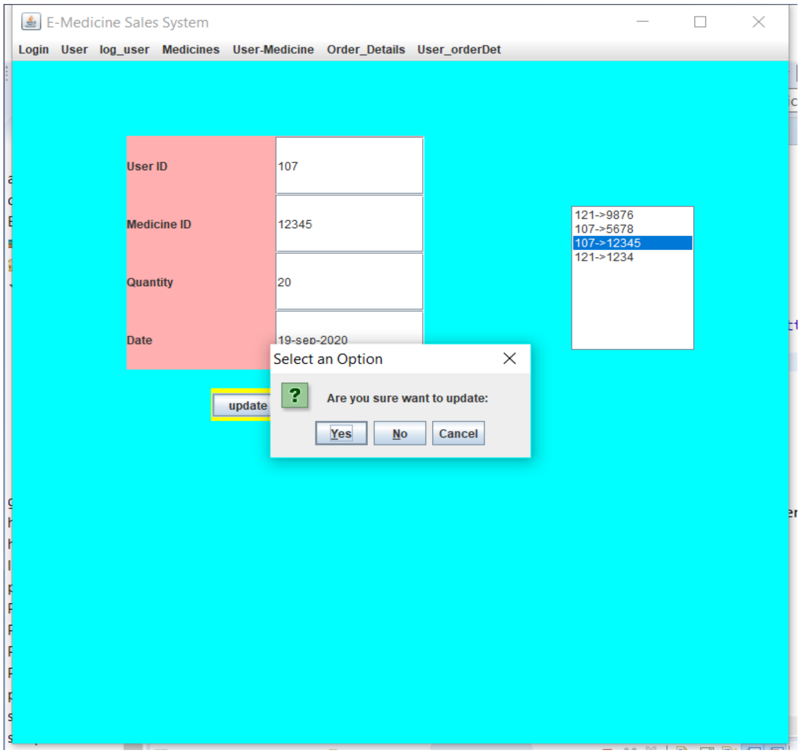
Quantity 5

Date 15-sep-2020

121->9876
107->5678
123->12345
107->12345
121->1234

Message

Deleted 1 rows succesfully



E-Medicine Sales System
Login User log_user Medicines User-Medicine Order_Details User_orderDet
update
Yes
No
Cancel

User ID 107

Medicine ID 12345

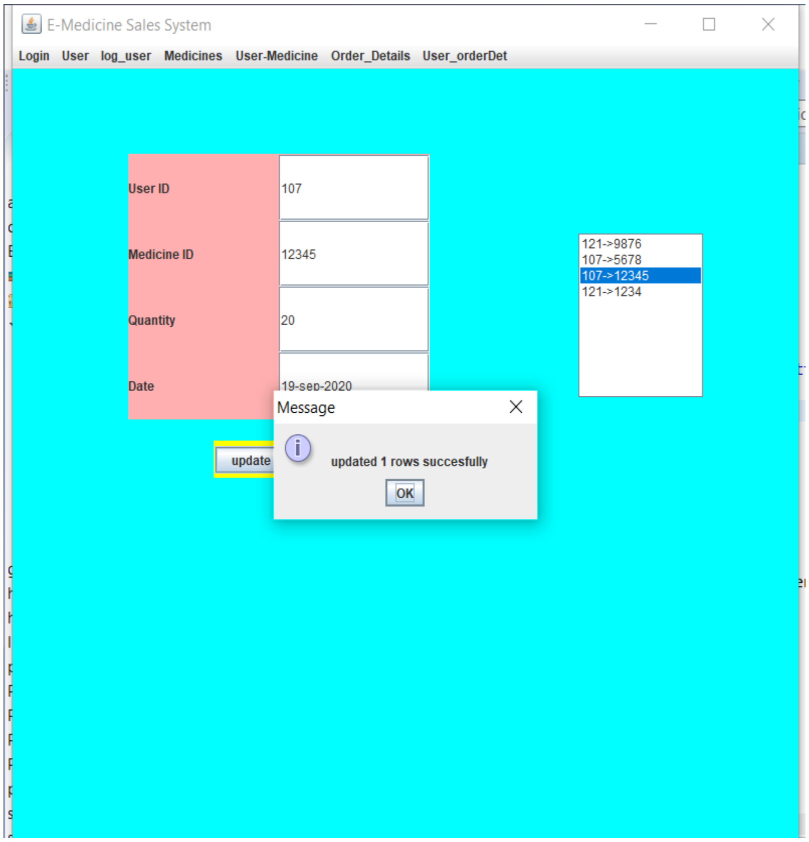
Quantity 20

Date 19-sep-2020

121->9876
107->5678
107->12345
121->1234

Select an Option

Are you sure want to update:



E-Medicine Sales System
Login User log_user Medicines User-Medicine Order_Details User_orderDet
update
OK

User ID 107

Medicine ID 12345

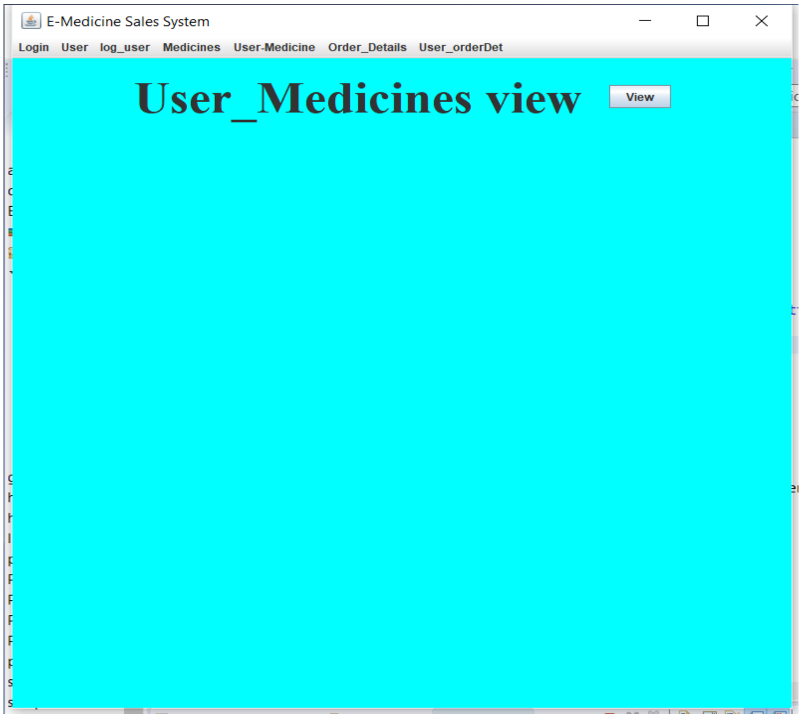
Quantity 20

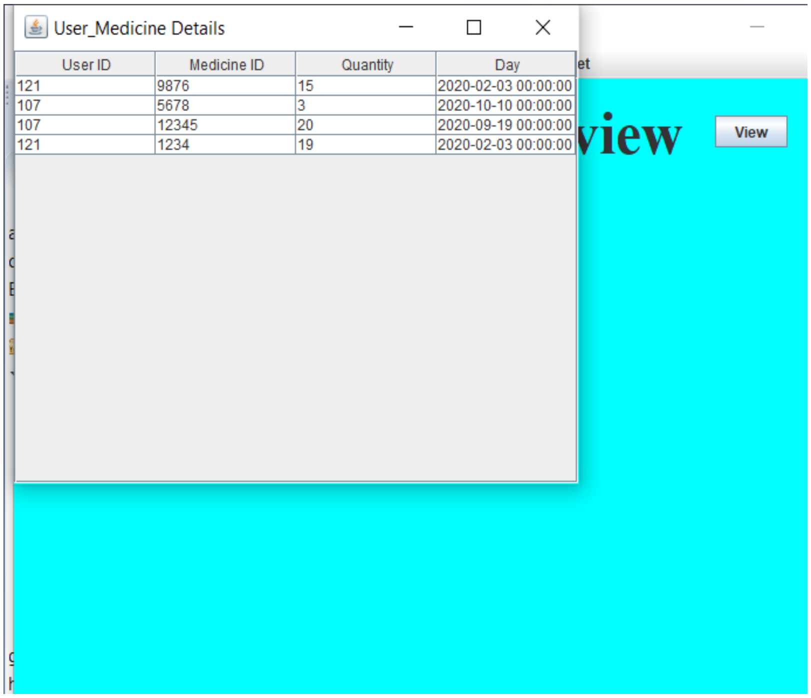
Date 19-sep-2020

121->9876
107->5678
107->12345
121->1234

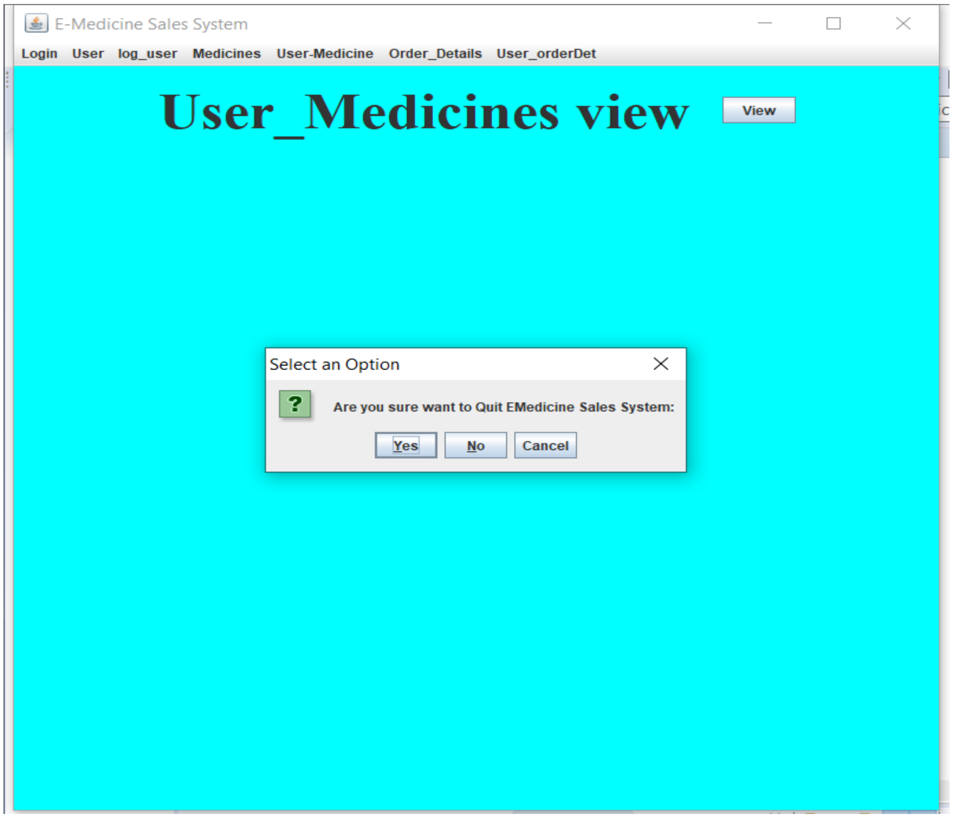
Message

updated 1 rows succesfully



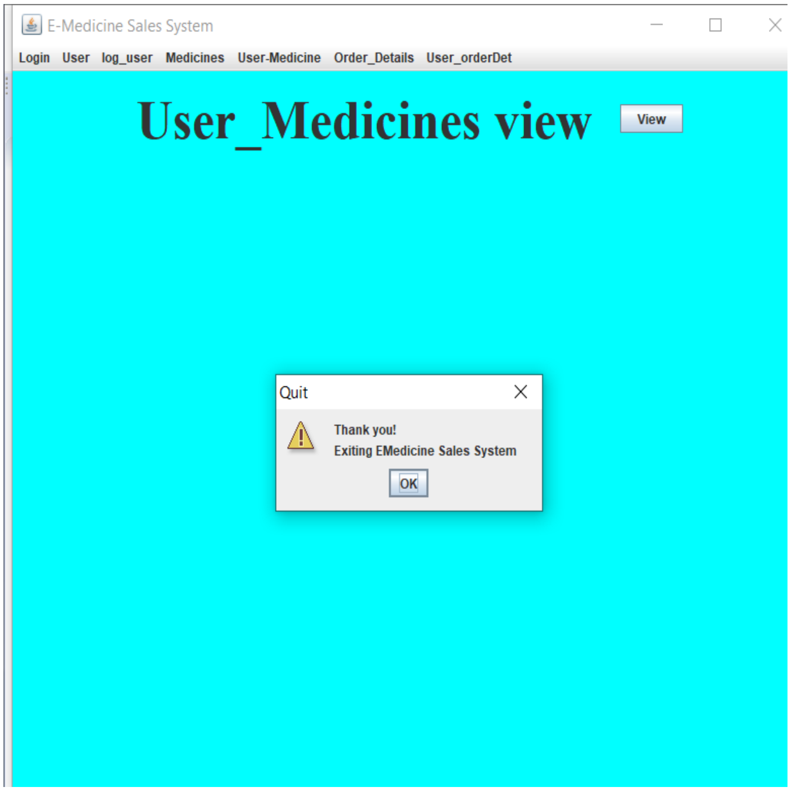


User ID	Medicine ID	Quantity	Day
121	9876	15	2020-02-03 00:00:00
107	5678	3	2020-10-10 00:00:00
107	12345	20	2020-09-19 00:00:00
121	1234	19	2020-02-03 00:00:00



Select an Option

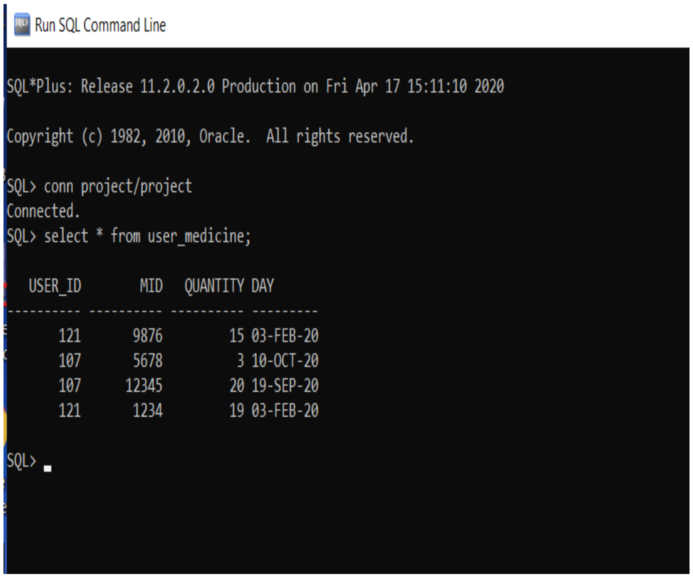
Are you sure want to Quit EMedicine Sales System:



Quit

Thank you!
Exiting EMedicine Sales System

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



```
SQL*Plus: Release 11.2.0.2.0 Production on Fri Apr 17 15:11:10 2020

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn project/project
Connected.
SQL> select * from user_medicine;

   USER_ID        MID   QUANTITY DAY
---------- ---------- ---------- --------
       121       9876         15 03-FEB-20
       107       5678          3 10-OCT-20
       107      12345         20 19-SEP-20
       121       1234         19 03-FEB-20

SQL> _
```

Results:

I had successfully completed MINI PROJECT on "E MEDICINE SALES SYSTEM".

Discussion and future Work:

This application provides the customer to select the medicines and can order them from respective medical shops and track the order. While working on this project I wanted to extend to make a app which is user friendly and provides accurate information. It stores the details and data of users, medicines and orders in appropriate manner.

CONCLUSION:

Thus, a Java SWING based E-MEDICINE SALES SYSTEM 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