

# **JAVASWINGBASED–INVENTORY MANAGEMENT SYSTEM-SQL CONNECTIVITY USING JDBC**

**A**

*Report*

*Submitted in partial fulfilment of the  
Requirements for the award of the Degree of  
**BACHELOR OF TECHNOLOGY**  
IN*

**INFORMATION TECHNOLOGY**

By

**ANTHATI KARTHIK <1602-19-737-310>**

**Under the Guidance of**

**B. Leelavathy**



**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020-2021**

## **BONAFIDE CERTIFICATE**

**This to Certify that the project report titled  
“INVENTORY MANAGEMENT SYSTEM” project work  
of Mr.ANTHATI KARTHIK bearing  
Roll.no:1602-19-737-310 who carried out this  
project under my supervision in the IV semester  
for the academic year 2020-2021.**

Signature  
*external examiner*

Signature  
*internal examiner*

# **Inventory Management System**

**ANTHATI KARTHIK**  
**IT-B**  
**1602-19-737-310**  
**DBMS**

## **ABSTRACT:**

Inventory Management System is important to ensure quality control in businesses that handle transactions revolving around consumer goods. Without proper inventory control, a large retail store may run out of stock on an important item. A good Inventory Management System will alert the retailer when it is time to reorder. Inventory Management System is also an important means of automatically tracking large shipments. For example, if a business orders ten pairs of socks for retail resale, but only receives nine pairs, this will be obvious upon inspecting the contents of the package, and error is not likely. On the other hand, say a wholesaler orders 100,000 pairs of socks and 10,000 are missing. Manually counting each pair of socks is likely to result in error. An automated Inventory Management System helps to minimize the risk of error. In retail stores, an Inventory Management System also helps track theft of retail merchandise, providing valuable information about store profits and the need for theft-prevention systems. Automated Inventory Management System work by scanning a barcode either on the item. A barcode scanner is used to read the barcode, and the information encoded by the barcode is read by the machine. This information is then tracked by a central computer system. For example, a purchase order may contain a list of items to be pulled for packing and shipping. The Inventory Management System can serve a variety of functions in this case. It can help a worker locate the items on the order list in the warehouse, it can encode shipping information like tracking numbers and delivery addresses, and it can remove these purchased items from the inventory tally to keep an accurate count of in-stock items. All of this data works in tandem to provide businesses with real-time inventory tracking information. Inventory Management System make it simple to locate and analyze inventory information in real-time with a simple database search.

## REQUIREMENT ANALYSIS

### List of tables:

- **BRANDS**
- **CATEGORY**
- **ORDERS**
- **PRODUCT**
- **USERS**

### List of attributes with their domain types:

- **BRANDS:**
  - BRAND ID: bid number (20)
  - BRAND NAME: bname varchar (40)
  - BRAND ACTIVE: bactive varchar (40)
  - BRAND STATUS: bstatus varchar (40)
- **CATEGORY:**
  - CATEGORY ID: cid number(20)
  - CATEGORY NAME: cname varchar(40)
  - CATEGORY ACTIVE: cactive varchar (40)
  - CATEGORY STATUS: cstatus varchar (40)

- **ORDERS**

ORDER ID: oid number(20)

PRODUCT NAME: pname varchar(40)

ORDER DATE: odate date

NOOFPRODUCT: nop number(20)

TOTAL: total varchar(20)

- **PRODUCT**

PRODUCT ID: pid number(20)

PRODUCT NAME: pname varchar(40)

QUANTITY: quantity varchar(40)

RATE: rate varchar(40)

STATUS: status varchar(20)

- **USERS**

USER: usid number(20)

USERNAME: uname varchar(40)

PASSWORD: pwd varchar(40)

## **THROUGH THE PROJECT:**

This project helps to store data in a efficient way and it can be achieved through various sql commands and we can also store this for any future use and also we can save our data in a many different areas so we cannot lost all the data at once. The quality and product details are must in now a days because quality matters every where, these project stores details in database so that whenever it is necessary to know how and when a product can be used.

## **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:

```
try
{
    Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
    System.err.println("Unable to find and load driver");
    System.exit(1);
}
public void connectToDB()
{
    try
    {
        connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","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.



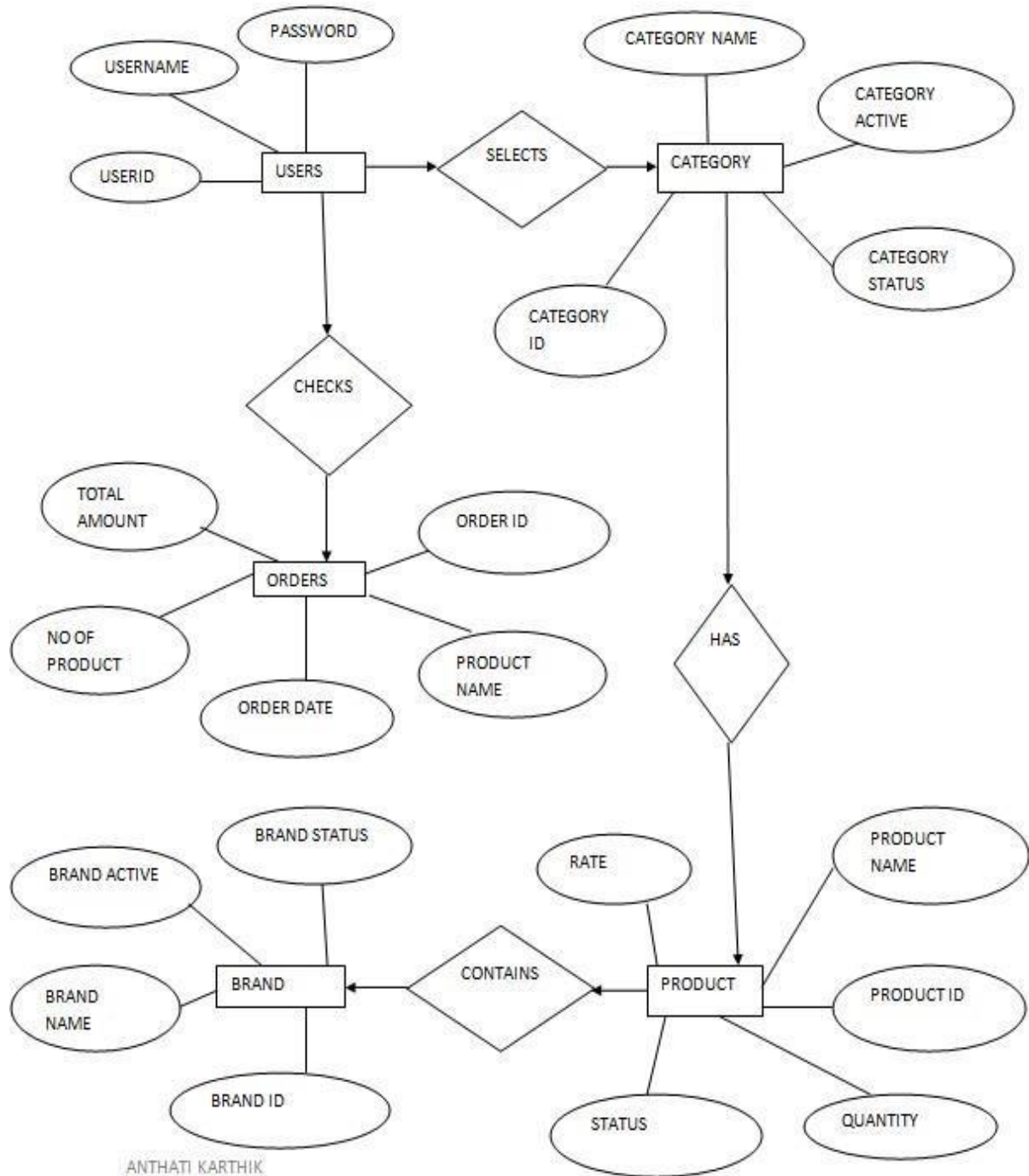
## Table Created in SQL for above mentioned purpose is as:

```
SQL> create table brand(bid number(5),bname varchar(20),bactive varchar(20),bstatus varchar(20));  
Table created.  
  
SQL> create table category(cid number(5),cname varchar(20),cactive varchar(20),cstatus varchar(20));  
Table created.  
  
SQL> create table orders(oid number(5),pname varchar(20),odate date,nop number(5),total number(5));  
Table created.  
  
SQL> create table product(pid number(5),pname varchar(20),quantity number(20),rate number(20),status varchar(20));  
Table created.  
  
SQL> create table users(usid number(20),uname varchar(20),pwd varchar(20));  
Table created.  
  
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
BRAND	TABLE	
CATEGORY	TABLE	
ORDERS	TABLE	
PRODUCT	TABLE	
USERS	TABLE	

## DESIGN:

### ER DIAGRAM:



### **Database Design:**

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
BRAND	TABLE	
CATEGORY	TABLE	
ORDERS	TABLE	
PRODUCT	TABLE	
USERS	TABLE	

```
SQL> desc brand;
```

Name	Null?	Type
BID		NUMBER(5)
BNAME		VARCHAR2(20)
BACTIVE		VARCHAR2(20)
BSTATUS		VARCHAR2(20)

```
SQL> desc category;
```

Name	Null?	Type
CID		NUMBER(5)
CNAME		VARCHAR2(20)
CACTIVE		VARCHAR2(20)
CSTATUS		VARCHAR2(20)

```
SQL> desc orders;
```

Name	Null?	Type
OID		NUMBER(5)
PNAME		VARCHAR2(20)
ODATE		DATE
NOP		NUMBER(5)
TOTAL		NUMBER(5)

```
SQL> desc product;
```

Name	Null?	Type
PID		NUMBER(5)
PNAME		VARCHAR2(20)
QUANTITY		NUMBER(20)
RATE		NUMBER(20)
STATUS		VARCHAR2(20)

```
SQL> desc users;
```

Name	Null?	Type
USID		NUMBER(20)
UNAME		VARCHAR2(20)
PWD		VARCHAR2(20)

## Implementation:

### Program:

#### Main.java:

```
package InventoryManagementSystem;
```

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

#### OracleCon.java:

```
package InventoryManagementSystem;
```

```
/*import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException; */
```

```
import java.sql.*;
```

```
class OracleCon
```

```
{  
    public static void main(String args[]){  
  
        try{  
            //step1 load the driver class  
            Class.forName("oracle.jdbc.driver.OracleDriver");    // or  
            Class.forName("oracle.jdbc.OracleDriver");  
  
            //step2 create the connection object  
            Connection  
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","vasavi  
            ");  
  
            //step3 create the statement object  
            Statement stmt=con.createStatement();  
  
            //step4 execute query  
            ResultSet rs=stmt.executeQuery("select * from sailors");  
            while(rs.next())  
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+"  
            "+rs.getString(3));  
  
            //step5 close the connection object  
            con.close();  
  
        }catch(Exception e){ System.out.println(e);}  
    }  
}
```

## ProjectUI.java:

```
package InventoryManagementSystem;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class ProjectUI extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JMenu mnubrand,mnuorders,mnucategory,mnuusers,mnuproduct;
    private JMenuBar mnuBar;
    private JMenuItem miInsert1,miUpdate1,midelete1,miView1;
    private JMenuItem miInsert2,miUpdate2,midelete2,miView2;
    private JMenuItem miInsert3,miUpdate3,midelete3,miView3;
    private JMenuItem miInsert4,miUpdate4,midelete4,miView4;
    private JMenuItem miInsert5,miUpdate5,midelete5,miView5;

    private JTextField txtField;

    static JPanel p1;

    void initialize()
    {
        //po=new JPanel();
        p1=new JPanel();
        mnubrand=new JMenu("Brand");
        mnucategory=new JMenu("Category");
        mnuorders=new JMenu("Orders");
        mnuproduct=new JMenu("Product");
        mnuusers=new JMenu("Users");
        mnuBar=new JMenuBar();

        miInsert1=new JMenuItem("Insert");
        miUpdate1=new JMenuItem("Update");
        midelete1=new JMenuItem("Delete");
        miView1=new JMenuItem("View");

        miInsert2=new JMenuItem("Insert");
        miUpdate2=new JMenuItem("Update");
        midelete2=new JMenuItem("Delete");
        miView2=new JMenuItem("View");

        miInsert3=new JMenuItem("Insert");
        miUpdate3=new JMenuItem("Update");
        midelete3=new JMenuItem("Delete");
        miView3=new JMenuItem("View");
    }
}
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```
miInsert4=new JMenuItem("Insert");  
miUpdate4=new JMenuItem("Update");  
midelete4=new JMenuItem("Delete");  
miView4=new JMenuItem("View");
```

```
miInsert5=new JMenuItem("Insert");  
miUpdate5=new JMenuItem("Update");  
midelete5=new JMenuItem("Delete");  
miView5=new JMenuItem("View");
```

```
txtField = new JTextField("Inventory Management System");  
txtField.setFont(new Font("Serif", Font.PLAIN, 25));  
txtField.setEditable(false);  
// po.setBackground(Color.MAGENTA);  
}  
void addComponentsToFrame()  
{  
    mnubrand.add(miInsert1);  
    mnubrand.add(miUpdate1);  
    mnubrand.add(midelete1);  
    mnubrand.add(miView1);  
  
    mnuorders.add(miInsert2);  
    mnuorders.add(miUpdate2);  
    mnuorders.add(midelete2);  
    mnuorders.add(miView2);  
  
    mnucategory.add(miInsert3);  
    mnucategory.add(miUpdate3);  
    mnucategory.add(midelete3);  
    mnucategory.add(miView3);  
  
    mnuusers.add(miInsert4);  
    mnuusers.add(miUpdate4);  
    mnuusers.add(midelete4);  
    mnuusers.add(miView4);  
  
    mnuproduct.add(miInsert5);  
    mnuproduct.add(miUpdate5);  
    mnuproduct.add(midelete5);  
    mnuproduct.add(miView5);  
  
    mnuBar.add(mnubrand);  
    mnuBar.add(mnuorders);
```

## DBMS ASSIGNMENT

### TITLE: INVENTORY MANAGEMENT SYSTEM

```
mnuBar.add(mnucategory);
mnuBar.add(mnuusers);
mnuBar.add(mnuproduct);

setJMenuBar(mnuBar);

p1.setLayout(new BorderLayout());
p1.add(txtField,BorderLayout.CENTER);

this.setLayout(new BorderLayout());
add(p1,BorderLayout.CENTER);

}
/**
 *
 */
void register()
{
    brand t1=new brand(p1,ProjectUI.this,miInsert1,miUpdate1,midelete1,miView1);
    t1.regis_inventory();
    orders t2=new orders(p1,ProjectUI.this,miInsert2,miUpdate2,midelete2,miView2);
    t2.regis_inventory();
    category t3=new category(p1,ProjectUI.this,miInsert3,miUpdate3,midelete3,miView3);
    t3.regis_inventory();
    Users t4=new Users(p1, ProjectUI.this, miInsert4, miUpdate4, midelete4, miView4);
    t4.regis_inventory();
    product t5=new product(p1,ProjectUI.this,miInsert5,miUpdate5,midelete5,miView5);
    t5.regis_inventory();

    addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent evt)
        {
            ImageIcon icon = new
            ImageIcon("src/t2s1.jpg");//C:\Users\DELL\Desktop\dbms\project\airlines Quality and Inf mgmt system\src
            int a=JOptionPane.showConfirmDialog(ProjectUI.this,"Are you sure?",
            "This will close", JOptionPane.OK_CANCEL_OPTION, JOptionPane.INFORMATION_MESSAGE, icon);
            if(a==JOptionPane.YES_OPTION)
                ProjectUI.this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
    });
}

public ProjectUI()
{
    initialize();
    addComponentsToFrame();
    register();
    pack();
    setTitle("Inventory Management System");
    setSize(600,500);
    setVisible(true);
    //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

## Brand.java:

```
package InventoryManagementSystem;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GridLayout;
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.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class brand {

    private JPanel p1;
    private JFrame frame;

    private JMenuItem miInsert1,miUpdate1,midelete1,miView1;
    private JLabel lblbid,lblbname,lblbactive,lblbstatus;
    private JTextField txtbid,txtbname,txtbactive,txtbstatus;
    private JButton btn;
    private JTextArea txtmsg;
    //private static Statement stmt;
    private Connection connection;
    private Statement statement;
    public brand(JPanel p1,JFrame frame,JMenuItem miInsert1,JMenuItem
miUpdate1,JMenuItem midelete1,JMenuItem miView1) {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
    }
```



DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```

        connectToDB();
        this.frame=frame;
        this.p1=p1;
        this.miInsert1=miInsert1;
        this.midelete1=midelete1;
        this.miUpdate1=miUpdate1;
        this.miView1=miView1;
        txtbid=new JTextField(20);
        txtbname=new JTextField(20);
        txtbactive=new JTextField(20);
        txtbstatus=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        lblbid=new JLabel("Brand Id:");
        lblbname=new JLabel("Brand Name:");
        lblbactive=new JLabel("Brand active:");
        lblbstatus=new JLabel("Brand status:");
        //queryHandler();
    }
    public void connectToDB()
    {
        try
        {
            connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","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);
        }
    }
    public void regis_inventory() {
        miInsert1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                p1.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                // p1.setBackground(Color.CYAN);
                JPanel p=new JPanel();
                txtbid=new JTextField(20);
                txtbname=new JTextField(20);
                txtbactive=new JTextField(20);
                txtbstatus=new JTextField(20);
                txtmsg=new JTextArea(8,50);
                btn=new JButton();
                //a grid of lbl and txtfield
                p.add(lblbid);
                p.add(txtbid);
                p.add(lblbname);
                p.add(txtbname);
                p.add(lblbactive);
                p.add(txtbactive);
                p.add(lblbstatus);
            }
        });
    }

```

```

        p.add(txtbstatus);
        p.setLayout(new GridLayout(4,2));

        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);//msg text area added to panel
        btn.setText("SUBMIT");
        p1.setLayout(new FlowLayout());
        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        //register listener
        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try {
                    Statement statement =
connection.createStatement();
                    String query= "INSERT INTO brand VALUES("
+ txtbid.getText() + ", " + "'" + txtbname.getText() + "'," + "'" +
txtbactive.getText() + "'," + txtbstatus.getText() + ")";
                    int i = statement.executeUpdate(query);
                    txtmsg.append("\nInserted " + i + "
rows successfully");

                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                    txtmsg.append(e1.getMessage());
                }
            }
        });

    }

});

miUpdate1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.LIGHT_GRAY);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtbid=new JTextField(20);
        txtbname=new JTextField(20);
        txtbactive=new JTextField(20);
        txtbstatus=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        txtmsg.setEditable(false);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT bid FROM
BRAND");

            while (rs.next())
            {
                idlist.add(rs.getString("bid"));
            }
        }
    }
});

```

```

        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lblbid);
        p.add(txtbid);
        p.add(lblbname);
        p.add(txtbname);
        p.add(lblbactive);
        p.add(txtbactive);
        p.add(lblbstatus);
        p.add(txtbstatus);
        p.setLayout(new GridLayout(4,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("MODIFY");
        p1.setLayout(new FlowLayout());

        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT
* FROM BRAND where bid ="+idlist.getSelectedItem());
                    rs.next();
                    txtbid.setText(rs.getString("bid"));
                    txtbname.setText(rs.getString("bname"));

                    txtbactive.setText(rs.getString("bactive"));

                    txtbstatus.setText(rs.getString("bstatus"));
                }
                catch (SQLException selectException)
                {

                    txtmsg.append(selectException.getMessage());
                }
            }
        });

        btn.addActionListener(new ActionListener() {

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

```

```

        try
        {
            Statement statement =
connection.createStatement();

            int i= statement.executeUpdate("UPDATE
BRAND "
            + "SET bname='" + txtbname.getText() +
            + "bactive =" + txtbactive.getText() +
            + "bstatus =" + txtbstatus.getText() + "
            + idlist.getSelectedItem());
            txtmsg.append("\nUpdated " + i + " rows
            //idlist.removeAll();
            //loadSailors();
        }
        catch (SQLException insertException)
        {
            txtmsg.append(insertException.getMessage());
        }
    }
});
});

midelete1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.BLUE);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtbid=new JTextField(20);
        txtbname=new JTextField(20);
        txtbactive=new JTextField(20);
        txtbstatus=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT bid FROM
BRAND");

            while (rs.next())
            {
                idlist.add(rs.getString("bid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lblbid);
        p.add(txtbid);
        p.add(lblbname);
        p.add(txtbname);
    }
});

```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```

        p.add(lblbactive);
        p.add(txtbactive);
        p.add(lblbstatus);
        p.add(txtbstatus);
        p.setLayout(new GridLayout(4,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("Delete");
        p1.setLayout(new FlowLayout());

        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT
* FROM BRAND where bid =" + idlist.getSelectedItemId());
                    rs.next();
                    txtbid.setText(rs.getString("bid"));
                    txtbname.setText(rs.getString("bname"));

                    txtbactive.setText(rs.getString("bactive"));

                    txtbstatus.setText(rs.getString("bstatus"));
                }
                catch (SQLException selectException)
                {

                    txtmsg.append(selectException.getMessage());
                }
            }
        });

        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try
                {
                    Statement statement =
connection.createStatement();

                    int i = statement.executeUpdate("delete
from BRAND where bid=" + idlist.getSelectedItemId());
                    txtmsg.append("\nDeleted " + i + " rows
successfully");

                    //idlist.removeAll();
                    //loadSailors();
                    txtbid.setText(null);
                    txtbname.setText(null);
                    txtbactive.setText(null);
                    txtbstatus.setText(null);
                    idlist.removeAll();

                }
            }
        });

```

```

        catch (SQLException insertException)
        {

txtmsg.append(insertException.getMessage());
        }

    }

});

});

miView1.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        p1.add(txtmsg);
        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        JTable j;
        DefaultTableModel S = new DefaultTableModel();
        j = new JTable(S);
        S.addColumn("BID");
        S.addColumn("BName");
        S.addColumn("BACTIVE");
        S.addColumn("BSTATUS");

        try {
            ResultSet rs=statement.executeQuery("select * from BRAND");
            while(rs.next())
                S.addRow(new
Object[]{rs.getString("bid"),rs.getString("bname"),rs.getString("bactive"),rs.getString("bstatus")});
        } catch (SQLException e1) {
            JOptionPane.showMessageDialog(frame,"Something Went Wrong");
        }
        JScrollPane sp = new JScrollPane(j);

        p1.add(sp, BorderLayout.NORTH);
        frame.invalidate();
        frame.validate();
        frame.repaint();

    }

});

```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

**Category.java:**

```
package InventoryManagementSystem;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
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.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

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.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class category {

    private JPanel p1;
    private JFrame frame;

    private JMenuItem miInsert3,miUpdate3,midelete3,miView3;
    private JLabel lblcid,lblcname,lbactive,lbstatus;
    private JTextField txtcid,txtcname,txtactive,txtstatus;
    private JButton btn;
    private JTextArea txtmsg;
    //private static Statement stmt;
    private Connection connection;
    private Statement statement;
    public category(JPanel p1,JFrame frame,JMenuItem miInsert3,JMenuItem miUpdate3,JMenuItem
midelete3,JMenuItem miView3) {
        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.p1=p1;
        this.miInsert3=miInsert3;
        this.midelete3=midelete3;
        this.miUpdate3=miUpdate3;
        this.miView3=miView3;
    }
}
```

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```
txtcid=new JTextField(20);
txtcname=new JTextField(20);
txtcactive=new JTextField(20);
txtcstatus=new JTextField(20);
txtmsg=new JTextArea(8,50);
lblcid=new JLabel("Category Id:");
lblcname=new JLabel("category Name:");
lblcactive=new JLabel("category active:");
lblcstatus=new JLabel("Category status:");
//queryHandler();
}
public void connectToDB()
{
    try
    {
        connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","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);
    }
}

public void regis_inventory() {
    miInsert3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            p1.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();
            //
            p1.setBackground(Color.CYAN);
            JPanel p=new JPanel();
            txtcid=new JTextField(20);
            txtcname=new JTextField(20);
            txtcactive=new JTextField(20);
            txtcstatus=new JTextField(20);
            txtmsg=new JTextArea(8,50);
            btn=new JButton();
            //a grid of lbl and txtfield
            p.add(lblcid);
            p.add(txtcid);
            p.add(lblcname);
            p.add(txtcname);
            p.add(lblcactive);
            p.add(txtcactive);
            p.add(lblcstatus);
            p.add(txtcstatus);
            p.setLayout(new GridLayout(4,2));

            p1.add(p);
            p1.add(btn);
            p1.add(txtmsg);//msg text area added to panel
            btn.setText("SUBMIT");
            p1.setLayout(new FlowLayout());
            frame.add(p1,BorderLayout.CENTER);
            frame.validate();

            //register listener
```



```

        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try {
                    Statement statement = connection.createStatement();
                    String query= "INSERT INTO category VALUES(" +
txtcid.getText() + ", " + "" + txtcname.getText() + ", " + "" + txtcactive.getText() + ", " + txtcstatus.getText() + ")";
                    int i = statement.executeUpdate(query);
                    txtmsg.append("\nInserted " + i + " rows successfully");
                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                    txtmsg.append(e1.getMessage());
                }
            }
        });

    });

    miUpdate3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            // p1.setBackground(Color.LIGHT_GRAY);
            p1.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();
            JPanel p=new JPanel();
            txtcid=new JTextField(20);
            txtcname=new JTextField(20);
            txtcactive=new JTextField(20);
            txtcstatus=new JTextField(20);
            txtmsg=new JTextArea(8,50);
            txtmsg.setEditable(false);
            btn=new JButton();
            List idlist =new List(10);
            try
            {
                ResultSet rs = statement.executeQuery("SELECT cid FROM category");
                while (rs.next())
                {
                    idlist.add(rs.getString("cid"));
                }
            }
            catch (SQLException e)
            {
                txtmsg.append(e.getMessage());
            }
            p1.add(idlist);
            p.add(lblcid);
            p.add(txtcid);
            p.add(lblcname);
            p.add(txtcname);
            p.add(lblcactive);
            p.add(txtcactive);
            p.add(lblcstatus);
            p.add(txtcstatus);
            p.setLayout(new GridLayout(4,2));
            p1.add(p);
            p1.add(btn);

```

# DBMS ASSIGNMENT

## TITLE: INVENTORY MANAGEMENT SYSTEM

```

p1.add(txtmsg);
btn.setText("MODIFY");
p1.setLayout(new FlowLayout());

frame.add(p1, BorderLayout.CENTER);
frame.validate();

idlist.addItemListener(new ItemListener()
{
    @Override
    public void itemStateChanged(ItemEvent arg0) {
        // TODO Auto-generated method stub

        try
        {
            ResultSet rs = statement.executeQuery("SELECT * FROM category
where cid =" + idlist.getSelectedItemId());

            rs.next();
            txtcid.setText(rs.getString("cid"));
            txtcname.setText(rs.getString("cname"));
            txtcactive.setText(rs.getString("cactive"));
            txtcstatus.setText(rs.getString("cstatus"));
        }
        catch (SQLException selectException)
        {
            txtmsg.append(selectException.getMessage());
        }
    }
});

btn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try
        {
            Statement statement = connection.createStatement();
            int i = statement.executeUpdate("UPDATE category "
            + "SET cname=" + txtcname.getText() + ", "
            + "cactive =" + txtcactive.getText() + ", "
            + "cstatus =" + txtcstatus.getText() + " WHERE cid = "
            + idlist.getSelectedItemId());
            txtmsg.append("\nUpdated " + i + " rows successfully");
            //idlist.removeAll();
            //loadSailors();
        }
        catch (SQLException insertException)
        {
            txtmsg.append(insertException.getMessage());
        }
    }
});

}

});

midelete3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.BLUE);

```

## DBMS ASSIGNMENT

### TITLE: INVENTORY MANAGEMENT SYSTEM

```
p1.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
JPanel p=new JPanel();
txtcid=new JTextField(20);
txtcname=new JTextField(20);
txtcactive=new JTextField(20);
txtcstatus=new JTextField(20);
txtmsg=new JTextArea(8,50);
btn=new JButton();
List idlist =new List(10);
try
{
    ResultSet rs = statement.executeQuery("SELECT cid FROM category");
    while (rs.next())
    {
        idlist.add(rs.getString("cid"));
    }
}
catch (SQLException e)
{
    txtmsg.append(e.getMessage());
}
p1.add(idlist);
p.add(lblcid);
p.add(txtcid);
p.add(lblcname);
p.add(txtcname);
p.add(lblcactive);
p.add(txtcactive);
p.add(lblcstatus);
p.add(txtcstatus);
p.setLayout(new GridLayout(4,2));
p1.add(p);
p1.add(btn);
p1.add(txtmsg);
btn.setText("Delete");
p1.setLayout(new FlowLayout());

frame.add(p1, BorderLayout.CENTER);
frame.validate();

idlist.addItemListener(new ItemListener()
{
    @Override
    public void itemStateChanged(ItemEvent arg0) {
        // TODO Auto-generated method stub

        try
        {
            ResultSet rs = statement.executeQuery("SELECT * FROM category
where cid =" +idlist.getSelectedItem());

            rs.next();
            txtcid.setText(rs.getString("cid"));
            txtcname.setText(rs.getString("cname"));
            txtcactive.setText(rs.getString("cactive"));
            txtcstatus.setText(rs.getString("cstatus"));
        }
        catch (SQLException selectException)
        {
            txtmsg.append(selectException.getMessage());
        }
    }
});
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```
        }
    }
});

btn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try
        {
            Statement statement = connection.createStatement();
            int i = statement.executeUpdate("delete from category where
cid=" + idlist.getSelectedItemId());

            txtmsg.append("\nDeleted " + i + " rows successfully");
            //idlist.removeAll();
            //loadSailors();
            txtcid.setText(null);
            txtcname.setText(null);
            txtcactive.setText(null);
            txtcstatus.setText(null);
            idlist.removeAll();
        }
        catch (SQLException insertException)
        {
            txtmsg.append(insertException.getMessage());
        }
    }
});

miView3.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        //p1.add(txtmsg);
        frame.add(p1, BorderLayout.NORTH);
        frame.validate();
        JTable j;
        DefaultTableModel S = new DefaultTableModel();
        j = new JTable(S);
        S.addColumn("CID");
        S.addColumn("CName");
        S.addColumn("CACTIVE");
        S.addColumn("CSTATUS");

        try {
            ResultSet rs=statement.executeQuery("select * from category");
            while(rs.next())
                S.addRow(new
Object[]{rs.getString("cid"),rs.getString("cname"),rs.getString("cactive"),rs.getString("cstatus")});
        } catch (SQLException e1) {
            JOptionPane.showMessageDialog(frame,"Something Went Wrong");
        }
    }
});
```

## DBMS ASSIGNMENT

### TITLE: INVENTORY MANAGEMENT SYSTEM

```
JScrollPane sp = new JScrollPane(j);

p1.add(sp,BorderLayout.NORTH);
frame.invalidate();
frame.validate();
frame.repaint();

    }

});

}

}
```

### Orders.java:

```
package InventoryManagementSystem;
```

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
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.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
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.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
```

```
public class orders {
```

```
    private JPanel p1;
    private JFrame frame;
```

```
    private JMenuItem miInsert2,miUpdate2,midelete2,miView2;
    private JLabel lbloid,lbldname,lblday,lbldbill,lbldnum;
    private JTextField txtoid,txtpname,txtday,txtbill,txtnum;
    private JButton btn;
    private JTextArea txtmsg;
    //private static Statement stmt;
    private Connection connection;
    private Statement statement;
    String s1;
    String s2;
    public orders(JPanel p1,JFrame frame,JMenuItem miInsert2,JMenuItem miUpdate2,JMenuItem
```

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```
midelete2,JMenuItem miView2) {
    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.p1=p1;
    this.miInsert2=miInsert2;
    this.midelete2=midelete2;
    this.miUpdate2=miUpdate2;
    this.miView2=miView2;
    txtoid=new JTextField(20);
    txtpname=new JTextField(20);
    txtday=new JTextField(20);
    txtbill=new JTextField(20);
    txtnum=new JTextField(20);
    txtmsg=new JTextArea(8,50);
    lbloid=new JLabel("Order Id:");
    lbpname=new JLabel("Product name:");
    lblday=new JLabel("Date:");
    lblnum = new JLabel("No. of products");
    lbllbill=new JLabel("Total Amount");
    //queryHandler();
}
public void connectToDB()
{
    try
    {
        connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","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);
    }
}

public void regis_inventory() {

    miInsert2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {

            p1.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();

            JPanel p=new JPanel();
            txtoid=new JTextField(20);
            txtpname=new JTextField(20);
            txtday=new JTextField(20);
            txtbill=new JTextField(20);
            txtnum=new JTextField(20);
```

## DBMS ASSIGNMENT

### TITLE: INVENTORY MANAGEMENT SYSTEM

```

        txtmsg=new JTextArea(8,50);
        btn=new JButton();
        //a grid of lbl and txtfield
        p.add(lbloid);
        p.add(txtoid);
        p.add(lblpname);
        p.add(txtpname);
        p.add(lblday);
        p.add(txtday);
        p.add(lblnum);
        p.add(txtnum);
        p.add(lblbill);
        p.add(txtbill);
        p.setLayout(new GridLayout(6,2));

        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);//msg text area added to panel
        btn.setText("SUBMIT");
        p1.setLayout(new FlowLayout());
        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        //register listener
        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try {
                    Statement statement = connection.createStatement();
                    String query= "INSERT INTO orders VALUES(" +
txtoid.getText() + "," + txtpname.getText() + "," + txtday.getText() + "," + txtnum.getText() + "," + txtbill.getText() + ")";
                    int i = statement.executeUpdate(query);
                    txtmsg.append("\nInserted " + i + " rows successfully");
                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    // e1.printStackTrace();
                    txtmsg.append(e1.getMessage());
                }
            }
        });

    }

});

miUpdate2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.LIGHT_GRAY);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtoid=new JTextField(20);
        txtpname=new JTextField(20);
        txtday=new JTextField(20);
        txtnum=new JTextField(20);
        txtbill=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        txtmsg.setEditable(false);
        btn=new JButton();

```

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```

        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT oid FROM orders");
            while (rs.next())
            {
                idlist.add(rs.getString("oid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lbloid);
        p.add(txtoid);
        p.add(lblpname);
        p.add(txtpname);
        p.add(lblday);
        p.add(txtday);
        p.add(lblnum);
        p.add(txtnum);
        p.add(lblbill);
        p.add(txtbill);
        p.setLayout(new GridLayout(6,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("MODIFY");
        p1.setLayout(new FlowLayout());

        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT * FROM orders
where oid =" +idlist.getSelectedItem());

                    rs.next();
                    txtoid.setText(rs.getString("oid"));
                    txtpname.setText(rs.getString("pname"));
                    txtday.setText(rs.getString("odate"));
                    txtnum.setText(rs.getString("nop"));
                    txtbill.setText(rs.getString("total"));
                }
                catch (SQLException selectException)
                {
                    txtmsg.append(selectException.getMessage());
                }
            }
        });

        btn.addActionListener(new ActionListener() {

            @Override

```



```

        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            try
            {
                Statement statement = connection.createStatement();
                int i = statement.executeUpdate("UPDATE orders "
                + "SET pname=" + txtpname.getText() + ", "
                + "odate =" + txtday.getText() + ", "
                + "nop =" + txtnum.getText() + ", "
                + "total =" + txtbill.getText() + " WHERE oid = "
                + idlist.getSelectedItem());
                txtmsg.append("\nUpdated " + i + " rows successfully");
                //idlist.removeAll();
                //loadSailors();
            }
            catch (SQLException insertException)
            {
                txtmsg.append(insertException.getMessage());
            }
        }
    });
}
});

```

```

midelete2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.BLUE);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtoid=new JTextField(20);
        txtpname=new JTextField(20);
        txtday=new JTextField(20);
        txtnum=new JTextField(20);
        txtbill=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT oid FROM orders");
            while (rs.next())
            {
                idlist.add(rs.getString("oid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lbloid);
        p.add(txtoid);
        p.add(lblpname);
        p.add(txtpname);
        p.add(lblday);
    }
});

```

## DBMS ASSIGNMENT

### TITLE: INVENTORY MANAGEMENT SYSTEM

```

        p.add(txtday);
        p.add(lblnum);
        p.add(txtnum);
        p.add(lblbill);
        p.setLayout(new GridLayout(6,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("Delete");
        p1.setLayout(new FlowLayout());

        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT * FROM orders
where oid =" + idlist.getSelectedItemId());

                    rs.next();
                    txtoid.setText(rs.getString("oid"));
                    txtpname.setText(rs.getString("pname"));
                    txtday.setText(rs.getString("odate"));
                    txtnum.setText(rs.getString("nop"));
                    txtbill.setText(rs.getString("total"));
                }
                catch (SQLException selectException)
                {
                    txtmsg.append(selectException.getMessage());
                }
            }
        });

        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try
                {
                    Statement statement = connection.createStatement();
                    int i = statement.executeUpdate("delete from orders where
oid=" + idlist.getSelectedItemId());

                    txtmsg.append("\nDeleted " + i + " rows successfully");
                    //idlist.removeAll();
                    //loadSailors();
                    txtoid.setText(null);
                    txtpname.setText(null);
                    txtday.setText(null);
                    txtnum.setText(null);
                    txtbill.setText(null);
                    idlist.removeAll();
                }
                catch (SQLException insertException)
                {

```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```
txtmsg.append(insertException.getMessage());
    }
}
});
}
});

miView2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        //p1.add(txtmsg);
        frame.add(p1, BorderLayout.NORTH);
        frame.validate();
        JTable j;
        DefaultTableModel S = new DefaultTableModel();
        j = new JTable(S);
        S.addColumn("OID");
        S.addColumn("PName");
        S.addColumn("ORDER DATE");
        S.addColumn("NO OF PRODUCT");
        S.addColumn("TOTAL AMOUNT");

        try {
            ResultSet rs=statement.executeQuery("select * from ORDERS");
            while(rs.next())
                S.addRow(new
Object[]{rs.getString("oid"),rs.getString("pname"),rs.getString("odate"),rs.getString("nop"),rs.getString("total")});
        } catch (SQLException e1) {
            JOptionPane.showMessageDialog(frame,"Something Went Wrong");
        }
        JScrollPane sp = new JScrollPane(j);

        p1.add(sp, BorderLayout.NORTH);
        frame.invalidate();
        frame.validate();
        frame.repaint();
    }
});
}
}
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

**Product.java:**

```
package InventoryManagementSystem;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
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.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
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.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class product {

    private JPanel p1;
    private JFrame frame;

    private JMenuItem miInsert5,miUpdate5,midelete5,miView5;
    private JLabel lblpid,lblpname,lblquantity,lblstatus,blbrate;
    private JTextField txtpid,txtpname,txtquantity,txtstatus,txtrate;
    private JButton btn;
    private JTextArea txtmsg;
    //private static Statement stmt;
    private Connection connection;
    private Statement statement;
    String s1;
    String s2;

    public product(JPanel p1,JFrame frame,JMenuItem miInsert5,JMenuItem miUpdate5,JMenuItem
midelete5,JMenuItem miView5) {
        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.p1=p1;
        this.miInsert5=miInsert5;
```

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```
this.midelete5=midelete5;
this.miUpdate5=miUpdate5;
this.miView5=miView5;
txtpid=new JTextField(20);
txtpname=new JTextField(20);
txtquantity=new JTextField(20);
txtstatus=new JTextField(20);
txtrate=new JTextField(20);
txtmsg=new JTextArea(8,50);
lblpid=new JLabel("Product Id:");
lblpname=new JLabel("Product name:");
lblquantity=new JLabel("Quantity:");
lblrate = new JLabel("Rate:");
lblstatus=new JLabel("Status:");
//queryHandler();
}
public void connectToDB()
{
    try
    {
        connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","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);
    }
}

public void regis_inventory() {

    miInsert5.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {

            p1.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();

            JPanel p=new JPanel();
            txtpid=new JTextField(20);
            txtpname=new JTextField(20);
            txtquantity=new JTextField(20);
            txtstatus=new JTextField(20);
            txtrate=new JTextField(20);
            txtmsg=new JTextArea(8,50);
            btn=new JButton();
            //a grid of lbl and txtfield
            p.add(lblpid);
            p.add(txtpid);
            p.add(lblpname);
            p.add(txtpname);
            p.add(lblquantity);
            p.add(txtquantity);
            p.add(lblrate);
            p.add(txtrate);
            p.add(lblstatus);
            p.add(txtstatus);
            p.setLayout(new GridLayout(6,2));
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```
p1.add(p);
p1.add(btn);
p1.add(txtmsg);//msg text area added to panel
btn.setText("SUBMIT");
p1.setLayout(new FlowLayout());
frame.add(p1, BorderLayout.CENTER);
frame.validate();

//register listener
btn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {
            Statement statement = connection.createStatement();
            String query= "INSERT INTO product VALUES(" +
txtpid.getText() + "," + txtpname.getText() + "," + txtquantity.getText() + "," + txtrate.getText() + "," +
txtstatus.getText() + ")";

            int i = statement.executeUpdate(query);
            txtmsg.append("\nInserted " + i + " rows successfully");
        } catch (SQLException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
            txtmsg.append(e1.getMessage());
        }
    }
});

});

}

miUpdate5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.LIGHT_GRAY);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtpid=new JTextField(20);
        txtpname=new JTextField(20);
        txtquantity=new JTextField(20);
        txtrate=new JTextField(20);
        txtstatus=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        txtmsg.setEditable(false);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT pid FROM product");
            while (rs.next())
            {
                idlist.add(rs.getString("pid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
    }
});
```

# DBMS ASSIGNMENT

## TITLE: INVENTORY MANAGEMENT SYSTEM

```

        p1.add(idlist);
        p.add(lblpid);
        p.add(txtpid);
        p.add(lblpname);
        p.add(txtpname);
        p.add(lblquantity);
        p.add(txtquantity);
        p.add(lblrate);
        p.add(txtrate);
        p.add(lblstatus);
        p.add(txtstatus);
        p.setLayout(new GridLayout(6,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("MODIFY");
        p1.setLayout(new FlowLayout());

        frame.add(p1, BorderLayout.CENTER);
        frame.validate();

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT * FROM product
where pid ="+idlist.getSelectedItemId());

                    rs.next();
                    txtpid.setText(rs.getString("pid"));
                    txtpname.setText(rs.getString("pname"));
                    txtquantity.setText(rs.getString("quantity"));
                    txtrate.setText(rs.getString("rate"));
                    txtstatus.setText(rs.getString("status"));
                }
                catch (SQLException selectException)
                {
                    txtmsg.append(selectException.getMessage());
                }
            }
        });

        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try
                {
                    Statement statement = connection.createStatement();
                    int i = statement.executeUpdate("UPDATE product "
                    + "SET pname=" + txtpname.getText() + ", "
                    + "quantity=" + txtquantity.getText() + ", "
                    + "rate =" + txtrate.getText() + ", "
                    + "status =" + txtstatus.getText() + " WHERE pid = "
                    + idlist.getSelectedItemId());
                    txtmsg.append("\nUpdated " + i + " rows successfully");
                    //idlist.removeAll();
                }
            }
        });
    
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```
                                //loadSailors();
                                }
                                catch (SQLException insertException)
                                {
                                    txtmsg.append(insertException.getMessage());
                                }
                                }
                                });
                                }
                                });
                                };
```

```
midelete5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.BLUE);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtpid=new JTextField(20);
        txtpname=new JTextField(20);
        txtquantity=new JTextField(20);
        txtrate=new JTextField(20);
        txtstatus=new JTextField(20);
        txtmsg=new JTextArea(8,50);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT pid FROM product");
            while (rs.next())
            {
                idlist.add(rs.getString("pid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lblpid);
        p.add(txtpid);
        p.add(lblpname);
        p.add(txtpname);
        p.add(lblquantity);
        p.add(txtquantity);
        p.add(lblrate);
        p.add(txtrate);
        p.add(lblstatus);
        p.add(txtstatus);
        p.setLayout(new GridLayout(6,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("Delete");
        p1.setLayout(new FlowLayout());

        frame.add(p1, BorderLayout.CENTER);
        frame.validate();
    }
});
```



# DBMS ASSIGNMENT

## TILTLE: INVENTORY MANAGEMENT SYSTEM

```

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT * FROM product
where pid =" + idlist.getSelectedItemId());

                    rs.next();
                    txtpid.setText(rs.getString("pid"));
                    txtpname.setText(rs.getString("pname"));
                    txtquantity.setText(rs.getString("quantity"));
                    txtrate.setText(rs.getString("rate"));
                    txtstatus.setText(rs.getString("status"));
                }
                catch (SQLException selectException)
                {
                    txtmsg.append(selectException.getMessage());
                }
            }
        });

        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try
                {
                    Statement statement = connection.createStatement();
                    int i = statement.executeUpdate("delete from product where
pid=" + idlist.getSelectedItemId());

                    txtmsg.append("\nDeleted " + i + " rows successfully");
                    //idlist.removeAll();
                    //loadSailors();
                    txtpid.setText(null);
                    txtpname.setText(null);
                    txtquantity.setText(null);
                    txtrate.setText(null);
                    txtstatus.setText(null);
                    idlist.removeAll();
                }
                catch (SQLException insertException)
                {
                    txtmsg.append(insertException.getMessage());
                }
            }
        });

        miView5.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                p1.removeAll();
                frame.invalidate();
            }
        });
    }
}

```

## DBMS ASSIGNMENT

### TITLE: INVENTORY MANAGEMENT SYSTEM

```
        frame.validate();
        frame.repaint();
        //p1.add(txtmsg);
        frame.add(p1,BorderLayout.NORTH);
        frame.validate();
        JTable j;
        DefaultTableModel S = new DefaultTableModel();
        j = new JTable(S);
        S.addColumn("PID");
        S.addColumn("PName");
        S.addColumn("QUANTITY");
        S.addColumn("RATE");
        S.addColumn("STATUS");

        try {
            ResultSet rs=statement.executeQuery("select * from product");
            while(rs.next())
                S.addRow(new
Object[] {rs.getString("pid"),rs.getString("pname"),rs.getString("quantity"),rs.getString("rate"),rs.getString("status")});
        } catch (SQLException e1) {
            JOptionPane.showMessageDialog(frame,"Something Went Wrong");
        }
        JScrollPane sp = new JScrollPane(j);

        p1.add(sp,BorderLayout.NORTH);
        frame.invalidate();
        frame.validate();
        frame.repaint();
    }
}
});
}
```

### Users.java:

```
package InventoryManagementSystem;
```

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
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.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
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;
```

ANTHATHI KARTHIK  
1602-19-737-310

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class Users {

    private JPanel p1;
    private JFrame frame;

    private JMenuItem miInsert4,miUpdate4,midelete4,miView4;
    private JLabel lblusid,lbluname,blpwd;
    private JTextField txtusid,txtuname,txtpwd;
    private JButton btn;
    private JTextArea txtmsg;
    //private static Statement stmt;
    private Connection connection;
    private Statement statement;
    public Users(JPanel p1,JFrame frame,JMenuItem miInsert4,JMenuItem miUpdate4,JMenuItem
midelete4,JMenuItem miView4) {
        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.p1=p1;
        this.miInsert4=miInsert4;
        this.midelete4=midelete4;
        this.miUpdate4=miUpdate4;
        this.miView4=miView4;
        txtusid=new JTextField(20);
        txtuname=new JTextField(20);
        txtpwd=new JTextField(20);

        txtmsg=new JTextArea(8,50);
        lblusid=new JLabel("User Id:");
        lbluname=new JLabel("UserName:");
        blpwd=new JLabel("Password:");

        //queryHandler();
    }
    public void connectToDB()
    {
        try
        {
            connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","karthik","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);
        }
    }
}
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```

}
    public void regis_inventory() {
        miInsert4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                p1.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                //
                p1.setBackground(Color.CYAN);
                JPanel p=new JPanel();
                txtusid=new JTextField(20);
                txtuname=new JTextField(20);
                txtpwd=new JTextField(20);

                txtmsg=new JTextArea(8,50);
                btn=new JButton();
                //a grid of lbl and txtfield
                p.add(lblusid);
                p.add(txtusid);
                p.add(lbluname);
                p.add(txtuname);
                p.add(lblpwd);
                p.add(txtpwd);

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

                p1.add(p);
                p1.add(btn);
                p1.add(txtmsg);//msg text area added to panel
                btn.setText("SUBMIT");
                p1.setLayout(new FlowLayout());
                frame.add(p1, BorderLayout.CENTER);
                frame.validate();

                //register listener
                btn.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
                        try {
                            Statement statement = connection.createStatement();
                            String query= "INSERT INTO users VALUES(" +
                                txtusid.getText() + ", " + "" + txtuname.getText() + ", " + "" + txtpwd.getText() + ")";
                            int i = statement.executeUpdate(query);
                            txtmsg.append("\nInserted " + i + " rows successfully");
                        } catch (SQLException e1) {
                            // TODO Auto-generated catch block
                            e1.printStackTrace();
                            txtmsg.append(e1.getMessage());
                        }
                    }
                });
            }
        });

        miUpdate4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                //
                p1.setBackground(Color.LIGHT_GRAY);
                p1.removeAll();
                frame.invalidate();

```

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtusid=new JTextField(20);
        txtuname=new JTextField(20);
        txtpwd=new JTextField(20);

        txtmsg=new JTextArea(8,50);
        txtmsg.setEditable(false);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT usid FROM users");
            while (rs.next())
            {
                idlist.add(rs.getString("usid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lblusid);
        p.add(txtusid);
        p.add(lbluname);
        p.add(txtuname);
        p.add(lblpwd);
        p.add(txtpwd);

        p.setLayout(new GridLayout(4,2));
        p1.add(p);
        p1.add(btn);
        p1.add(txtmsg);
        btn.setText("MODIFY");
        p1.setLayout(new FlowLayout());

        frame.add(p1,BorderLayout.CENTER);
        frame.validate();

        idlist.addItemListener(new ItemListener()
        {
            @Override
            public void itemStateChanged(ItemEvent arg0) {
                // TODO Auto-generated method stub

                try
                {
                    ResultSet rs = statement.executeQuery("SELECT * FROM users
where usid =" +idlist.getSelectedItem());

                    rs.next();
                    txtusid.setText(rs.getString("usid"));
                    txtuname.setText(rs.getString("uname"));
                    txtpwd.setText(rs.getString("pwd"));

                }
                catch (SQLException selectException)
                {
                    txtmsg.append(selectException.getMessage());
                }
            }
        })
    }
```

```
});

btn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try
        {
            Statement statement = connection.createStatement();
            int i = statement.executeUpdate("UPDATE users "
            + "SET uname=" + txtuname.getText() + ", "
            + "pwd=" + txtpwd.getText() + " WHERE usid = "
            + idlist.getSelectedItem());
            txtmsg.append("\nUpdated " + i + " rows successfully");
            //idlist.removeAll();
            //loadSailors();
        }
        catch (SQLException insertException)
        {
            txtmsg.append(insertException.getMessage());
        }
    }

});

});

midelete4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // p1.setBackground(Color.BLUE);
        p1.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        JPanel p=new JPanel();
        txtusid=new JTextField(20);
        txtuname=new JTextField(20);
        txtpwd=new JTextField(20);

        txtmsg=new JTextArea(8,50);
        btn=new JButton();
        List idlist =new List(10);
        try
        {
            ResultSet rs = statement.executeQuery("SELECT usid FROM users");
            while (rs.next())
            {
                idlist.add(rs.getString("usid"));
            }
        }
        catch (SQLException e)
        {
            txtmsg.append(e.getMessage());
        }
        p1.add(idlist);
        p.add(lblusid);
        p.add(txtusid);
        p.add(lbluname);
        p.add(txtuname);
    }
});
```

## DBMS ASSIGNMENT

### TILTLE: INVENTORY MANAGEMENT SYSTEM

```
p.add(lblpwd);
p.add(txtpwd);

p.setLayout(new GridLayout(4,2));
p1.add(p);
p1.add(btn);
p1.add(txtmsg);
btn.setText("Delete");
p1.setLayout(new FlowLayout());

frame.add(p1, BorderLayout.CENTER);
frame.validate();

idlist.addItemListener(new ItemListener()
{
    @Override
    public void itemStateChanged(ItemEvent arg0) {
        // TODO Auto-generated method stub

        try
        {
            ResultSet rs = statement.executeQuery("SELECT * FROM users
where usid =" + idlist.getSelectedItemId());

            rs.next();
            txtusid.setText(rs.getString("usid"));
            txtuname.setText(rs.getString("uname"));
            txtpwd.setText(rs.getString("pwd"));

        }
        catch (SQLException selectException)
        {
            txtmsg.append(selectException.getMessage());
        }
    }
});

btn.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try
        {
            Statement statement = connection.createStatement();
            int i = statement.executeUpdate("delete from users where
usid=" + idlist.getSelectedItemId());

            txtmsg.append("\nDeleted " + i + " rows successfully");
            //idlist.removeAll();
            //loadSailors();
            txtusid.setText(null);
            txtuname.setText(null);
            txtpwd.setText(null);

            idlist.removeAll();

        }
        catch (SQLException insertException)
        {
            txtmsg.append(insertException.getMessage());
        }
    }
});
```

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

```
        });  
    }  
});  
  
miView4.addActionListener(new ActionListener() {  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        // TODO Auto-generated method stub  
        p1.removeAll();  
        frame.invalidate();  
        frame.validate();  
        frame.repaint();  
        //p1.add(txtmsg);  
        frame.add(p1, BorderLayout.NORTH);  
        frame.validate();  
        JTable j;  
        DefaultTableModel S = new DefaultTableModel();  
        j = new JTable(S);  
        S.addColumn("USID");  
        S.addColumn("USERNAME");  
        S.addColumn("PASSWORD");  
  
        try {  
            ResultSet rs=statement.executeQuery("select * from users");  
            while(rs.next())  
                S.addRow(new  
Object[]{rs.getString("usid"),rs.getString("uname"),rs.getString("pwd")});  
        } catch (SQLException e1) {  
            JOptionPane.showMessageDialog(frame,"Something Went Wrong");  
        }  
        JScrollPane sp = new JScrollPane(j);  
  
        p1.add(sp, BorderLayout.NORTH);  
        frame.invalidate();  
        frame.validate();  
        frame.repaint();  
    }  
});  
  
}
```



## GitHub links and folder structure

<https://github.com/ANTHATIKARTHIK/DBMS-InventoryManagementSystem>

This PC > Windows (C:) > Users > 91918 > eclipse-workspace > testoneDBMSprjt

Name	Date modified	Type	Size
.settings	06-06-2021 18:16	File folder	
bin	06-06-2021 19:18	File folder	
src	06-06-2021 18:16	File folder	
.classpath	06-06-2021 18:30	CLASSPATH File	1 KB
.project	06-06-2021 18:16	PROJECT File	1 KB

This PC > Windows (C:) > Users > 91918 > eclipse-workspace > testoneDBMSprjt > src > testoneDBMSprjt

Name	Date modified	Type	Size
brand	10-06-2021 14:45	JAVA File	11 KB
category	10-06-2021 14:49	JAVA File	11 KB
Main	06-06-2021 18:25	JAVA File	1 KB
OracleCon	06-06-2021 18:25	JAVA File	1 KB
orders	10-06-2021 14:52	JAVA File	11 KB
product	10-06-2021 14:59	JAVA File	11 KB
ProjectUI	10-06-2021 15:02	JAVA File	5 KB
Users	10-06-2021 15:02	JAVA File	10 KB
















This PC > Windows (C:) > Users > 91918 > eclipse-workspace > testoneDBMSprjt > bin > testoneDBMSprjt

brand\$1\$1	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 1.94 KB
brand\$1	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 2.33 KB
brand\$2\$1	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 2.01 KB
brand\$2\$2	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 2.04 KB
brand\$2	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 3.24 KB
brand\$3\$1	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 2.01 KB
brand\$3\$2	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 2.01 KB
brand\$3	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 3.19 KB
brand\$4	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 2.33 KB
brand	Type: CLASS File	Date modified: 10-06-2021 14:45 Size: 3.40 KB
category\$1\$1	Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 1.97 KB
category\$1	Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 2.36 KB
category\$2\$1	Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 2.04 KB
category\$2\$2	Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 2.07 KB
category\$2	Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 2.33 KB
















# DBMS ASSIGNMENT













## TITLE: INVENTORY MANAGEMENT SYSTEM

This PC > Windows (C:) > Users > 91918 > eclipse-workspace > testoneDBMSprjt > bin > testoneDBMSprjt

	category\$4 Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 2.35 KB
	category Type: CLASS File	Date modified: 10-06-2021 14:49 Size: 3.45 KB
	Main Type: CLASS File	Date modified: 06-06-2021 19:18 Size: 433 bytes
	OradeCon Type: CLASS File	Date modified: 06-06-2021 19:18 Size: 1.77 KB
	orders\$1\$1 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 1.97 KB
	orders\$1 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 2.41 KB
	orders\$2\$1 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 2.06 KB
	orders\$2\$2 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 2.10 KB
	orders\$2 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 3.32 KB
	orders\$3\$1 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 2.06 KB
	orders\$3\$2 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 2.05 KB
	orders\$3 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 3.28 KB
	orders\$4 Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 2.38 KB
	orders Type: CLASS File	Date modified: 10-06-2021 14:52 Size: 3.54 KB
	product\$1\$1 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 1.98 KB

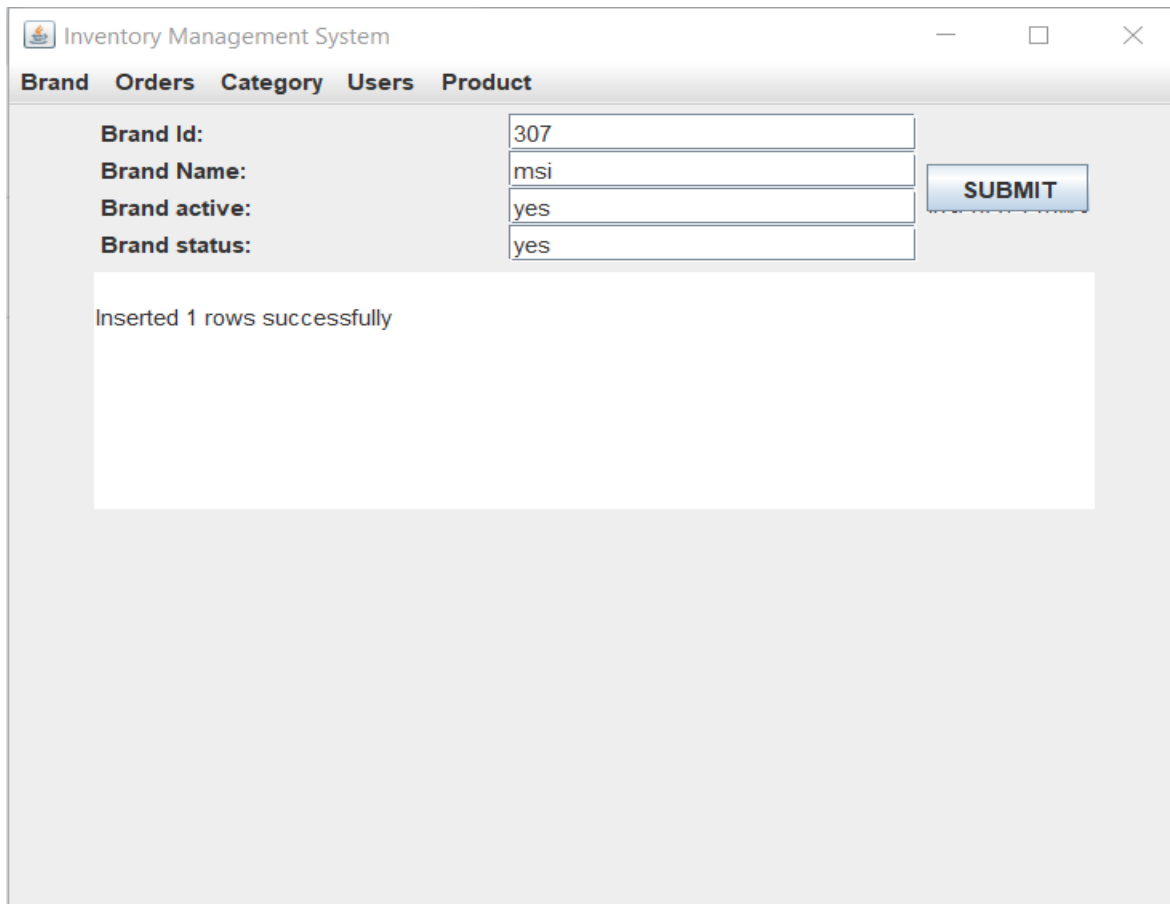
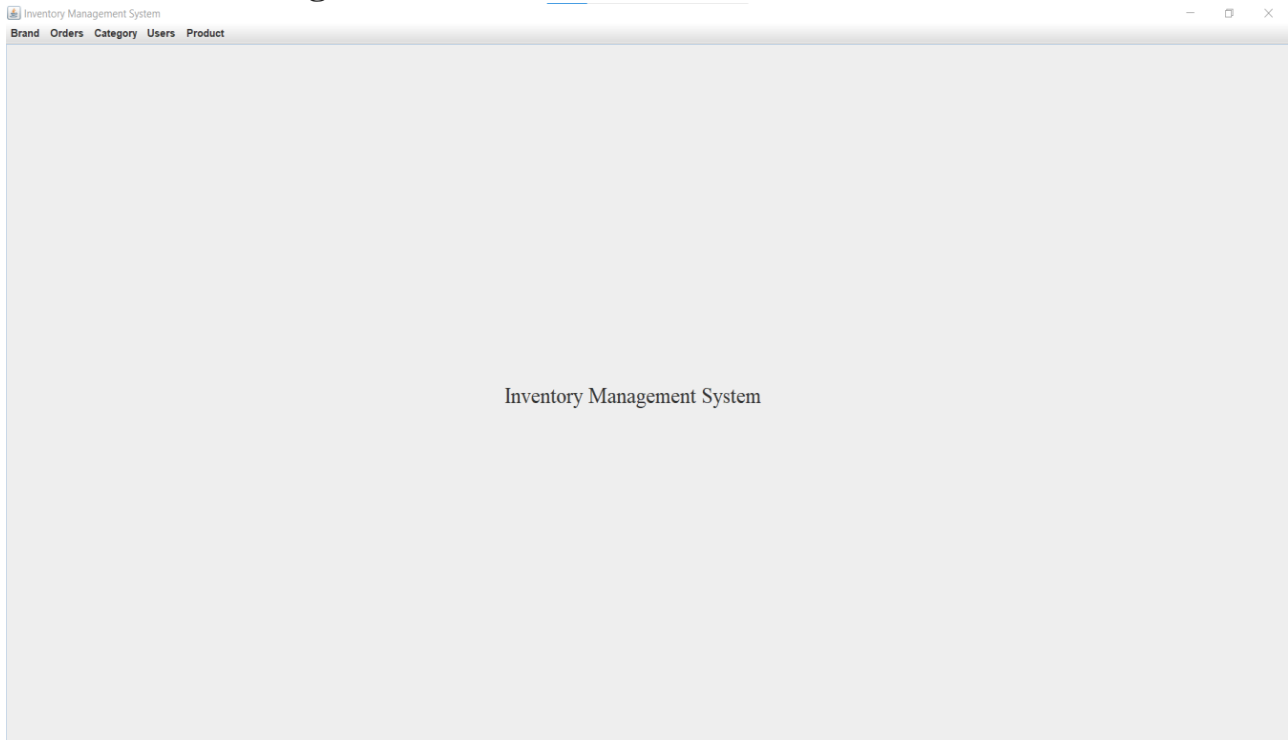
This PC > Windows (C:) > Users > 91918 > eclipse-workspace > testoneDBMSprjt > bin > testoneDBMSprjt

	product\$1\$1 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 1.98 KB
	product\$1 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 2.44 KB
	product\$2\$1 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 2.08 KB
	product\$2\$2 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 2.12 KB
	product\$2 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 3.35 KB
	product\$3\$1 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 2.08 KB
	product\$3\$2 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 2.07 KB
	product\$3 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 3.30 KB
	product\$4 Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 2.39 KB
	product Type: CLASS File	Date modified: 10-06-2021 14:59 Size: 3.56 KB
	ProjectUI\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.09 KB
	ProjectUI Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 4.34 KB
	Users\$1\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.89 KB
	Users\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 2.23 KB
	Users\$2\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.95 KB

	ProjectUI\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.09 KB
	ProjectUI Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 4.34 KB
	Users\$1\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.89 KB
	Users\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 2.23 KB
	Users\$2\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.95 KB
	Users\$2\$2 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.96 KB
	Users\$2 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 3.14 KB
	Users\$3\$1 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.95 KB
	Users\$3\$2 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 1.97 KB
	Users\$3 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 3.09 KB
	Users\$4 Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 2.30 KB
	Users Type: CLASS File	Date modified: 10-06-2021 15:02 Size: 3.27 KB

## Testing:

### Java GUI Testing:



DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

Inventory Management System

Brand Orders Category Users Product

1	Brand Id:	3
2	Brand Name:	nike
153	Brand active:	yes
3	Brand status:	nopeeeeeeee
310		

Delete

Inventory Management System

Brand Orders Category Users Product

1	Brand Id:	478
2	Brand Name:	BMW
153	Brand active:	yes
3	Brand status:	no
231		
310		
23		
64546		
307		

MODIFY

Updated 1 rows successfully

DBMS ASSIGNMENT  
TILTLE: INVENTORY MANAGEMENT SYSTEM

Inventory Management System			
Brand	Orders	Category	Users
BID	BName	BACTIVE	BSTATUS
1	adiddas	yes	no
2	titan	yes	yes
153	realme	yes	no
3	nike	yes	yes
231	bikes	yes	YES
310	Apple iOS	yes	no
23	ferari	yes	yes
153	brand	yes	yes
307	msi	yes	yes
5	PANUGANTI	YES	YES

Inventory Management System			
Brand	Orders	Category	Users
BID	BName	BACTIVE	BSTATUS
1	adiddas	yes	no
2	titan	yes	yes
153	realme	yes	no
3	nike	yes	yes
231	bikes	yes	YES
310	Apple iOS	yes	no
23	ferari	yes	yes
153	bra		yes
307	ms		yes
5	PA		YES

This will close

Are you sure?

OK Cancel

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

 Select Run SQL Command Line

```
SQL> select * from brand;
```

BID	BNAME	BACTIVE	BSTATUS
1	adiddas	yes	no
2	titan	yes	yes
153	realme	yes	no
3	nike	yes	yes
231	bikes	yes	no
310	Apple iOS	yes	no
23	ferari	yes	yes
153	brand	yes	yes
307	msi	yes	yes
3654	fesdddsd	esgf	efs

```
10 rows selected.  
SQL>
```

## Results:

I successfully completed this Assignment "Inventory Management system".

## Discussion and Future work

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create a android app by which I can control my project on my hand and connect to it.

## References:

- <https://docs.oracle.com/javase/7/docs/index.html>
- <https://www.javatpoint.com/dbms-tutorial>
- [http://www.sqlines.com/articles/java/sql\\_server\\_jdbc\\_connection](http://www.sqlines.com/articles/java/sql_server_jdbc_connection)