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

Maram Tanmayee <1602-19-737-119>

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 "***ClotheWith***" project work of **Maram Tanmayee** bearing Roll.no:**1602-19-737-119** who carried out this project under my supervision in the IV semester for the academic year 2019-2020.

Signature

external examiner

Signature

internal examiner

ABSTRACT

The ClotheWith is a management system that provides a database where all the activities of a clothing are stored. It is very important to keep track of all the people involved and the duties performed by them in the process of manufacturing to maintain the process efficiently. There are about 7 tables in this database where all the information starting from production of clothes to reaching out to the customer to sell them is stored. This database describes how the clothes are processed to provide a good product to a customer.

REQUIREMENTS ANALYSIS

List Of Tables:

1. Designer
2. Suppliers
3. Items
4. Store
5. Manager
6. Employees
7. Customer

List Of Attributes With Their Domain Types:

Designer:

Name of the Designer: D_Name varchar2(20)
ID of the Designer: D_id number(5)
Mobile number of the Designer: D_Phone number(10)

Supplier:

Name of the supplier: S_Name varchar2(20)
Id of the Supplier: S_Id number(5)
Shipment ID: Shipment_Id number(5)

Item:

Item code: Item_No number(5)
Cost of the item: Prize number(5)
Description of the item: Description varchar2(20)

Store:

Item code: Item_No number(5)
Quantity: Qty number(5)

Manager:

Name of the manager: M_Name varchar2(20)
Id of the manager: M_Id number(5)
Mobile number of the manager: M_Phone number(10)
Address of the manager: M_Address Varchar2(30)

Employees:

Name of the employee: E_Name varchar2(20)
Id of the employee: E_Id number(5)
Mobile number of the employee: E_Phone number(10)

Customer:

Membership ID: Membersip_Id number(5)
Discount percentage : Discount number(3)

THROUGH THIS PROJECT: It develops an online record application for clothing store which is management friendly. This project helps to record the data of people involved in managing a store and types of clothes. The data can be stored for future purpose in case the user forgets the shipment details. User can easily update his/her information/details very easily. As there is track of employees, items and customers, this makes this project more efficient.

AIM:

To create a **Java GUI based Store Management System** which consists complete information of the people involved in managing a clothing store. This information needs to be updated from time to time which is done 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.

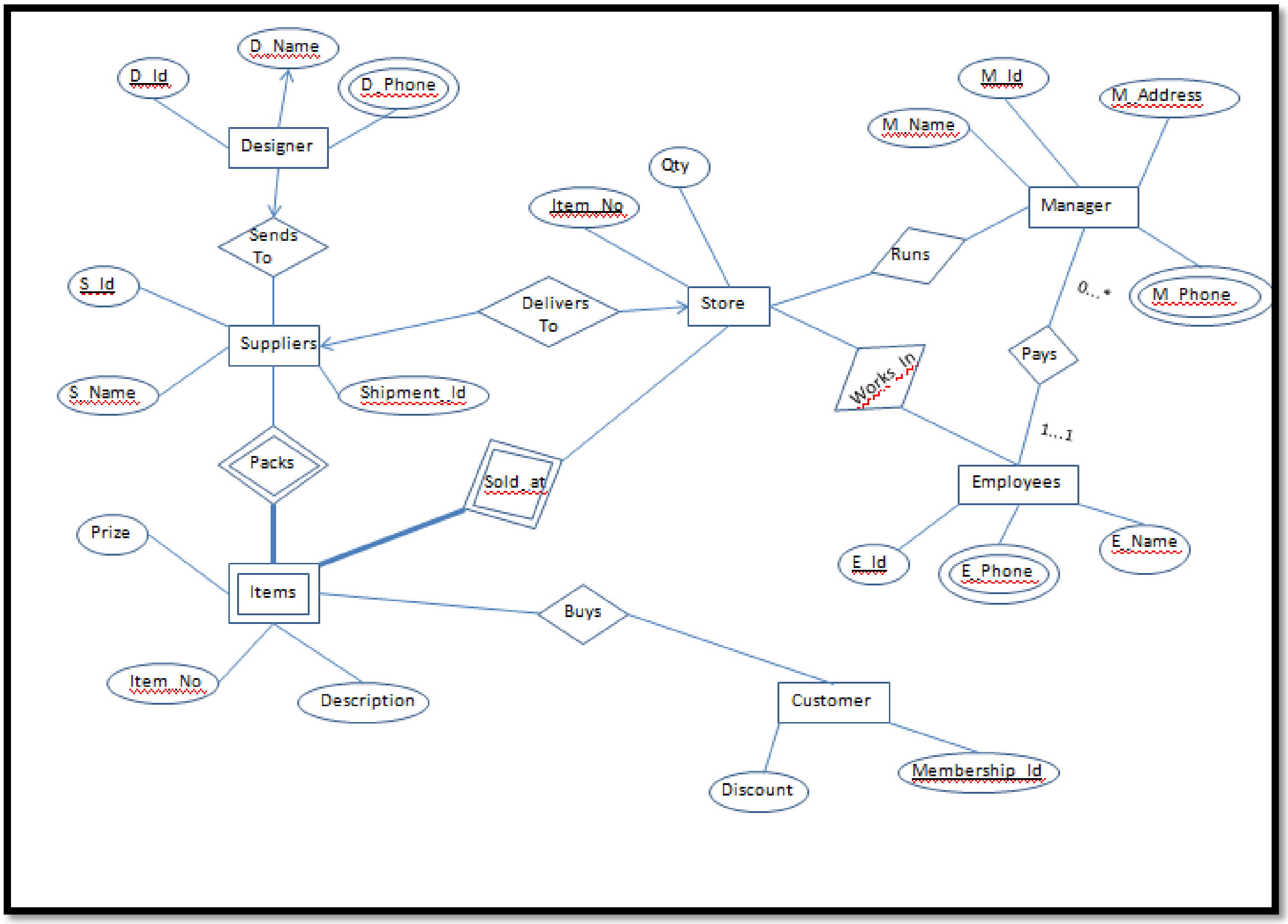
The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()
    {
             try {


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


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


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

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:

1. An employee works under a manager. A manager is alloted to a set of employees which means an employee must have 1 manger under which he works.
   (One to Many participation)
2. A supplier can provide items to store and a store can have many suppliers.
   (Many to Many participation)
3. A supplier get designs from many designers and a designer can supply to many suppliers.
   (Many to Many participation)

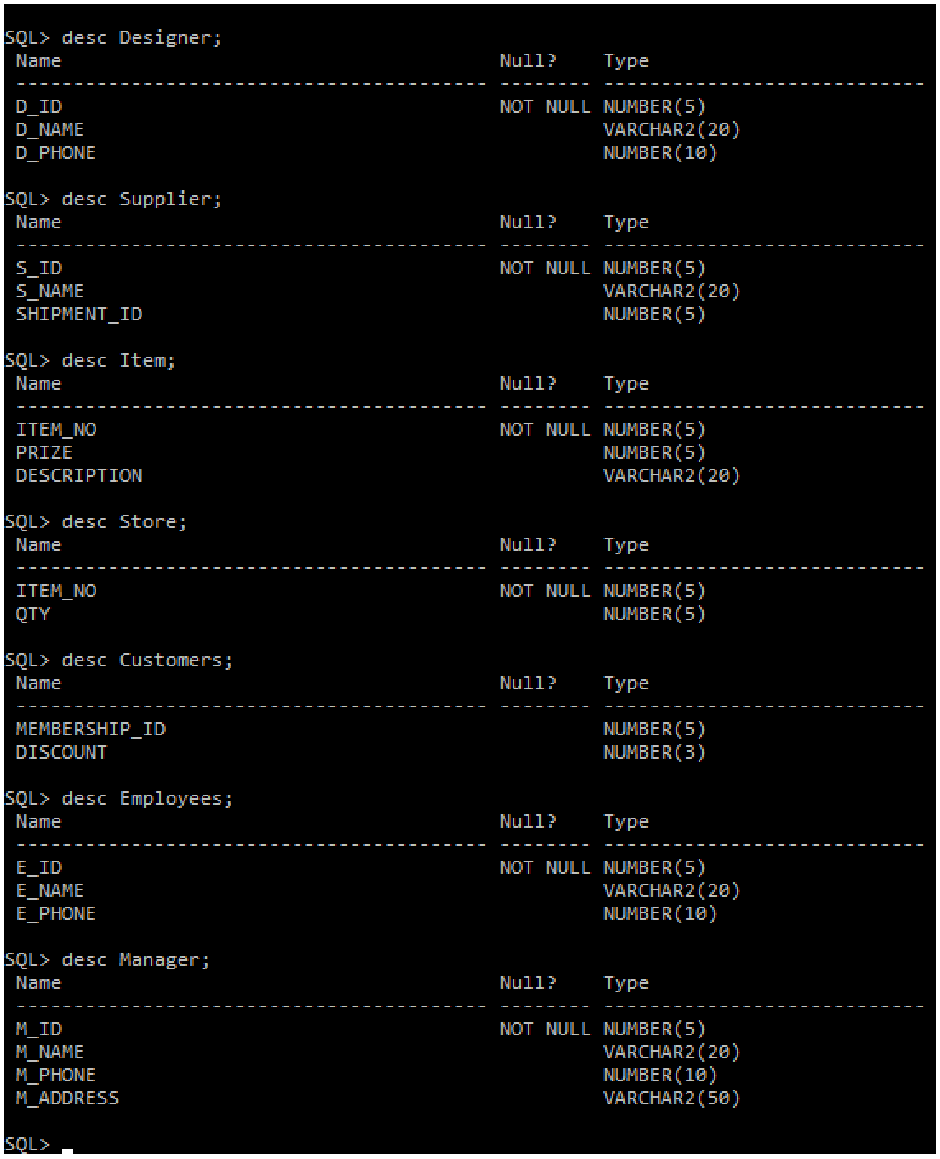
DDL COMMANDS:

Table Created in SQL for above mentioned purpose is as:

```
Create table designer(
D_id  number(5)  primary key, D_Name varchar2(20), D_Phone  number(10));

Create table supplier(
S_Id  number(5) primary key, S_Name  varchar2(20), Shipment_Id  number(5));

Create table item(
Item_No  number(5) primary key, Prize number(5), Description  varchar2(20));

Create table store(
Item_No  number(5) primary key, Qty  number(5));

Create table manager(
 M_Id  number(5) primary key, M_Name  varchar2(20), M_Phone  number(10), M_Address  Varchar2(30));

Create table employees(
E_Id  number(5) primary key, E_Name  varchar2(20), E_Phone  number(10));

Create table customers(
Membersip_Id  number(5), Discount  number(3));
```

```
SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
CUSTOMERS                       TABLE
DESIGNER                        TABLE
EMPLOYEES                      TABLE
ITEM                           TABLE
STORE                          TABLE
SUPPLIER                        TABLE
MANAGER                         TABLE

7 rows selected.
```

Description of the Tables:



```
SQL> desc Designer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 D_ID                                      NOT NULL NUMBER(5)
 D_NAME                                             VARCHAR2(20)
 D_PHONE                                            NUMBER(10)

SQL> desc Supplier;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 S_ID                                      NOT NULL NUMBER(5)
 S_NAME                                             VARCHAR2(20)
 SHIPMENT_ID                                        NUMBER(5)

SQL> desc Item;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_NO                                   NOT NULL NUMBER(5)
 PRIZE                                              NUMBER(5)
 DESCRIPTION                                        VARCHAR2(20)

SQL> desc Store;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_NO                                   NOT NULL NUMBER(5)
 QTY                                                NUMBER(5)

SQL> desc Customers;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEMBERSHIP_ID                                      NUMBER(5)
 DISCOUNT                                           NUMBER(3)

SQL> desc Employees;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 E_ID                                      NOT NULL NUMBER(5)
 E_NAME                                             VARCHAR2(20)
 E_PHONE                                            NUMBER(10)

SQL> desc Manager;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 M_ID                                      NOT NULL NUMBER(5)
 M_NAME                                             VARCHAR2(20)
 M_PHONE                                            NUMBER(10)
 M_ADDRESS                                          VARCHAR2(50)

SQL> _
```

Implementation:

Program:

User Interface:

```
package ClotheWith;
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.BorderFactory;
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;
import java.awt.Dimension;
import javax.swing.border.Border;
public class CW extends JFrame{
	/**
	 *                  */
	private static final long serialVersionUID = 1L;

	 private JMenuBar cw;

	private JMenu cwdesigner;
	private JMenu cwsupplier;
	private JMenu cwitems;
	private JMenu cwstore;
	private JMenu cwmanager;
	private JMenu cwemployees;
	private JMenu cwcustomers;

	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() {
```

```
            cw=new JMenuBar();
            cwdesigner= new JMenu("Designer");
            cwsupplier= new JMenu("Supplier");
            cwitems= new JMenu("Items");
            cwstore= new JMenu("Store");
            cwmanager= new JMenu("Manager");
            cwemployees= new JMenu("Employees");
            cwcustomers= new JMenu("Customers");
      labelName=new JLabel("ClotheWith", JLabel.CENTER);
      labelName.setPreferredSize(new Dimension(1000,100));
      labelName.setFont(new Font("Serif",Font.BOLD+Font.ITALIC,24));
      Border border= BorderFactory.createLineBorder(Color.BLACK);
      labelName.setBorder(border);
      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() {
                  cwdesigner.add(insert1);
                  cwdesigner.add(delete1);
                  cwdesigner.add(update1);
                  cwdesigner.add(view1);
```

```
                    cwsupplier.add(insert2);
                    cwsupplier.add(delete2);
                    cwsupplier.add(update2);
                    cwsupplier.add(view2);
                    cwitems.add(insert3);
                    cwitems.add(delete3);
                    cwitems.add(update3);
                    cwitems.add(view3);
                    cwstore.add(insert4);
                    cwstore.add(delete4);
                    cwstore.add(update4);
                    cwstore.add(view4);
                    cwmanager.add(insert5);
                    cwmanager.add(delete5);
                    cwmanager.add(update5);
                    cwmanager.add(view5);
                    cwemployees.add(insert6);
                    cwemployees.add(delete6);
                    cwemployees.add(update6);
                    cwemployees.add(view6);
                    cwcustomers.add(insert7);
                    cwcustomers.add(delete7);
                    cwcustomers.add(update7);
                    cwcustomers.add(view7);
                    cw.add(cwdesigner);
                    cw.add(cwsupplier);
                    cw.add(cwitems);
                    cw.add(cwstore);
                    cw.add(cwmanager);
                    cw.add(cwemployees);
                    cw.add(cwcustomers);
                    setJMenuBar(cw);
               p1.add(labelName);
               p1.setAlignmentY(CENTER_ALIGNMENT);
               p1.setBounds(500,500,8000,1000);
               p0.add(p1);
               p0.setBackground(Color.ORANGE);
               add(p0);
       }
/*******************************************************************************************
***********/
void closeWindow(){
             try {
                    int a=JOptionPane.showConfirmDialog(this,"Are you sure want to Quit
                    ClotheWith:");
                    if(a==JOptionPane.YES_OPTION){
                          JOptionPane.showMessageDialog(this,
                                   "Thank you!\nExiting ClotheWith","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() {
				Designer deg=new Designer(p0,CW.this,insert1,delete1,update1,view1);
				deg.buildGUI();
				Item ite=new Item(p0,CW.this,insert3,delete3,update3,view3);
				ite.buildGUI();
				Supplier sup=new Supplier(p0,CW.this,insert2,delete2,update2,view2);
				sup.buildGUI();
				Store sto=new Store(p0,CW.this,insert4,delete4,update4,view4);
				sto.buildGUI();
				Manager man=new Manager(p0,CW.this,insert5,delete5,update5,view5);
				man.buildGUI();
				Employee emp=new Employee(p0,CW.this,insert6,delete6,update6,view6);

				emp.buildGUI();
				Customers cus=new Customers(p0,CW.this,insert7,delete7,update7,view7);

				cus.buildGUI();
			addWindowListener(new WindowAdapter(){
				public void windowClosing(WindowEvent we)
				{
					closeWindow();
				}
			});

}
public CW() {
			initialize();
			addComponentsToFrame();
			register();
			pack();
		//	setBackground(Color.PINK);
			setTitle("Clothe With- Store");
			setSize(800,800);
			setVisible(true);
		}
}
```

Manager TABLE

```
package ClotheWith;

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

import 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 Manager{
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	private JButton insertButton,deleteButton,updateButton,viewButton;
	private JPanel p1,p2,p3,p;
	private JLabel lblM_Name,lblM_id,lblM_Phone,lblM_Address;
	private JTextField txtM_Name,txtM_id,txtM_Phone,txtM_Address;

	private List ManagerIDList;
	Connection con;ResultSet rs;
	Statement statement;
	private JFrame frame;
	private JMenuItem insert,delete,update,view;
	public Manager(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;

            lblM_id=new JLabel("Manager ID");
            lblM_Name=new JLabel("Manager Name");
            lblM_Phone=new JLabel("Phone");
            lblM_Address=new JLabel("Address");


            txtM_id=new JTextField(5);
            txtM_Name=new JTextField(20);
            txtM_Phone=new JTextField(10);
            txtM_Address=new JTextField(50);

            this.p=p;



     }

     public void connectToDB()
   {
            try {


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


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


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


      }
      public void loadManager() {
            try {
                  ManagerIDList.removeAll();
                  rs=statement.executeQuery("select M_id from Manager");
                  while(rs.next()) {
                        ManagerIDList.add(rs.getString("M_id"));
                  }
                  }
            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/Submit");
                        txtM_id.setText(null);
                        txtM_Name.setText(null);
                        txtM_Phone.setText(null);
                        txtM_Address.setText(null);


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


                        p1=new JPanel();

                         p1.setLayout(new GridLayout(4,2));
                         p1.add(lblM_id);
                         p1.add(txtM_id);
                         p1.add(lblM_Name);
                         p1.add(txtM_Name);
                         p1.add(lblM_Phone);
                         p1.add(txtM_Phone);
                         p1.add(lblM_Address);
                         p1.add(txtM_Address);

                         p3=new JPanel(new FlowLayout());
                         p3.add(insertButton);
```

```
                    //p1.add(txtf1);
                    p3.setBackground(Color.orange);
                    p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
                    p1.setBackground(Color.pink) ;




                    p2 = new JPanel(new FlowLayout());

                          ManagerIDList=new List(10);
                          loadManager();
                          p2.add(ManagerIDList);p2.setBackground(Color.cyan) ;

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


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


                    p.setLayout(new BorderLayout());

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

                    insertButton.addActionListener(new ActionListener() {
                          @Override
                               public void actionPerformed(ActionEvent e) {
                                     // TODO Auto-generated method stub
                   try {
                         String query="INSERT INTO Manager
VALUES("+txtM_id.getText()+",'"+txtM_Name.getText()+"',"+txtM_Phone.getText()+",'"+txtM_Address
.getText()+"')";

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



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

                    }


                         });
               }
               });
```

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

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

                txtM_id.setText(null);
                txtM_Name.setText(null);
                txtM_Phone.setText(null);
                txtM_Address.setText(null);


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


                p1=new JPanel();

                 p1.setLayout(new GridLayout(4,2));
                 p1.add(lblM_id);
                 p1.add(txtM_id);
                 p1.add(lblM_Name);
                 p1.add(txtM_Name);
                 p1.add(lblM_Phone);
                 p1.add(txtM_Phone);
                 p1.add(lblM_Address);
                 p1.add(txtM_Address);

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



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

                 p2 = new JPanel(new FlowLayout());

                        ManagerIDList=new List(10);
                        loadManager();
                        p2.add(ManagerIDList);p2.setBackground(Color.cyan) ;

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


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

```
                                ManagerIDList.addItemListener(new ItemListener()
                                     {
                                            public void itemStateChanged(ItemEvent e)
                                            {
                                                   try
                                                   {
                                                          rs=statement.executeQuery("select * from
Manager");
                                                          while (rs.next())
                                                          {
                                                                 if
(rs.getString("M_id").equals(ManagerIDList.getSelectedItem()))
                                                                 break;
                                                          }
                                                          if (!rs.isAfterLast())
                                                          {

       txtM_id.setText(rs.getString("M_id"));

       txtM_Name.setText(rs.getString("M_Name"));

       txtM_Phone.setText(rs.getString("M_Phone"));

       txtM_Address.setText(rs.getString("M_Address"));
                                                          }
                                                   }
                                                   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 you want
to delete:");
                                     if(a==JOptionPane.YES_OPTION){
                                     String query="DELETE FROM Manager WHERE
M_iD="+ManagerIDList.getSelectedItem();

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

```
                                }


                        }
                        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/Modify");
                        txtM_id.setText(null);
                        txtM_Name.setText(null);
                        txtM_Phone.setText(null);
                        txtM_Address.setText(null);


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


                        p1=new JPanel();

                         p1.setLayout(new GridLayout(4,2));
                         p1.add(lblM_id);
                         p1.add(txtM_id);
                         p1.add(lblM_Name);
                         p1.add(txtM_Name);
                         p1.add(lblM_Phone);
                         p1.add(txtM_Phone);
                         p1.add(lblM_Address);
                         p1.add(txtM_Address);
                         p3=new JPanel(new FlowLayout());
                         p3.add(updateButton);
                         //p1.add(txtf1);
                         p3.setBackground(Color.orange);
                         p1.setBounds(115,80,300,250);p3.setBounds(200,350,75,35);
                         p1.setBackground(Color.pink) ;

                         p2 = new JPanel(new FlowLayout());

                                ManagerIDList=new List(10);
```

```
                                    loadManager();
                                    p2.add(ManagerIDList);p2.setBackground(Color.cyan) ;

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


                             p. add(p1);p.add(p3);
                             p. add(p2);
                              ManagerIDList.addItemListener(new ItemListener()
                                   {
                                          public void itemStateChanged(ItemEvent e)
                                          {
                                                 try
                                                 {
                                                        rs=statement.executeQuery("select * from
Manager");
                                                        while (rs.next())
                                                        {
                                                               if
(rs.getString("M_id").equals(ManagerIDList.getSelectedItem()))
                                                               break;
                                                        }
                                                        if (!rs.isAfterLast())
                                                        {

       txtM_id.setText(rs.getString("M_id"));

       txtM_Name.setText(rs.getString("M_Name"));

       txtM_Phone.setText(rs.getString("M_Phone"));

       txtM_Address.setText(rs.getString("M_Address"));
                                                        }
                                                 }
                                                 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 you want to update:");
                                                   if(a==JOptionPane.YES_OPTION){
                                                   String query="update Manager set
M_Name='"+txtM_Name.getText()+"',M_Phone="+txtM_Phone.getText()+",M_Address='"+txtM_Address.get
Text()+"' WHERE M_id="+ManagerIDList.getSelectedItem();

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


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

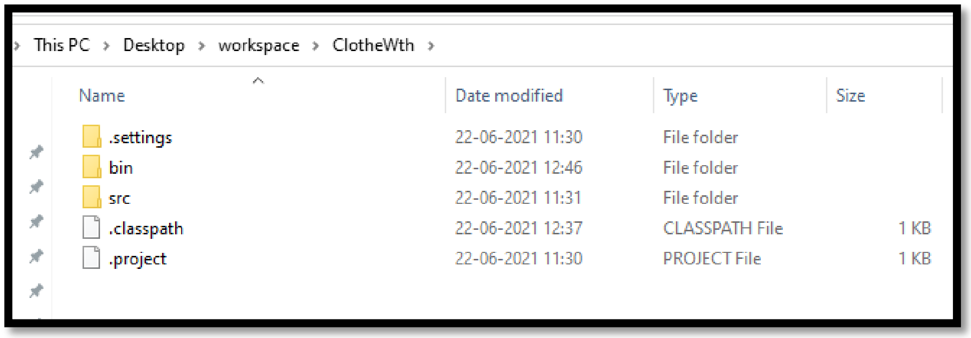
```
                                            JFrame f;

                                               JTable j;

                                                   f = new JFrame();


                                                   f.setTitle("Manager details");


                                                   DefaultTableModel model = new
DefaultTableModel();

                                                   j = new JTable(model);
                                                  model.addColumn("Manager id");
                                                  model.addColumn("Manager Name");
                                                  model.addColumn("PhoneNo");
                                                  model.addColumn("Address");


                                                  try {

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

                                                              while(rs.next()) {
                                                                     model.addRow(new
Object[]{rs.getString("M_id"),
rs.getString("M_Name"),rs.getString("M_Phone"),rs.getString("M_Address")});
                                                              }
                                                              }
                                                        catch(SQLException viewException) {
                                                              displaySQLErrors(viewException);
                                                        }
                                                        j.setEnabled(false);
                                                   j.setBounds(30, 40, 300, 300);


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

                                                   f.setSize(800, 400);

                                                   f.setVisible(true);


                                               }




                                    });
```

```
                    }
            });
        }


}
```

GITHUB LINK:

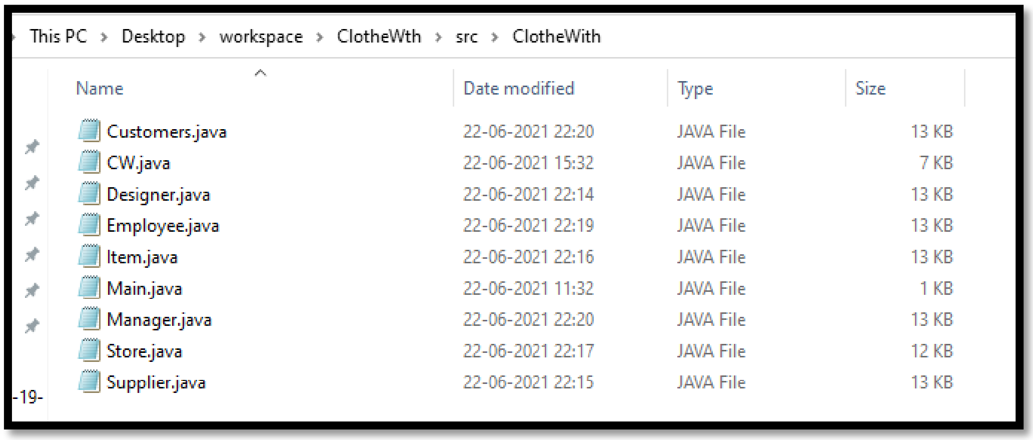
https://github.com/tanmayee043/ClotheWith-DBMS

FOLDER STRUCTURE:



This PC › Desktop › workspace › ClotheWth ›

Name	Date modified	Type	Size
.settings	22-06-2021 11:30	File folder	
bin	22-06-2021 12:46	File folder	
src	22-06-2021 11:31	File folder	
.classpath	22-06-2021 12:37	CLASSPATH File	1 KB
.project	22-06-2021 11:30	PROJECT File	1 KB



This PC › Desktop › workspace › ClotheWth › src › ClotheWith

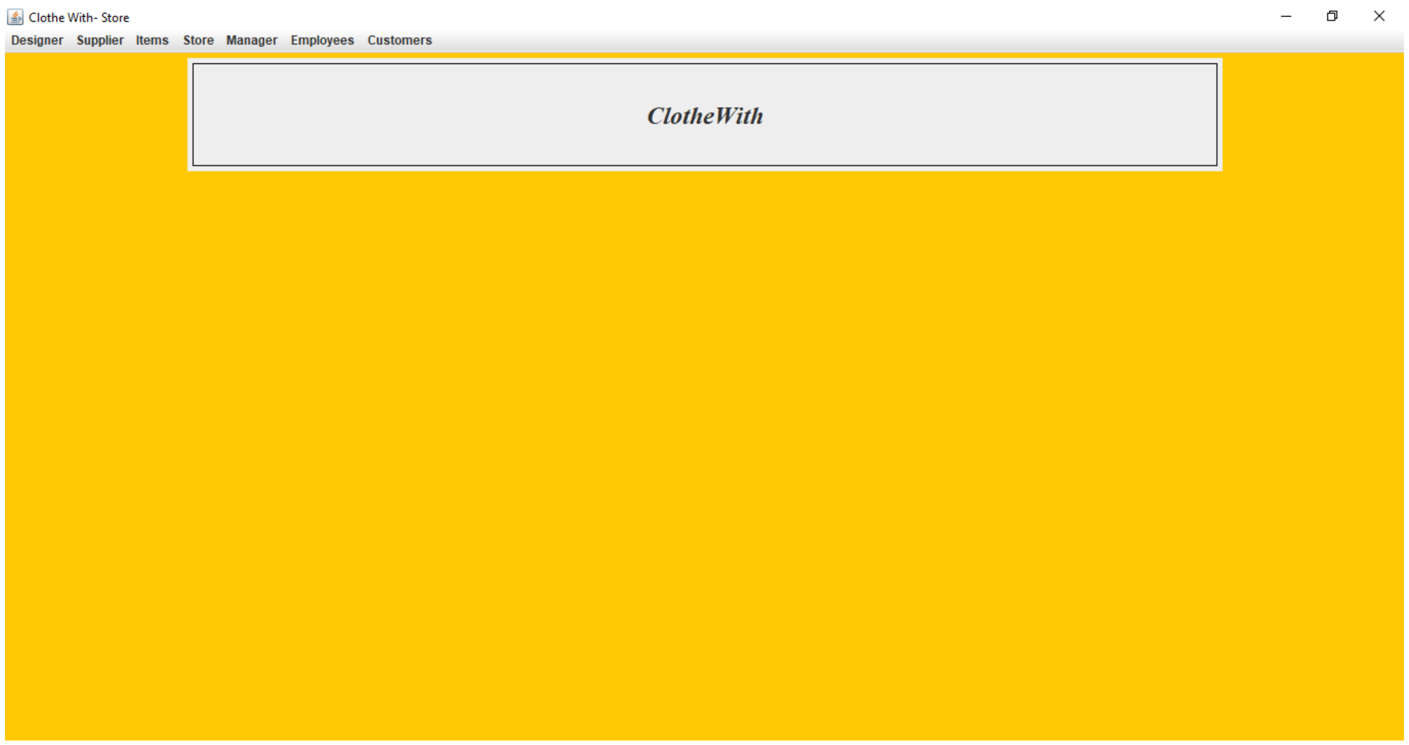
Name	Date modified	Type	Size
Customers.java	22-06-2021 22:20	JAVA File	13 KB
CW.java	22-06-2021 15:32	JAVA File	7 KB
Designer.java	22-06-2021 22:14	JAVA File	13 KB
Employee.java	22-06-2021 22:19	JAVA File	13 KB
Item.java	22-06-2021 22:16	JAVA File	13 KB
Main.java	22-06-2021 11:32	JAVA File	1 KB
Manager.java	22-06-2021 22:20	JAVA File	13 KB
Store.java	22-06-2021 22:17	JAVA File	12 KB
Supplier.java	22-06-2021 22:15	JAVA File	13 KB

-19-

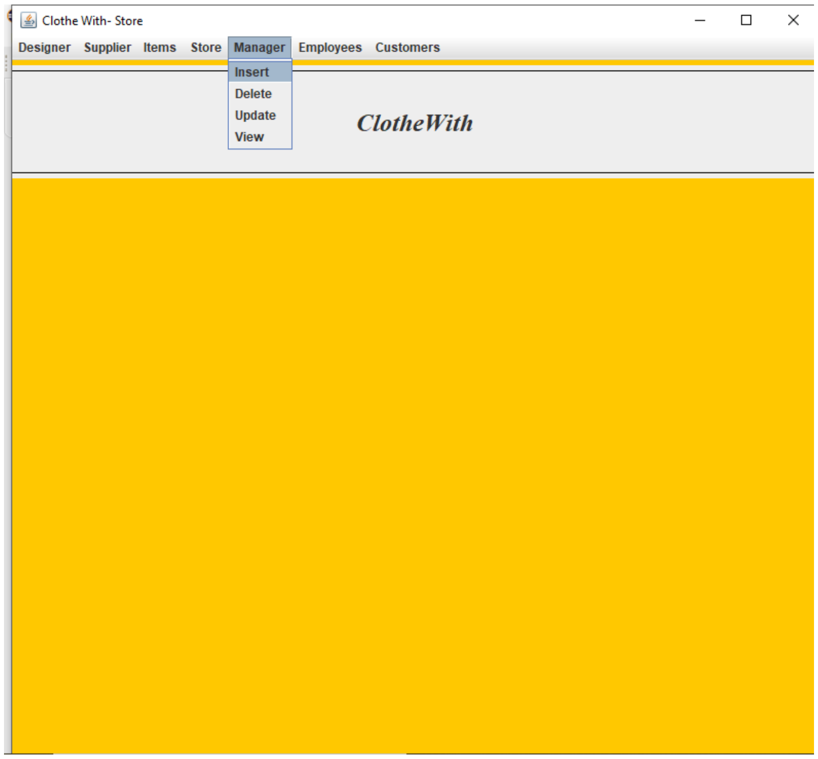
Testing:

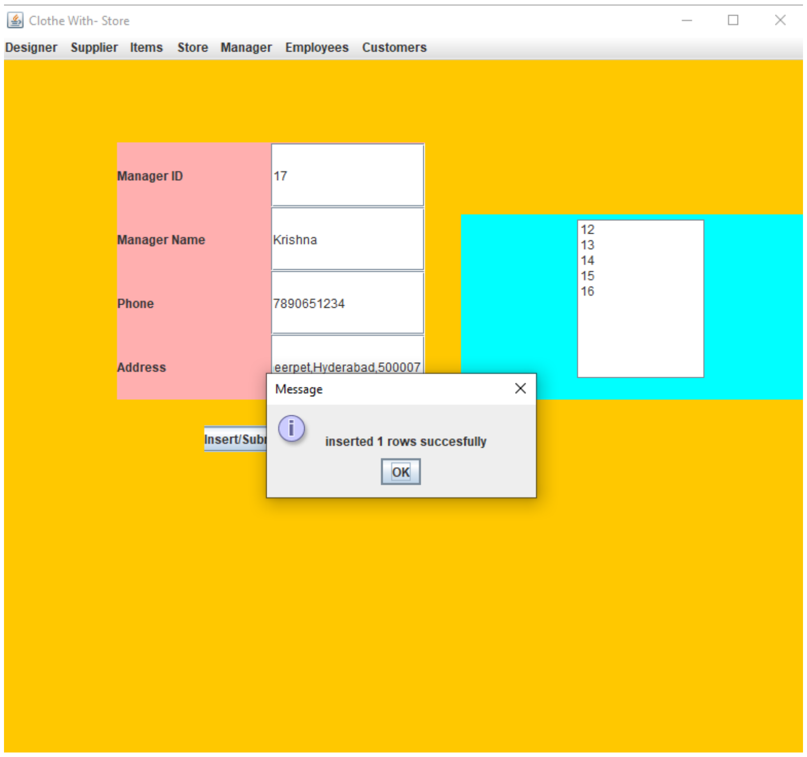
OUTPUT SCREENSHOTS:

Java GUI Screenshot:



Insertion:



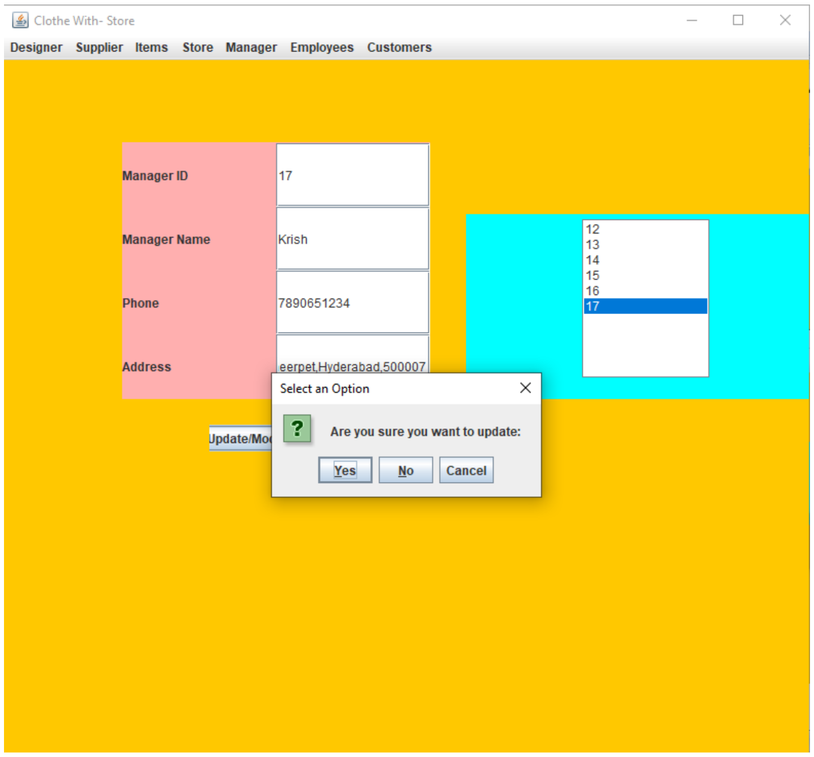


Manager ID 17
Manager Name Krishna
Phone 7890651234
Address eerpet,Hyderabad,500007

```
12
13
14
15
16
```

inserted 1 rows succesfully

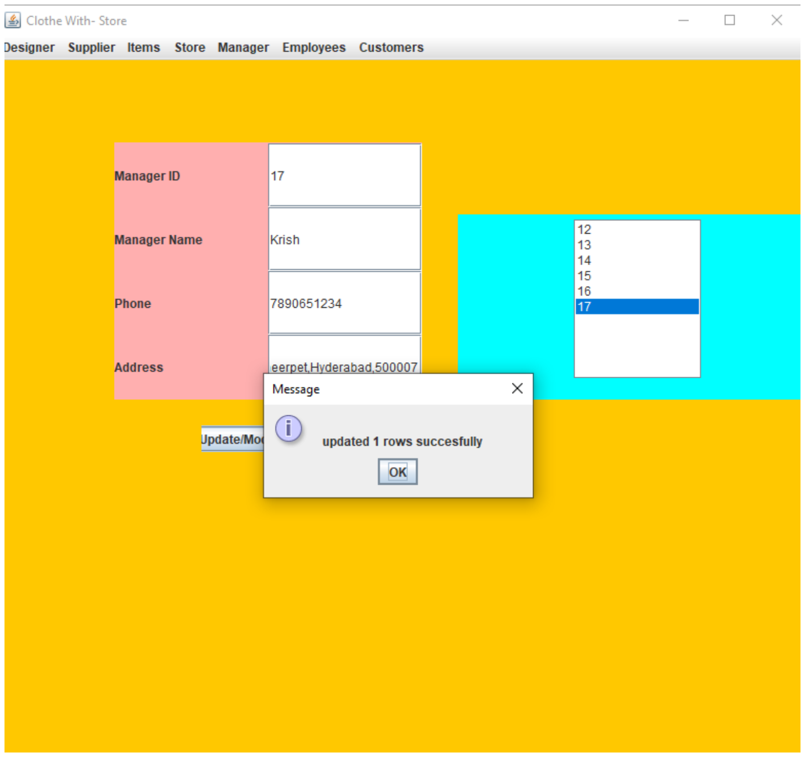
Updation:



Manager ID 17
Manager Name Krish
Phone 7890651234
Address eerpet,Hyderabad,500007

12
13
14
15
16
17

Are you sure you want to update:

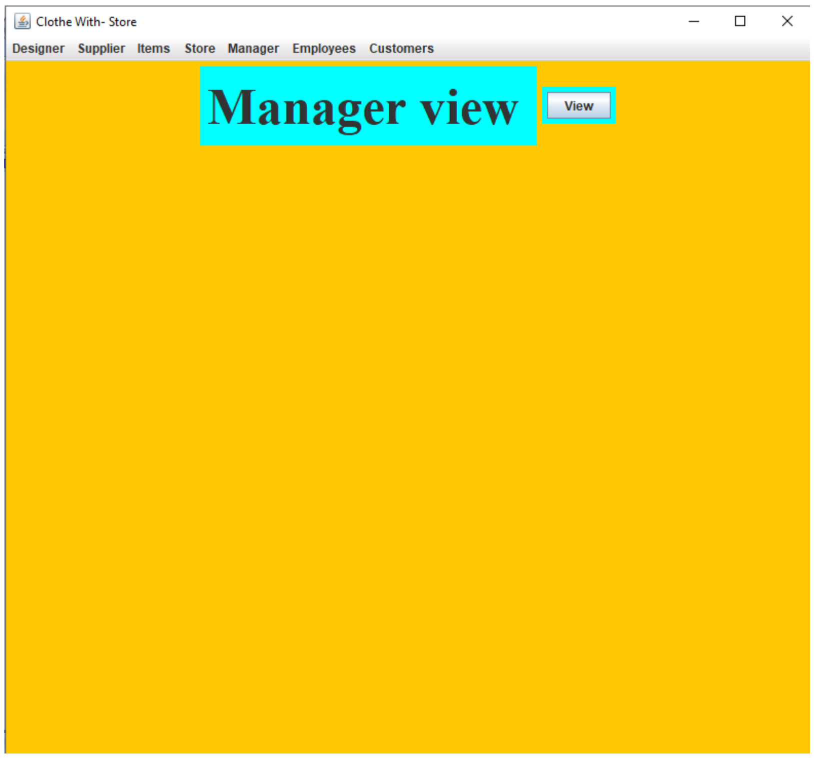


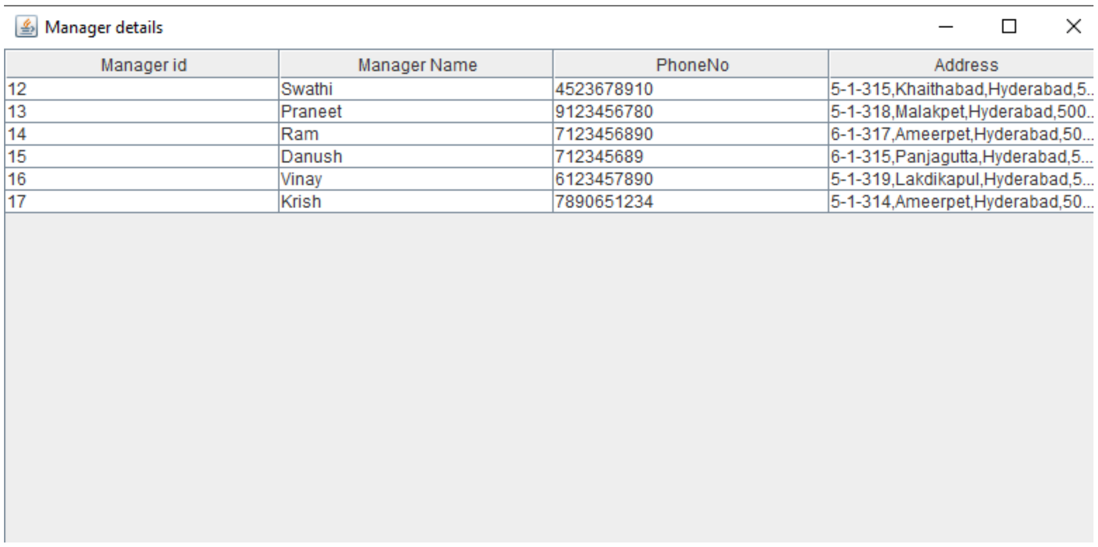
Manager ID 17
Manager Name Krish
Phone 7890651234
Address eerpet,Hyderabad,500007

12
13
14
15
16
17

updated 1 rows succesfully

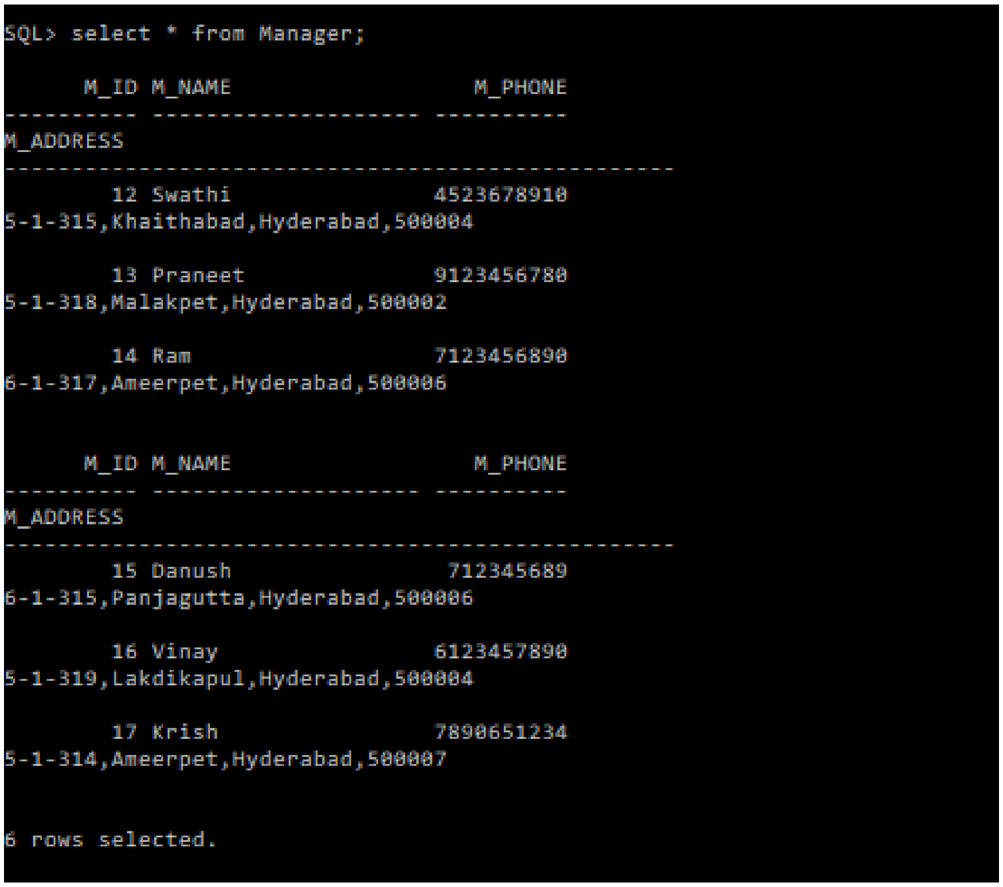
Table View:




Manager details

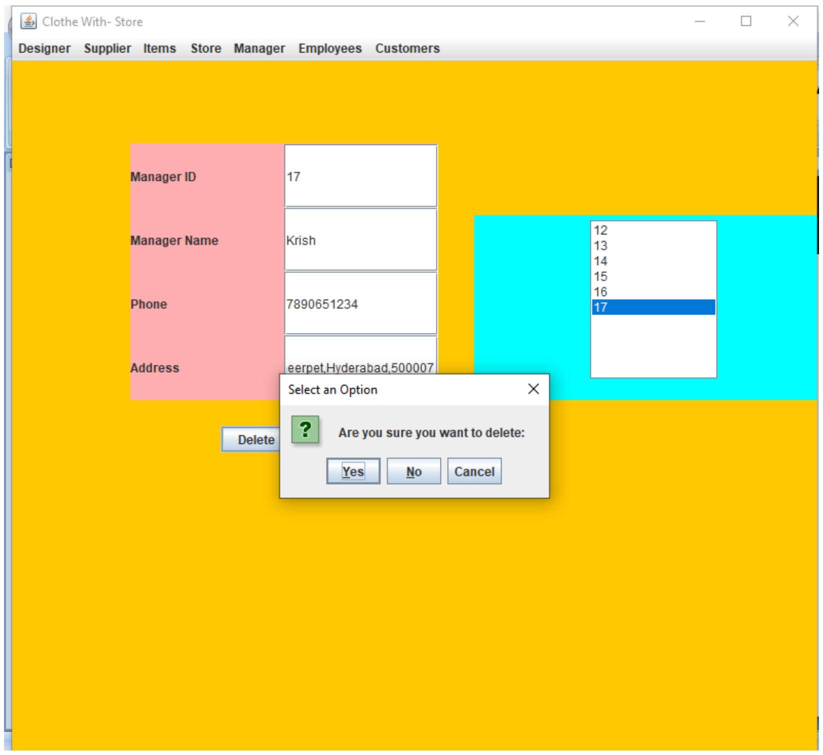
Manager id	Manager Name	PhoneNo	Address
12	Swathi	4523678910	5-1-315,Khaithabad,Hyderabad,5...
13	Praneet	9123456780	5-1-318,Malakpet,Hyderabad,500...
14	Ram	7123456890	6-1-317,Ameerpet,Hyderabad,50...
15	Danush	712345689	6-1-315,Panjagutta,Hyderabad,5...
16	Vinay	6123457890	5-1-319,Lakdikapul,Hyderabad,5...
17	Krish	7890651234	5-1-314,Ameerpet,Hyderabad,50...

SQL View:



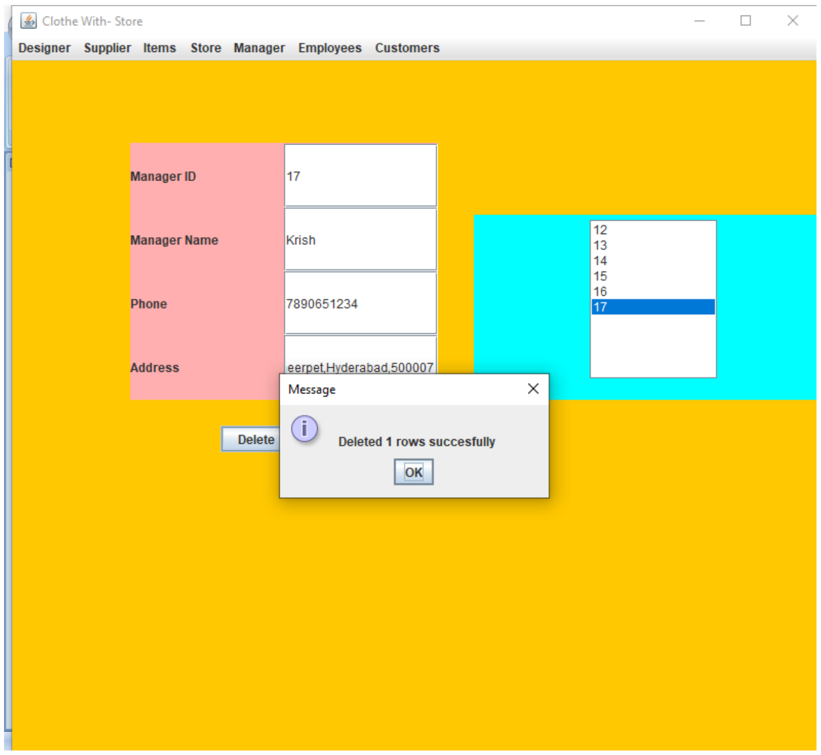
```
SQL> select * from Manager;

      M_ID M_NAME                   M_PHONE
---------- -------------------- ----------
M_ADDRESS
--------------------------------------------------
        12 Swathi               4523678910
5-1-315,Khaithabad,Hyderabad,500004

        13 Praneet              9123456780
5-1-318,Malakpet,Hyderabad,500002

        14 Ram                  7123456890
6-1-317,Ameerpet,Hyderabad,500006


      M_ID M_NAME                   M_PHONE
---------- -------------------- ----------
M_ADDRESS
--------------------------------------------------
        15 Danush                712345689
6-1-315,Panjagutta,Hyderabad,500006

        16 Vinay                6123457890
5-1-319,Lakdikapul,Hyderabad,500004

        17 Krish                7890651234
5-1-314,Ameerpet,Hyderabad,500007


6 rows selected.
```

Deletion:



Manager ID: 17
Manager Name: Krish
Phone: 7890651234
Address: eerpet,Hyderabad,500007

Are you sure you want to delete:



Manager ID 17

Manager Name Krish

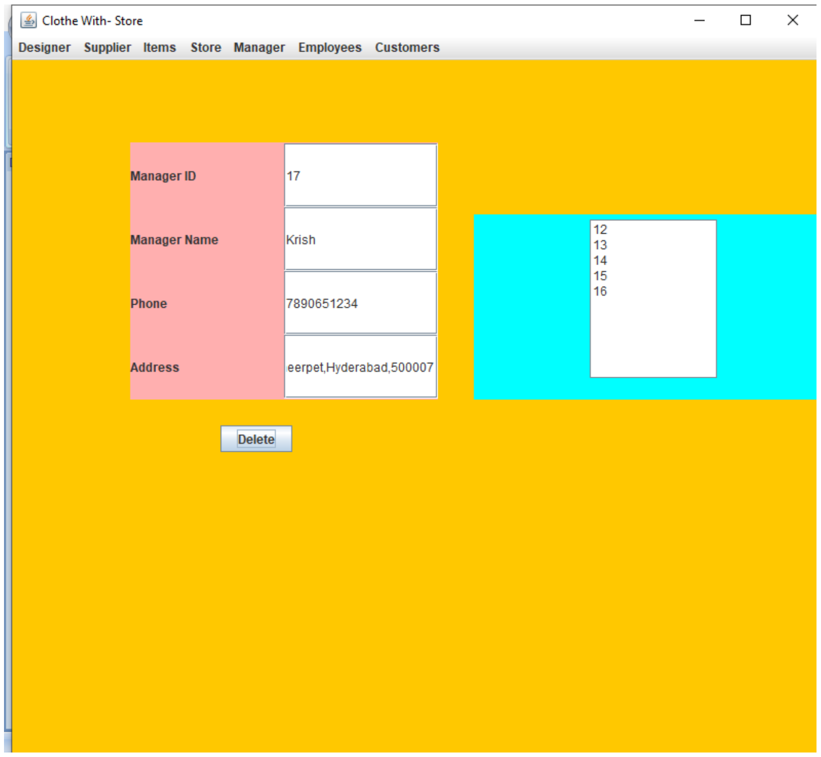
Phone 7890651234

Address eerpet,Hyderabad,500007

12
13
14
15
16
17

Deleted 1 rows succesfully

After Deletion:



Manager ID 17

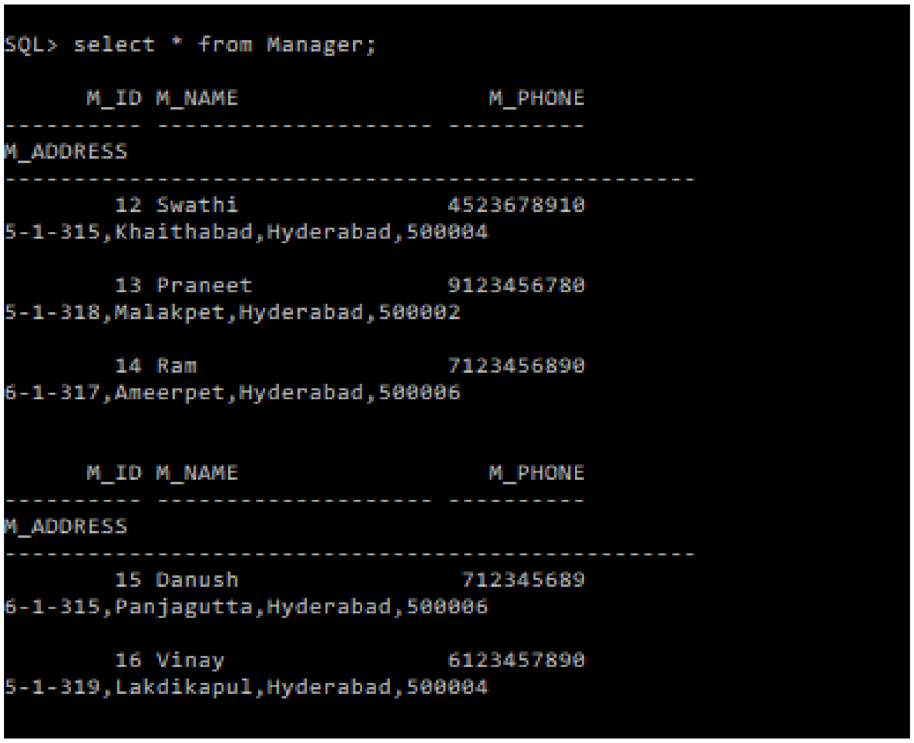
Manager Name Krish

Phone 7890651234

Address eerpet,Hyderabad,500007

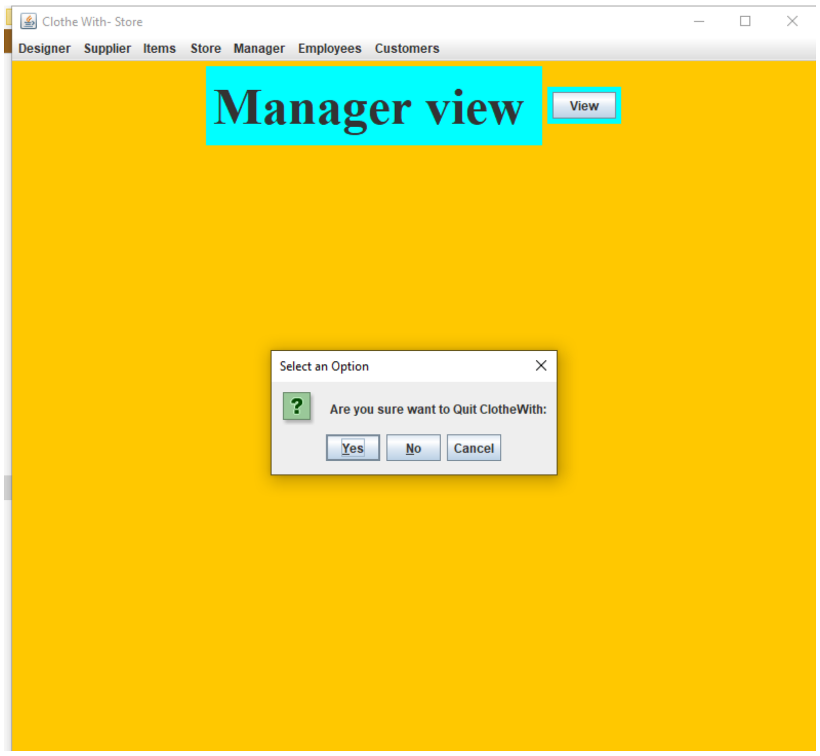
12
13
14
15
16

SQL View After Deletion:

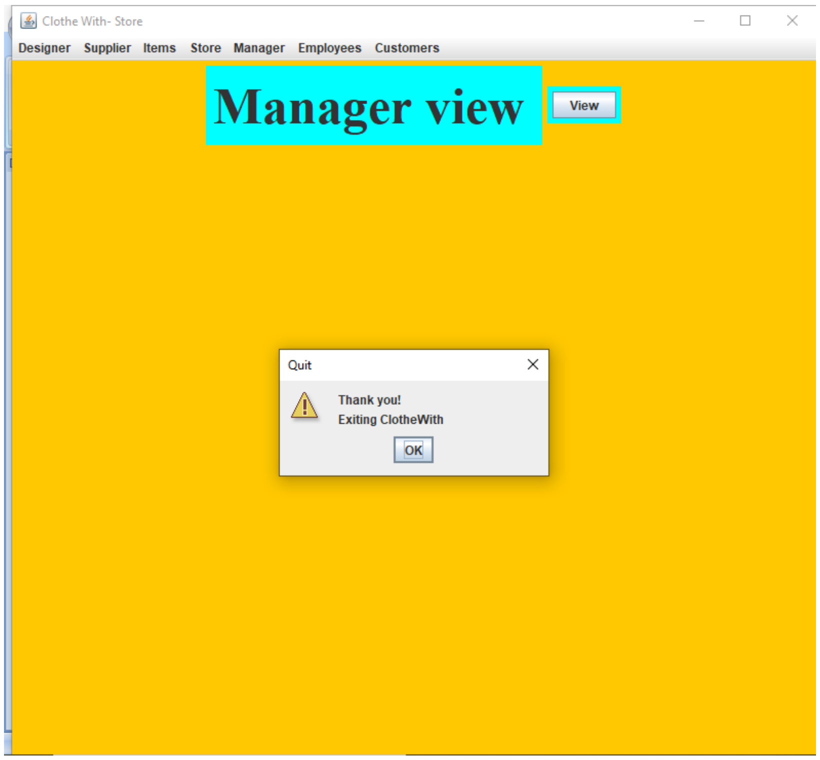


```
SQL> select * from Manager;

      M_ID M_NAME                         M_PHONE
---------- -------------------- ----------
M_ADDRESS
--------------------------------------------------
        12 Swathi                    4523678910
5-1-315,Khaithabad,Hyderabad,500004

        13 Praneet                   9123456780
5-1-318,Malakpet,Hyderabad,500002

        14 Ram                       7123456890
6-1-317,Ameerpet,Hyderabad,500006


      M_ID M_NAME                         M_PHONE
---------- -------------------- ----------
M_ADDRESS
--------------------------------------------------
        15 Danush                     712345689
6-1-315,Panjagutta,Hyderabad,500006

        16 Vinay                     6123457890
5-1-319,Lakdikapul,Hyderabad,500004
```

Closing ClotheWith



Are you sure want to Quit ClotheWith:



Thank you!
Exiting ClotheWith

Results:

I had successfully completed MINI PROJECT on “CLOTHEWITH”.

Discussion and future Work:

This application provides the management to select the details of employees and can keep track of clothe items present in the store and the information of customers. While working on this project I wanted to extend to make an app which is management friendly and provides accurate information.
It stores the details and data of employees, customers and items in appropriate manner.

CONCLUSION:

Thus, a Java SWING based ‘CLOTHEWITH’ 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