ANALYSIS AND DESIGN OF INVENTORY MANAGEMENT SYSTEM

Particulars of the Student:

Name: Prabhat M Reddy

Roll No: 18

Admn No: 4093

Class: XII A

Acknowledgement

It is with great pleasure that I find myself penning down these lines to express my sincere thanks to the various people who helped me a long way in completing the project.

The harmonious climate in our school provided a proper guide for preparing the project. It was a privilege to have been guided by Mr. Madhavacharya, and Ms. Y. Sandhya

Contents

Contents

Project Description

This project is a model Inventory Management System for a Musical Instruments Store ‘Rock Rules’. It also acts as an Order Management System. It has two sides to it. The first is the Customer side, where the Customer can browse through products, place orders, check their invoices, and edit their details. The other is the Administrator side. Here, designated employees of the store can obtain data related to Customers, Suppliers, Products, Inventories, and Invoices.

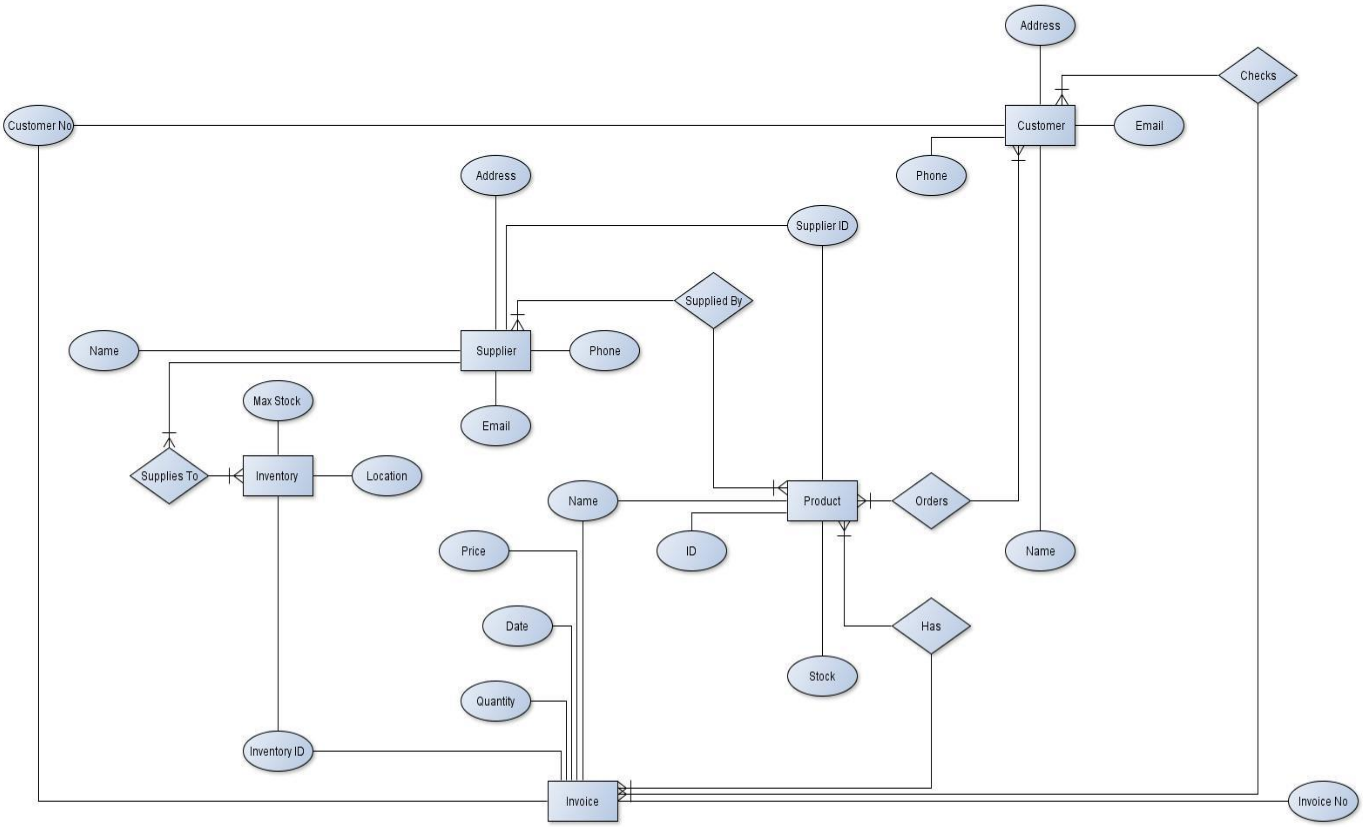
To access either side, one must login as a registered user. Prospective customers can create a new ID and register it, but the Admin side does not have this facility. Once logged in, the user can use the interface to work with the data in the SQL database.

Technology Used

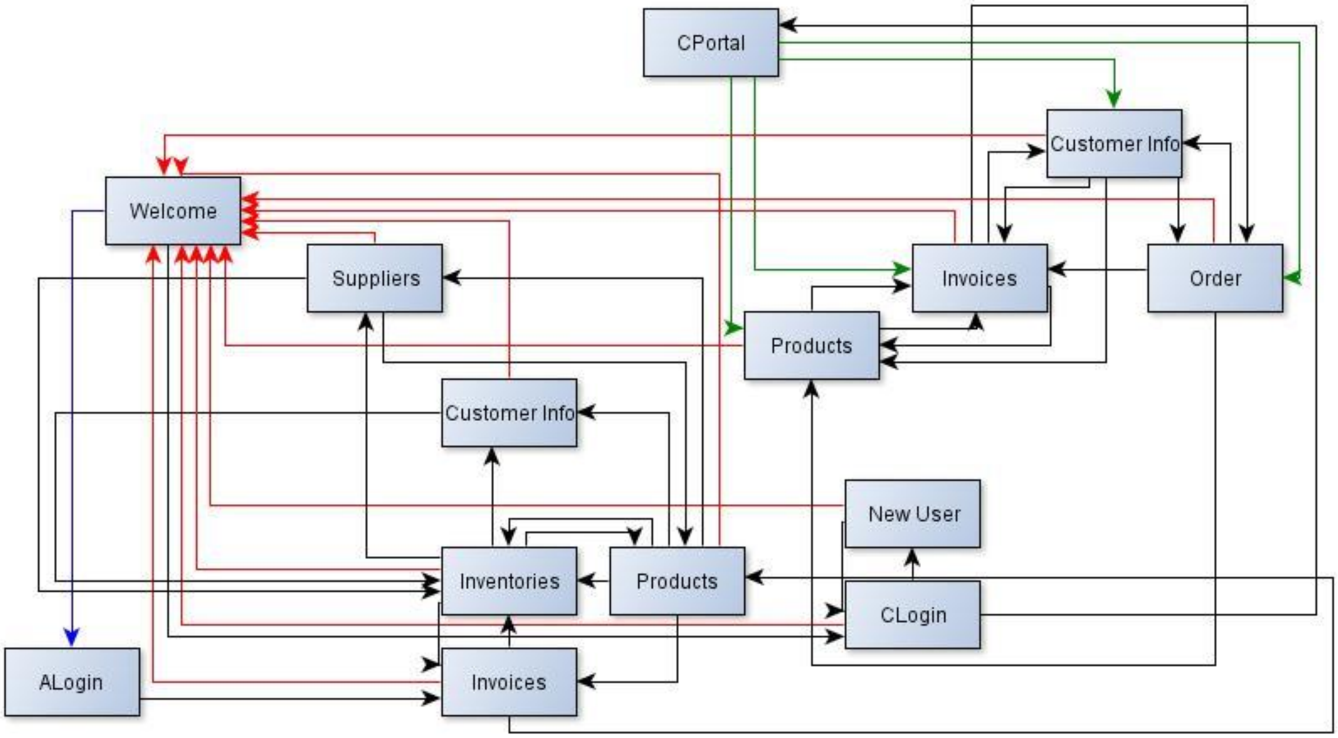
The back end of the project (DBMS), was done through MySQL Server 5.1. The front end of the project (GUI), was done through Net Beans IDE 7.0.1. The documentation of the project was done through MS Word 2007, and MS Paint 2007.

Project Design

ER Diagram



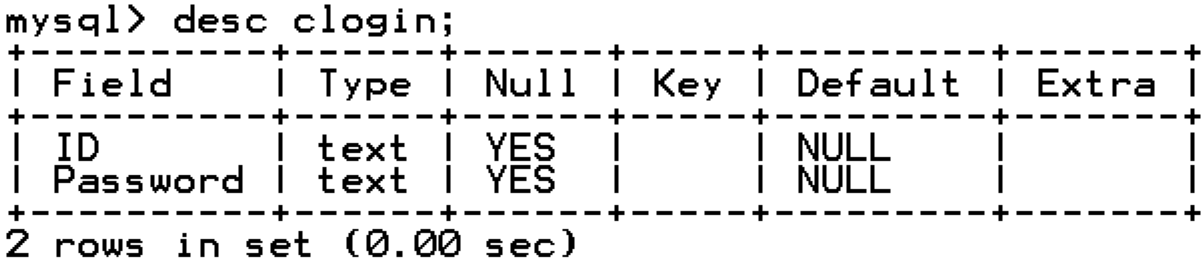
Flow Chart


CPortal
Customer Info
Welcome
Suppliers
Invoices
Order
Products
Customer Info
New User
Inventories
Products
CLogin
ALogin
Invoices

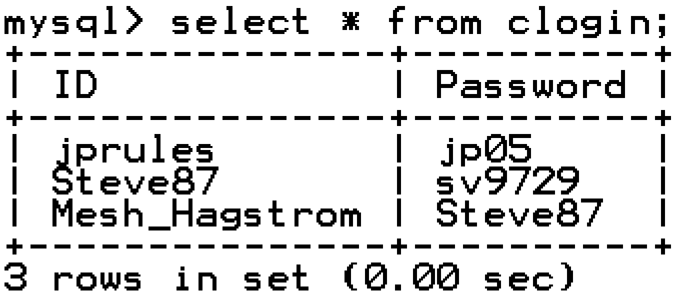
Screen Shots

MySQL

Customer Login:

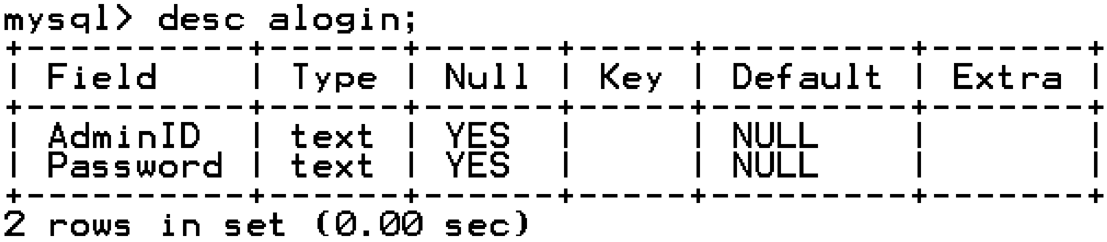


```
mysql> desc clogin;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| ID       | text | YES  |     | NULL    |       |
| Password | text | YES  |     | NULL    |       |
+----------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

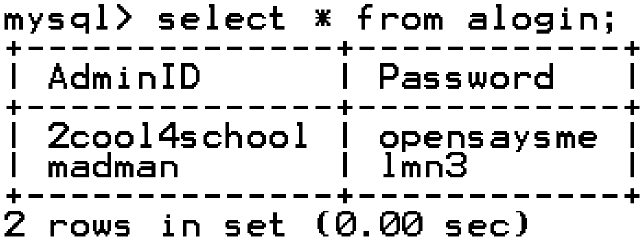


```
mysql> select * from clogin;
+---------------+----------+
| ID            | Password |
+---------------+----------+
| jprules       | jp05     |
| Steve87       | sv9729   |
| Mesh_Hagstrom | Steve87  |
+---------------+----------+
3 rows in set (0.00 sec)
```

Administrator Login:

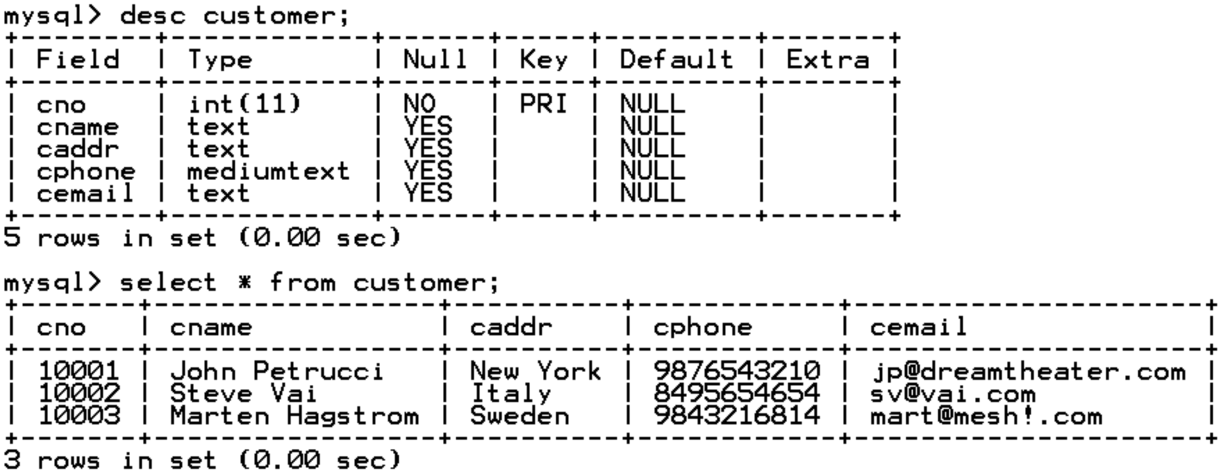


```
mysql> desc alogin;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| AdminID  | text | YES  |     | NULL    |       |
| Password | text | YES  |     | NULL    |       |
+----------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



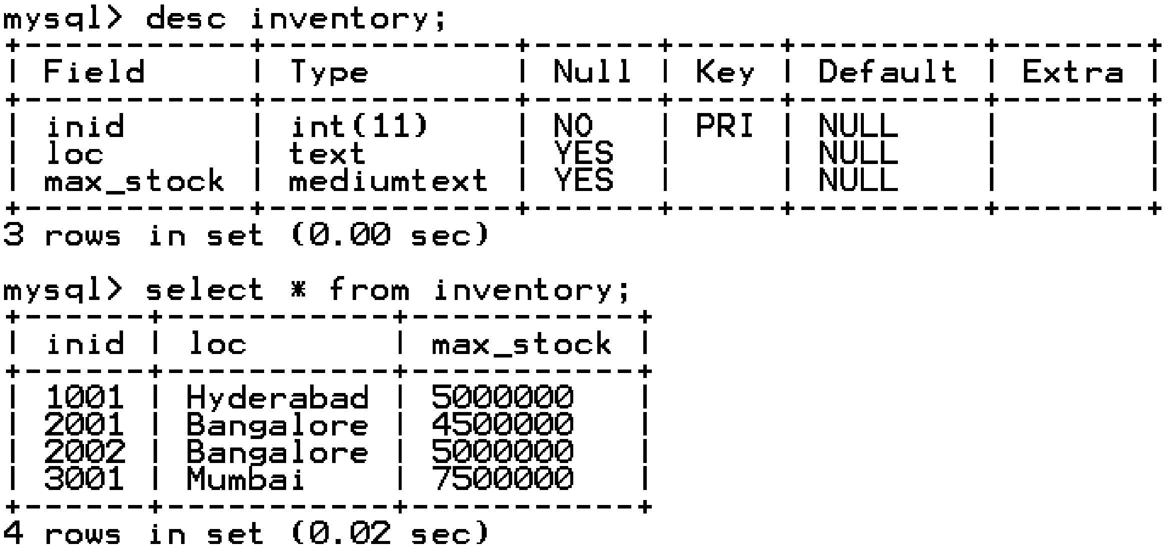
```
mysql> select * from alogin;
+-------------+-------------+
| AdminID     | Password    |
+-------------+-------------+
| 2cool4school | opensaysme |
| madman      | lmn3        |
+-------------+-------------+
2 rows in set (0.00 sec)
```

Customer:



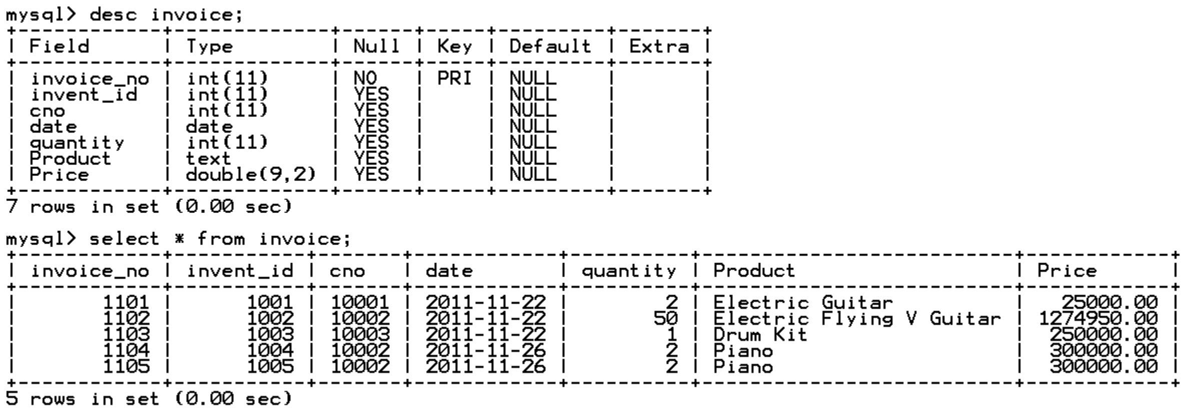
```
mysql> desc customer;
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| cno    | int(11)    | NO   | PRI | NULL    |       |
| cname  | text       | YES  |     | NULL    |       |
| caddr  | text       | YES  |     | NULL    |       |
| cphone | mediumtext | YES  |     | NULL    |       |
| cemail | text       | YES  |     | NULL    |       |
+--------+------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> select * from customer;
+-------+-----------------+----------+------------+----------------------+
| cno   | cname           | caddr    | cphone     | cemail               |
+-------+-----------------+----------+------------+----------------------+
| 10001 | John Petrucci   | New York | 9876543210 | jp@dreamtheater.com  |
| 10002 | Steve Vai       | Italy    | 8495654654 | sv@vai.com           |
| 10003 | Marten Hagstrom | Sweden   | 9843216814 | mart@mesh!.com       |
+-------+-----------------+----------+------------+----------------------+
3 rows in set (0.00 sec)
```

Inventory:



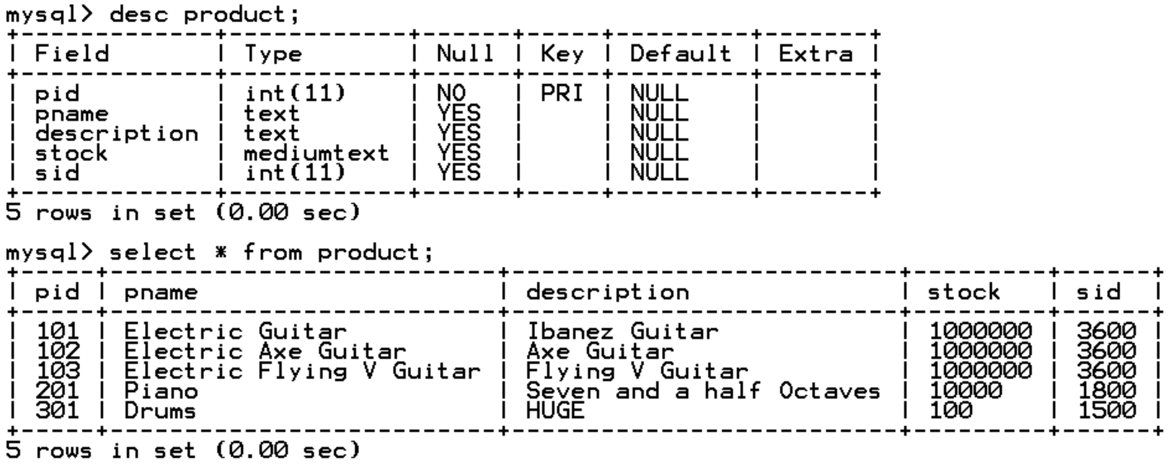
```
mysql> desc inventory;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| inid      | int(11)    | NO   | PRI | NULL    |       |
| loc       | text       | YES  |     | NULL    |       |
| max_stock | mediumtext | YES  |     | NULL    |       |
+-----------+------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> select * from inventory;
+------+-----------+-----------+
| inid | loc       | max_stock |
+------+-----------+-----------+
| 1001 | Hyderabad | 5000000   |
| 2001 | Bangalore | 4500000   |
| 2002 | Bangalore | 5000000   |
| 3001 | Mumbai    | 7500000   |
+------+-----------+-----------+
4 rows in set (0.02 sec)
```

Invoice:



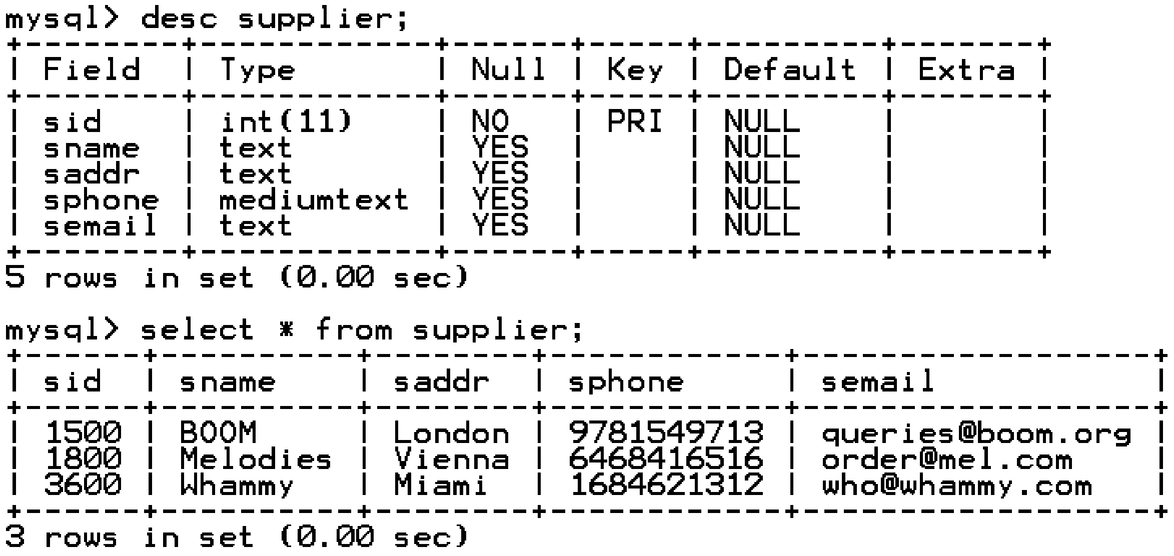
```
mysql> desc invoice;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| invoice_no | int(11)     | NO   | PRI | NULL    |       |
| invent_id  | int(11)     | YES  |     | NULL    |       |
| cno        | int(11)     | YES  |     | NULL    |       |
| date       | date        | YES  |     | NULL    |       |
| quantity   | int(11)     | YES  |     | NULL    |       |
| Product    | text        | YES  |     | NULL    |       |
| Price      | double(9,2) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> select * from invoice;
+------------+-----------+-------+------------+----------+--------------------------+------------+
| invoice_no | invent_id | cno   | date       | quantity | Product                  | Price      |
+------------+-----------+-------+------------+----------+--------------------------+------------+
|       1101 |      1001 | 10001 | 2011-11-22 |        2 | Electric Guitar          |   25000.00 |
|       1102 |      1002 | 10002 | 2011-11-22 |       50 | Electric Flying V Guitar | 1274950.00 |
|       1103 |      1003 | 10003 | 2011-11-22 |        1 | Drum Kit                 |  250000.00 |
|       1104 |      1004 | 10002 | 2011-11-26 |        2 | Piano                    |  300000.00 |
|       1105 |      1005 | 10002 | 2011-11-26 |        2 | Piano                    |  300000.00 |
+------------+-----------+-------+------------+----------+--------------------------+------------+
5 rows in set (0.00 sec)
```

Product:



```
mysql> desc product;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| pid         | int(11)    | NO   | PRI | NULL    |       |
| pname       | text       | YES  |     | NULL    |       |
| description | text       | YES  |     | NULL    |       |
| stock       | mediumtext | YES  |     | NULL    |       |
| sid         | int(11)    | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> select * from product;
+-----+--------------------------+---------------------------+---------+------+
| pid | pname                    | description               | stock   | sid  |
+-----+--------------------------+---------------------------+---------+------+
| 101 | Electric Guitar          | Ibanez Guitar             | 1000000 | 3600 |
| 102 | Electric Axe Guitar      | Axe Guitar                | 1000000 | 3600 |
| 103 | Electric Flying V Guitar | Flying V Guitar           | 1000000 | 3600 |
| 201 | Piano                    | Seven and a half Octaves  | 10000   | 1800 |
| 301 | Drums                    | HUGE                      | 100     | 1500 |
+-----+--------------------------+---------------------------+---------+------+
5 rows in set (0.00 sec)
```

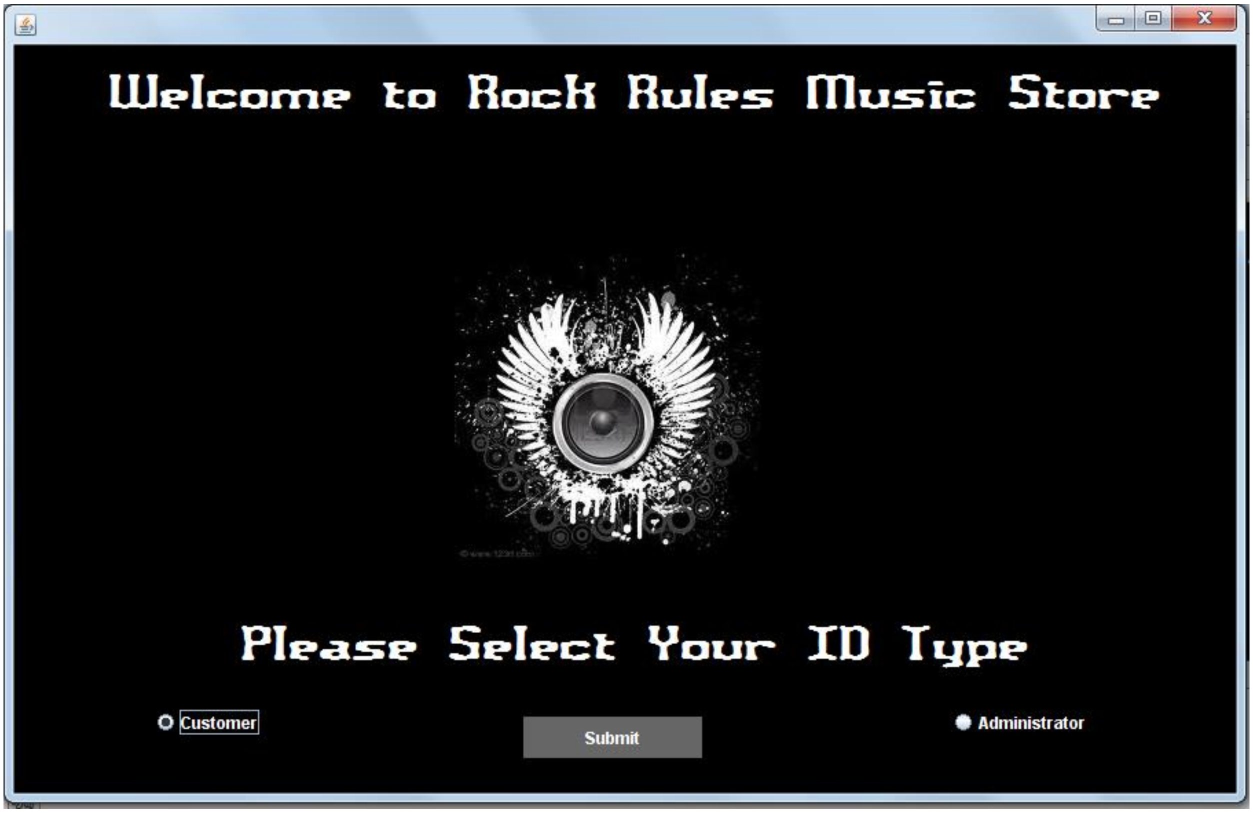
Supplier:



```
mysql> desc supplier;
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| sid    | int(11)    | NO   | PRI | NULL    |       |
| sname  | text       | YES  |     | NULL    |       |
| saddr  | text       | YES  |     | NULL    |       |
| sphone | mediumtext | YES  |     | NULL    |       |
| semail | text       | YES  |     | NULL    |       |
+--------+------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> select * from supplier;
+------+----------+--------+------------+------------------+
| sid  | sname    | saddr  | sphone     | semail           |
+------+----------+--------+------------+------------------+
| 1500 | BOOM     | London | 9781549713 | queries@boom.org |
| 1800 | Melodies | Vienna | 6468416516 | order@mel.com    |
| 3600 | Whammy   | Miami  | 1684621312 | who@whammy.com   |
+------+----------+--------+------------+------------------+
3 rows in set (0.00 sec)
```

Net Beans

Welcome:

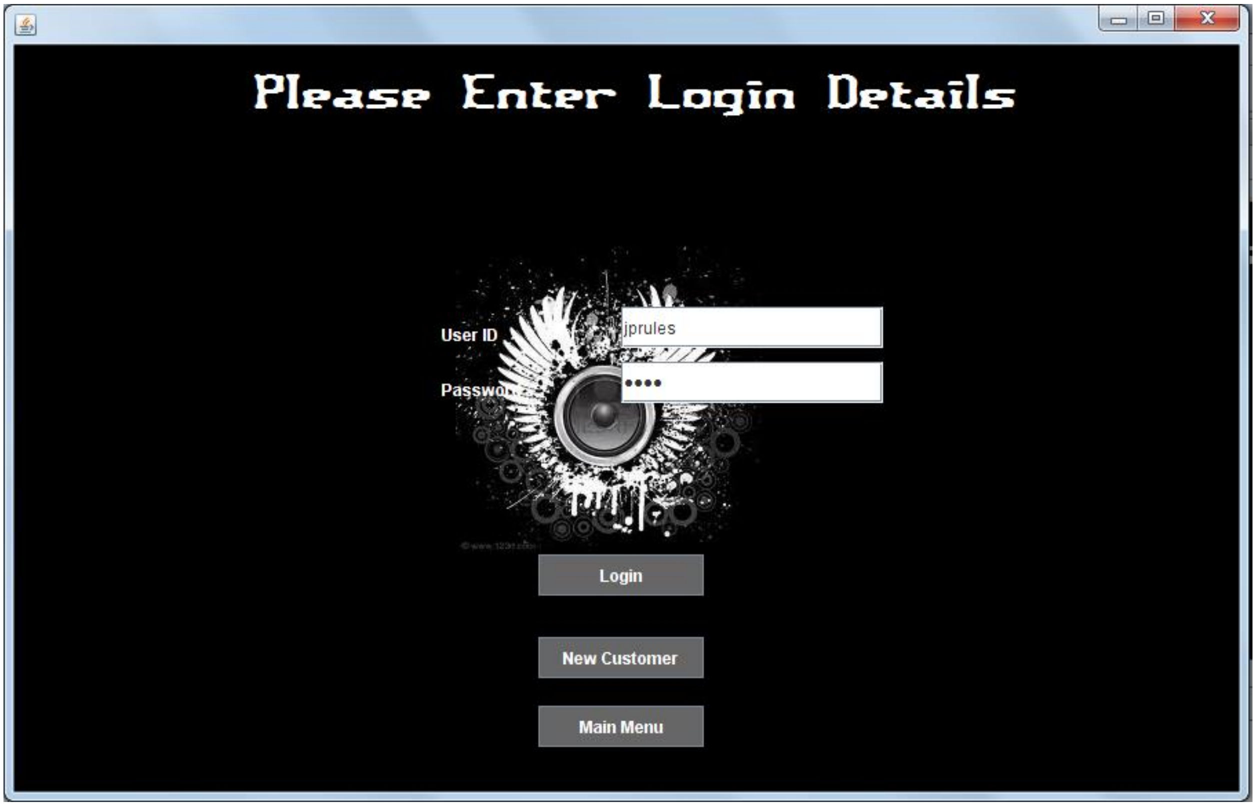


Welcome to Rock Rules Music Store

Please Select Your ID Type

```
import javax.swing.JOptionPane;

import javax.swing.JOptionPane.*;

private void SubmitBtnActionPerformed(java.awt.event.ActionEvent evt) {

    if(CustRB.isSelected()){

      CLogin c = new CLogin();

      c.setVisible(true);

      this.dispose();

    }

    else if(AdminRB.isSelected()){

      ALogin a = new ALogin();

      a.setVisible(true);

      this.dispose();

    }

    else{

      JOptionPane.showMessageDialog(null, "Select an ID type");

    }

  }
```

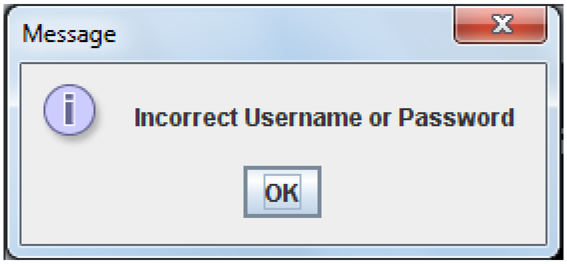
CLogin:



Please Enter Login Details

User ID: jprules

Password: ••••

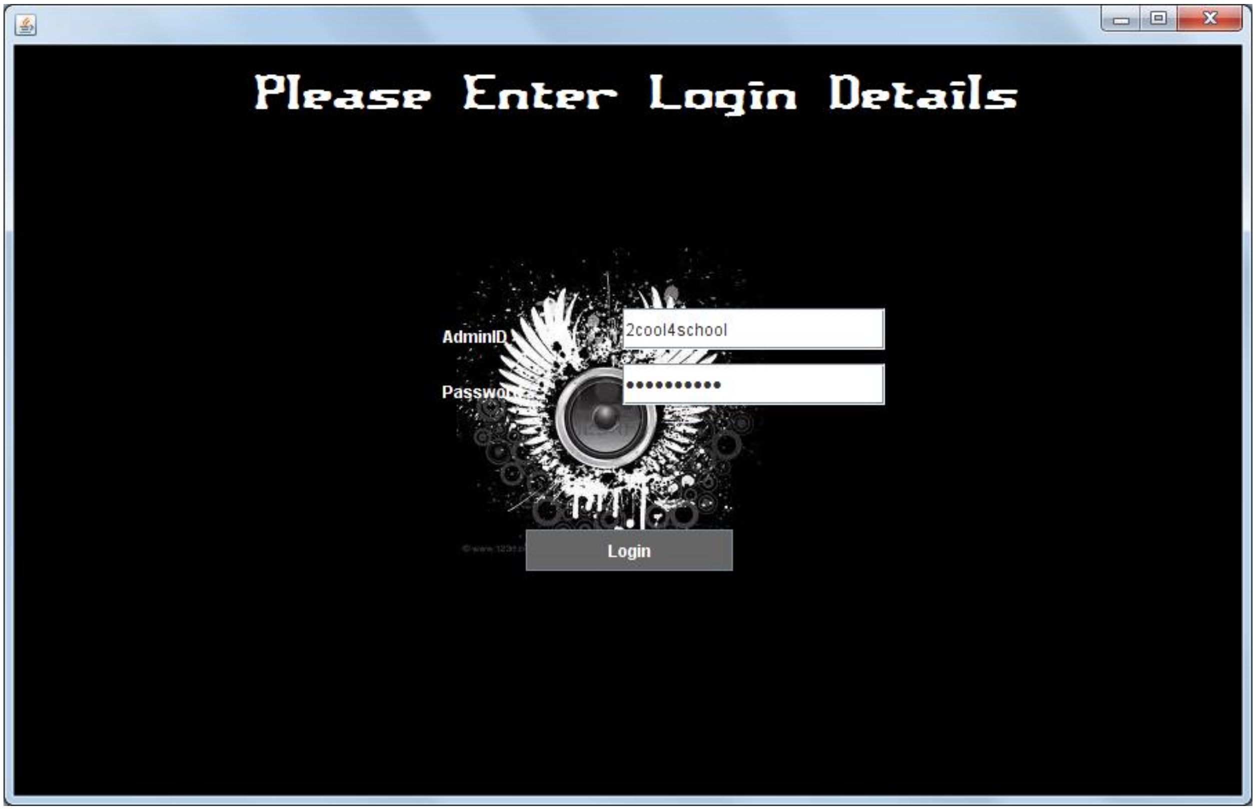


Message

Incorrect Username or Password

```
import javax.swing.*;

import java.sql.*;

private void LoginBtnActionPerformed(java.awt.event.ActionEvent evt) {

    String id = idTF.getText();

    String pwd = pwdTF.~~getText~~();

    try {

      Class.forName("java.sql.Driver");

      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

      Statement stmt = conn.createStatement();

      String sql = "Select * from Clogin where ID = '" + id + "';";

      ResultSet rs = stmt.executeQuery(sql);

      rs.next();

      String str = rs.getString("Password");

      if(str.equals(pwd)){

        CPortal c = new CPortal();

        c.setVisible(true);

        this.dispose();

        }

      else{

        JOptionPane.showMessageDialog(null, "Incorrect Username or Password");

      }

    }
catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Incorrect Username or Password");

    }

  }


  private void NewUserBtnActionPerformed(java.awt.event.ActionEvent evt) {

    NewUser n = new NewUser();

    n.setVisible(true);

    this.dispose();

  }


  private void MainMenuBtnActionPerformed(java.awt.event.ActionEvent evt) {

    Welcome w = new Welcome();

    w.setVisible(true);

    this.dispose();

  }
```

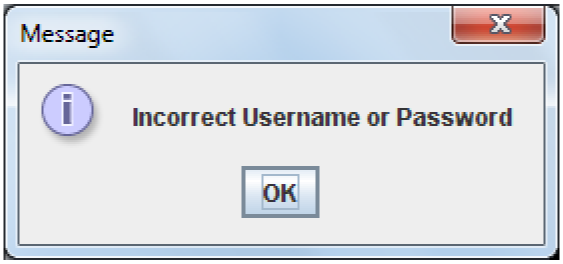
ALogin:



Please Enter Login Details

AdminID: 2cool4school

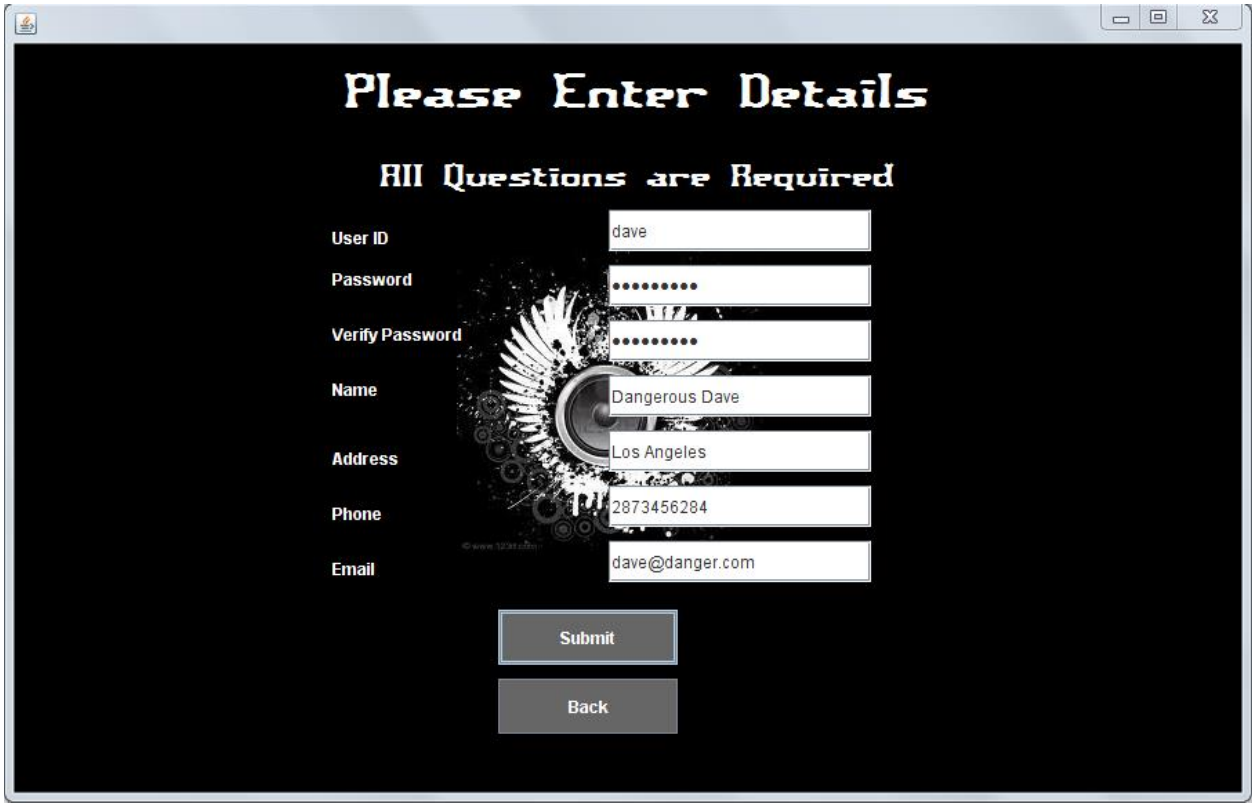
Password: •••••••••••



Incorrect Username or Password

```
import java.sql.*;

import javax.swing.JOptionPane;

private void LoginBtnActionPerformed(java.awt.event.ActionEvent evt) {

        String id = idTF.getText();

        String pwd = pwdTF.~~getText~~();

        try {

            Class.forName("java.sql.Driver");

            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root",
"0000");

            Statement stmt = conn.createStatement();

            String sql = "Select * from Alogin where AdminID = '" + id + "';";

            ResultSet rs = stmt.executeQuery(sql);

            rs.next();

            String str = rs.getString("Password");

            if(str.equals(pwd)){

                AInvoice a = new AInvoice();

                a.setVisible(true);

                this.dispose();

                }

            else{

                JOptionPane.showMessageDialog(null, "Incorrect Username or Password");

            }

        }

        catch (Exception e) { JOptionPane.showMessageDialog(null, "Incorrect Username or Password"); }

 }
```

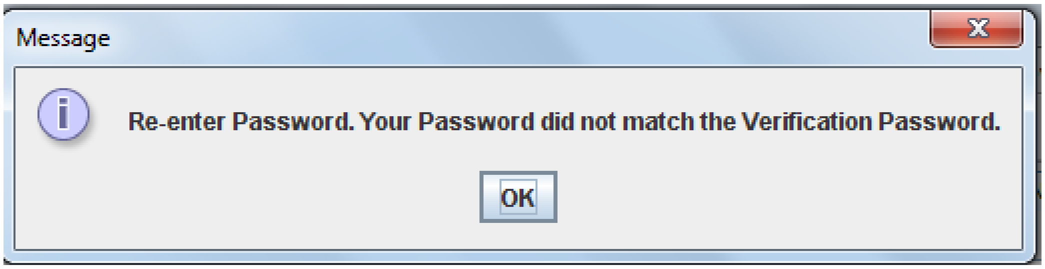
New User:



Please Enter Details

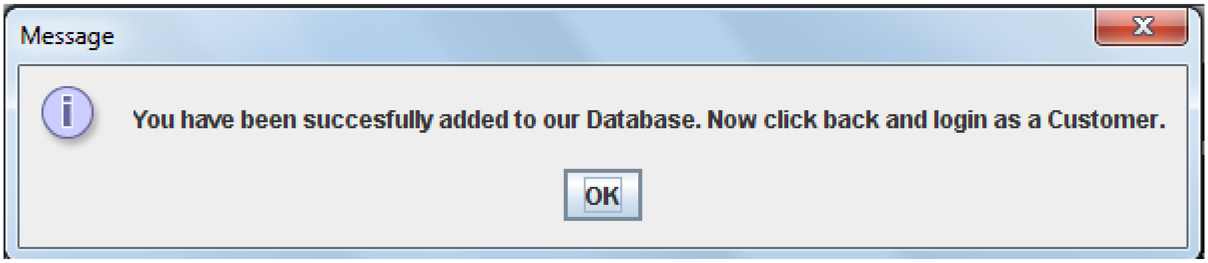
All Questions are Required

User ID	dave
Password	••••••••••
Verify Password	••••••••••
Name	Dangerous Dave
Address	Los Angeles
Phone	2873456284
Email	dave@danger.com



Message

Re-enter Password. Your Password did not match the Verification Password.



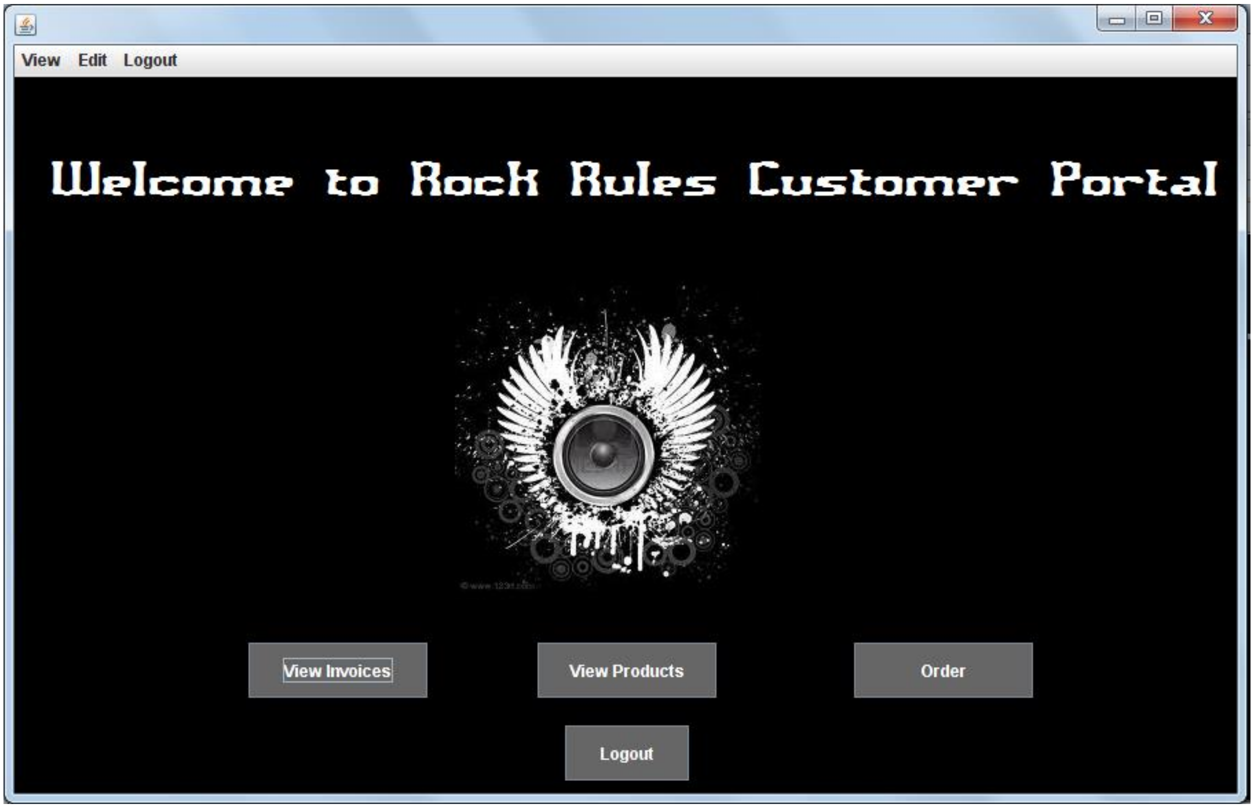
Message

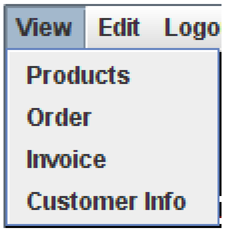
You have been succesfully added to our Database. Now click back and login as a Customer.

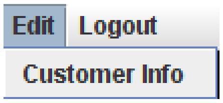
```
import java.sql.*;

import javax.swing.JOptionPane;

private void SubmitBtnActionPerformed(java.awt.event.ActionEvent evt) {

    String id = idTF.getText();

    String pwd = pwdTF.~~getText~~();

    String vpwd = vpwdTF.~~getText~~();

    String name = nameTF.getText();

    String addr = AddrTF.getText();

    String email = EmailTF.getText();

    long phone = Long.parseLong(PhoneTF.getText());

    if(pwd.equals(vpwd)){

      try {

      Class.forName("java.sql.Driver");

      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root",
"0000");

      Statement stmt = conn.createStatement();

        String sql1 = "Insert into CLogin values('" + id + "', '" + pwd + "');";
        stmt.executeUpdate(sql1);

      String sql2 = "Select * from Customer;”;

      ResultSet rs = stmt.executeQuery(sql2);

      rs.last();

      int i = rs.getInt("Cno");

      i++;

      String sql3 = "Insert into Customer values('" + i + "', '" + name + "', '" + addr + "', '" + phone + "', '"
+ email + "');";

      stmt.executeUpdate(sql3);
```

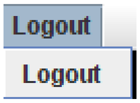
```
 JOptionPane.showMessageDialog(null, "You have been successfully added to our Database. Now click back and login as a Customer.");

    }

      catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database");

    }

    }

    else{

      JOptionPane.showMessageDialog(null, "Re-enter Password. Your Password did not match the Verification Password.");

    }

  }


  private void BackBtnActionPerformed(java.awt.event.ActionEvent evt) {

    Welcome w = new Welcome();

    w.setVisible(true);

    this.dispose();

  }
```

CPortal:


View
Edit
Logout
Welcome to Rock Rules Customer Portal
View Invoices
View Products
Order
Logout


View
Edit
Logo
Products
Order
Invoice
Customer Info


Edit
Logout
Customer Info

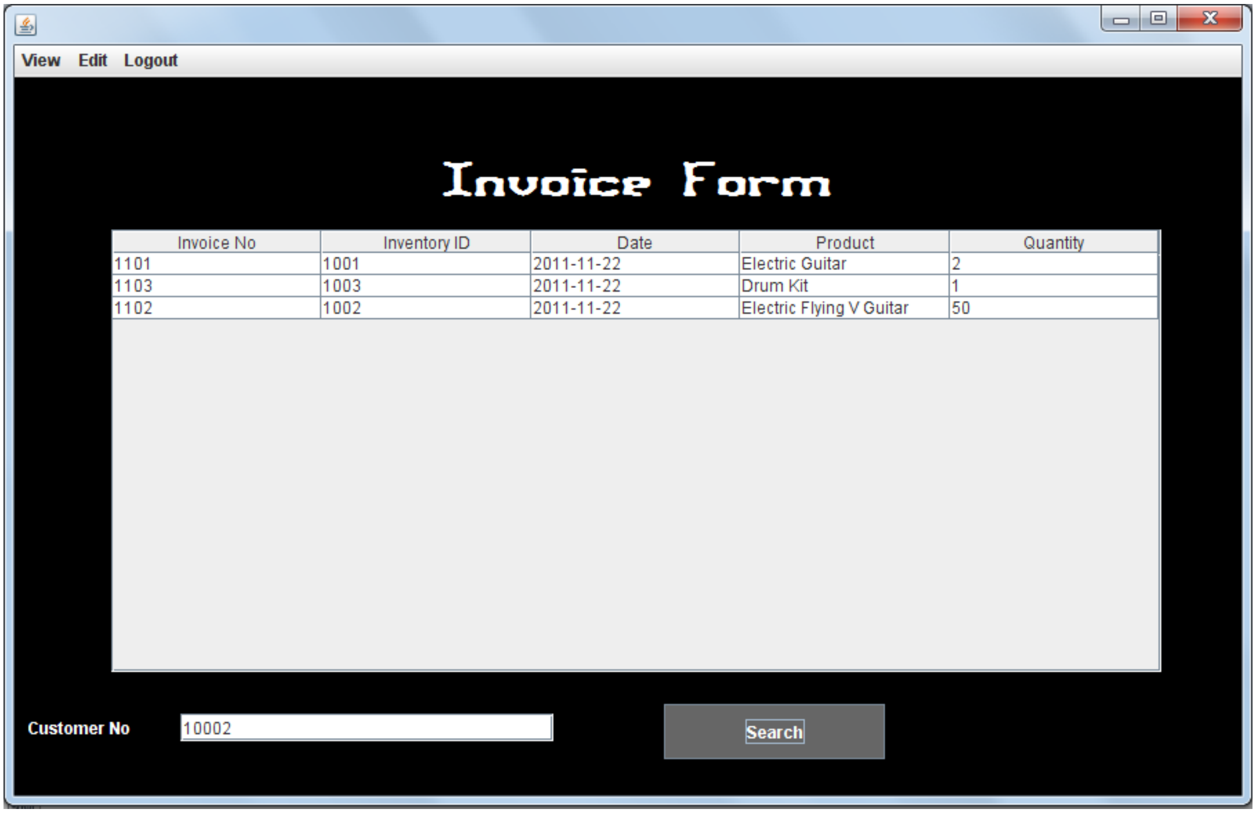

Logout
Logout

```
private void LogoutBtnActionPerformed(java.awt.event.ActionEvent evt) {

    Welcome w = new Welcome();

    w.setVisible(true);

    this.dispose();

  }


  private void InvoiceBtnActionPerformed(java.awt.event.ActionEvent evt) {

    CInvoice i = new CInvoice();

    i.setVisible(true);

    this.dispose();

  }


  private void ProductBtnActionPerformed(java.awt.event.ActionEvent evt) {

    CProducts p = new CProducts();

    p.setVisible(true);

    this.dispose();

  }


  private void OrderBtnActionPerformed(java.awt.event.ActionEvent evt) {

    COrder o = new COrder();

   o.setVisible(true);

   this.dispose();

  }
```

```
private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {
     CProducts p = new CProducts();
     p.setVisible(true);
     this.dispose();
  }

  private void OrderMItemActionPerformed(java.awt.event.ActionEvent evt) {
    COrder o = new COrder();
    o.setVisible(true);
    this.dispose();
  }

  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
     CInvoice i = new CInvoice();
     i.setVisible(true);
     this.dispose();
  }

  private void CustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
     CCustomerInfo c = new CCustomerInfo();
     c.setVisible(true);
     this.dispose();
  }
```

```
private void EditCustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {

     CCustomerInfo c = new CCustomerInfo();

     c.setVisible(true);

     this.dispose();

  }


  private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {

     Welcome w = new Welcome();

     w.setVisible(true);

     this.dispose();

  }
```

CInvoice:



Invoice No	Inventory ID	Date	Product	Quantity
1101	1001	2011-11-22	Electric Guitar	2
1103	1003	2011-11-22	Drum Kit	1
1102	1002	2011-11-22	Electric Flying V Guitar	50

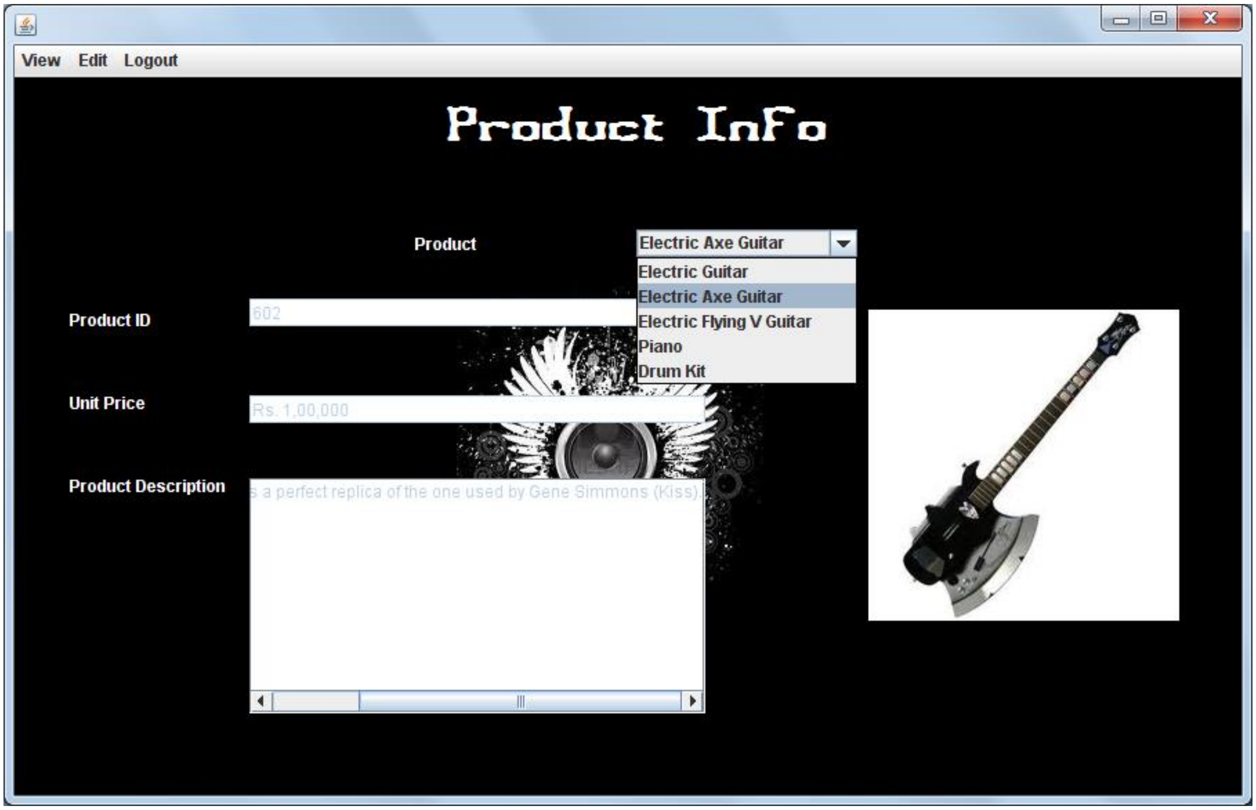
Customer No 10002

```
import javax.swing.JOptionPane;
import java.sql.*;
import javax.swing.table.DefaultTableModel;
private void SearchBtnActionPerformed(java.awt.event.ActionEvent evt) {
    int CustNo = Integer.parseInt(cnoTF.getText());
    try {
      Class.forName("java.sql.Driver");
      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root",
"0000");
      Statement stmt = conn.createStatement();
      String sql1 = "Select * from Invoice where Cno = '" + CustNo + "'";
      ResultSet rs1 = stmt.executeQuery(sql1);
      rs1.first();
      DefaultTableModel tb = (DefaultTableModel) InvoiceTbl.getModel();
      tb.addRow(new Object[]{rs1.getInt("invoice_no"), rs1.getInt("invent_id"), rs1.getDate("date"),
rs1.getString("Product") , rs1.getInt("quantity")});
    }
    catch (Exception e) {
      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
  }
  private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {
    CProducts p = new CProducts();
    p.setVisible(true);
    this.dispose();
  }
```

```
private void OrderMItemActionPerformed(java.awt.event.ActionEvent evt) {
    COrder o = new COrder();
    o.setVisible(true);
    this.dispose();
  }

  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
    CInvoice i = new CInvoice();
    i.setVisible(true);
    this.dispose();
  }

  private void CustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
    CCustomerInfo c = new CCustomerInfo();
    c.setVisible(true);
    this.dispose();
  }

  private void EditCustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
    CCustomerInfo c = new CCustomerInfo();
    c.setVisible(true);
    this.dispose();
  }
```

```
private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {

    Welcome w = new Welcome();

    w.setVisible(true);

    this.dispose();

  }
```

CProducts:



Product Info

Product: Electric Axe Guitar

Product ID: 602

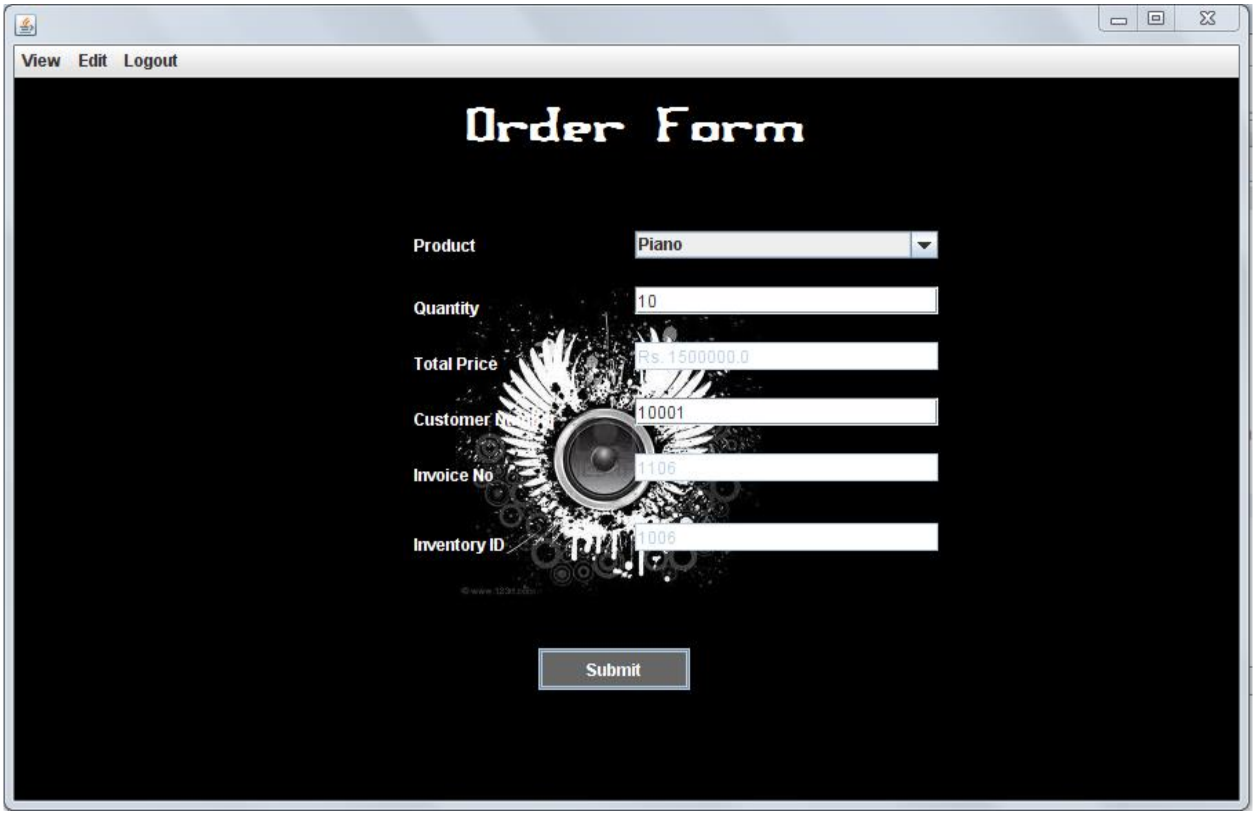
Unit Price: Rs. 1,00,000

Product Description: s a perfect replica of the one used by Gene Simmons (Kiss).

```
public CProducts() {
    initComponents();
    PIDTF.~~enable~~(false);
    UPTF.~~enable~~(false);
    PDTA.~~enable~~(false);
  }
private void ProductCBItemStateChanged(java.awt.event.ItemEvent evt) {
     String product = ProductCB.getSelectedItem().toString();
     if(product.equals("Electric Guitar")){
       PIDTF.setText("601");
       UPTF.setText("Rs. 12,500");
       PDTA.setText("This Ibanez guitar is a great buy for its price.");
       PicLbl.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Guitar.jpg")));
     }
     else if(product.equals("Electric Axe Guitar")){
       PIDTF.setText("602");
       UPTF.setText("Rs. 1,00,000");
       PDTA.setText("This Axe guitar is a perfect replica of the one used by Gene Simmons (Kiss).");
       PicLbl.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Axe Guitar.jpg")));
     }
     else if(product.equals("Electric Flying V Guitar")){
       PIDTF.setText("603");
       UPTF.setText("Rs. 25,499");
       PDTA.setText("This Flying V will make you feel like Dave Mustaine (Megadeth).");
       PicLbl.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Flying V Guitar.jpg")));        }
  else if(product.equals("Piano")){

     PIDTF.setText("801");

     UPTF.setText("Rs. 1,50,000");

     PDTA.setText("A Grand Piano with all you could wish for.");

     PicLbl.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Piano.jpg")));

   }

   else if(product.equals("Drum Kit")){

     PIDTF.setText("901");

     UPTF.setText("Rs. 2,50,000");

     PDTA.setText("Dave Lombardo (Slayer) might envy this kit.");

     PicLbl.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Drums.jpg")));

   }

}


private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {

  CProducts p = new CProducts();

  p.setVisible(true);

  this.dispose();

}


private void OrderMItemActionPerformed(java.awt.event.ActionEvent evt) {

  COrder o = new COrder();

 o.setVisible(true);

 this.dispose();

}
```

```
private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {

   CInvoice i = new CInvoice();

   i.setVisible(true);

   this.dispose();

}


private void CustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {

   CCustomerInfo c = new CCustomerInfo();

   c.setVisible(true);

   this.dispose();

}


private void EditCustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {

   CCustomerInfo c = new CCustomerInfo();

   c.setVisible(true);

   this.dispose();

}


private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {

   Welcome w = new Welcome();

   w.setVisible(true);

   this.dispose();

}
```

COrder:



Order Form

Product: Piano

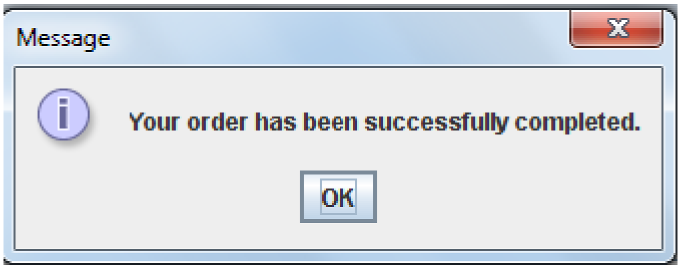
Quantity: 10

Total Price: Rs. 1500000.0

Customer No: 10001

Invoice No: 1106

Inventory ID: 1006



Message

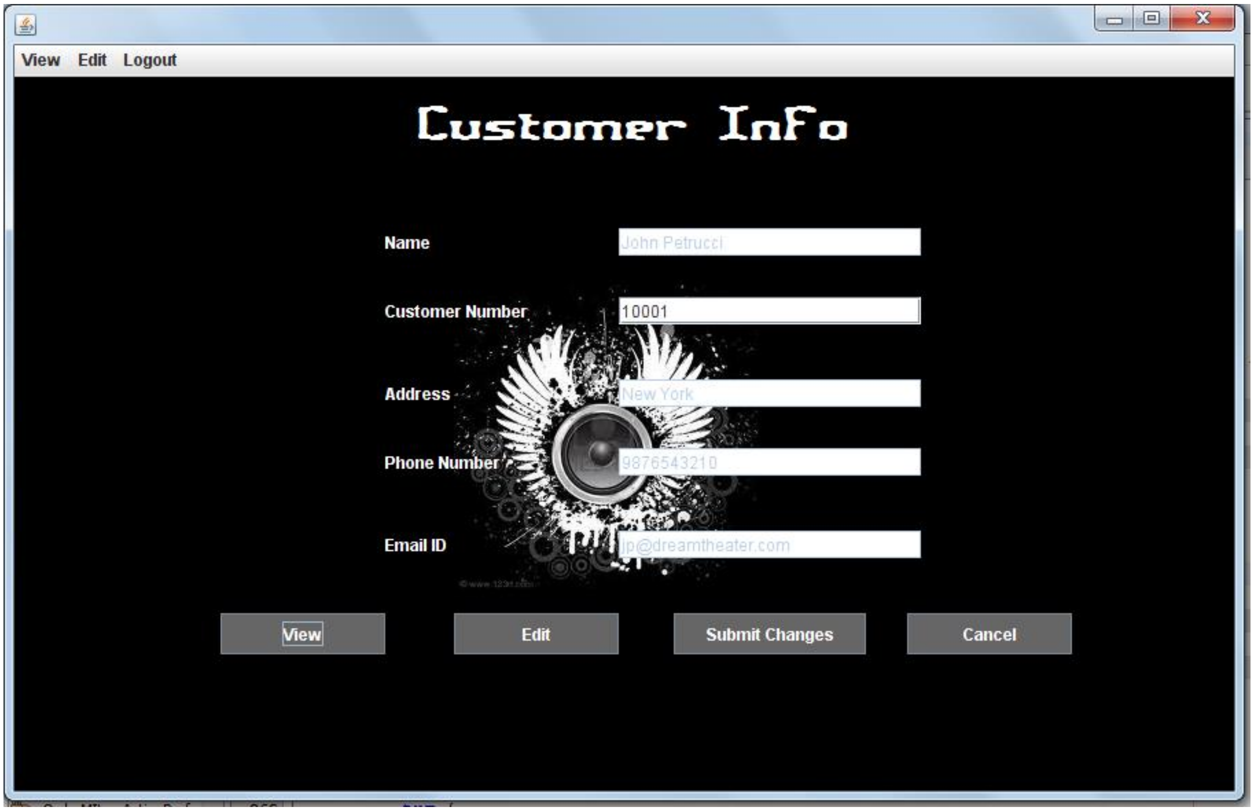
Your order has been successfully completed.

```
import javax.swing.*;

import java.sql.*;

import java.util.*;

public COrder() {

    initComponents();

    PriceTF.~~enable~~(false);

    InvoiceTF.~~enable~~(false);

    InventTF.~~enable~~(false);

  }

private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {

    CProducts p = new CProducts();

    p.setVisible(true);

    this.dispose();

  }

  private void OrderMItemActionPerformed(java.awt.event.ActionEvent evt) {

   COrder o = new COrder();

   o.setVisible(true);

   this.dispose();

  }

  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {

    CInvoice i = new CInvoice();

    i.setVisible(true);

    this.dispose();

  }
```

```
    private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {

        Welcome w = new Welcome();

        w.setVisible(true);

        this.dispose();

    }


    private void SubmitBtnActionPerformed(java.awt.event.ActionEvent evt) {

        String product = ProductCB.getSelectedItem().toString();

        int qty = Integer.parseInt(qtyTF.getText());

        int Cno = Integer.parseInt(CnoTF.getText());

        double totalprice = price(product, qty);

        Calendar cal = Calendar.getInstance();

        String date = cal.get(Calendar.YEAR) + "-" + (cal.get(Calendar.MONTH) + 1) + "-" + cal.get(Calendar.DATE);

        PriceTF.setText("Rs. " + totalprice);

        try {

            Class.forName("java.sql.Driver");

            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

            Statement stmt = conn.createStatement();

            String sql1 = "Select invoice_no, invent_id from Invoice;";

            ResultSet rs = stmt.executeQuery(sql1);

            rs.last();

            int i = rs.getInt("Invoice_no");

            i++;

            InvoiceTF.setText("" + i);
    int j = rs.getInt("Invent_id");
        j++;
        InventTF.setText("" + j);
        String sql2 = "Insert into Invoice values('" + i + "', '" + j + "', '" + Cno + "', '" + date + "', '" + qty + "', 
'" + product + "', '" + totalprice + "');";
        stmt.executeUpdate(sql2);
        JOptionPane.showMessageDialog(null, "Your order has been successfully completed.");
      }
        catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error Connecting to Database");
      }
    }

    private void CustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
      CCustomerInfo c = new CCustomerInfo();
      c.setVisible(true);
      this.dispose();
    }

    private void EditCustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
      CCustomerInfo c = new CCustomerInfo();
      c.setVisible(true);
      this.dispose();
    }
```

```
double price(String product,int qty){
    double uprice = 0.0;
    double tprice = 0.0;
    if(product.equals("Electric Guitar")){
        uprice = 12500;
        tprice = uprice * qty;
     }
     else if(product.equals("Electric Axe Guitar")){
        uprice = 100000;
        tprice = uprice * qty;
     }
     else if(product.equals("Electric Flying V Guitar")){
        uprice = 25499;
        tprice = uprice * qty;
     }
     else if(product.equals("Piano")){
        uprice = 150000;
        tprice = uprice * qty;
     }
     else if(product.equals("Drum Kit")){
        uprice = 250000;
        tprice = uprice * qty;
     }
    return tprice;
 }
```

CCustomerInfo:



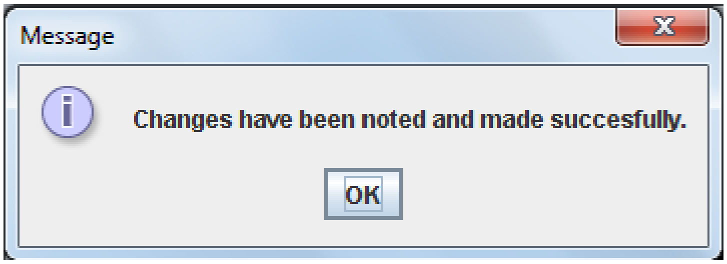
Name: John Petrucci

Customer Number: 10001

Address: New York

Phone Number: 9876543210

Email ID: jp@dreamtheater.com



Changes have been noted and made succesfully.

```
import java.sql.*;

import javax.swing.JOptionPane;

public class **CCustomerInfo** extends javax.swing.JFrame {

        int cno;

        String addr;

        String email;

        long phone;

        String name;

    public CCustomerInfo() {

        initComponents();

        NameTF.~~enable~~(false);

        PhoneTF.~~enable~~(false);

        AddressTF.~~enable~~(false);

        EmailID.~~enable~~(false);

        SubmitBtn.~~enable~~(false);

    }
```

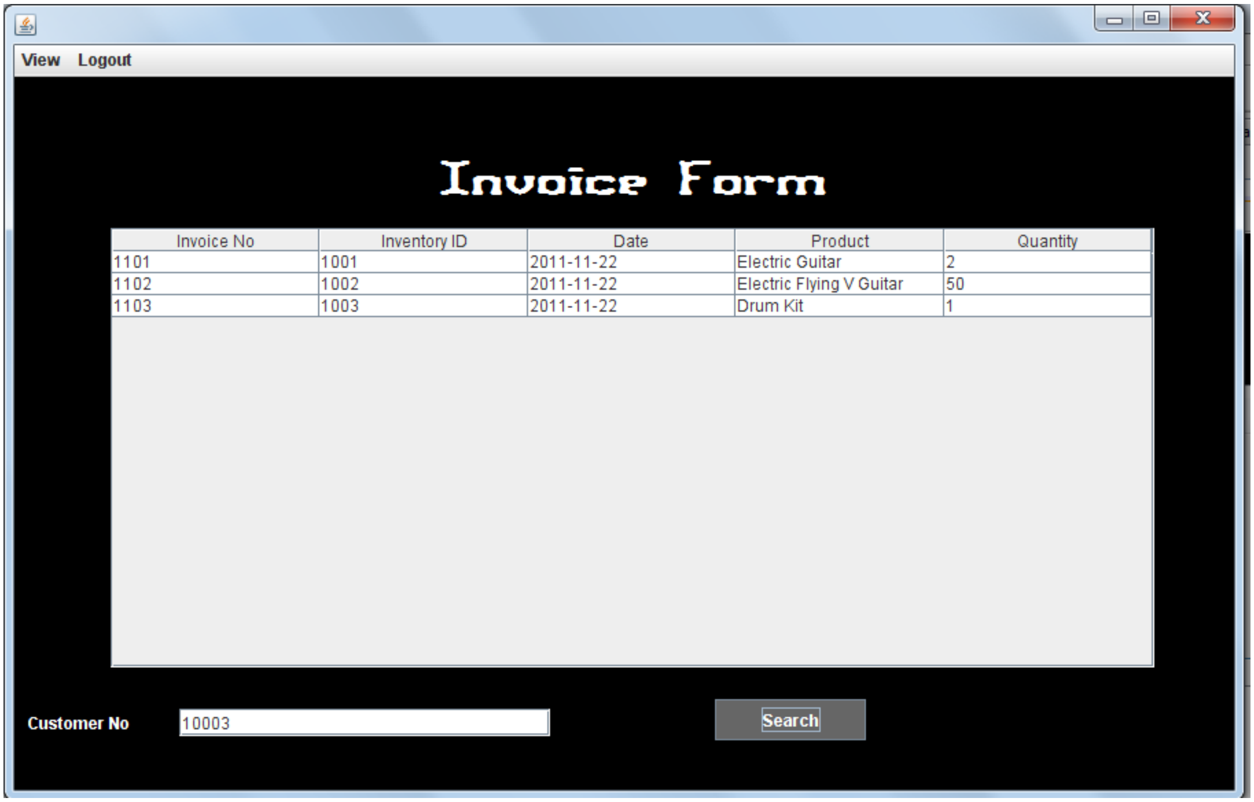
```
private void ViewBtnActionPerformed(java.awt.event.ActionEvent evt) {

    cno = Integer.parseInt(CnoTF.getText());

    try {

      Class.forName("java.sql.Driver");

      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

      Statement stmt = conn.createStatement();

      String sql1 = "Select * from Customer where cno = '" + cno + "';";

      ResultSet rs = stmt.executeQuery(sql1);

      rs.first();

      name = rs.getString("Cname");

      addr = rs.getString("Caddr");

      email = rs.getString("Cemail");

      phone = rs.getLong("cphone");

      NameTF.setText("" + name);

      PhoneTF.setText("" + phone);

      AddressTF.setText("" + addr);

      EmailID.setText("" + email);

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "You have typed your Customer Number wrong. Please type the name you did while registering.");

    }

  }
```

```
private void EditBtnActionPerformed(java.awt.event.ActionEvent evt) {

        NameTF.~~enable~~();

        PhoneTF.~~enable~~();

        AddressTF.~~enable~~();

        EmailID.~~enable~~();

        SubmitBtn.~~enable~~();

        NameTF.requestFocus();

    }

    private void SubmitBtnActionPerformed(java.awt.event.ActionEvent evt) {

        String n = NameTF.getText();

        int cn = Integer.parseInt(CnoTF.getText());

        long ph = Long.parseLong(PhoneTF.getText());

        String ad = AddressTF.getText();

        String eid = EmailID.getText();

        try {

            Class.forName("java.sql.Driver");

            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

            Statement stmt = conn.createStatement();

            String sql = "Update Customer set Cname = '" + n + "', cno = '" + cn + "', caddr = '" + ad + "', cphone = '" + ph + "', cemail = '" + eid + "' where cname = '" + name + "';";

            stmt.executeUpdate(sql);

            JOptionPane.showMessageDialog(null, "Changes have been noted and made successfully.");

        }

        catch (Exception e) {

            JOptionPane.showMessageDialog(null, "Error. Please make sure you have entered all fields correctly.");       }    }
```

```
  }

  private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {
      CProducts p = new CProducts();
      p.setVisible(true);
      this.dispose();
  }
  private void OrderMItemActionPerformed(java.awt.event.ActionEvent evt) {
     COrder o = new COrder();
     o.setVisible(true);
     this.dispose();
  }
  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
      CInvoice i = new CInvoice();
      i.setVisible(true);
      this.dispose();
  }
private void CustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
      CCustomerInfo c = new CCustomerInfo();
      c.setVisible(true);
      this.dispose();
  }
  private void EditCustInfoMItemActionPerformed(java.awt.event.ActionEvent evt) {
      CCustomerInfo c = new CCustomerInfo();
      c.setVisible(true);
      this.dispose();
  }
```

```
private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {
    Welcome w = new Welcome();
    w.setVisible(true);
    this.dispose();
}

private void CancelBtnActionPerformed(java.awt.event.ActionEvent evt) {
    CPortal c = new CPortal();
    c.setVisible(true);
    this.dispose();
}
```

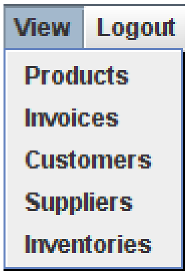
AInvoice:

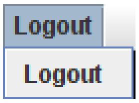


Invoice Form

Invoice No	Inventory ID	Date	Product	Quantity
1101	1001	2011-11-22	Electric Guitar	2
1102	1002	2011-11-22	Electric Flying V Guitar	50
1103	1003	2011-11-22	Drum Kit	1

Customer No: 10003

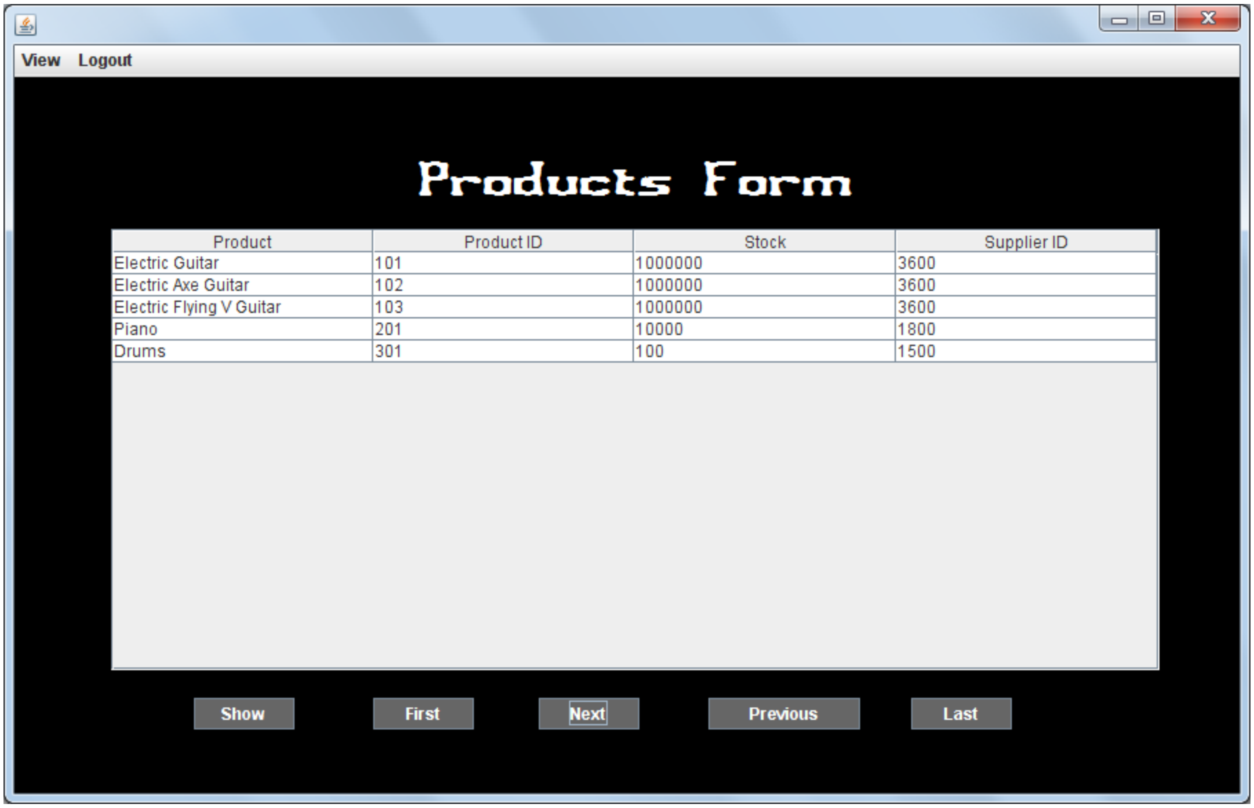




```
import java.sql.*;

import javax.swing.JOptionPane;

import javax.swing.table.*;

private void SearchBtnActionPerformed(java.awt.event.ActionEvent evt) {

        int CustNo = Integer.parseInt(cnoTF.getText());

        try {

            Class.forName("java.sql.Driver");

            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root",
"0000");

            Statement stmt = conn.createStatement();

            String sql1 = "Select * from Invoice where Cno = '" + CustNo + "';";

            ResultSet rs1 = stmt.executeQuery(sql1);

            rs1.first();

            DefaultTableModel tb = (DefaultTableModel) InvoiceTbl.getModel();

            tb.addRow(new Object[]{rs1.getInt("invoice_no"), rs1.getInt("invent_id"), rs1.getDate("date"),
rs1.getString("Product") , rs1.getInt("quantity")});

        }

        catch (Exception e) {

            JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

        }

    }

 private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {

        AProducts p = new AProducts();

        p.setVisible(true);

        this.dispose();

    }
```

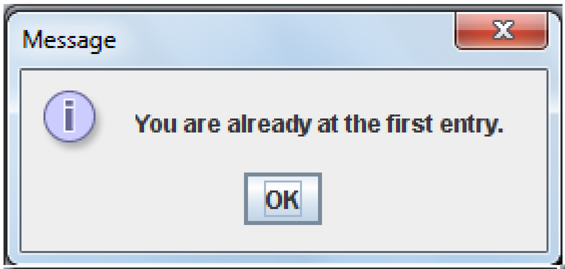
```
  }

private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInvoice i = new AInvoice();
    i.setVisible(true);
    this.dispose();
  }
  private void CustMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ACustomerInfo c = new ACustomerInfo();
    c.setVisible(true);
    this.dispose();
  }
  private void SupplierMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ASuppliers s = new ASuppliers();
    s.setVisible(true);
    this.dispose();
  }
  private void InventMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInventory i = new AInventory();
    i.setVisible(true);
    this.dispose();
  }
 private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {
    Welcome w = new Welcome();
    w.setVisible(true);
    this.dispose();
  }
```

AProducts:

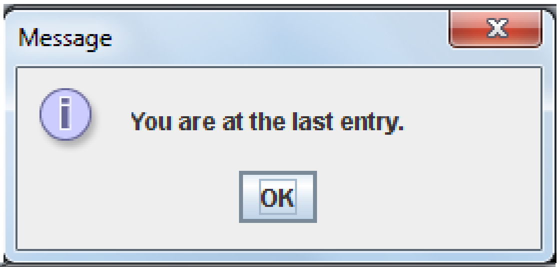


Products Form

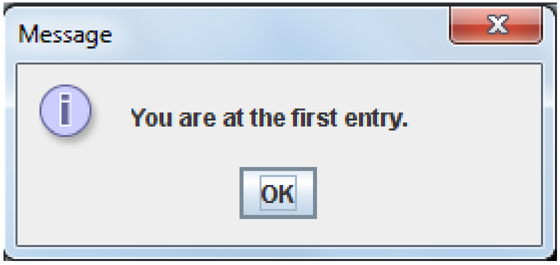
Product	Product ID	Stock	Supplier ID
Electric Guitar	101	1000000	3600
Electric Axe Guitar	102	1000000	3600
Electric Flying V Guitar	103	1000000	3600
Piano	201	10000	1800
Drums	301	100	1500


Message

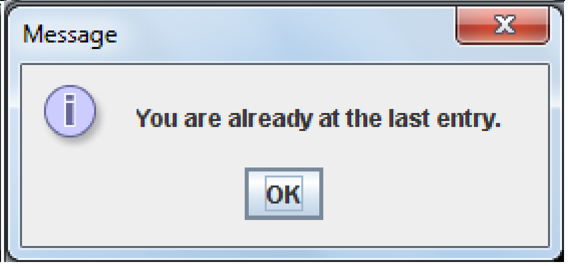
You are already at the first entry.


Message

You are at the last entry.


Message

You are at the first entry.


Message

You are already at the last entry.

```
import java.sql.*;

import javax.swing.JOptionPane;

import javax.swing.table.*;

public class AProducts extends javax.swing.JFrame {

    Connection conn;

    Statement stmt;

    ResultSet rs1;

  public AProducts() {

    initComponents();

  }

private void ShowBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      Class.forName("java.sql.Driver");

      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

      stmt = conn.createStatement();

      String sql1 = "Select * from product;";

      rs1 = stmt.executeQuery(sql1);

      rs1.first();

      DefaultTableModel tb = (DefaultTableModel) ProductTbl.getModel();

      tb.addRow(new Object[]{rs1.getString("pname"), rs1.getInt("pid"), rs1.getLong("stock"),
rs1.getInt("sid")});

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
```

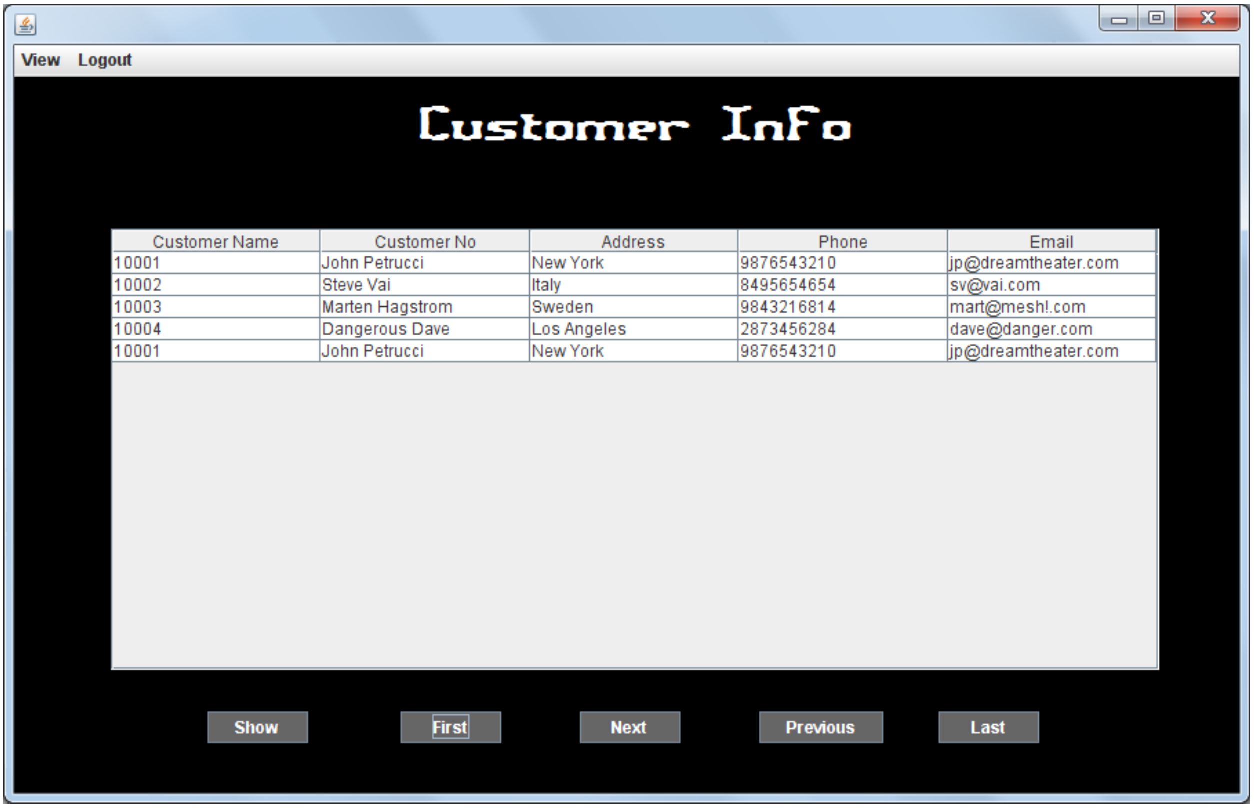
```
private void FirstBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
      if(rs1.isFirst()){
        JOptionPane.showMessageDialog(null, "You are already at the first entry.");
      }
      else{
        rs1.first();
        DefaultTableModel tb = (DefaultTableModel) ProductTbl.getModel();
        tb.addRow(new Object[]{rs1.getString("pname"), rs1.getInt("pid"), rs1.getLong("stock"), rs1.getInt("sid")});
      }
    }
    catch (Exception e) {
      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
  }
```

```
 private void NextBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      if(rs1.isLast()){

        JOptionPane.showMessageDialog(null, "You are at the last entry.");

      }

      else{

        rs1.next();

        DefaultTableModel tb = (DefaultTableModel) ProductTbl.getModel();

        tb.addRow(new Object[]{rs1.getString("pname"), rs1.getInt("pid"), rs1.getLong("stock"),
rs1.getInt("sid")});

      }

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
```

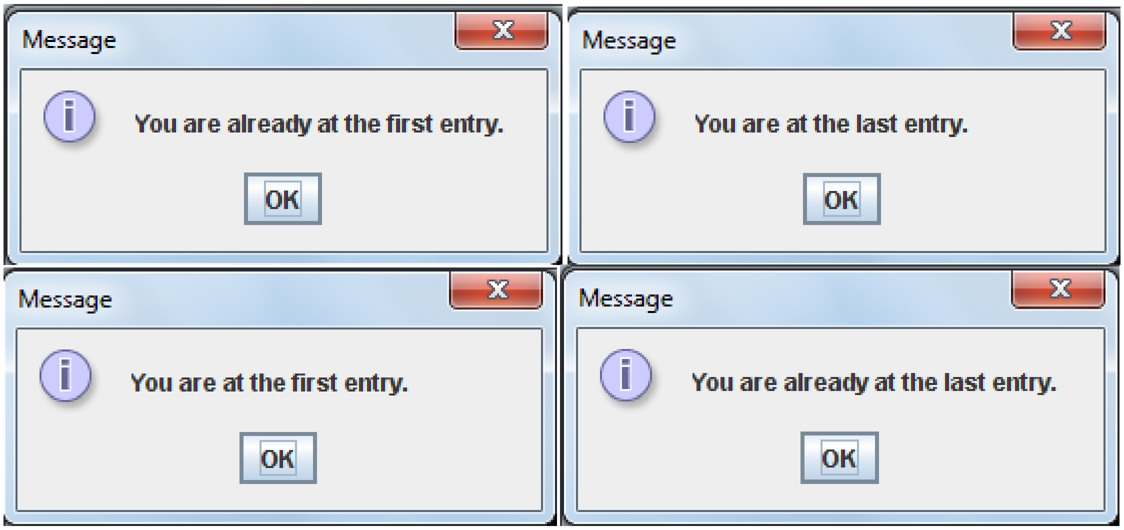
```
private void PreviousBtnActionPerformed(java.awt.event.ActionEvent evt) {

        try {

            if(rs1.isFirst()){

                JOptionPane.showMessageDialog(null, "You are at the first entry.");

            }

            else{

                rs1.previous();

                DefaultTableModel tb = (DefaultTableModel) ProductTbl.getModel();

                tb.addRow(new Object[]{rs1.getString("pname"), rs1.getInt("pid"), rs1.getLong("stock"),
rs1.getInt("sid")});

            }

        }

        catch (Exception e) {

            JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

        }

    }
    }

private void LastBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
      if(rs1.isLast()){
        JOptionPane.showMessageDialog(null, "You are already at the last entry.");
      }
      else{
        rs1.last();
        DefaultTableModel tb = (DefaultTableModel) ProductTbl.getModel();
        tb.addRow(new Object[]{rs1.getString("pname"), rs1.getInt("pid"), rs1.getLong("stock"),
rs1.getInt("sid")});
      }
    }
    catch (Exception e) {
      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
  }
  private void ProductMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AProducts p = new AProducts();
    p.setVisible(true);
    this.dispose();
  }
  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInvoice i = new AInvoice();
    i.setVisible(true);
    this.dispose();
  }
```

```
private void CustMItemActionPerformed(java.awt.event.ActionEvent evt) {

        ACustomerInfo c = new ACustomerInfo();

        c.setVisible(true);

        this.dispose();

    }


    private void SuppMItemActionPerformed(java.awt.event.ActionEvent evt) {

        ASuppliers s = new ASuppliers();

        s.setVisible(true);

        this.dispose();

    }


    private void InventMItemActionPerformed(java.awt.event.ActionEvent evt) {

        AInventory i = new AInventory();

        i.setVisible(true);

        this.dispose();

    }


    private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {

        Welcome w = new Welcome();

        w.setVisible(true);

        this.dispose();

    }
```

ACustomerInfo:



Customer Name	Customer No	Address	Phone	Email
10001	John Petrucci	New York	9876543210	jp@dreamtheater.com
10002	Steve Vai	Italy	8495654654	sv@vai.com
10003	Marten Hagstrom	Sweden	9843216814	mart@mesh!.com
10004	Dangerous Dave	Los Angeles	2873456284	dave@danger.com
10001	John Petrucci	New York	9876543210	jp@dreamtheater.com



You are already at the first entry.

You are at the last entry.

You are at the first entry.

You are already at the last entry.

```
import java.sql.*;

import javax.swing.JOptionPane;

import javax.swing.table.*;

public class ACustomerInfo extends javax.swing.JFrame {

    Connection conn;

    Statement stmt;

    ResultSet rs;

  public ACustomerInfo() {

    initComponents();

  }

private void ShowBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      Class.forName("java.sql.Driver");

      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

      stmt = conn.createStatement();

      String sql1 = "Select * from customer;";

      rs = stmt.executeQuery(sql1);

      rs.first();

      DefaultTableModel tb = (DefaultTableModel) CustTbl.getModel();

      tb.addRow(new Object[]{rs.getInt("cno"), rs.getString("cname"), rs.getString("caddr"),
rs.getLong("cphone"), rs.getString("cemail")});

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
```

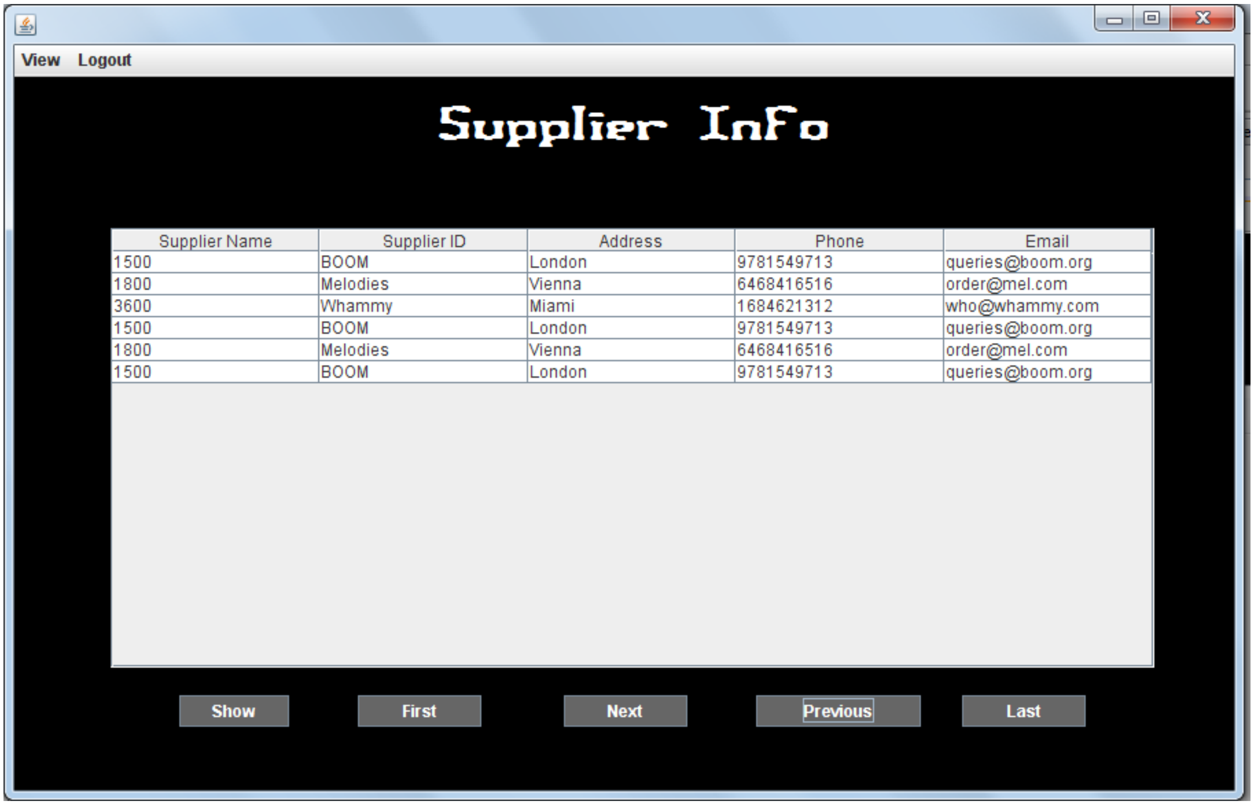
```
private void FirstBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
      if(rs.isFirst()){
        JOptionPane.showMessageDialog(null, "You are already at the first entry.");
      }
      else{
        rs.first();
        DefaultTableModel tb = (DefaultTableModel) CustTbl.getModel();
        tb.addRow(new Object[]{rs.getInt("cno"), rs.getString("cname"), rs.getString("caddr"),
rs.getLong("cphone"), rs.getString("cemail")});
      }
    }
    catch (Exception e) {
      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
  }
```

```
    private void NextBtnActionPerformed(java.awt.event.ActionEvent evt) {

        try {

            if(rs.isLast()){

                JOptionPane.showMessageDialog(null, "You are at the last entry.");

            }

            else{

                rs.next();

                DefaultTableModel tb = (DefaultTableModel) CustTbl.getModel();

                tb.addRow(new Object[]{rs.getInt("cno"), rs.getString("cname"), rs.getString("caddr"),
rs.getLong("cphone"), rs.getString("cemail")});

            }

        }

        catch (Exception e) {

            JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

        }

    }
    private void PreviousBtnActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            if(rs.isFirst()){
                JOptionPane.showMessageDialog(null, "You are at the first entry.");
            }
            else{
                rs.previous();
                DefaultTableModel tb = (DefaultTableModel) CustTbl.getModel();
                tb.addRow(new Object[]{rs.getInt("cno"), rs.getString("cname"), rs.getString("caddr"),
rs.getLong("cphone"), rs.getString("cemail")});
            }
        }
        catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
        }
    }
```

```
      }
  private void LastBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
      if(rs.isLast()){
        JOptionPane.showMessageDialog(null, "You are already at the last entry.");
      }
      else{
        rs.last();
        DefaultTableModel tb = (DefaultTableModel) CustTbl.getModel();
        tb.addRow(new Object[]{rs.getInt("cno"), rs.getString("cname"), rs.getString("caddr"),
rs.getLong("cphone"), rs.getString("cemail")});
      }
    }
    catch (Exception e) {
      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
  }
  private void ProductsMitemActionPerformed(java.awt.event.ActionEvent evt) {
    AProducts p = new AProducts();
    p.setVisible(true);
    this.dispose();
  }
  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInvoice i = new AInvoice();
    i.setVisible(true);
    this.dispose();
  }
```

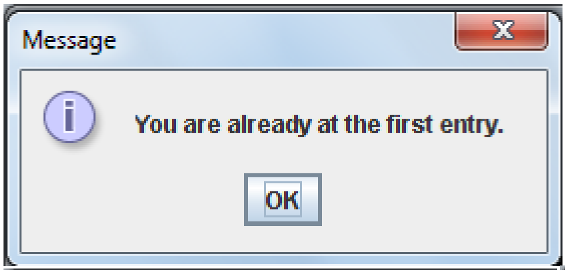
```java
private void CustMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ACustomerInfo c = new ACustomerInfo();
    c.setVisible(true);
    this.dispose();
  }
  private void SuppMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ASuppliers s = new ASuppliers();
    s.setVisible(true);
    this.dispose();
  }
  private void InventMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInventory i = new AInventory();
    i.setVisible(true);
    this.dispose();
  }
  private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {
    Welcome w = new Welcome();
    w.setVisible(true);
    this.dispose();
  }
```

ASuppliers:

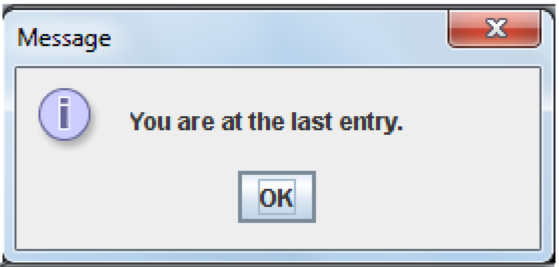


Supplier Info

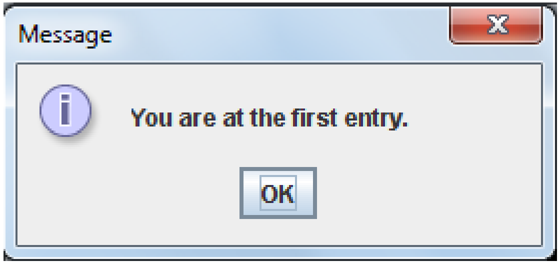
Supplier Name	Supplier ID	Address	Phone	Email
1500	BOOM	London	9781549713	queries@boom.org
1800	Melodies	Vienna	6468416516	order@mel.com
3600	Whammy	Miami	1684621312	who@whammy.com
1500	BOOM	London	9781549713	queries@boom.org
1800	Melodies	Vienna	6468416516	order@mel.com
1500	BOOM	London	9781549713	queries@boom.org



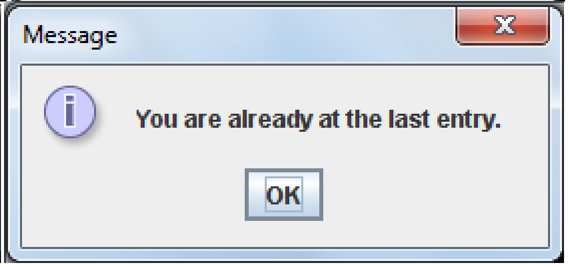
You are already at the first entry.



You are at the last entry.



You are at the first entry.



You are already at the last entry.

```
    }

import java.sql.*;

import javax.swing.JOptionPane;

import javax.swing.table.*;

public class ASuppliers extends javax.swing.JFrame {

        Connection conn;

        Statement stmt;

        ResultSet rs;

    public ASuppliers() {

        initComponents();

    }

private void ProductsMitemActionPerformed(java.awt.event.ActionEvent evt) {

        AProducts p = new AProducts();

        p.setVisible(true);

        this.dispose();

    }

    private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {

        AInvoice i = new AInvoice();

        i.setVisible(true);

        this.dispose();

    }

    private void CustMItemActionPerformed(java.awt.event.ActionEvent evt) {

        ACustomerInfo c = new ACustomerInfo();

        c.setVisible(true);

        this.dispose();

    }
```

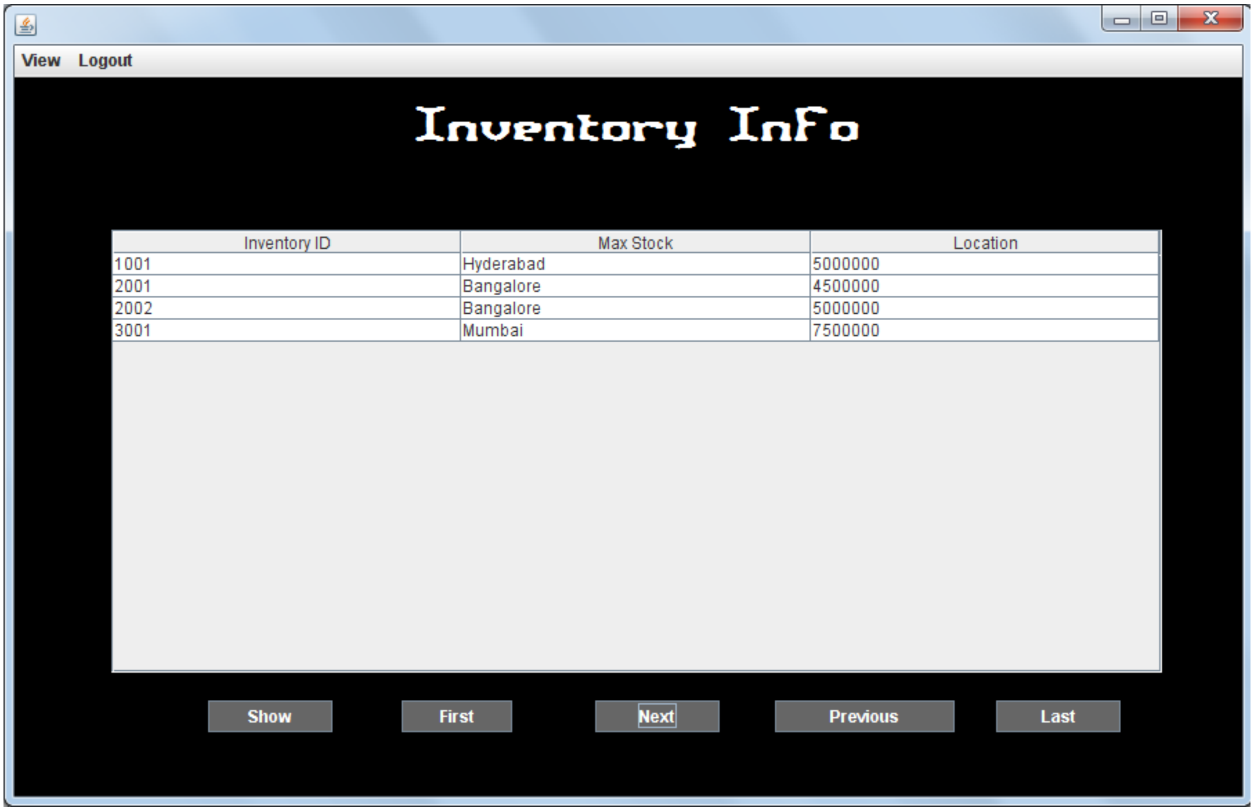
```
private void SuppMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ASuppliers s = new ASuppliers();
    s.setVisible(true);
    this.dispose();
  }

  private void InventMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInventory i = new AInventory();
    i.setVisible(true);
    this.dispose();
  }

  private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {
    Welcome w = new Welcome();
    w.setVisible(true);
    this.dispose();
  }
```

```
private void ShowBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      Class.forName("java.sql.Driver");

      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

      stmt = conn.createStatement();

      String sql1 = "Select * from supplier;";

      rs = stmt.executeQuery(sql1);

      rs.first();

      DefaultTableModel tb = (DefaultTableModel) SuppTbl.getModel();

      tb.addRow(new Object[]{rs.getInt("sid"), rs.getString("sname"), rs.getString("saddr"),
rs.getLong("sphone"), rs.getString("semail")});

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
  private void FirstBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      if(rs.isFirst()){

        JOptionPane.showMessageDialog(null, "You are already at the first entry.");

      }

      else{

        rs.first();

        DefaultTableModel tb = (DefaultTableModel) SuppTbl.getModel();

        tb.addRow(new Object[]{rs.getInt("sid"), rs.getString("sname"), rs.getString("saddr"),
rs.getLong("sphone"), rs.getString("semail")});

      }

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
  private void NextBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      if(rs.isLast()){

        JOptionPane.showMessageDialog(null, "You are at the last entry.");

      }

      else{

        rs.next();

        DefaultTableModel tb = (DefaultTableModel) SuppTbl.getModel();

        tb.addRow(new Object[]{rs.getInt("sid"), rs.getString("sname"), rs.getString("saddr"), rs.getLong("sphone"), rs.getString("semail")});

      }

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
```

```
private void PreviousBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(rs.isFirst()){
            JOptionPane.showMessageDialog(null, "You are at the first entry.");
        }
        else{
            rs.previous();
            DefaultTableModel tb = (DefaultTableModel) SuppTbl.getModel();
            tb.addRow(new Object[]{rs.getInt("sid"), rs.getString("sname"), rs.getString("saddr"), rs.getLong("sphone"), rs.getString("semail")});
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
}
```

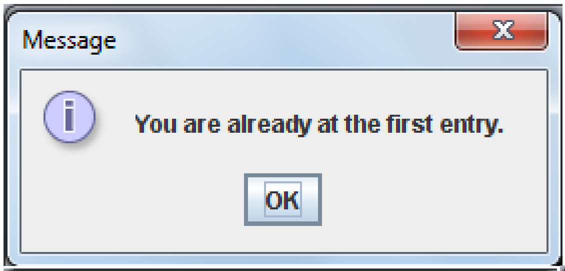
```
    private void LastBtnActionPerformed(java.awt.event.ActionEvent evt) {
      try {
        if(rs.isLast()){
          JOptionPane.showMessageDialog(null, "You are already at the last entry.");
        }
        else{
          rs.last();
          DefaultTableModel tb = (DefaultTableModel) SuppTbl.getModel();
          tb.addRow(new Object[]{rs.getInt("sid"), rs.getString("sname"), rs.getString("saddr"),
rs.getLong("sphone"), rs.getString("semail")});
        }
      }
      catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
      }
    }
```

AInventory:

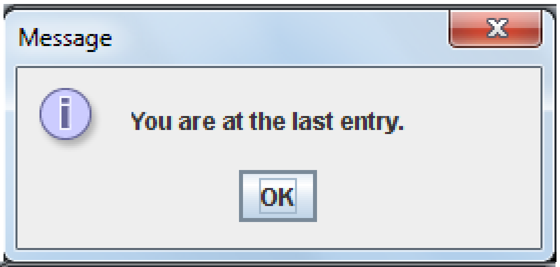


Inventory Info

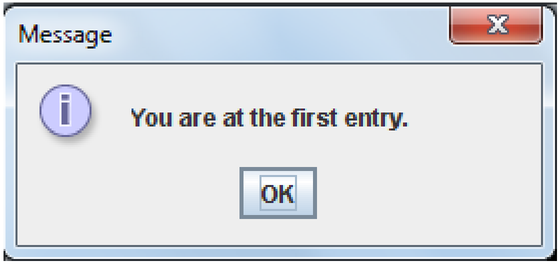
Inventory ID	Max Stock	Location
1001	Hyderabad	5000000
2001	Bangalore	4500000
2002	Bangalore	5000000
3001	Mumbai	7500000



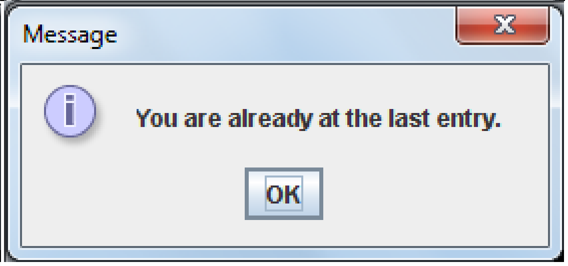
You are already at the first entry.



You are at the last entry.



You are at the first entry.



You are already at the last entry.

```
import java.sql.*;

import javax.swing.JOptionPane;

import javax.swing.table.*;

public class AInventory extends javax.swing.JFrame {

    Connection conn;

    Statement stmt;

    ResultSet rs;

  public AInventory() {

    initComponents();

  }

private void ShowBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

      Class.forName("java.sql.Driver");

      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/ims", "root", "0000");

      stmt = conn.createStatement();

      String sql1 = "Select * from inventory;";

      rs = stmt.executeQuery(sql1);

      rs.first();

      DefaultTableModel tb = (DefaultTableModel) InventTbl.getModel();

      tb.addRow(new Object[]{rs.getInt("inid"), rs.getString("loc"), rs.getLong("max_stock")});

    }

    catch (Exception e) {

      JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

  }
}

private void ProductsMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AProducts p = new AProducts();
    p.setVisible(true);
    this.dispose();
  }
  private void InvoiceMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInvoice i = new AInvoice();
    i.setVisible(true);
    this.dispose();
  }
  private void CustMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ACustomerInfo c = new ACustomerInfo();
    c.setVisible(true);
    this.dispose();
  }
  private void SuppMItemActionPerformed(java.awt.event.ActionEvent evt) {
    ASuppliers s = new ASuppliers();
    s.setVisible(true);
    this.dispose();
  }
  private void InventMItemActionPerformed(java.awt.event.ActionEvent evt) {
    AInventory i = new AInventory();
    i.setVisible(true);
    this.dispose();
  }
```

```java
private void LogoutMItemActionPerformed(java.awt.event.ActionEvent evt) {
    Welcome w = new Welcome();
    w.setVisible(true);
    this.dispose();
}
private void FirstBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(rs.isFirst()){
            JOptionPane.showMessageDialog(null, "You are already at the first entry.");
        }
        else{
            rs.first();
            DefaultTableModel tb = (DefaultTableModel) InventTbl.getModel();
            tb.addRow(new Object[]{rs.getInt("inid"), rs.getString("loc"), rs.getLong("max_stock")});
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
}
```

```
private void NextBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(rs.isLast()){
            JOptionPane.showMessageDialog(null, "You are at the last entry.");
        }
        else{
            rs.next();
            DefaultTableModel tb = (DefaultTableModel) InventTbl.getModel();
            tb.addRow(new Object[]{rs.getInt("inid"), rs.getString("loc"), rs.getLong("max_stock")});
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
}
```

```
private void PreviousBtnActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        if(rs.isFirst()){

            JOptionPane.showMessageDialog(null, "You are at the first entry.");

        }

        else{

            rs.previous();

            DefaultTableModel tb = (DefaultTableModel) InventTbl.getModel();

            tb.addRow(new Object[]{rs.getInt("inid"), rs.getString("loc"), rs.getLong("max_stock")});

        }

    }

    catch (Exception e) {

        JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");

    }

}
```

```
private void LastBtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(rs.isLast()){
            JOptionPane.showMessageDialog(null, "You are already at the last entry.");
        }
        else{
            rs.last();
            DefaultTableModel tb = (DefaultTableModel) InventTbl.getModel();
            tb.addRow(new Object[]{rs.getInt("inid"), rs.getString("loc"), rs.getLong("max_stock")});
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error Connecting to Database. Try again.");
    }
}
```