

# ***BAKERY MANAGEMENT SYSTEM***



**SUBMITTED AS A PART OF C.B.S.E. CURRICULUM  
FOR THE YEAR 2010-11**

***SUBMITTED TO :-***

***Ms. NEELAM***

***SUBMITTED BY :-***

***ROLL NO. -***

# CONTENTS

- ***CERTIFICATE***
- ***ACKNOWLEDGEMENT***
- ***PROJECT PREAMBLE***
- ***PROJECT OVERVIEW***
- ***DATABASE STRUCTURE***
- ***SOURCE CODE***
- ***OUTPUT***

# CERTIFICATE

This is to certify that

**Roll No. :** .....

of Class XII, KV NO-2 School,

have worked on the project titled

**BAKERY MANAGEMENT SYSTEM**

and have completed the project to my satisfaction.

**Teacher's Signature**

.....

.....

**Student's Signature**

.....

.....

# **ACKNOWLEDGEMENT**

**It gives me immense pleasure in expressing my deep sense of gratitude to my respected teacher,**

**Ms. Neelam** for her inspiring guidance and thought in the preparation of the project.

**This project would not have been completed without her valuable guidance and her tremendous effort.**

# **PROJECT PREAMBLE**

**The project titled “ .....**

**has been developed as per the requirement of**

**CBSE for subject Informatics Practices (065)**

**for CBSE – 2011. The source code has been**

**developed in JAVA NETBEANS and database is in MYSQL.**

**The project deals with creation of forms in JAVA**

**NETBEANS,**

**inserting data in Mysql tables, deleting,**

**searching and report generation.**

# PROJECT OVERVIEW

The following project has been designed with a deep insight into the working of the BAKERY Management System. Some of the salient features of the project are :-

The following project can be used to manage Products, Orders details.

The Product Id and Order Id are assigned through the project to prevent any tamper or loss of information.

It includes options such as modifying and deleting records to update the Product and Order database.

Product and Order details can be searched when required.

The project also generates the report for bill charged.

The code for calculating the bill eliminates any possibility of error.

While taking any input from the user all possible validation checks have been taken care of to increase the robustness of the project.

The project can be customized according to the needs of the users without much trouble.

Although this project has been developed with a great deal of care, research, thought and hard work but still some smaller and insignificant areas may have been left un-addressed, so scope has been left for customization.

# **DATABASE STRUCTURE**

**FOLLOWING TABLES HAVE BEEN USED IN THE PROJECT  
TO STORE THE DATA :-**

**DATABASE NAME: BAKERY**

**CREATE TABLE ITEM**

```
(  
    ITEMID      INT(4) PRIMARY KEY,  
    ITEMNAME    VARCHAR(20),  
    PRICE       INT(4)  
)
```

**CREATE TABLE ORDERS**

```
(  
    ORDERID     INT(3),  
    DATOFORD    DATE,  
    ITEMID      INT(4),  
    QTY         INT(3),  
    AMOUNT      DECIMAL(7,2),  
    PRIMARY KEY(ORDERID,ITEMID )  
)
```

# SOURCE

# CODE





## ***CODING OF ITEM ENTRY FORM***

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

try
{
    Class.forName("java.sql.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/
                                bakery","root","");
    Statement st=con.createStatement();
    String qry="select * from item;";
    ResultSet rs=st.executeQuery(qry);
    rs.first();
        //while(rs.next())
    int id=rs.getInt("itemid");
    String itemname = rs.getString("itemname");
    int price = rs.getInt("price");
    idtxt.setText(""+id);
    nametxt.setText(itemname);
    pricetxt.setText(""+price);
    rs.close();
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"ERROR IN
                                CONNECTIVITY");
}

}

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

```

try
{
    Class.forName("java.sql.Driver");
    Connection con = DriverManager. getConnection ("jdbc:
        mysql://localhost/ bakery","root","");
    Statement st=con.createStatement();
    int id=Integer.parseInt(idtxt.getText());
    id++;
    String qry="select * from item where itemid="+id+"";
    ResultSet rs=st.executeQuery(qry);
    rs.next();
    id=rs.getInt("itemid");
    String itemname = rs.getString("itemname");
    int price = rs.getInt("price");
    idtxt.setText(""+id);
    nametxt.setText(itemname);
    pricetxt.setText(""+price);
}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"ERROR IN
        CONNECTIVITY");
}

}

private void LASTActionPerformed(java.awt.event.ActionEvent
evt) {
    try
    {
        Class.forName("java.sql.Driver");

```

```
Connection con = DriverManager. getConnection ("jdbc:
                                     mysql://localhost/bakery","root","");
Statement st=con.createStatement();
String qry="select * from item;";
ResultSet rs=st.executeQuery(qry);
    rs.last();
int id=rs.getInt("itemid");
String itemname = rs.getString("itemname");
int price = rs.getInt("price");
idtxt.setText(""+id);
nametxt.setText(itemname);
pricetxt.setText(""+price);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(null,"ERROR IN
                                     CONNECTIVITY");
```

```
}
```

```
}
```

```
private void ADDBTN ActionPerformed (java.awt.event. Action
Event evt) {
```

```
try
```

```
{
```

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

```
Connection con = DriverManager. getConnection ("jdbc:
                                     mysql://localhost/bakery","root","");
```

```
Statement st=con.createStatement();
```

```
String qry="select itemid from item;";
```

```
ResultSet rs=st.executeQuery(qry);
    rs.last();
int count=rs.getInt("itemid");
++count;
idtxt.setText(count+"");
nametxt.setText("");
pricetxt.setText("");
savebtn.enable(false);
```

```
JOptionPane.showMessageDialog(null,"ENTER THE ITEM
    NAME AND ITEMPRICE THEN CLICK SAVE BUTTON");
rs.close();
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(null,"ERROR IN
    CONNECTIVITY");
```

```
}
```

```
}
```

```
private void MODIFYBTNActionPerformed (java.awt.event.
Action
```

```
Event evt) {
```

```
try
```

```
{
```

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

```
Connection con = DriverManager.getConnection("jdbc:
    mysql://localhost/bakery","root","");
```

```
Statement st=con.createStatement();
```

```
int id=Integer.parseInt (JOptionPane.showInputDialog
    ("ENTER THE ITEM ID TO MODIFY"));
```

```
String qry="select * from item where itemid="+id+";";
```

```
ResultSet rs=st.executeQuery(qry);
```

```
rs.first();
```

```

        idtxt.setText(""+rs.getInt("itemid"));
        nametxt.setText(rs.getString("itemname"));
        pricetxt.setText(""+rs.getInt("price"));
        JOptionPane.showMessageDialog(null,"CHANGE THE
                                INFORMATION U WANT TO CHANGE");
        nametxt.getFocusAccelerator();
        rs.close();

    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"ERROR IN

CONNECTIVITY");
    }
}

```

```

private void DELETEActionPerformed (java.awt.event. Action
Event evt) {
try
{
    Class.forName("java.sql.Driver");
    Connection con=DriverManager.getConnection("jdbc:
                                mysql://localhost/bakery","root","");
    Statement st=con.createStatement();
    int id=Integer.parseInt(JOptionPane.showInputDialog
                                ("ENTER THE ITEM ID TO DELETE"));
    String qry="select * from item where itemid="+id+"";
    ResultSet rs=st.executeQuery(qry);
        rs.first();
    String itemname = rs.getString("itemname");
    int price = rs.getInt("price");
    idtxt.setText(""+id);
    nametxt.setText(itemname);
    pricetxt.setText(""+price);
    qry="delete from item where itemid="+id+"";

```

```

    st.executeUpdate(qry);
    JOptionPane.showMessageDialog(null,"ITEM DELETED");
    idtxt.setText("");
    nametxt.setText("");
    pricetxt.setText("");
    st.close();
}

catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"ERROR IN
                                   CONNECTIVITY");
}
}

private void BACKBTNActionPerformed (java.awt.event.Action
Event evt) {
MENU i = new MENU();
    i.setVisible(true);
    this.setVisible(false);
}

private void SAVEActionPerformed(java.awt.event.ActionEvent
evt) {
try
{
    Class.forName("java.sql.Driver");
    Connection con=DriverManager.getConnection("jdbc:
                                             mysql://localhost/bakery","root","");
    Statement st=con.createStatement();
    int id=Integer.parseInt(idtxt.getText());
    String name=nametxt.getText();
    int price=Integer.parseInt(pricetxt.getText());

```

```

String qry="insert into item
                values("+id+", ""+name+", "+price+");";
st.executeUpdate(qry);
JOptionPane.showMessageDialog(null,"ITEM SAVED");
st.close();
}

catch(Exception e)
{
JOptionPane.showMessageDialog(null,"ERROR IN
                                CONNECTIVITY");
}
}

```

```

private void UPDATEActionPerformed(java.awt.event.Action
Event evt) {
try
{
    Class.forName("java.sql.Driver");
    Connection con=DriverManager.getConnection("jdbc:
                                                mysql://localhost/bakery","root","");
    Statement st=con.createStatement();
    int id=Integer.parseInt(idtxt.getText());
    String name=nametxt.getText();
    int price=Integer.parseInt(pricetxt.getText());
    String qry="update item set itemname = ""+name+" ,
                price ="+price+" where itemid="+id+";";
    st.executeUpdate(qry);
    JOptionPane.showMessageDialog(null,"ITEM UPDATE");
    st.close();

}

catch(Exception e)
{

```

```

JOptionPane.showMessageDialog(null,"ERROR IN
                                CONNECTIVITY");
    }
}

```

## ***CODING OF ORDER ENTRY FORM***

```

private void PURCHASE morebtnActionPerformed (java.awt.
event.ActionEvent evt) {
    try
    {
        Class.forName("java.sql.Driver");
        Connection con=DriverManager.getConnection("jdbc:
                                                    mysql://localhost/bakery","root","");
        Statement st=con.createStatement();
        Object id=(itemcmb.getSelectedItem());
        String qry="select price from item where itemid="+id+"";
        ResultSet rs=st.executeQuery(qry);
            rs.first();
        int p=rs.getInt("price");
        int qty=Integer.parseInt(qtytxt.getText());
        listtxt.append("\n"+id+"      "+p+"      "+qty);
        total=total+(p*qty);
        int orderid=Integer.parseInt(orderidtxt.getText());
        int price=p*qty;
        qry="Select * from newdate";
        rs=st.executeQuery(qry);
            rs.first();
        Date cd=rs.getDate("dt");

```





}

}

```
private void SAVEBTNActionPerformed(java.awt.event.Action
Event evt) {
    try
    {
        Class.forName("java.sql.Driver");
        Connection con=DriverManager.getConnection("jdbc:
            mysql://localhost/bakery","root","");
        Statement st=con.createStatement();
        Object id=(itemcmb.getSelectedItem());
        String qry="select price from item where itemid="+id+";";
        ResultSet rs=st.executeQuery(qry);
            rs.first();
        int p=rs.getInt("price");
        int qty=Integer.parseInt(qtytxt.getText());
        listtxt.append("\n"+id+"      "+p+"      "+qty);
        total=total+(p*qty);
        int orderid=Integer.parseInt(orderidtxt.getText());
        int price=p*qty;
        qry="Select * from newdate";
        rs=st.executeQuery(qry);
            rs.first();
        Date cd=rs.getDate("dt");
        qry="insert into orders values("+orderid+", "+cd+ " ", "
            +id+", " +qty+", "+price+");";
        st.executeUpdate(qry);
        totalbilltxt.setText(""+total);
        rs.close();
        JOptionPane.showMessageDialog(null,"ORDER IS SAVED");
```

```

}
catch(Exception e)
{JOptionPane.showMessageDialog(null,"ERROR IN
                                CONNECTIVITY");
}}

```

```

private void BACKTOMENU ActionPerformed(java.awt.event.
ActionEvent evt)
{
    MENU i = new MENU();
    i.setVisible(true);
    this.setVisible(false);
}

```

## ***CODING OF ITEM SEARCH***

```

private void ITEMmenuitem1ActionPerformed(java.awt.event.
ActionEvent evt) {
DefaultTableModel tm=(DefaultTableModel)itemtable.getModel();
int id=Integer.parseInt(JOptionPane. showInputDialog
                                (null,"ENTER ITEM ID TO SEARCH"));
    opanel.setVisible(false);
try
{
while(tm.getRowCount(>0)
tm.removeRow(0);
Class.forName("java.sql.Driver");
    Connection con=DriverManager.getConnection("jdbc:
                                                mysql://localhost/bakery","root","");
    Statement st=con.createStatement();
    String qry="select * from item where itemid="+id+"";
    ResultSet rs=st.executeQuery(qry);
        rs.first();
    int price=rs.getInt("price");
    String name=rs.getString("itemname");
    tpanel.setVisible(true);
}
}

```

```

        tm.addRow(new Object[]{id,name,price});
        rs.close();
    }

    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"RECORD NOT FOUND");
    }

    private void ORDERBTNItem1ActionPerformed(java.awt.event.
   (ActionEvent evt) {
        DefaultTableModel tm=(DefaultTableModel)ordertable.
            getModel();
        int id=Integer.parseInt(JOptionPane. showInputDialog
            (null,"ENTER ORDER ID TO SEARCH"));
        tpanel.setVisible(false);
        try
        {
            while(tm.getRowCount(>0)
            tm.removeRow(0);
            Class.forName("java.sql.Driver");
            Connection con=DriverManager.getConnection("jdbc:
                mysql://localhost/bakery","root","");
            Statement st=con.createStatement();
            String qry="select * from orders where orderid="+id+"";
            ResultSet rs=st.executeQuery(qry);
                opanel.setVisible(true);
                while(rs.next())}
            int amt=rs.getInt("amount");
            Date d=rs.getDate("datoford");
            int qty=rs.getInt("qty");
            int i=rs.getInt("itemid");
            tm.addRow(new Object[]{id,i,qty,amt,d});
            rs.close();
        }
        catch(Exception e)

```

```
{JOptionPane.showMessageDialog(null,"RECORD NOT  
FOUND");  
    } }
```

## ***CODING OF SEARCH / REPORT FORM***

### **REPORT**

```
private void ITEMREPORT MenuItem1ActionPerformed  
(java.awt.event.ActionEvent evt)  
{ DefaultTableModel tm=(DefaultTableModel)itemtable.  
                                                                    getModel();  
    opanel.setVisible(false);  
    try  
{  
    while(tm.getRowCount(>0)  
    tm.removeRow(0);  
        Class.forName("java.sql.Driver");  
        Connection con=DriverManager.getConnection("jdbc:  
                                                    mysql://localhost/bakery","root","");  
        Statement st=con.createStatement();  
        String qry="select * from item;";  
        ResultSet rs=st.executeQuery(qry);  
            while(rs.next())  
{  
                int id=rs.getInt("itemid");  
                int price=rs.getInt("price");  
                String name=rs.getString("itemname");  
                tpanel.setVisible(true);  
                tm.addRow(new Object[]{id,name,price});  
                rs.close();  
            }  
        }  
    catch(Exception e)  
{  
    JOptionPane.showMessageDialog(null,"RECORD NOT FOUND");  
    }  
}
```

```
}
```

```
private void ORDERmenuitem2ActionPerformed  
(java.awt.event. ActionEvent evt) {  
DefaultTableModel tm=(DefaultTableModel) ordertable.  
                                                                    getModel();  
tpanel.setVisible(false);  
  
    try  
{  
while(tm.getRowCount(>0)  
tm.removeRow(0);  
Class.forName("java.sql.Driver");  
    Connection con=DriverManager.getConnection("jdbc:  
                                                                    mysql://localhost/bakery","root","");  
    Statement st=con.createStatement();  
    String qry="select * from orders ;";  
    ResultSet rs=st.executeQuery(qry);  
    opanel.setVisible(true);  
    while(rs.next()){  
int id=rs.getInt("orderid");  
int amt=rs.getInt("amount");  
Date d=rs.getDate("datoford");  
int qty=rs.getInt("qty");  
int i=rs.getInt("itemid");  
tm.addRow(new Object[]{id,i,qty,amt,d});  
rs.close();  
    }  
catch(Exception e)  
{  
JOptionPane.showMessageDialog(null,"RECORD NOT FOUND");  
    }  
}
```

## ***CODING OF FORM EXIT FORM***

```
private void EXITActionPerformed(java.awt.event.ActionEvent  
evt) {  
System.exit(0);  
}
```

The word "OUTPUT" is rendered in a large, 3D, serif font. The letters are blue with a thick orange outline, giving them a blocky, three-dimensional appearance. The text is centered horizontally in the lower half of the page.

**MAIN FRAME**

ITEM ORDER **SEARCH** REPORT EXIT

## BAKERY MANAGEMENT

ITEM ID	ITEM NAME	PRICE

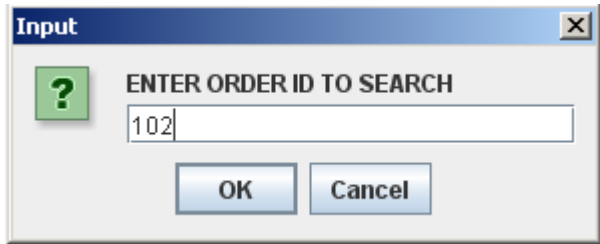
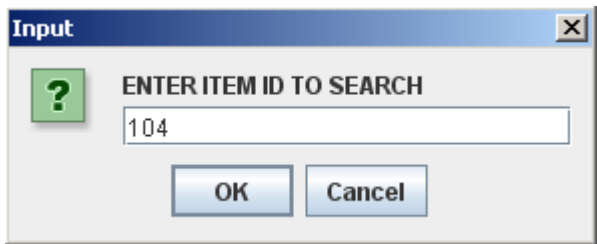
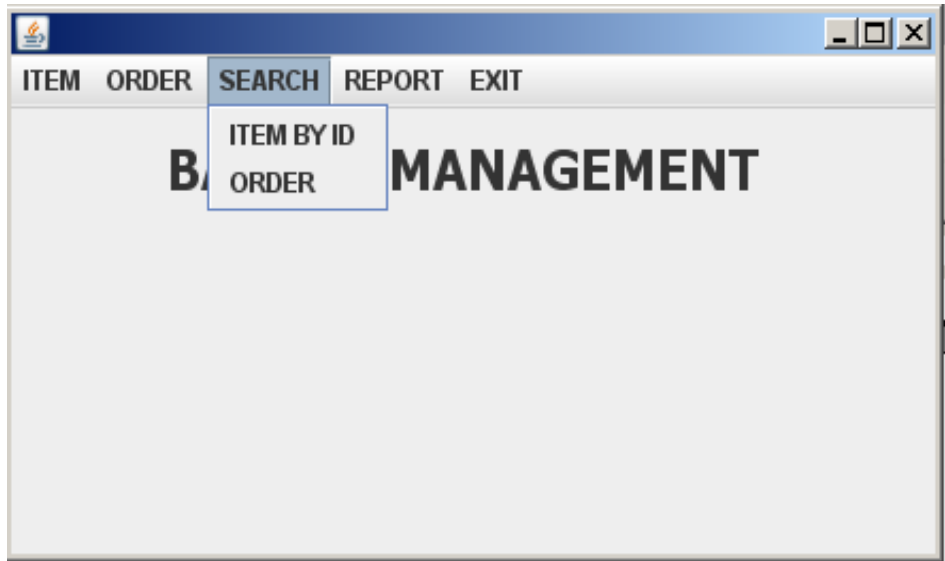
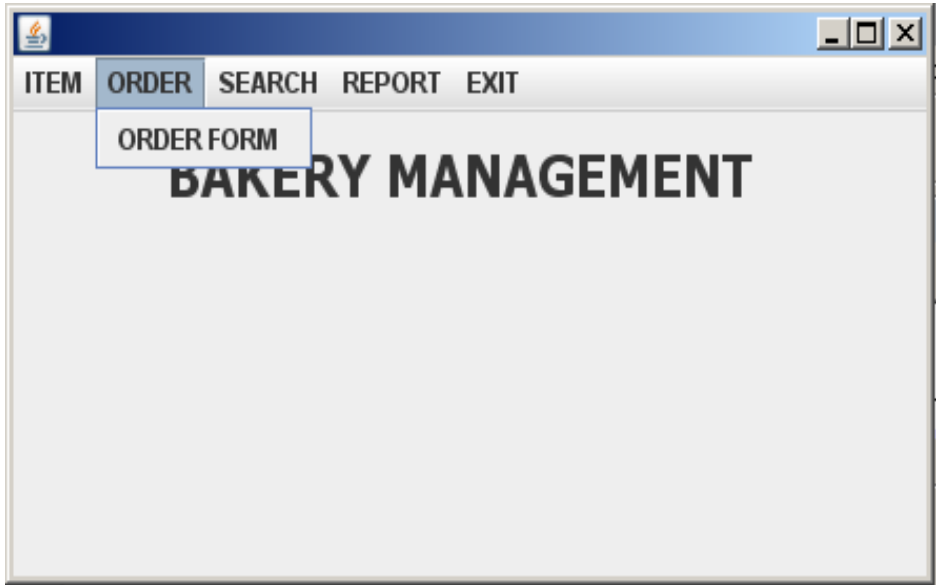
ORDER ID	ITEM ID	QUANTITY	AMOUNT	DATE

ITEM ORDER SEARCH REPORT EXIT

ITEM FORM

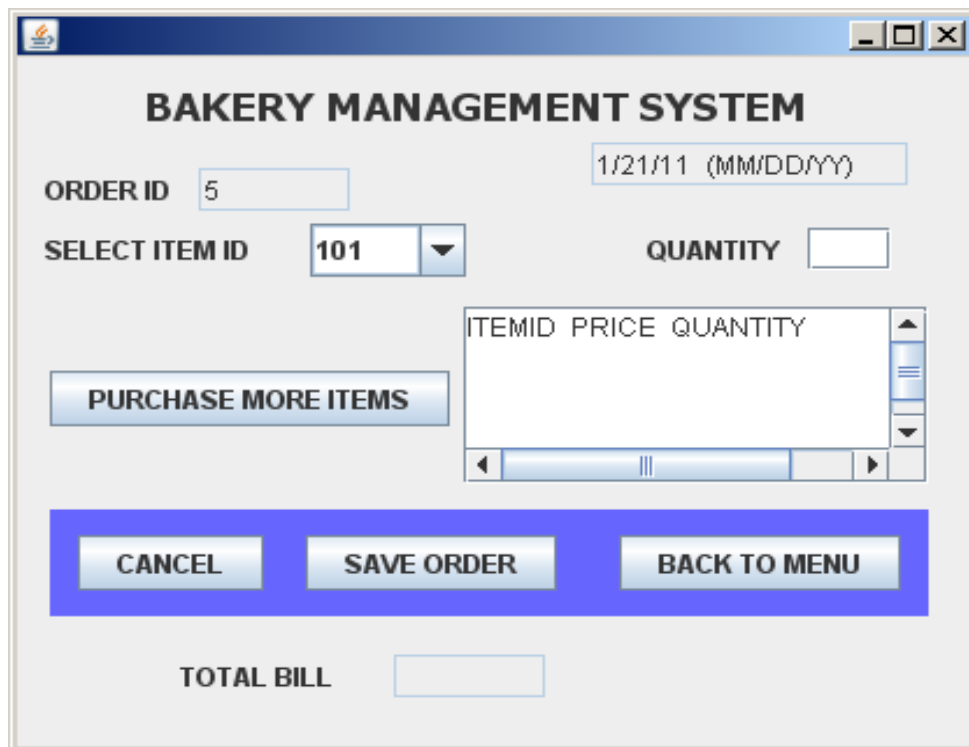
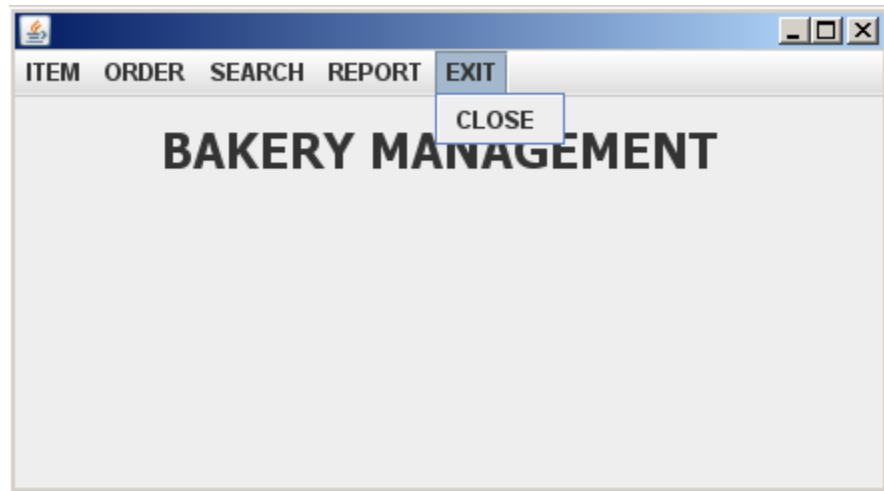
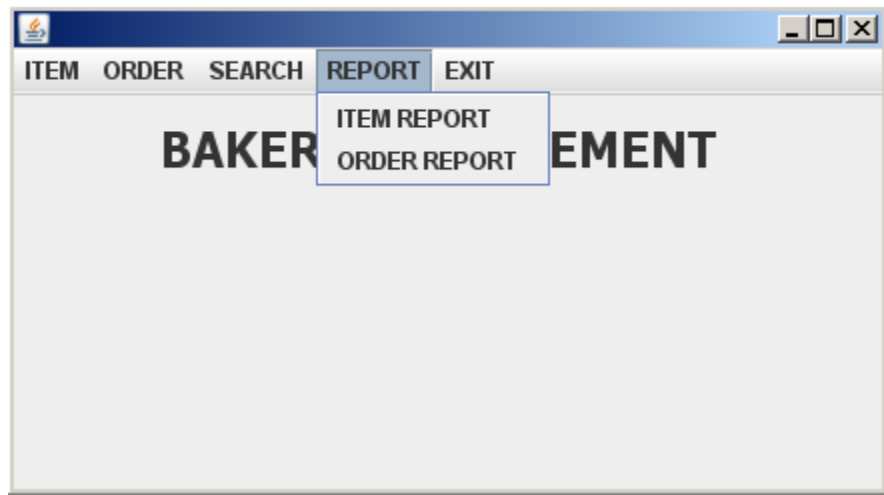
## BAKERY MANAGEMENT





BAKERY MANAGEMENT		
ITEM ID	ITEM NAME	PRICE
101	biscuit	50

BAKERY MANAGEMENT				
ORDER ID	ITEM ID	QUANTITY	AMOUNT	DATE
1	101	2	90	2011-01-17



**BAKERY MANAGEMENT SYSTEM**

ITEM ID

ITEM NAME

PRICE

SAVE

UPDATE

ADD NEW ITEM

MODIFY ITEM

DELETE ITEM

BACK TO MENU

FIRST ITEM

NEXT ITEM

LAST ITEM

ITEM ORDER SEARCH REPORT EXIT

**BAKERY MANAGEMENT**

ITEM ID	ITEM NAME	PRICE
101	biscuit	50
102	chocolate	130
103	ice cream	100
104	jam	155
105	shampoo	96
106	shoes	2200
107	bags	3020
108	clothes	1800

ORDER ID	ITEM ID	QUANTITY	AMOUNT	DATE	
1	101	2	90	2011-01-17	▲
2	101	2	100	2011-01-17	
2	102	5	600	2011-01-17	
3	101	1	50	2011-01-17	
3	102	2	240	2011-01-17	▼

