**ACKNOWLEDGEMENT**

I would like to add a few heartfelt words for the folks who were part of this project ” The Agro services ” in numerous to express my deepest gratitude to all those who have contributed in this project either directly or indirectly.

At the very outset we take opportunity to express our deepest gratitude and heartily thanks to our Head of department Mr.Nikam Sir and project in charge Miss. Rohini Daund for her helping and encouragement from time to time.

Last but not the list I must thank to my friends, well wishers for their full support and prompt help without which the completion of this project was impossible.

Thank you all,

Monika Bafana & Pranali Sonje. T.Y.B.Sc(comp sci.)

**INTRODUCTION**

The project is about the system of ‘Agro Services’ which is deals with the Agricultural Products.

The term ‘Agro Services’ can be specifically defined as “Managing the daily transaction of Fertilizers, Pesticides &seeds which can be purchased from supplier and sale to customer.”

In this system we maintain the Fertilizer database in which all information about product, customer is stored. From this database we give input to system for process.

At process time, the particular information is given from database for creation of purchase Bill, sale bill, supplier details, customer details, Stock details.

As result the system gives us the report, which is nothing but information about daily sale bill, Parches bill & Member entry

**REQUIREMENT ANALYSIS**

**HARDWARE & SOFTWARE USED:-**

To install Project successfully the hardware and software

Requirements are as follows

1 . Operating environment:

Operating system :- WINDOWS XP OR ABOVE

2 . Software tools used:

Front-end tool used :- JAVA 1.6, SWING, AWT

Back-end tool used :- MICROSOFT ACCESS 2000 OR ABOVE

Reports :- Data reports

3 .System configuration:

CPU type :- PENTIUM III OR ABOVE(PROCESSOR)

RAM :- 2 GB

Hard Disk :- 160 GB

**EXISTING SYSTEM**

**C**urrently the system involves manual work .The customer checkin entries are first recorded in a rough register. After particular interval day

Or a week the entries are registered into main catalog. Close entries are also registered in main register to calculate net profit.

The worker bill registered in same way .When customer finish his/her function, one copyof bill is given to him and the other is kept for record.

The entries in main catalog checked periodically. These registered have to maintain for calculation of net profit or loss.

**Drawbacks of Existing System**

1. Time consuming
2. Lack of security
3. Dose not full fill the all requirements
4. Difficult to Analysis
5. Dose not have any backup
6. Dose not generate daily reports

**PROPOSED SYSTEM**

Proposed system is computerized system which minimizes the drawbacks of manual system

It is user friendly interface which will help the organization to contribute to perform the task more easily.

In this system for the records of many clients, we have also provided Validation to all fields in form based on their data types in database.

We are providing many buttons like Insert, Update, Remove and Search that help user to retrieve data. All records are correctly save in database so there is no chance of ambiguity. Relationships of tables are well maintained

* **The following are the Advantage of proposed System:**
* The proposed system is computerized that’s why all data is entered on disk the retrieve of information is very easy.
* The system is not time consuming.
* Modification is done very easily on existing record.
* The system is cheaper.
* The input given through database to system. In processing phase error doesn’t occur.
* It generates output which is the cash memo. This is nothing but the details of bill of Fertilizer. It also generates the daily cash memo register, member register, stock register, bill reports register etc.
* This system also used by the user who is not highly qualified but has basic knowledge of computer.

**SYSTEM ANALYSIS**

**DEFINITION OF FEASIBILITY STUDY**

An outline of the programming investigation is the determination that the system. Suggested is really feasible or not. There are three main aspects in the feasibility study portion of the preliminary investigation.

**TYPE OF FEASIBILITY STUDY**

* **TECHNICAL FEASIBILITY.**
* **ECONOMICAL FEASIBILITY.**
* **OPERATIONAL FEASIBILITY.**
* **Technical feasibility:**

Proposed system is developed in relational database management System like ‘MS Access’ as backend and ‘Java 1.6’ as frontend. Technical validations are made all over in the system in every screen and for every entry. So the system guarantees the data security, easy access and accuracy of the reports.

* **Operational feasibility:**

The relational database management provides an environment to user develop user friendly applications. In this project the screens are developed with concern of every aspect of the user, every problem and every query of the user. A tool tip is given to the every command button and every operational object which are display in the status bar at the bottom of the every form.

* **Economical feasibility:**

The cost effectiveness of the project can be measured in terms of tangible and intangible benefits. Here we don’t have worry about the hardware and software. Therefore no extra cost needed or no any special devices are required. It requires less manpower than manual system & also provides accurate managerial information.

**SCOPE OF THE SYSTEM**

* The scope of the work involves studying existing computerized Management System and to simplify complex agro services
* In this project all the activities which were done manually are done with great reliability, accuracy and speed.
* It helps to store record for longer period without any data loss.
* In this data retrieval and insertion or data deletion is very easy and fast.
* Updating to any record is done in negligible amount of time.
* The efforts are reduce and required manpower is less

**DATABASE DESIGN**

**Login**

**Company**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| User\_ID | AutoNumber | Primary key |
| User\_Name | Text |  |
| User\_Password | Text |  |
| User\_Type | Text |  |

**Product**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| Pro\_No | AutoNumber | Primary key |
| Pro\_Name | Text |  |
| Wt\_per\_unit | Number |  |
| Batch\_no | Text |  |
| Mfg\_date | Text |  |
| Exp\_date | Text |  |
| Price | Text |  |

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| comp\_id | AutoNumber | Primary key |
| comp\_name | Text |  |
| comp\_addr | Text |  |
| comp\_phone | Text |  |
| comp\_email | Text |  |

**Supplier**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| Sup\_ID | AutoNumber | Primary key |
| Sup\_Name | Text |  |
| Address | Text |  |
| Sup\_Company | Text |  |
| Mobile\_no | Text |  |
| Place | Text |  |

**Customer**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| Cust\_ID | AutoNumber | Primary key |
| Cust\_Name | Text |  |
| Address | Text |  |
| Profession | Text |  |
| Mobile\_no | Text |  |
| Email\_ID | Text |  |
| Nationality | Text |  |
| Place | Text |  |

**PO\_Header**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| Bill\_no | AutoNumber | Primary Key |
| Total\_Qty | Number |  |
| Total\_amount | Text |  |
| Supp\_id | Number | Foreign Key |
| Paid\_amt | Text |  |
| Bal\_amt | Text |  |
| Bill\_date | Text |  |

**PO\_Details**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| poid | AutoNumber | Primary Key |
| Prod\_name | Text |  |
| qty | Text |  |
| Bill\_no | text | Foreign Key |

**Sale\_Header**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| saleBill\_no | AutoNumber | Primary Key |
| Total\_Qty | Number |  |
| Total\_amount | Text |  |
| Cust\_id | Number | Foreign Key |
| Paid\_amt | Text |  |
| Bal\_amt | Text |  |
| sBill\_date | Text |  |

**Sale\_Details**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Constraint** |
| saled | AutoNumber | Primary Key |
| Prod\_name | Text |  |
| qty | Text |  |
| sBill\_no | text | Foreign Key |

**LOGIN SCREEN CODE**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

class Login1 extends JFrame implements ActionListener

{

JLabel l1,l2,l3,img;

JTextField t1,t2;

JButton b1,b2,b3;

Container c1=getContentPane();

Connection conn; ResultSet rs;

Font f1 = new Font("Arial",Font.PLAIN,14);

public Login1()

{

super("Login : Agro Services");

setBounds(300,200,400,300);

setVisible(true);

setLayout(null);

// Image img = getImage(getDocumentBase(),"bk.jpg");

img=new JLabel(new ImageIcon("sss.jpg"));

img.setBounds(8,0,90,90);

c1.add(img);

l3=new JLabel("WEL-COME");

l3.setBounds(120,20,500,30);

Font f2=new Font("Bell MT",Font.BOLD+Font.ITALIC,30);

l3.setFont(f2);

c1.add(l3);

l1 = new JLabel("User Name :");

l1.setForeground(Color.black);

l1.setBounds(80,70,500,30);

l1.setFont(f1);

c1.add(l1);

l2 = new JLabel("Password :");

l2.setForeground(Color.black);

l2.setBounds(80,120,500,30);

l2.setFont(f1);

c1.add(l2);

t1 = new JTextField();

t1.setBounds(200,75,150,22);

t1.setToolTipText("Enter your user name here.");

c1.add(t1);

t1.requestFocus();

t2 = new JPasswordField();

t2.setBounds(200,125,150,22);

t2.setToolTipText("Enter your password here.");

c1.add(t2);

JLabel lb = new JLabel("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"+

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

lb.setForeground(Color.BLACK);

lb.setBounds(0,125,400,125);

c1.add(lb);

b1 = new JButton("Ok");

b1.setMnemonic('O');

b1.setBounds(100,220,80,30);

c1.add(b1);

b2 = new JButton("Exit");

b2.setMnemonic('x');

b2.setBounds(200,220,80,30);

c1.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

setSize(410,310);

setVisible(true) ;

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

String usernm=t1.getText().trim();

String pswd=t2.getText().trim();

try

{

if(usernm.equals("") && pswd.equals(""))

{

String msg="Please Enter User name & password";

int j = JOptionPane.showConfirmDialog

(this,msg,"Login : Agro Services",

JOptionPane.OK\_CANCEL\_OPTION,

JOptionPane.WARNING\_MESSAGE);

if(j == JOptionPane.OK\_OPTION)

{

t2.setText(null);

t1.setText(null);

t1.requestFocusInWindow();

}

if(j == JOptionPane.CANCEL\_OPTION)

{

System.exit(0);

}

}

if (conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

// Query for checking user

String qry = "Select User\_Password,User\_Type,User\_ID"+

" From Login Where User\_Name="+"'"+usernm+"'";

PreparedStatement Check\_User = conn.prepareStatement(qry);

//chku.setString(1,usernm);

ResultSet rs = Check\_User.executeQuery(); // execute query

rs.next();

int i = rs.getRow();

if(i == 0) // User Does not exist

{

String msg = "User "+"'"+usernm+"'"+" dose not exist.";

int j = JOptionPane.showConfirmDialog

(this,msg,"Login : Agro services",

JOptionPane.OK\_CANCEL\_OPTION,

JOptionPane.WARNING\_MESSAGE);

if(j == JOptionPane.OK\_OPTION)

{

conn.close();

t2.setText(null);

t1.setText(null);

t1.requestFocusInWindow();

}

if(j == JOptionPane.CANCEL\_OPTION)

{

conn.close();

System.exit(0);

}

}

else

if(i > 0) // User exist

{

String UserPass = rs.getString(1);

// Functions Fun = new Functions();

if(pswd.equalsIgnoreCase(UserPass))

{

// Password Match grant Access.

String msg = "Succesful Login.";

int j = JOptionPane.showConfirmDialog

(this,msg,"Login : Agro Services",

JOptionPane.DEFAULT\_OPTION,

JOptionPane.INFORMATION\_MESSAGE);

if(j == JOptionPane.OK\_OPTION)

{

// Exit Login and display main window

dispose();

master MW = new master();

MW.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

MW.setExtendedState(JFrame.MAXIMIZED\_BOTH);

MW.setVisible(true);

MW.setResizable(false);

}

}

else

{

// Password not Match.

String msg = "Password dose not match.";

int j = JOptionPane.showConfirmDialog

(this,msg,"Login : Agro Services",

JOptionPane.OK\_CANCEL\_OPTION,

JOptionPane.WARNING\_MESSAGE);

if(j == JOptionPane.OK\_OPTION)

{

conn.close();

t2.setText(null);

t1.setText(null);

t1.requestFocusInWindow();

}

if(j == JOptionPane.CANCEL\_OPTION)

{

conn.close();

System.exit(0);

}

}

}

}

catch(Exception e1){}

}

else

if(e.getSource()==b2)

{

System.exit(0);

}

}

public static void main(String args[])

{

Login1 l=new Login1();

}

}

**USER SCREEN CODE**

import java.sql.\*;

import java.util.Date;

import java.io.\*;

import java.util.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.border.\*;

import java.text.\*;

import javax.swing.text.\*;

import javax.swing.table.\*;

import javax.sql.rowset.\*;

import com.sun.rowset.\*;

import java.awt.print.\*;

class User extends JInternalFrame implements ActionListener

{

JLabel l1,l2,l3,l4,l5;

JTextField tf1,tf2; JPasswordField tf3;

JComboBox cmb1; JButton b1,b2,b3,b4,b5;

Font f1 = new Font("Times New Roman",Font.BOLD,22);

JPanel panDescription,panComponent,panButton; Container c1;

Connection conn; ResultSet rs;

int framemode = 0; // 1 : Addmode 2 : Editmode 3 : Viewmode

public User()

{

super("Master : User"); setLayout(null); c1 = getContentPane();

c1.setBackground(Color.white);

panDescription = new JPanel(); panComponent = new JPanel();

panButton = new JPanel(); panDescription.setLayout(null);

panComponent.setLayout(null); panButton.setLayout(null);

panDescription.setBorder(new LineBorder(new Color(100,100,200),1,true));

panComponent.setBorder(new LineBorder(new Color(100,100,200),1,true));

panButton.setBorder(new LineBorder(new Color(100,100,200),1,true));

panDescription.setBounds(0,0,492,60);

panComponent.setBounds(0,60,492,200);

panButton.setBounds(0,260,492,60);

l1 = new JLabel("Enter Login Detail of User");

l2 = new JLabel("User ID :");

l3 = new JLabel("User Name :");

l4 = new JLabel("User Password :");

l5 = new JLabel("User Type :");

l1.setBounds(120,13,400,30); l2.setBounds(60,20,150,22);

l3.setBounds(60,60,150,22); l4.setBounds(60,100,150,22);

l5.setBounds(60,140,150,22); l1.setFont(f1);

tf1 = new JTextField(); tf2 = new JTextField();

tf3 = new JPasswordField(); tf1.setBounds(200,20,100,22);

tf2.setBounds(200,60,200,22); tf3.setBounds(200,100,200,22);

b1 = new JButton("Add"); b1.setMnemonic('A');

b2 = new JButton("Modify"); b2.setMnemonic('M');

b3 = new JButton("Delete"); b3.setMnemonic('D');

b4 = new JButton("Ok"); b4.setMnemonic('O');

b5 = new JButton("Cancel"); b5.setMnemonic('C');

b1.setToolTipText("Add"); b2.setToolTipText("Modify");

b3.setToolTipText("Delete"); b4.setToolTipText("Ok");

b5.setToolTipText("Cancel");

b1.setBounds(10,15,75,30); b2.setBounds(90,15,75,30);

b3.setBounds(170,15,75,30); b4.setBounds(10,15,75,30);

b5.setBounds(90,15,75,30);

cmb1 = new JComboBox(); cmb1.setBounds(200,140,200,22);

cmb1.addItem("Administrator"); cmb1.addItem("User");

panDescription.add(l1); panComponent.add(l2);

panComponent.add(l3); panComponent.add(l4);

panComponent.add(l5); panComponent.add(tf1);

panComponent.add(tf2); panComponent.add(tf3);

panComponent.add(cmb1); panButton.add(b1);

panButton.add(b2); panButton.add(b3); panButton.add(b4);

panButton.add(b5);

c1.add(panDescription);c1.add(panComponent,"Center");c1.add(panButton);

b1.addActionListener(this); b2.addActionListener(this);

b3.addActionListener(this); b4.addActionListener(this);

b5.addActionListener(this);

framemode = 3; enabledisable\_Button(framemode); // View Mode

display\_Record(4); // Display last records

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource() == b1) // Add is clicked.

{

framemode = 1;

tf1.setText(null); tf2.setText(null); tf3.setText(null);

enabledisable\_Button(framemode); // Add Mode

tf2.requestFocus();

}

if(ae.getSource() == b2) // Modify is clicked.

{

/\*if(tf1.getText().equals(""))

{

String msg = "There is no User to Modify";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

}

else

{

tf1.setText(null); tf2.setText(null); tf3.setText(null);

framemode = 2; enabledisable\_Button(framemode); // Edit Mode

tf1.requestFocus();

}\*/

tf1.setText(null); tf2.setText(null); tf3.setText(null);

framemode = 2; enabledisable\_Button(framemode); // Edit Mode

tf1.requestFocus();

tf1.addKeyListener(new KeyAdapter()

{

public void keyPressed(KeyEvent EVT)

{

if(!tf1.getText().equals(""))

{

String ID = tf1.getText().trim();

String qry = "";

try

{

if(conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

// Query for Searching member name

qry = "Select User\_Name,User\_Password,User\_Type,User\_ID"+

" From Login Where User\_ID="+""+ID+"";

PreparedStatement Record = conn.prepareStatement(qry);

ResultSet rs = Record.executeQuery(); // execute query

rs.next(); int i = rs.getRow();

if(i == 0) // There is no record

{

String msg = "There is no User to display.";

JOptionPane.showMessageDialog((Component)null,msg,"Agro Services",

JOptionPane.ERROR\_MESSAGE);

conn.close();

tf1.requestFocus();

}

else

{

tf1.setEditable(false);

tf2.setText(rs.getString(1));

tf3.setText(rs.getString(2));

cmb1.setSelectedItem(rs.getString(3));

tf2.setEditable(true);

tf3.setEditable(true);

cmb1.setEnabled(true);

}

} // End of try

catch(SQLException sqle)

{

System.out.println("SQL Error"+sqle);

} // End of catch

}

}

});

}

if(ae.getSource() == b5) // Cancel is clicked.

{

framemode = 3; enabledisable\_Button(framemode); // View Mode

display\_Record(4); // Display Last Record

}

if(ae.getSource() == b4) // Ok is clicked

{

String un= tf2.getText().trim(); // User Name

String up = tf3.getText().trim(); // User Password

String ut = (String)cmb1.getSelectedItem(); // User Type

if(un.equals("") || up.equals(""))

{

String msg = "Please fill all Necessory Data";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

}

else

{

try

{

if (conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

if(framemode == 1) // Addmode

{

// Query for checking user

String qry = "Select User\_Name,User\_Password,User\_Type,User\_ID"+

" From Login Where User\_Name="+"'"+un+"'";

PreparedStatement Check\_User = conn.prepareStatement(qry);

//chku.setString(1,usernm);

ResultSet rs = Check\_User.executeQuery(); // execute query

rs.next();

int i = rs.getRow();

if(i == 0) // User Does not exist

{

// Query for Inserting record

String qry1="INSERT INTO Login(User\_Name,User\_Password,"+

"User\_Type) VALUES(?,?,?)";

PreparedStatement Insert\_Record=conn.prepareStatement(qry1);

Insert\_Record.setString(1,un);Insert\_Record.setString(2,up);

Insert\_Record.setString(3,ut);

Insert\_Record.execute(); // execute query

String msg = "Record Saved Sucessfully!";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

conn.close();

display\_Record(4);

}

else

{

String msg = "User Already exist...!!!";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

}

}

else if(framemode == 2) // Editmode

{

String qry="UPDATE Login Set User\_Name=?,User\_Password=?,"+

"User\_Type=?"+

" WHERE Login.User\_ID="+tf1.getText()+";";

PreparedStatement Update\_Record=conn.prepareStatement(qry);

Update\_Record.setString(1,un);

Update\_Record.setString(2,up);

Update\_Record.setString(3,ut);

Update\_Record.execute(); // execute query

String msg = "Record Updated Sucessfully!";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

conn.close();

}

else if(framemode == 4) // Delete mode

{

if(tf1.getText().equals(""))

{

String msg = "There is no User to Delete";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

}

else

{

String msg = "Are you sure!";

int i = JOptionPane.showConfirmDialog

(this,msg,"Agro services",JOptionPane.YES\_NO\_OPTION,

JOptionPane.WARNING\_MESSAGE);

if(i==JOptionPane.YES\_OPTION)

{

try

{

if (conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

String qry="DELETE \* FROM Login"+

" WHERE Login.User\_ID="+tf1.getText()+";";

PreparedStatement Delete\_Record=conn.prepareStatement(qry);

Delete\_Record.execute(); // execute query

msg = "Record Deleted Sucessfully!";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

tf1.setText("");tf2.setText("");tf3.setText("");

display\_Record(4); // Display last records

conn.close();

}

catch(SQLException sqle)

{

System.out.println("SQL Error"+sqle);

}

}

}

}

framemode = 3; enabledisable\_Button(framemode);

}

catch(SQLException sqle)

{

System.out.println("SQL Error"+sqle);

}

}

}

if(ae.getSource() == b3) // Delete is clicked

{

tf1.setText(null); tf2.setText(null); tf3.setText(null);

framemode = 4; enabledisable\_Button(framemode); // Delete Mode

tf1.requestFocus();

tf1.addKeyListener(new KeyAdapter()

{

public void keyPressed(KeyEvent EVT)

{

if(!tf1.getText().equals(""))

{

String ID = tf1.getText().trim();

String qry = "";

try

{

if(conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

// Query for Searching member name

qry = "Select User\_Name,User\_Password,User\_Type,User\_ID"+

" From Login Where User\_ID="+""+ID+"";

PreparedStatement Record = conn.prepareStatement(qry);

ResultSet rs = Record.executeQuery(); // execute query

rs.next(); int i = rs.getRow();

if(i == 0) // There is no record

{

String msg = "There is no User to display.";

JOptionPane.showMessageDialog((Component)null,msg,"Agro Services",

JOptionPane.ERROR\_MESSAGE);

conn.close();

tf1.requestFocus();

}

else

{

tf1.setEditable(false);

tf2.setText(rs.getString(1));

tf3.setText(rs.getString(2));

cmb1.setSelectedItem(rs.getString(3));

tf2.setEditable(true);

tf3.setEditable(true);

cmb1.setEnabled(true);

}

} // End of try

catch(SQLException sqle)

{

System.out.println("SQL Error"+sqle);

} // End of catch

}

}

});

}

}

void enabledisable\_Button(int Frame\_Mode)

{

if(Frame\_Mode == 1)

{ // Frame Mode is addmode

b1.setVisible(false); b2.setVisible(false); b3.setVisible(false);

b4.setVisible(true); b5.setVisible(true);

tf1.setEditable(false);tf2.setEditable(true); tf3.setEditable(true);

cmb1.setEnabled(true);

this.closable = false;

}

else if(Frame\_Mode == 2)

{ // Frame Mode is edit mode

b1.setVisible(false); b2.setVisible(false); b3.setVisible(false);

b4.setVisible(true); b5.setVisible(true);

tf1.setEditable(true);tf2.setEditable(false); tf3.setEditable(false);

cmb1.setEnabled(false);

this.closable = false;

}

else

if(Frame\_Mode == 3) // Frame Mode is view Mode

{

b1.setVisible(true); b2.setVisible(true); b3.setVisible(true);

b4.setVisible(false); b5.setVisible(false);

tf1.setEditable(false);tf2.setEditable(false);

tf3.setEditable(false);cmb1.setEnabled(false);

this.closable = true;

}

else if(Frame\_Mode == 4)

{ // Frame Mode is delete mode

b1.setVisible(false); b2.setVisible(false); b3.setVisible(false);

b4.setVisible(true); b5.setVisible(true);

tf1.setEditable(true);tf2.setEditable(false); tf3.setEditable(false);

cmb1.setEnabled(false);

this.closable = false;

}

}

// This method display the record according to button clicked in view mode.

void display\_Record(int flag)

{

String U\_ID = tf1.getText().trim();

if(framemode!=3)

{

tf1.setText(null);tf2.setText(null);tf3.setText(null);

}

String qry = "";

try

{

if(conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

if(flag==1)

{ // Query for Selecting First record

qry = "SELECT Login.\*, Login.User\_ID"+

" FROM Login WHERE (((Login.User\_ID)="+

"(SELECT Min(Login.User\_ID) AS Expr1 FROM Login;)));";

}

if(flag==2)

{ // Query for Selecting Previous record

qry = "SELECT Login.\*, Login.User\_ID FROM Login "+

"WHERE (((Login.User\_ID)<"+U\_ID+"))"+

" ORDER BY Login.User\_ID DESC;";

}

if(flag==3)

{ // Query for Selecting Next record

qry = "SELECT Login.\*, Login.User\_ID FROM Login "+

"WHERE (((Login.User\_ID)>"+U\_ID+"))"+

"ORDER BY Login.User\_ID;";

}

if(flag==4)

{ // Query for Selecting Last record

qry = "SELECT Login.\*, Login.User\_ID"+

" FROM Login WHERE (((Login.User\_ID)="+

"(SELECT Max(Login.User\_ID) AS Expr1 FROM Login;)));";

}

PreparedStatement Record = conn.prepareStatement(qry);

ResultSet rs = Record.executeQuery(); // execute query

rs.next(); int i = rs.getRow();

if(i == 0) // There is no record

{

String msg = "There is no user to display.";

JOptionPane.showMessageDialog((Component)null,msg,"Agro services",

JOptionPane.INFORMATION\_MESSAGE);

conn.close();

}

else

{

tf1.setText((String)rs.getString("User\_ID"));

tf2.setText(rs.getString(2));

tf3.setText(rs.getString(3));

String UT =(String)rs.getString(4);

if(UT.equals("Administrator"))

cmb1.setSelectedIndex(0);

else

cmb1.setSelectedIndex(1);

conn.close();

}

} // End of try

catch(SQLException sqle)

{

System.out.println("SQL Error"+sqle);

} // End of catch

} // End of display\_Record

} // End of User Frame

**STOCK REPORT**

import java.sql.\*;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.io.\*;

import javax.sql.rowset.\*;

import javax.swing.table.DefaultTableModel;

public class dispProductRecord extends JFrame

{

private JPanel jpShow = new JPanel ();

private DefaultTableModel dtmCustomer;

private JTable tbCustomer;

private JScrollPane jspTable;

private int row = 0;

private int total = 0;

private String rowData[][];

private FileInputStream fis;

private DataInputStream dis;

Connection conn; ResultSet rs;

dispProductRecord ()

{

setSize (700, 365);

jpShow.setLayout (null);

populateArray ();

tbCustomer = makeTable ();

jspTable = new JScrollPane (tbCustomer);

jspTable.setBounds (20, 20, 700, 300);

//Adding the Table to Panel.

jpShow.add (jspTable);

//Adding Panel to Window.

getContentPane().add (jpShow);

//In the End Showing the New Account Window.

setVisible (true);

}

//Function use to load all Records from File when Window Open.

void populateArray ()

{

// Query to find total no of records

try

{

if (conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

// Query to get village names

String qry1 = "SELECT \* FROM product;";

PreparedStatement get\_producerl = conn.prepareStatement(qry1);

ResultSet rs1 = get\_producerl.executeQuery();

for(total=0;rs1.next();total++);

rs1.close();conn.close();

}

catch(Exception e)

{

System.out.println(e);

}

String rows[][] = new String [500][10];

rowData = new String [total][10];

try

{

if (conn!= null)

conn.close();

conn = ConnectTODB.getConnection();

// Query to get village names

String qry = "SELECT product.prod\_id, product.prod\_name,"+

" product.wt\_per\_unit, product.price, product.ava\_qty, product.batch\_no, product.mfg\_date, product.exp\_date"+

" FROM product;";

PreparedStatement get\_producer = conn.prepareStatement(qry);

ResultSet rs = get\_producer.executeQuery(); // execute query

int i;

for (i = 0; rs.next(); i++)

{

rowData[i][0] = rs.getString(1);

rowData[i][1] = rs.getString(2);

rowData[i][2] = rs.getString(3);

rowData[i][3] = rs.getString(4);

rowData[i][4] = rs.getString(5);

rowData[i][5] = rs.getString(6);

rowData[i][6] = rs.getString(7);

rowData[i][7] = rs.getString(8);

}

rs.close();conn.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

//Function to Create the Table and Add Data to Show.

private JTable makeTable ()

{

//String Type Array use to Give Table Column Names.

String cols[] = {"Prod\_no.", "Prod\_name", "wt\_per\_unit", "price","ava\_qty","Batch No","Mfg\_Date","Exp\_Date"};

dtmCustomer = new DefaultTableModel (rowData, cols);

tbCustomer = new JTable (dtmCustomer)

{

public boolean isCellEditable (int iRow, int iCol)

{

return false; //Disable All Columns of Table.

}

};

//Sizing the Columns of Table.

(tbCustomer.getColumnModel().getColumn(0)).setPreferredWidth (180);

(tbCustomer.getColumnModel().getColumn(1)).setPreferredWidth (275);

(tbCustomer.getColumnModel().getColumn(2)).setPreferredWidth (275);

(tbCustomer.getColumnModel().getColumn(3)).setPreferredWidth (200);

(tbCustomer.getColumnModel().getColumn(4)).setPreferredWidth (180);

(tbCustomer.getColumnModel().getColumn(5)).setPreferredWidth (180);

(tbCustomer.getColumnModel().getColumn(6)).setPreferredWidth (180);

(tbCustomer.getColumnModel().getColumn(7)).setPreferredWidth (180);

tbCustomer.setRowHeight (20);

tbCustomer.setSelectionMode (ListSelectionModel.SINGLE\_SELECTION);

return tbCustomer;

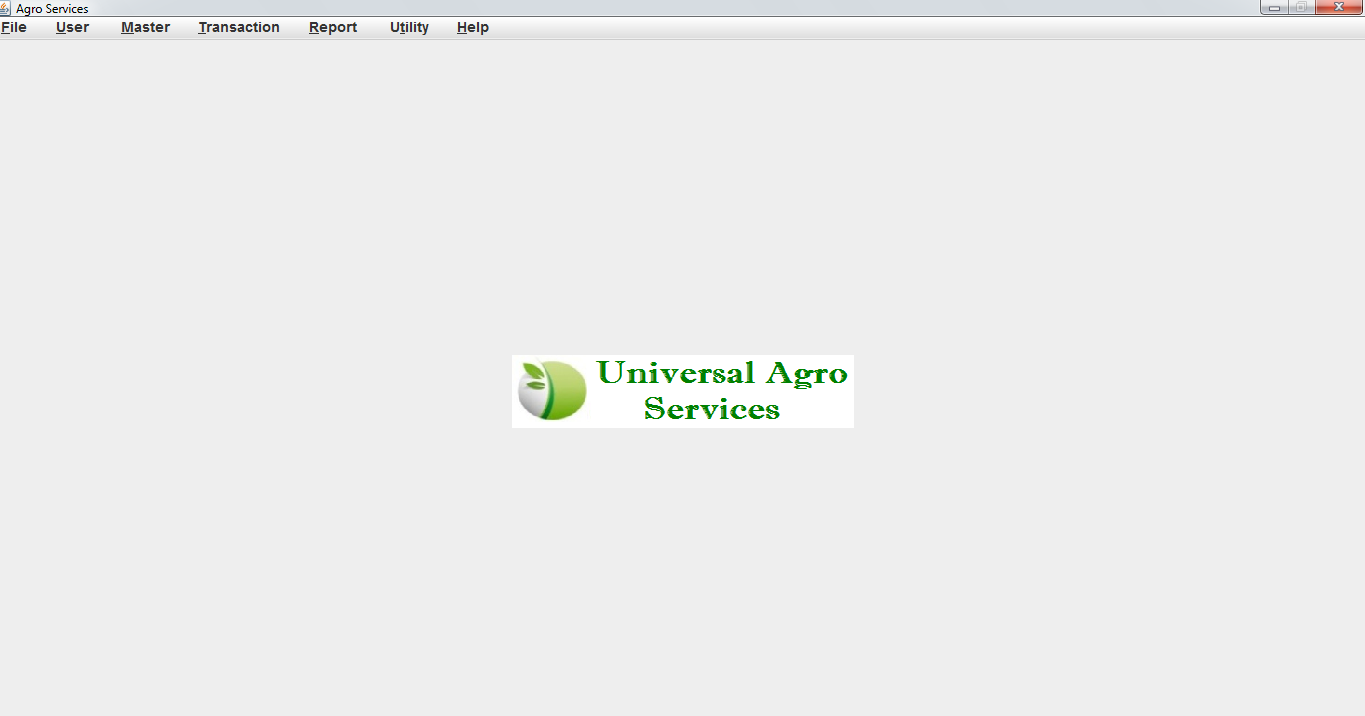
}

}

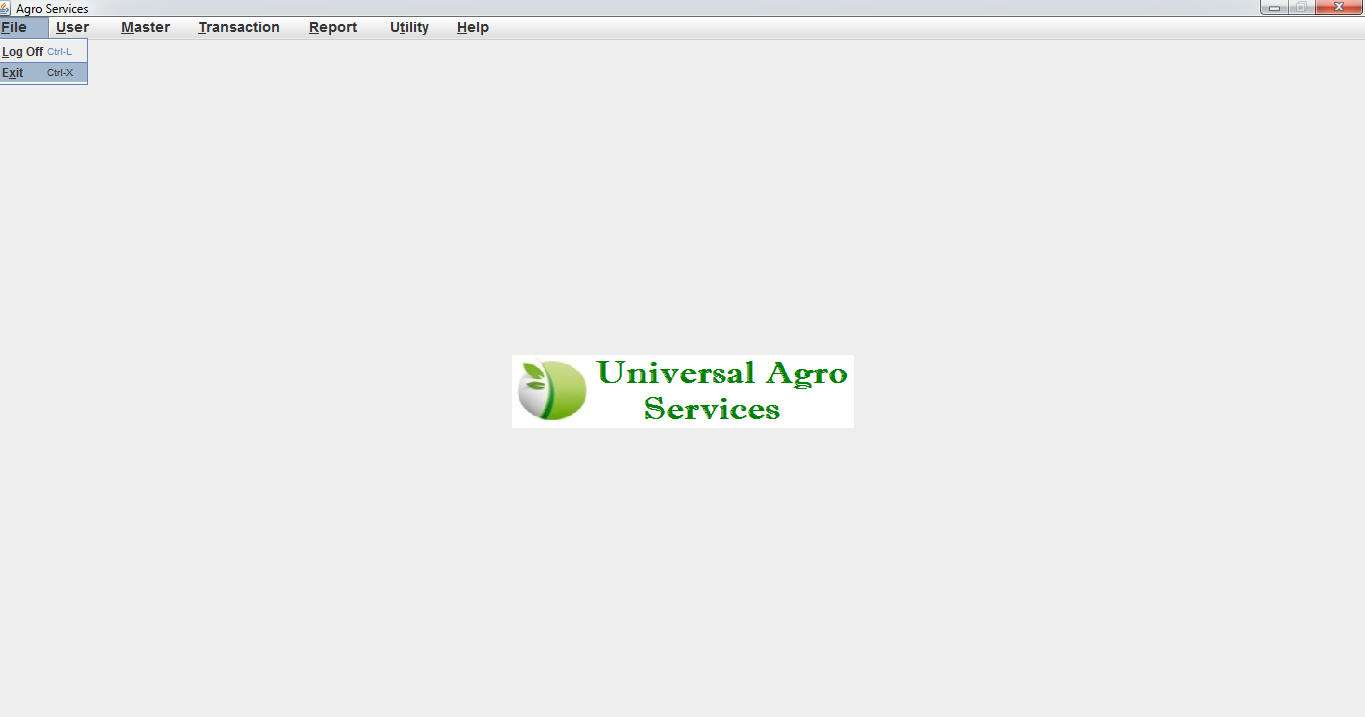
**LOGIN SCREEN**

****

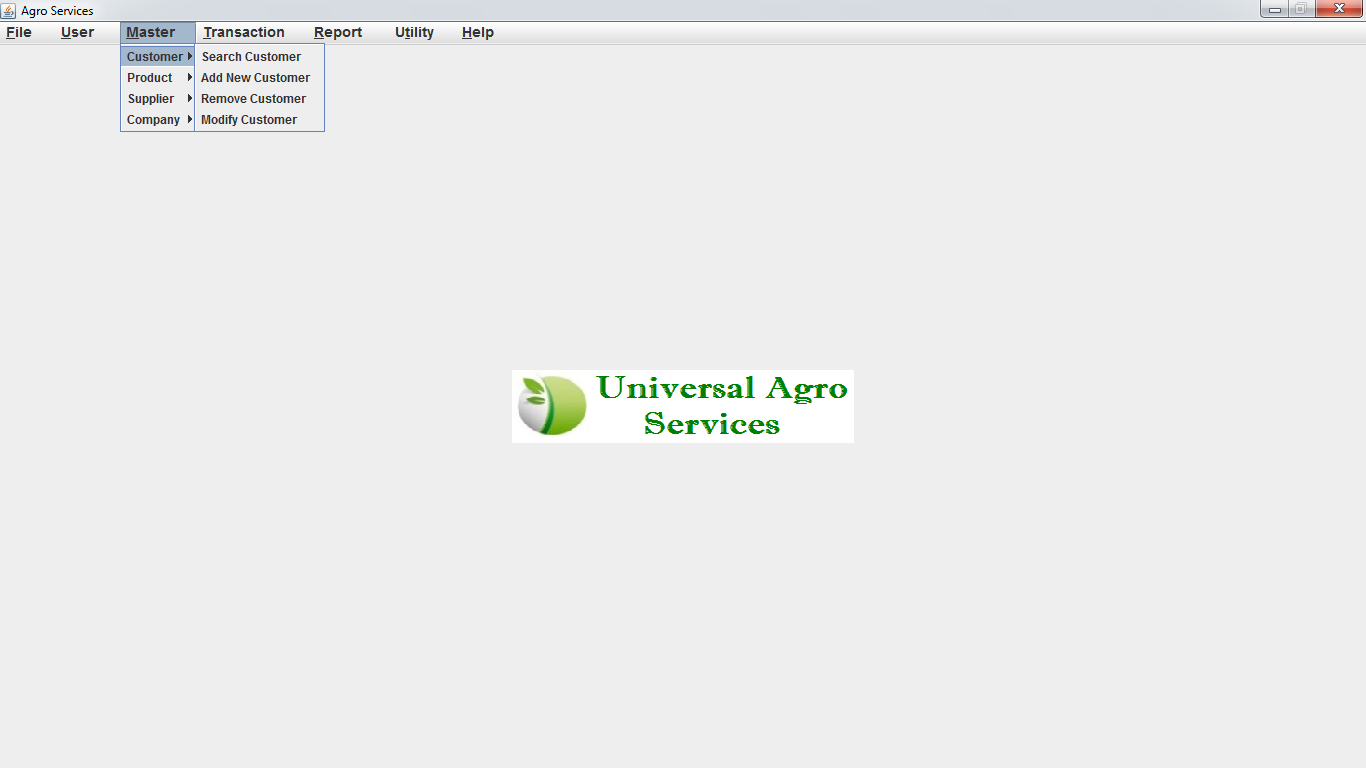
**MASTER ENTRY**

****

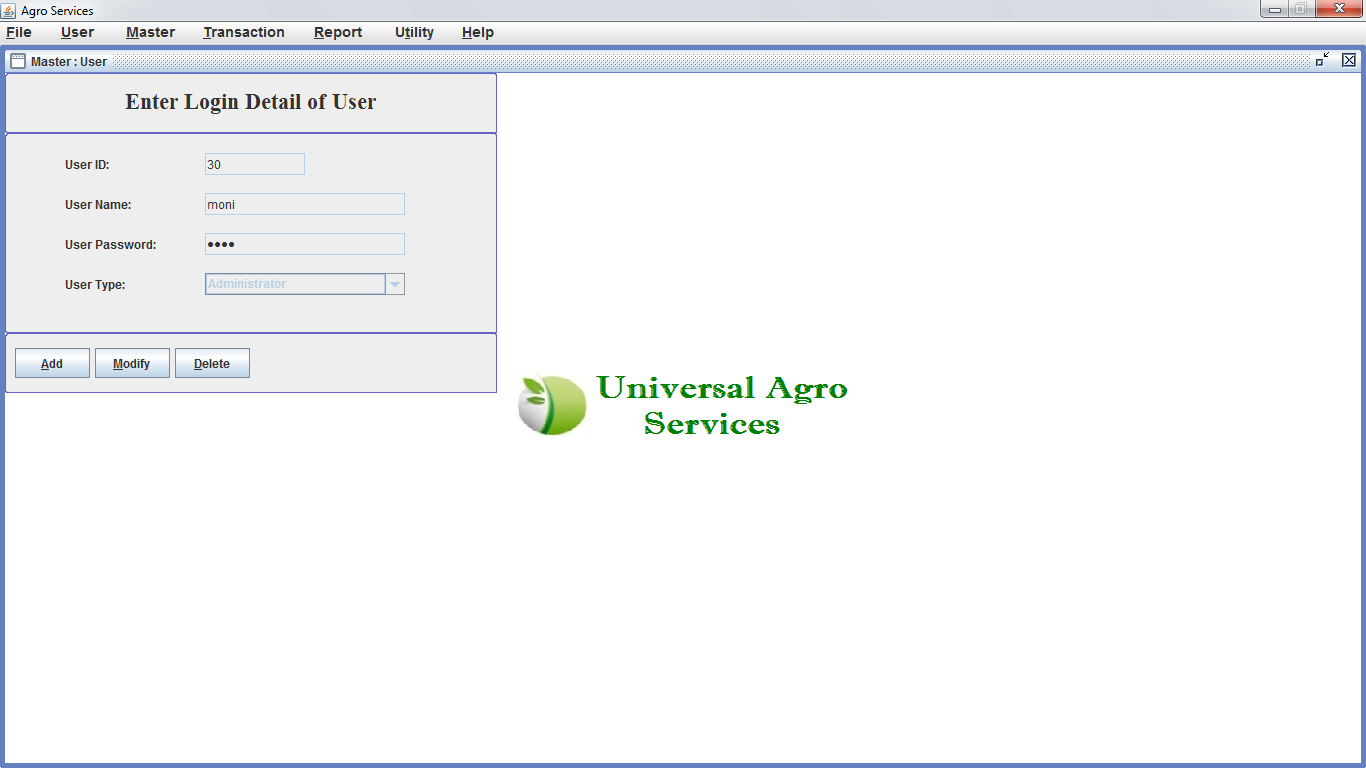
**FILE MENU**

****

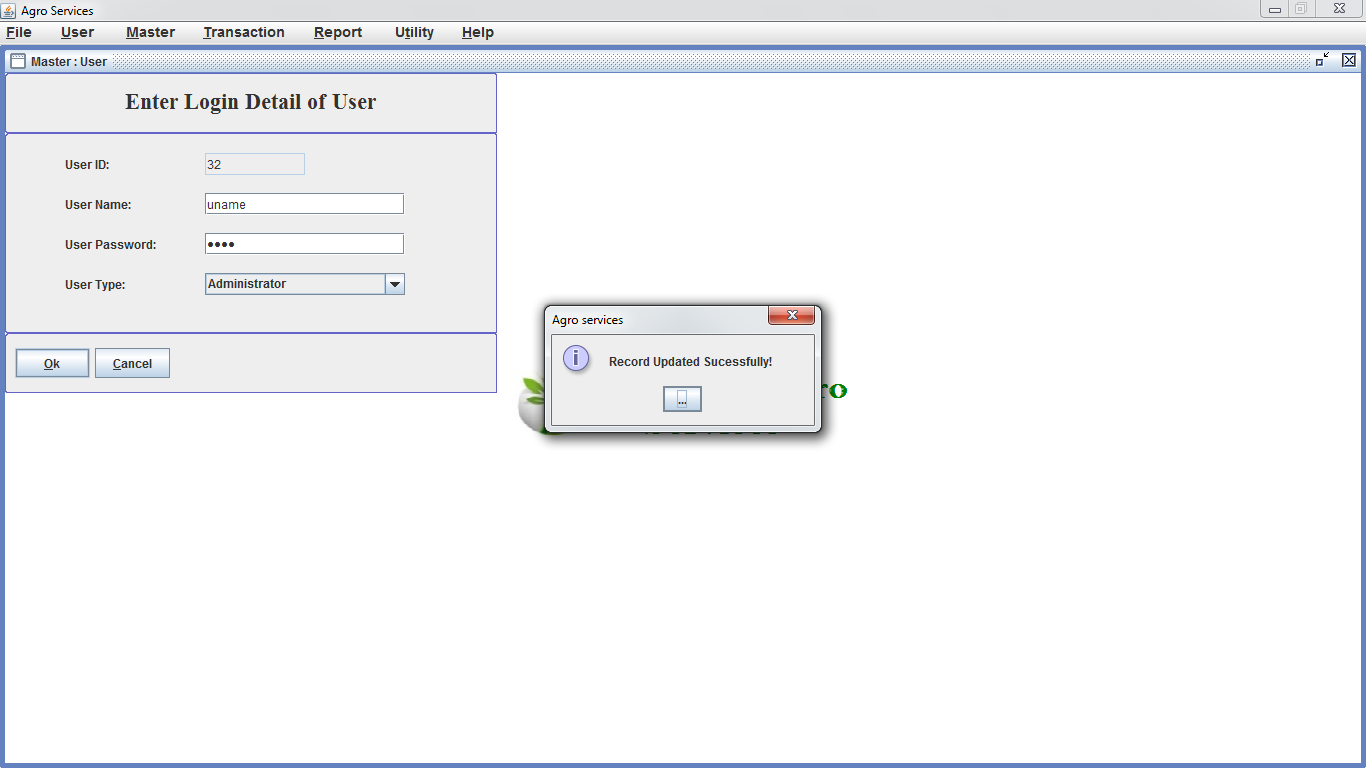
**MASTER ENTRY**

****

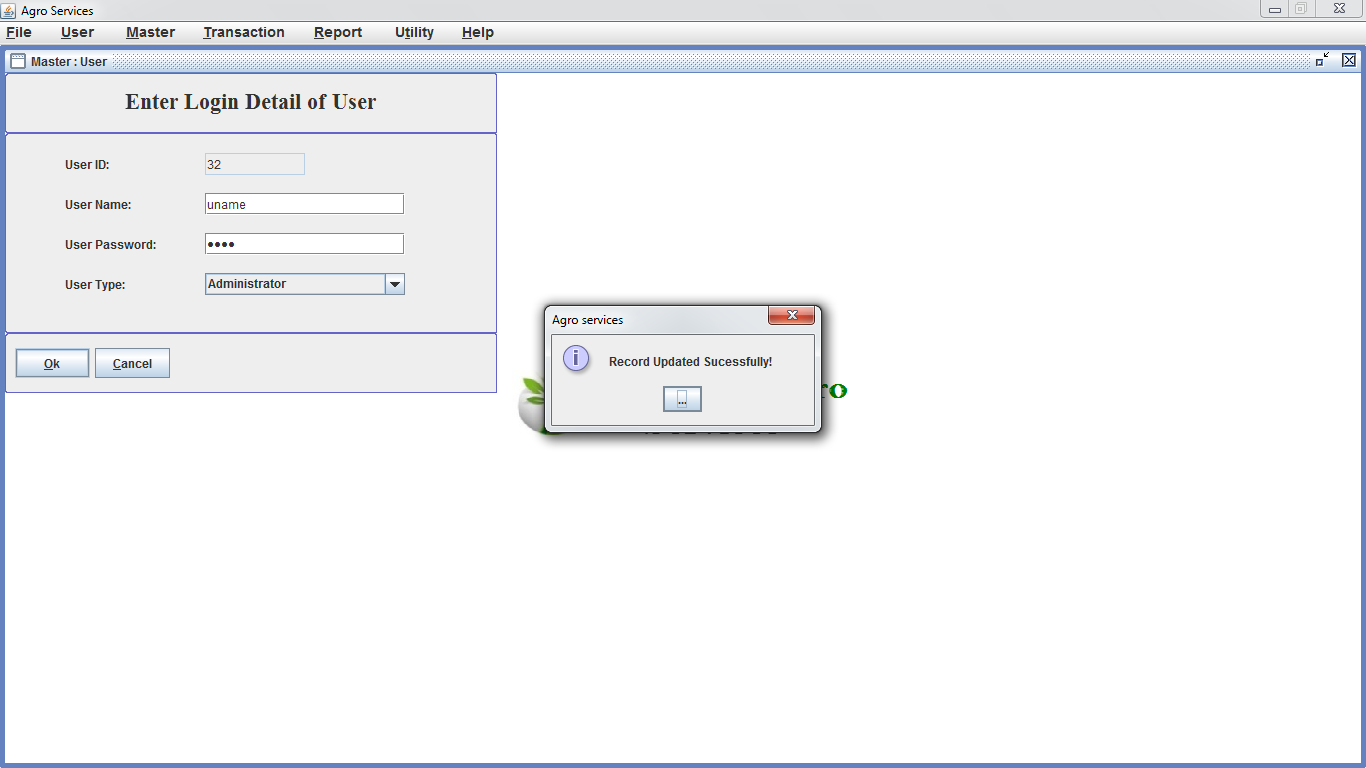
**USER MASTER**

****

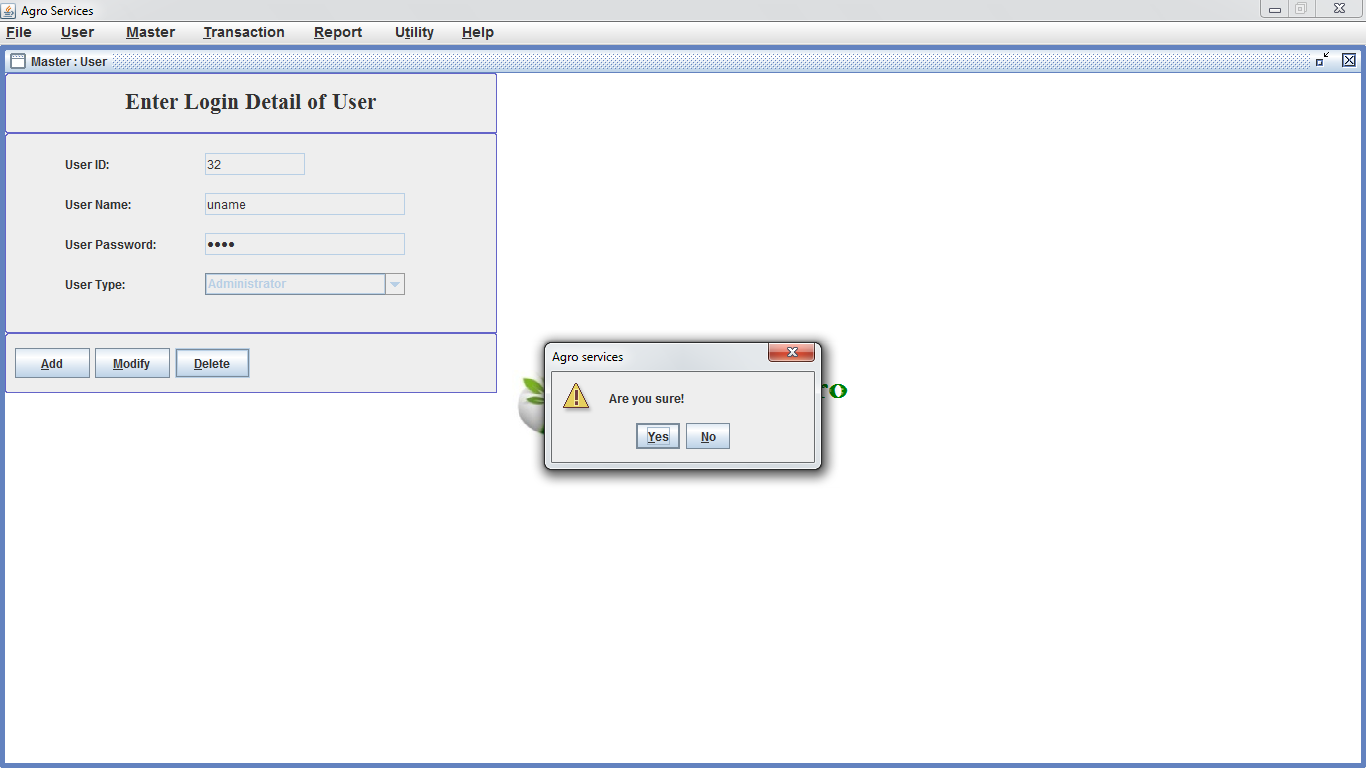
**USER ENTRY**

****

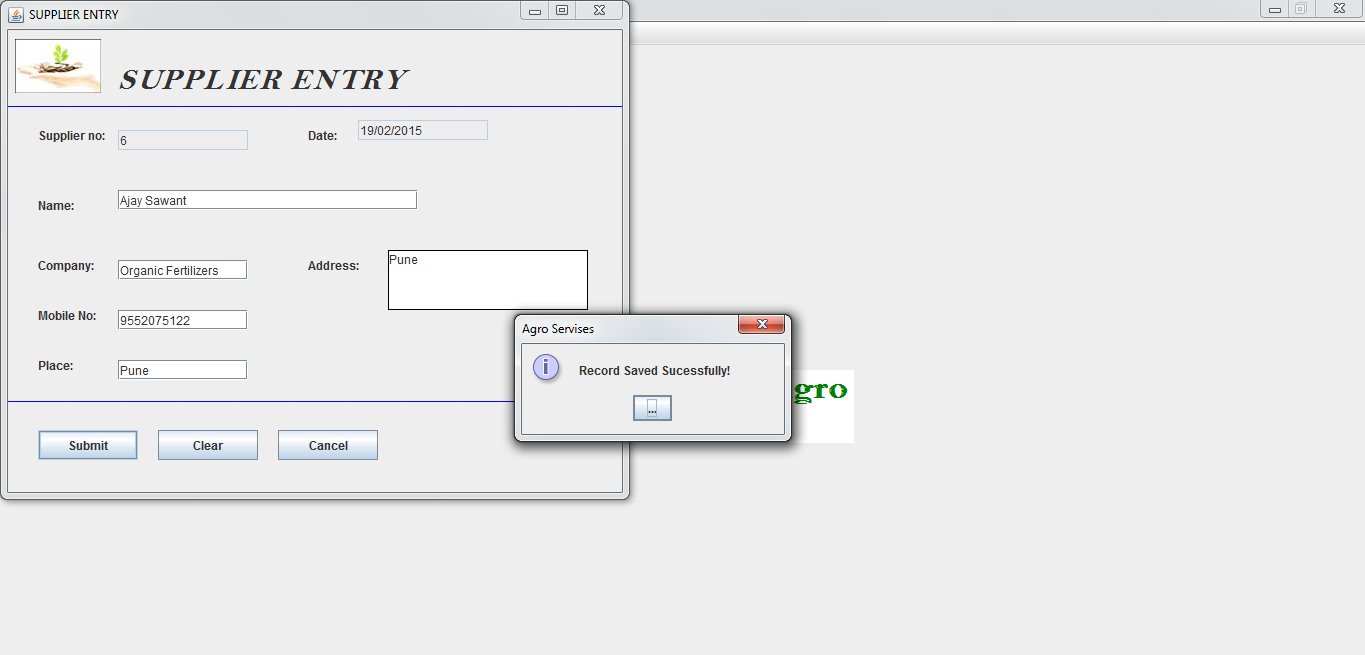
**MODIFY USER**

****

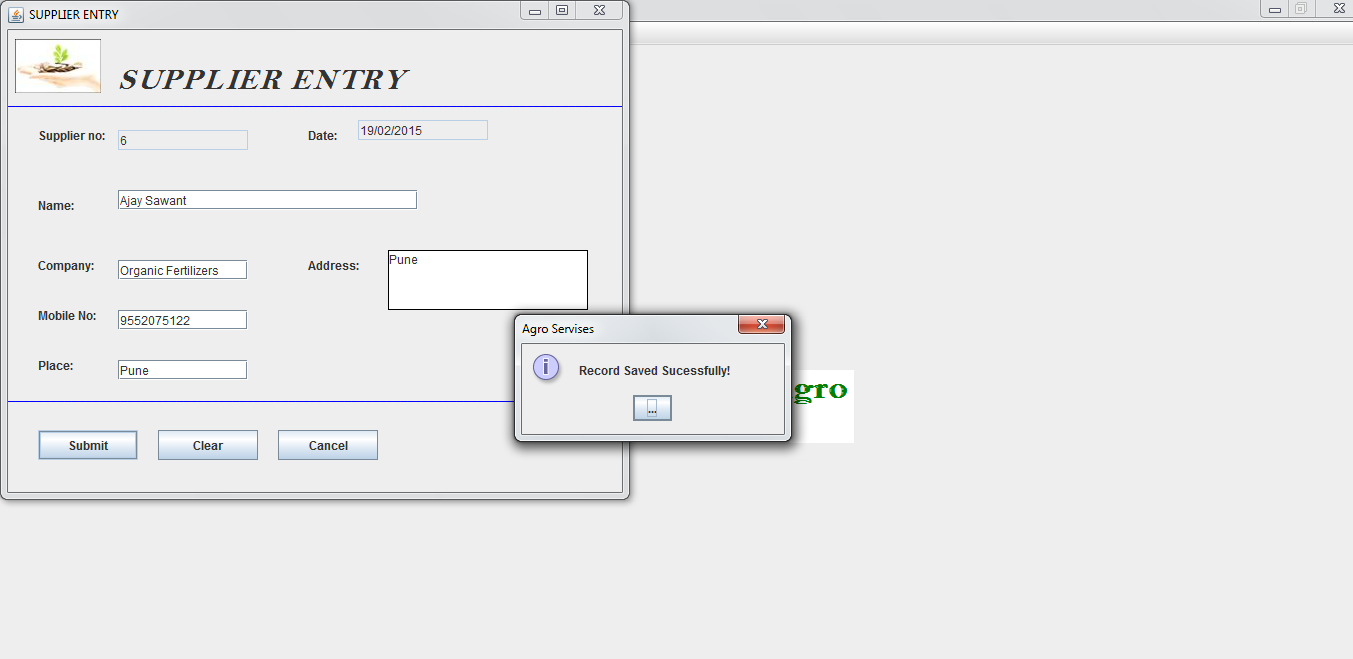
**DELETE USER**

****

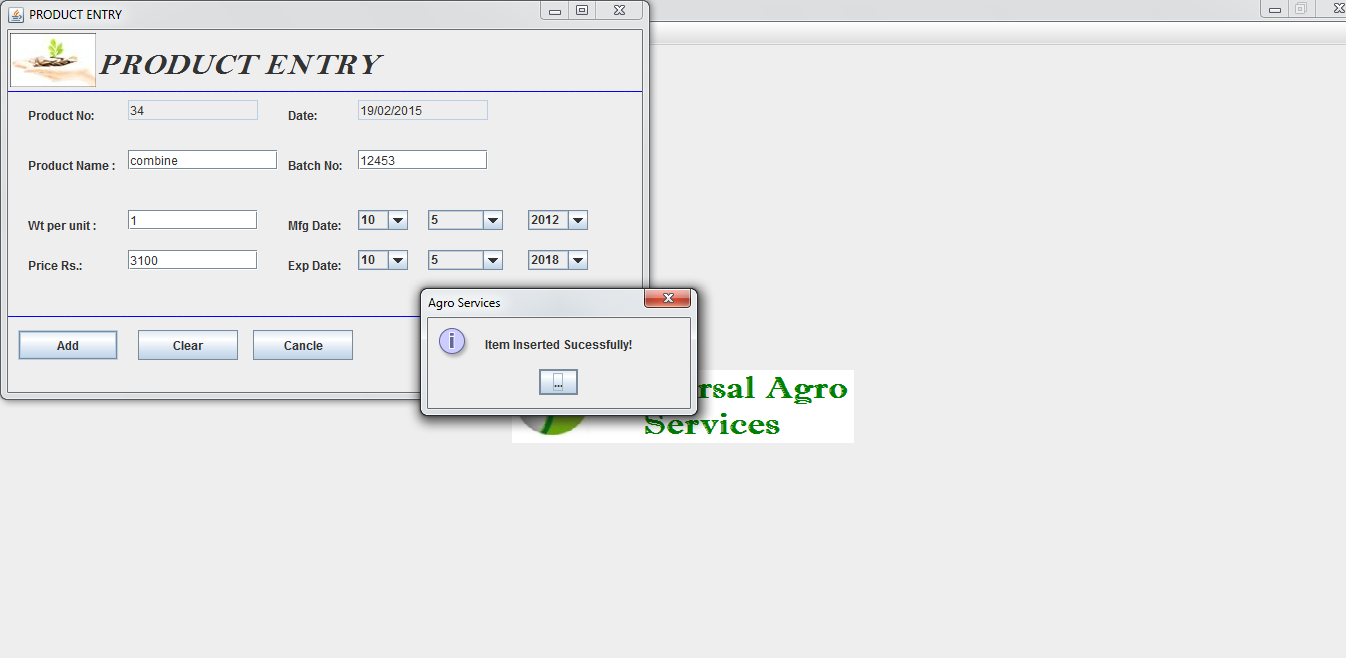
**ADD NEW COMPANY**

****

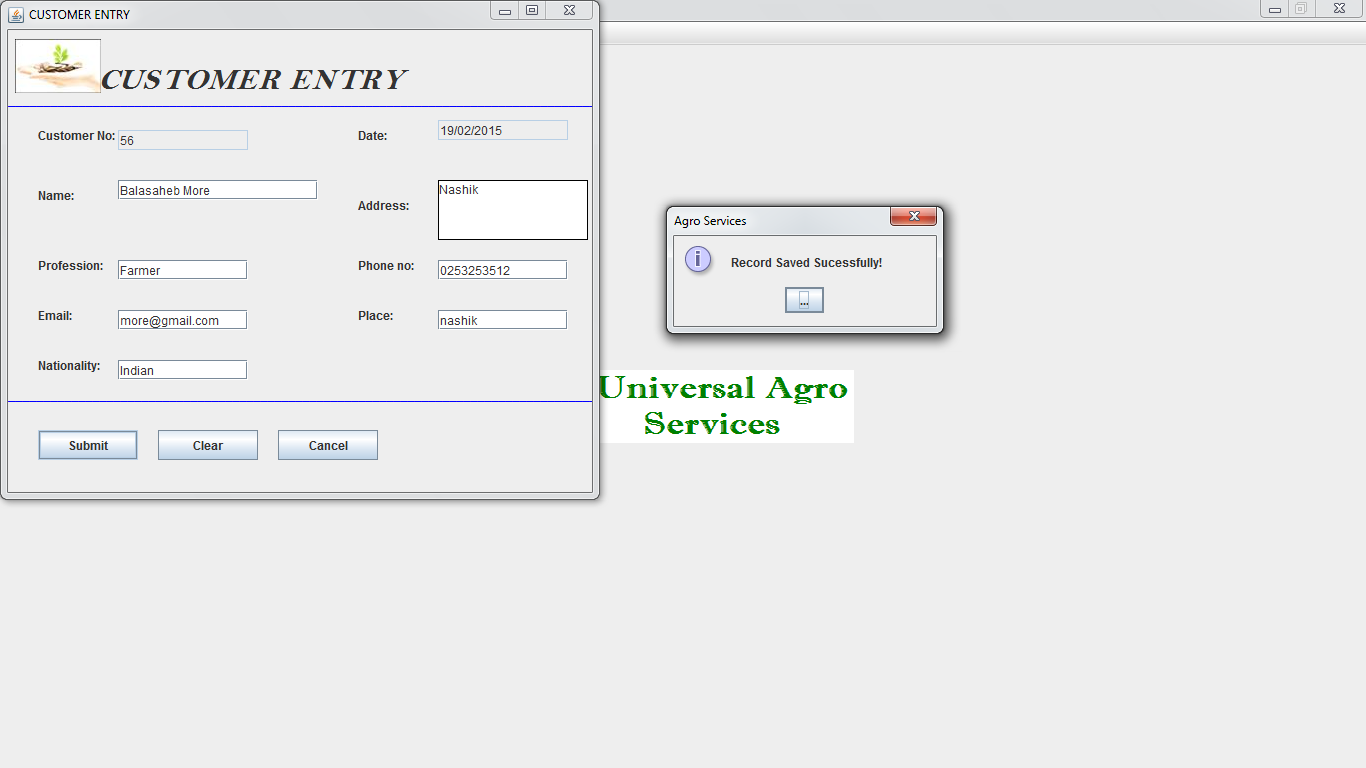
**ADD NEW SUPPLIER**

****

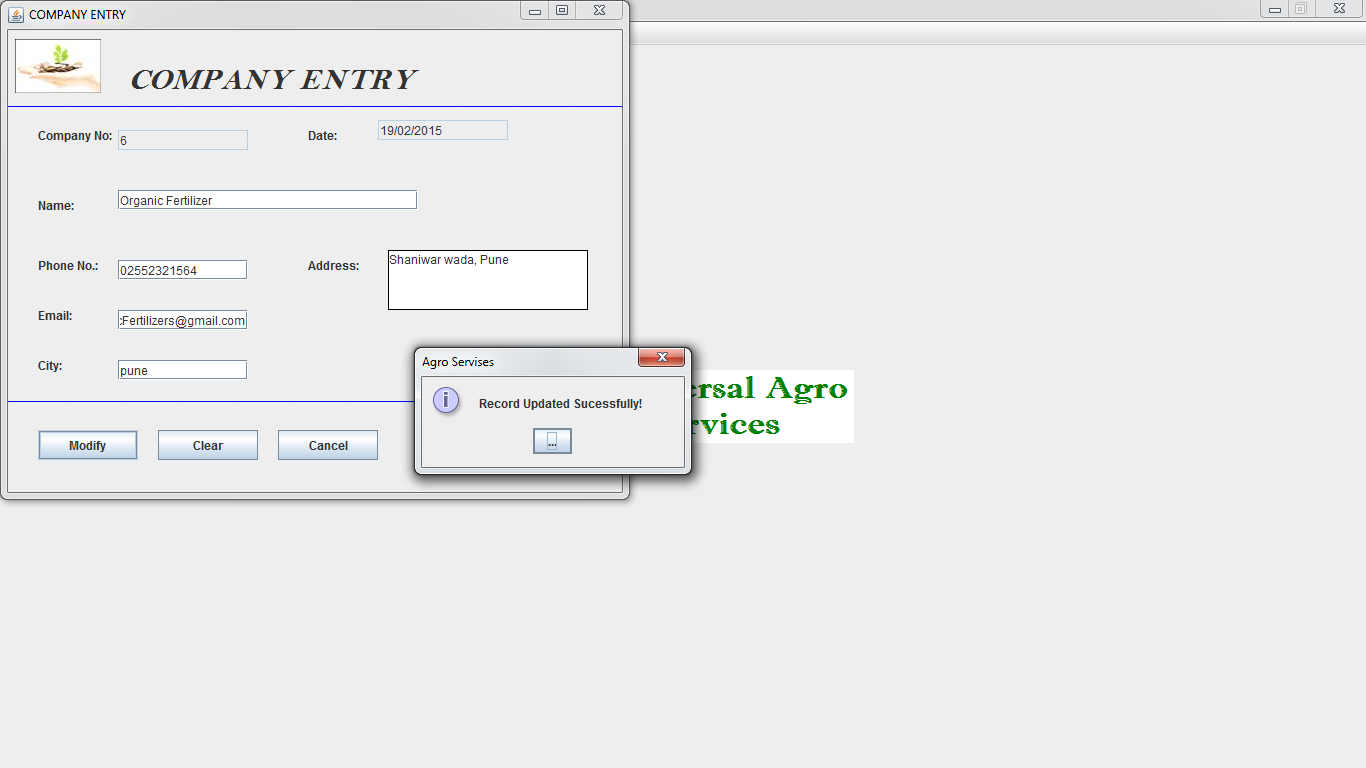
**ADD NEW PRODUCT**

****

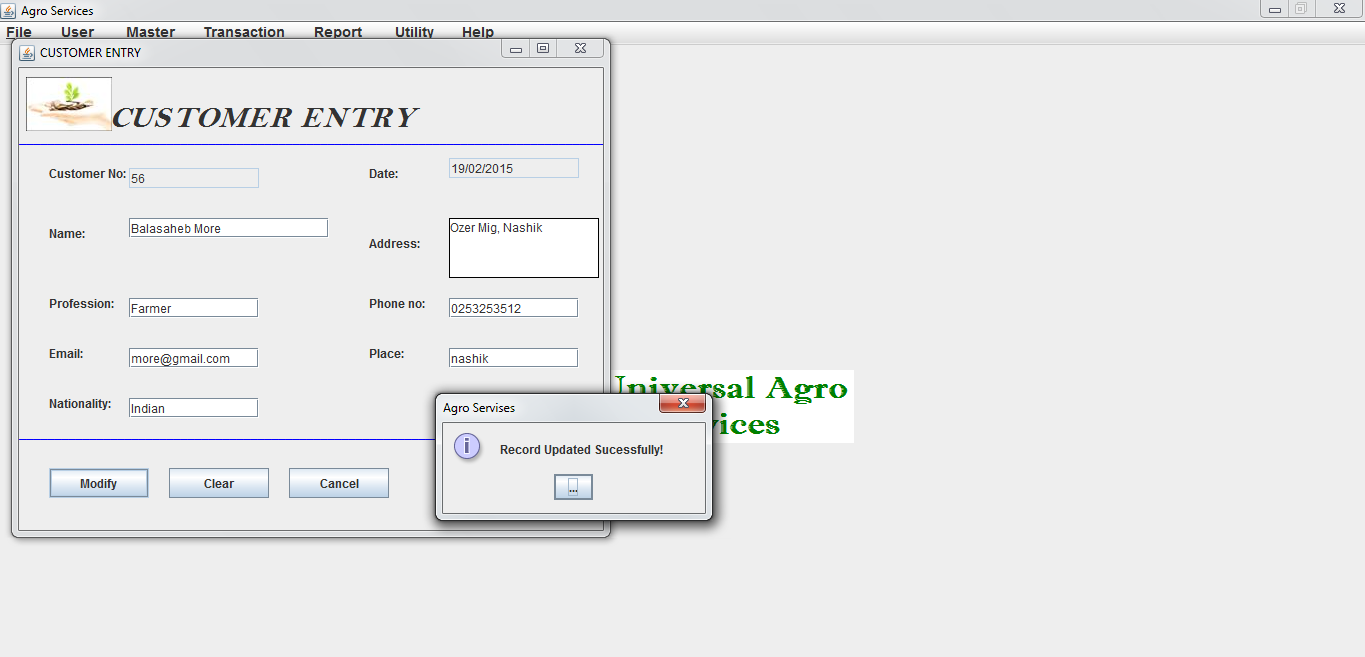
**ADD NEW CUSTOMER**

****

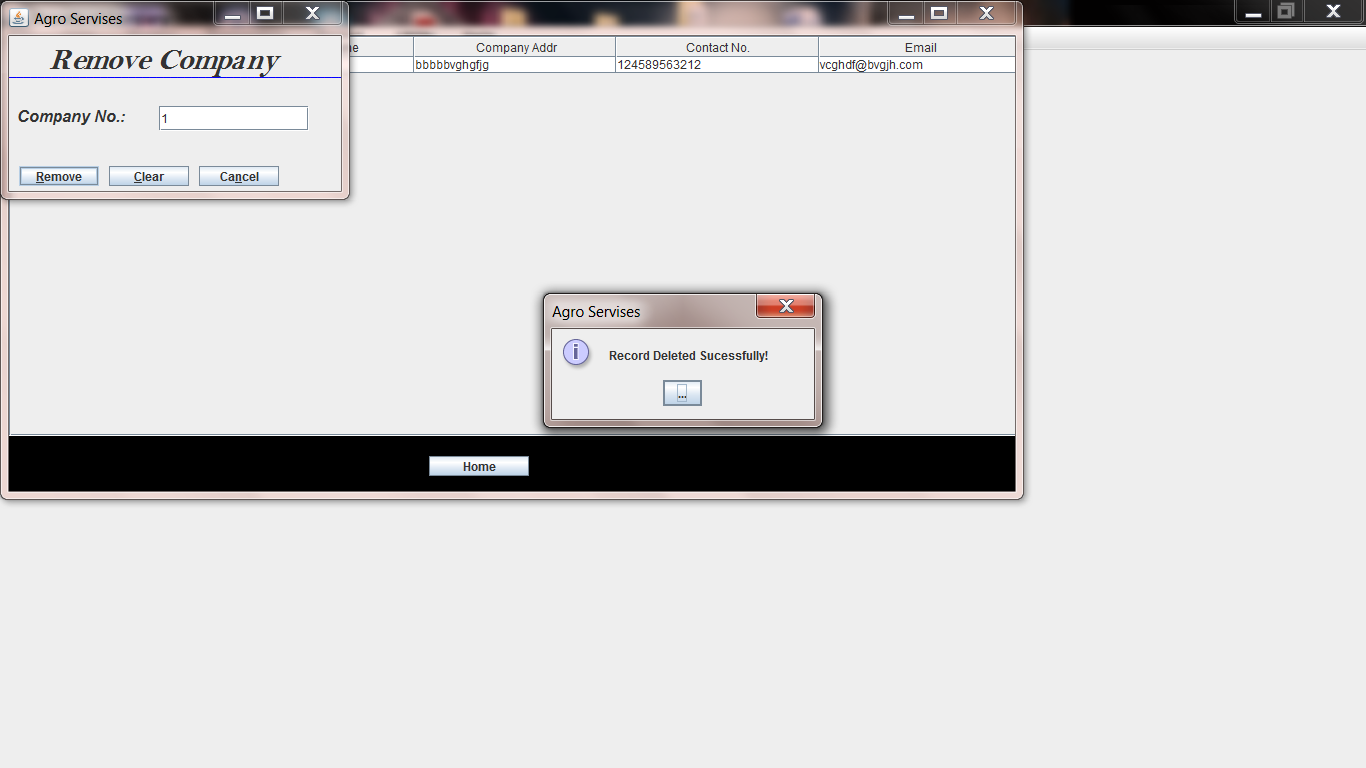
**MODIFY COMPANY**

****

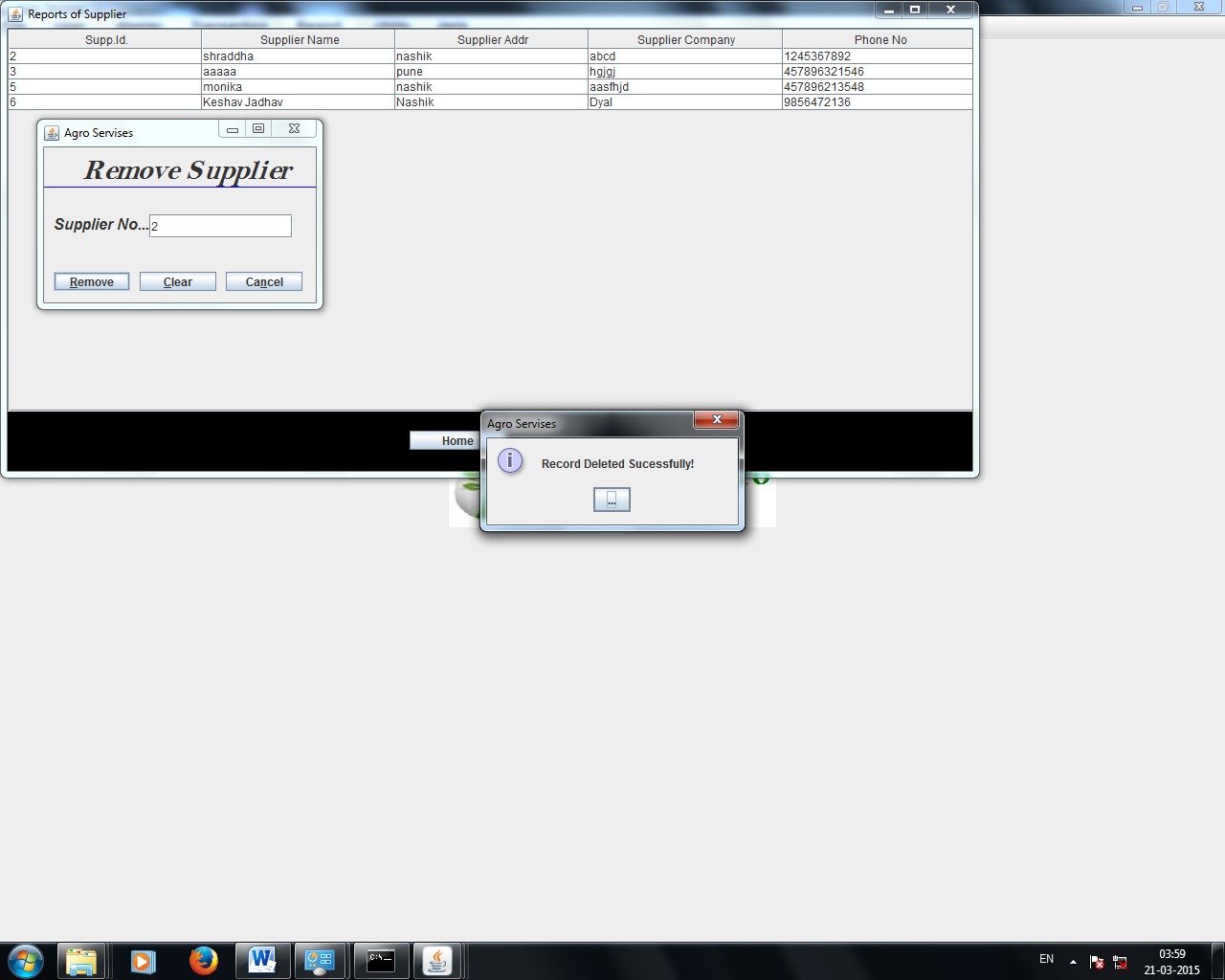
**MODIFY CUSTOMER**

****

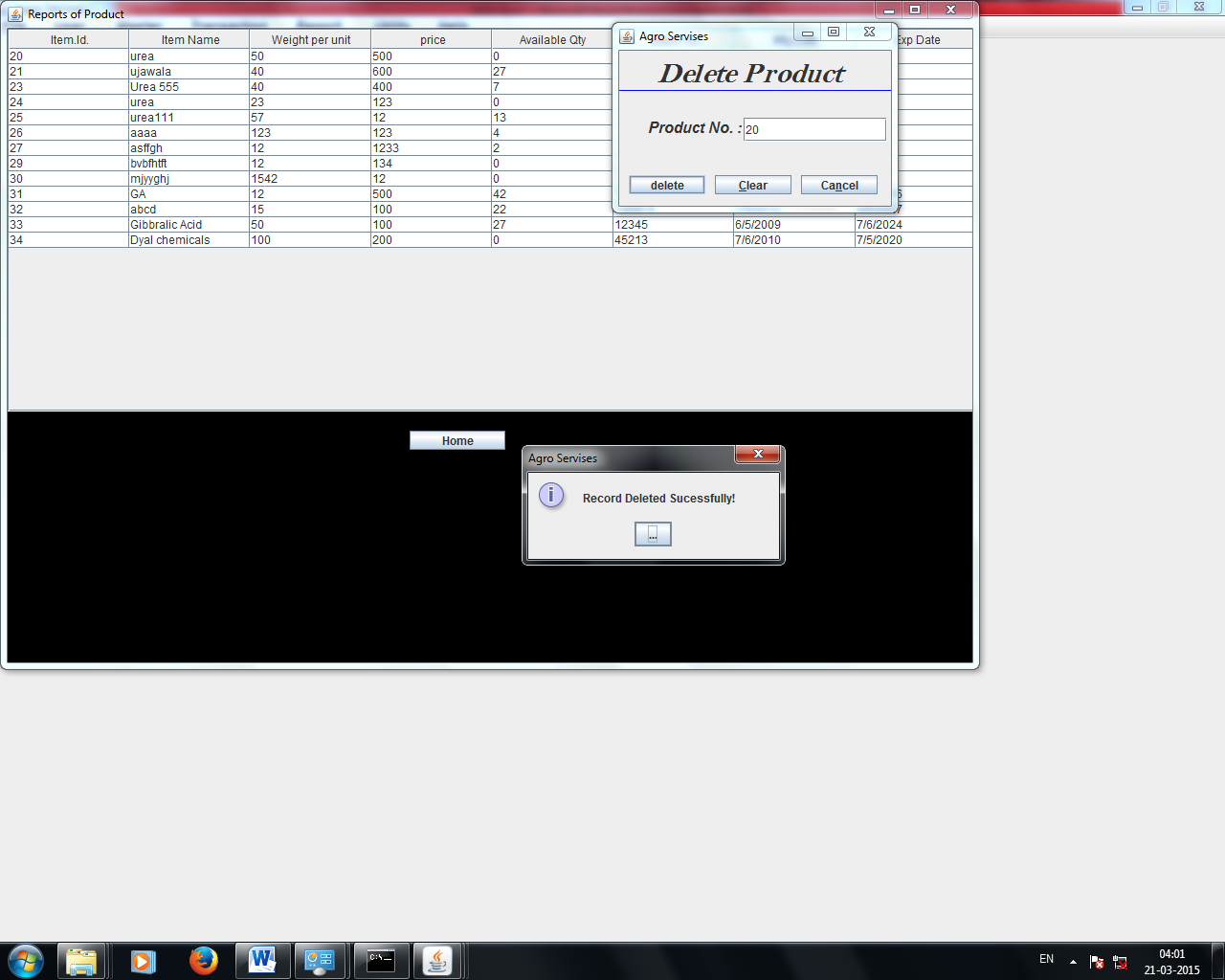
**REMOVE COMPANY**



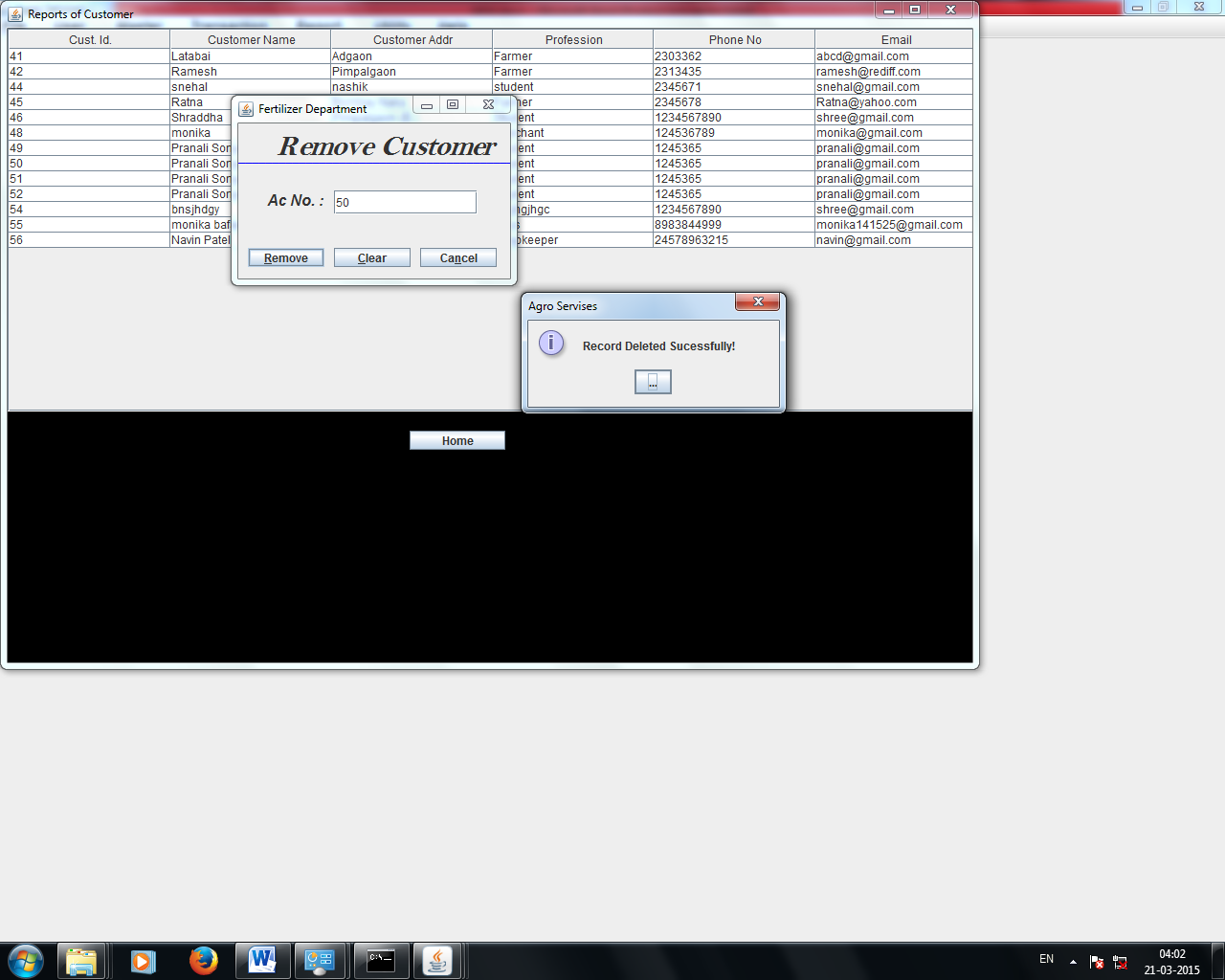
**REMOVE SUPPLIER**



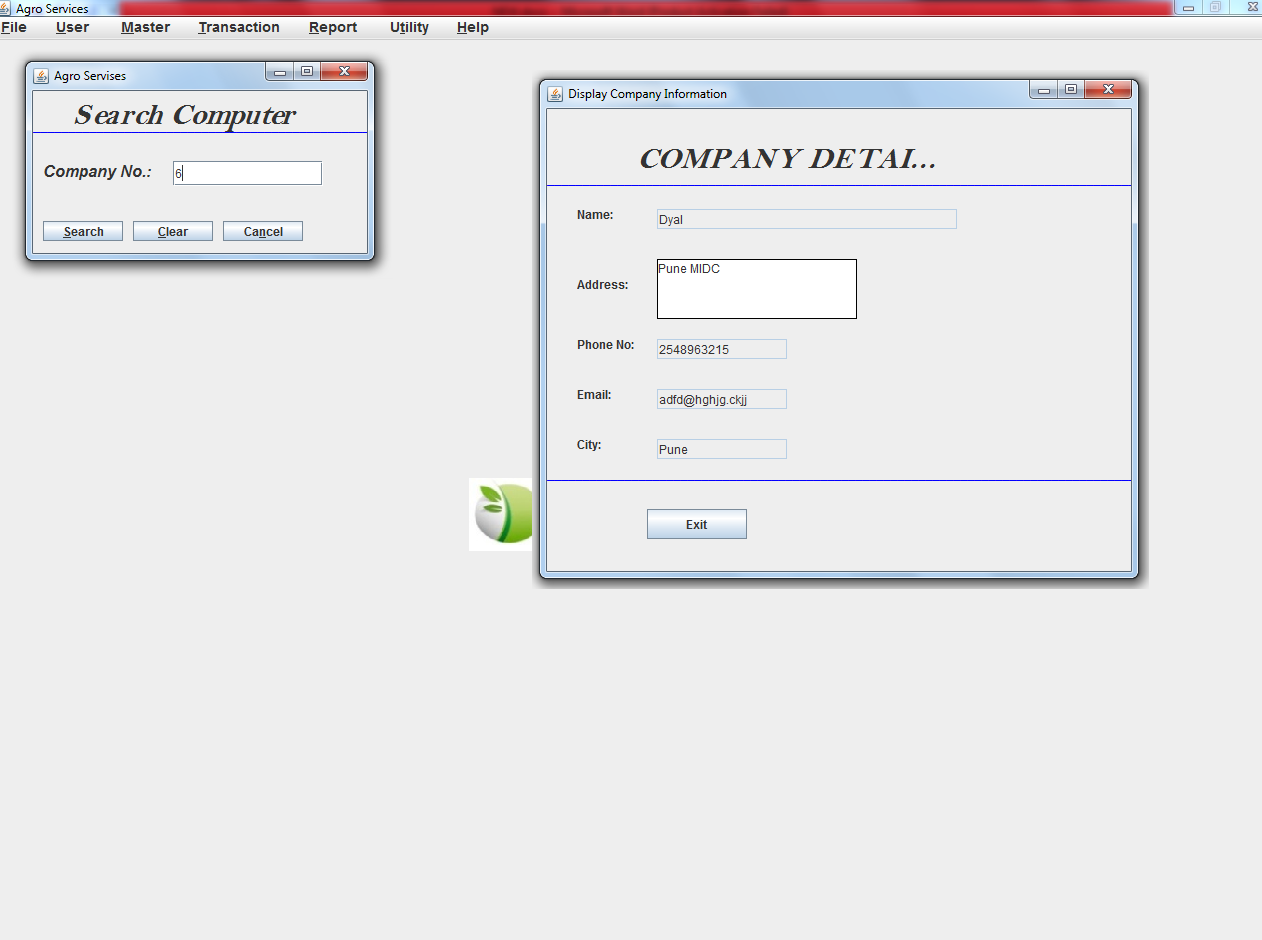
**REMOVE PRODUCT**



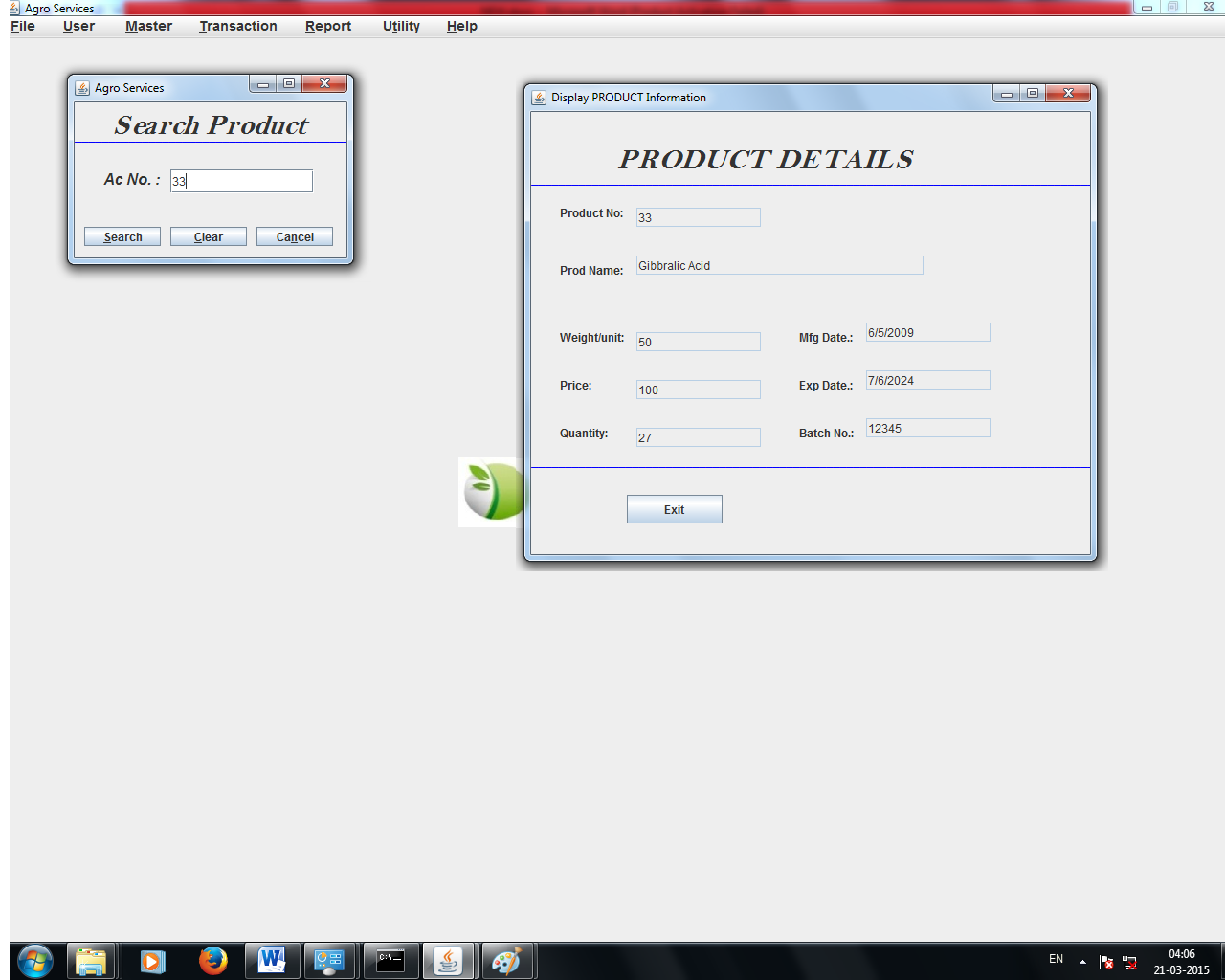
**REMOVE CUSTOMER**



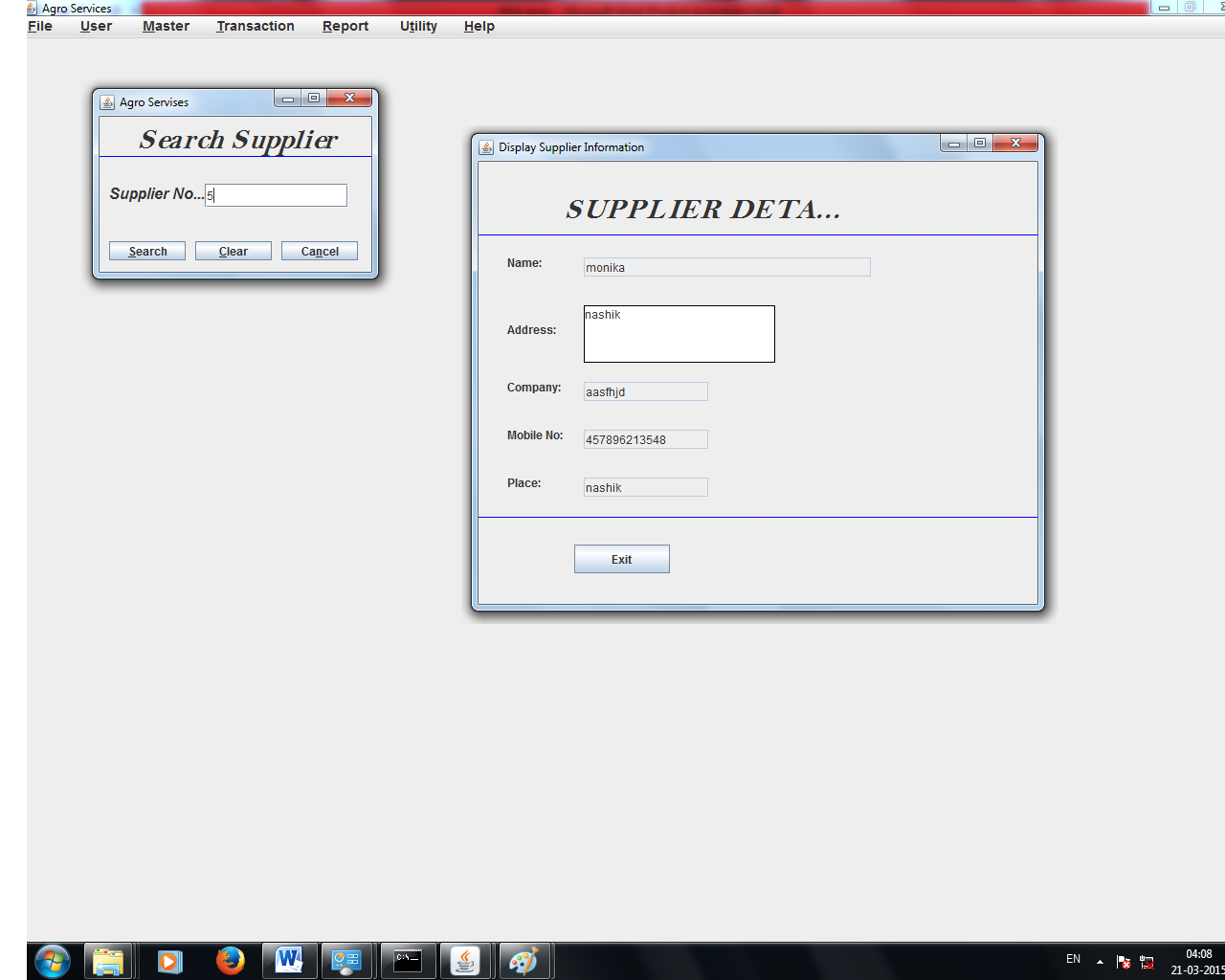
**SEARCH COMPANY**

****

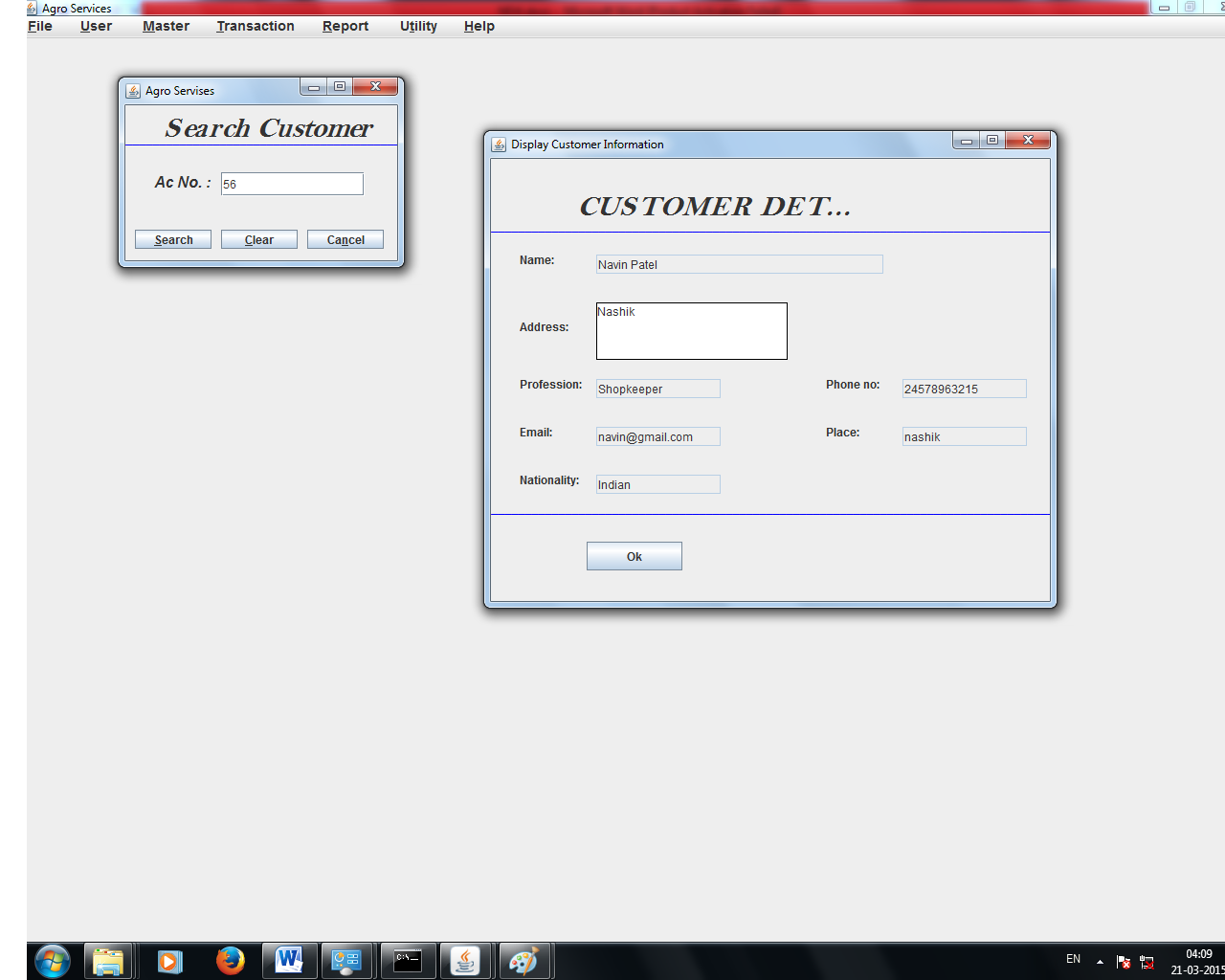
**SEARCH PRODUCT**

****

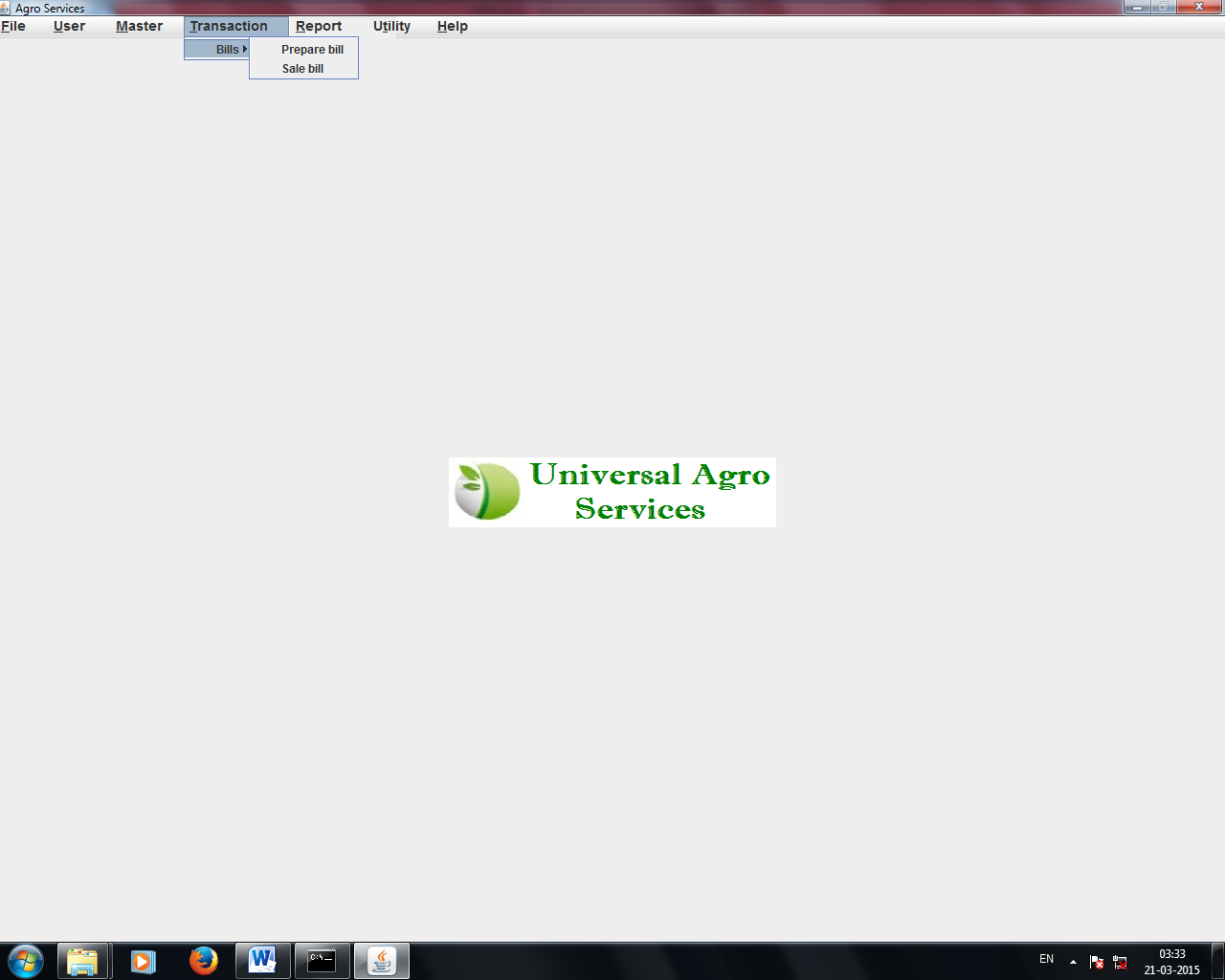
**SEARCH SUPPLIER**

****

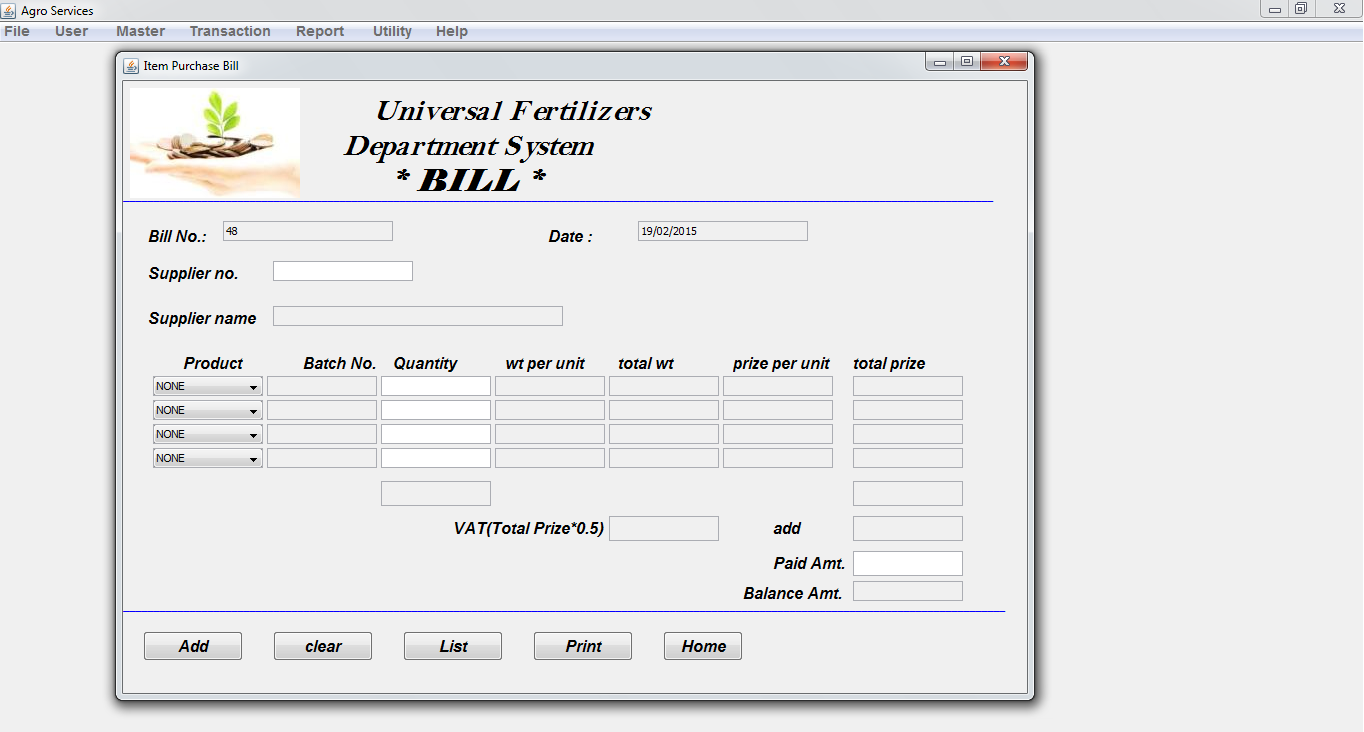
**SEARCH CUSTOMER**

****

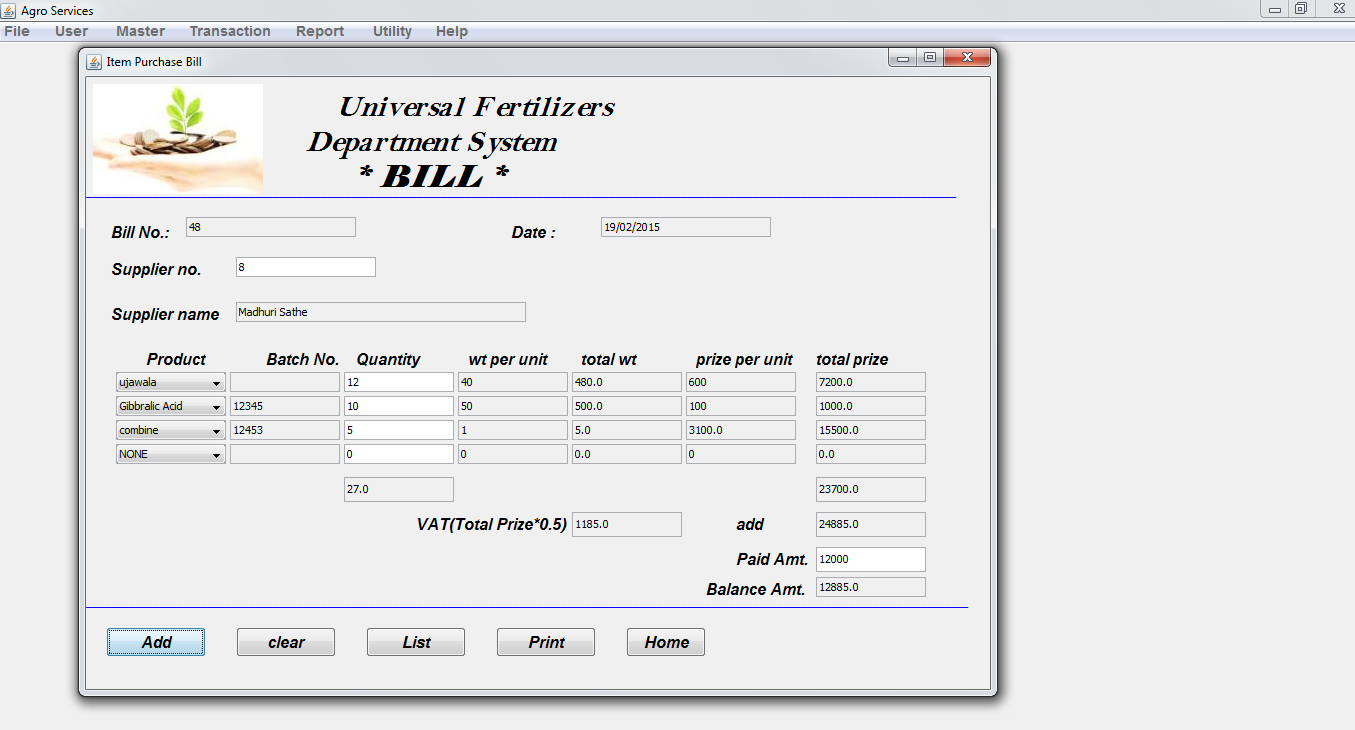
**TRANSACTIONS**



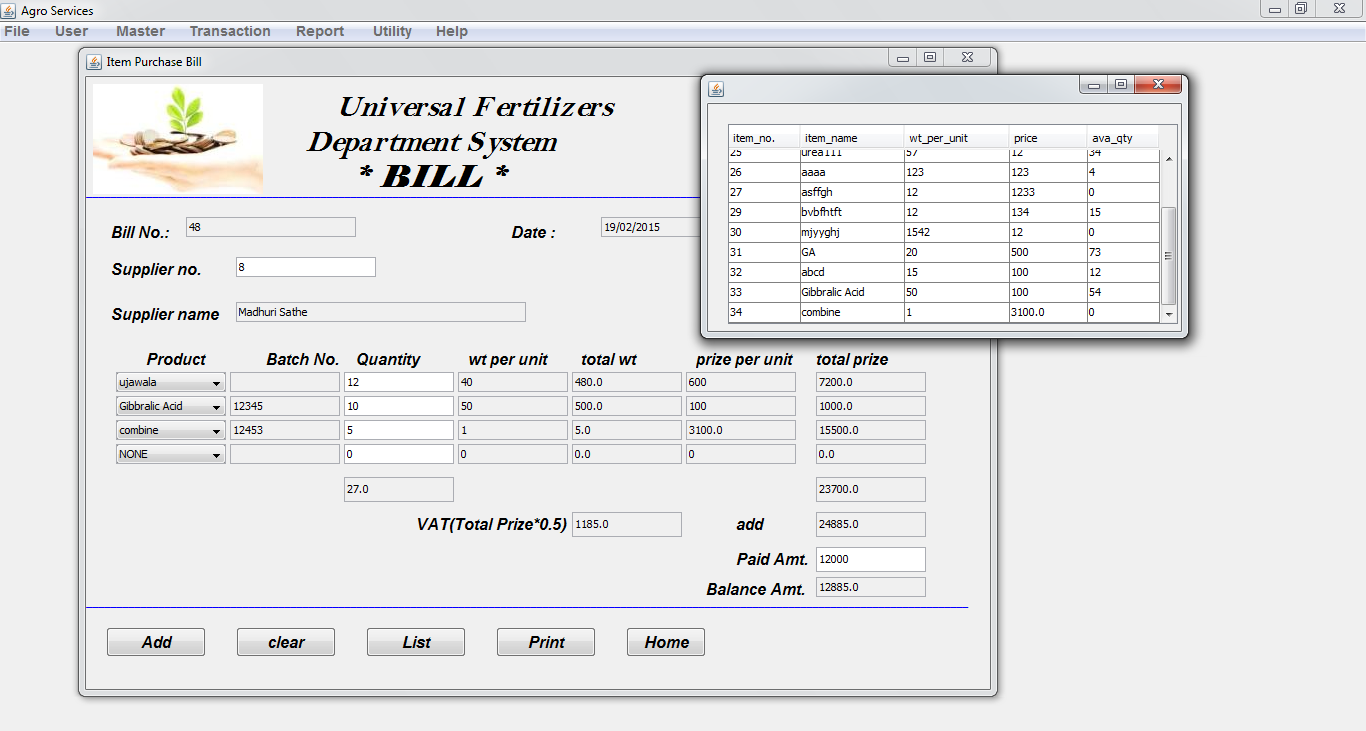
**PO SCREEN**

****

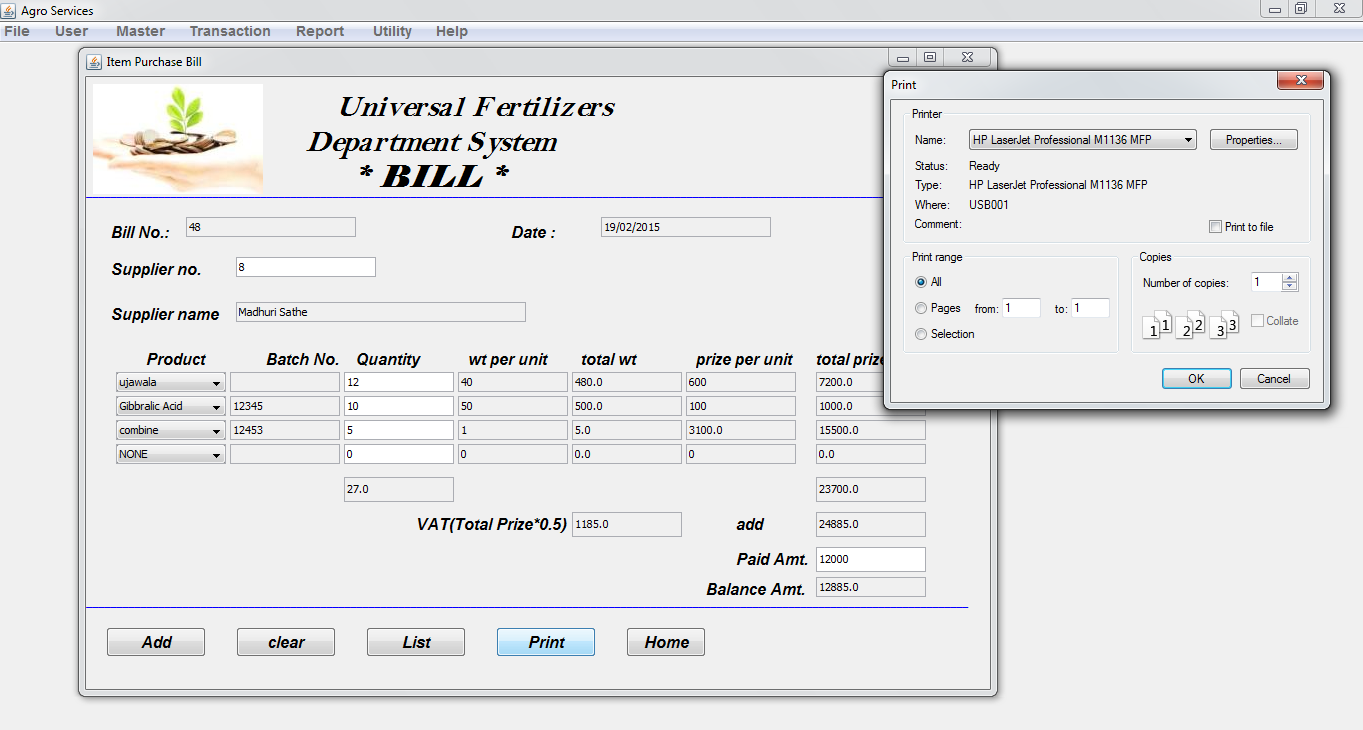
**PO BILL Entry**

****

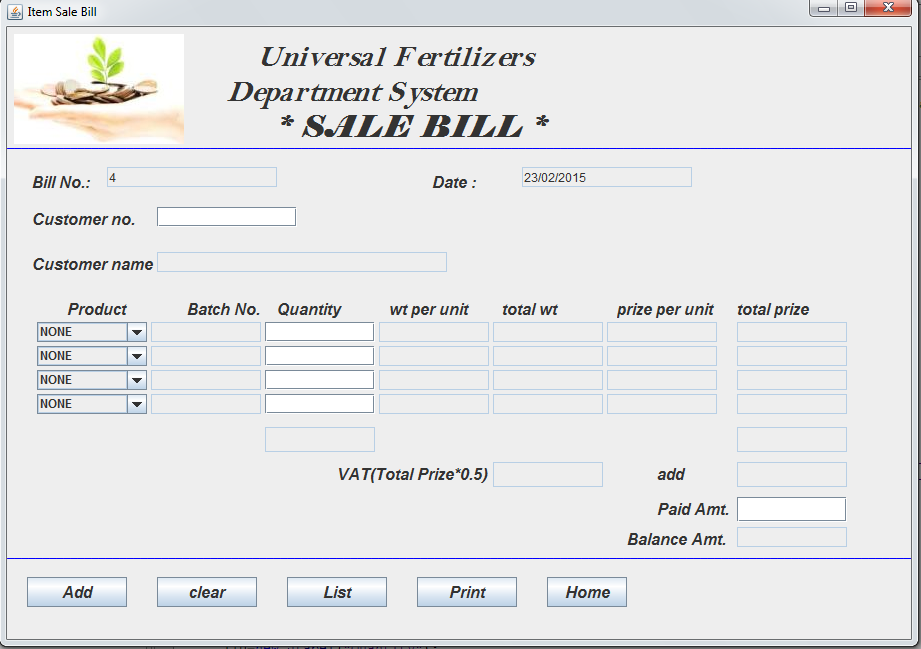
**PO BILL LIST**

****

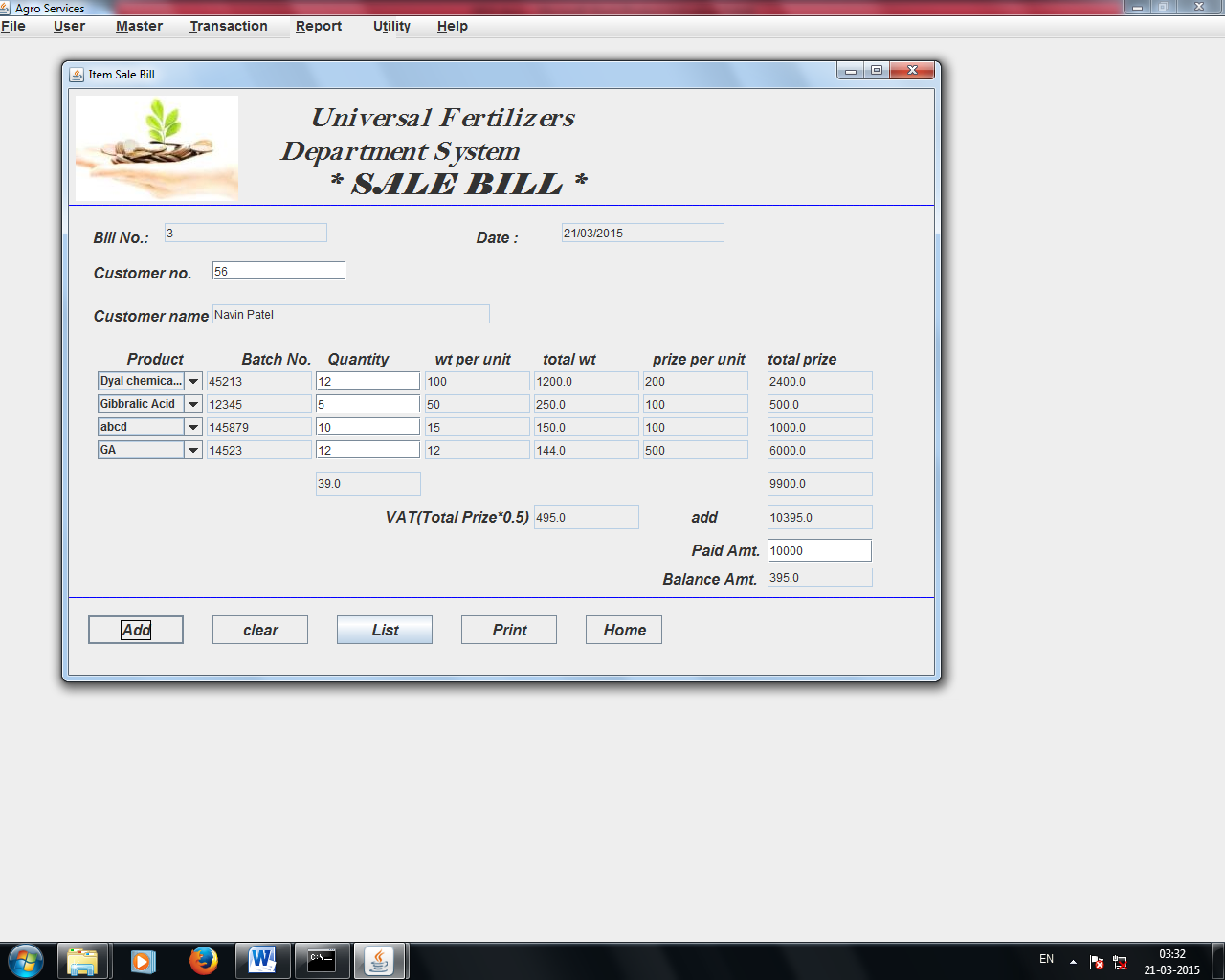
**PO BILL PRINT**

****

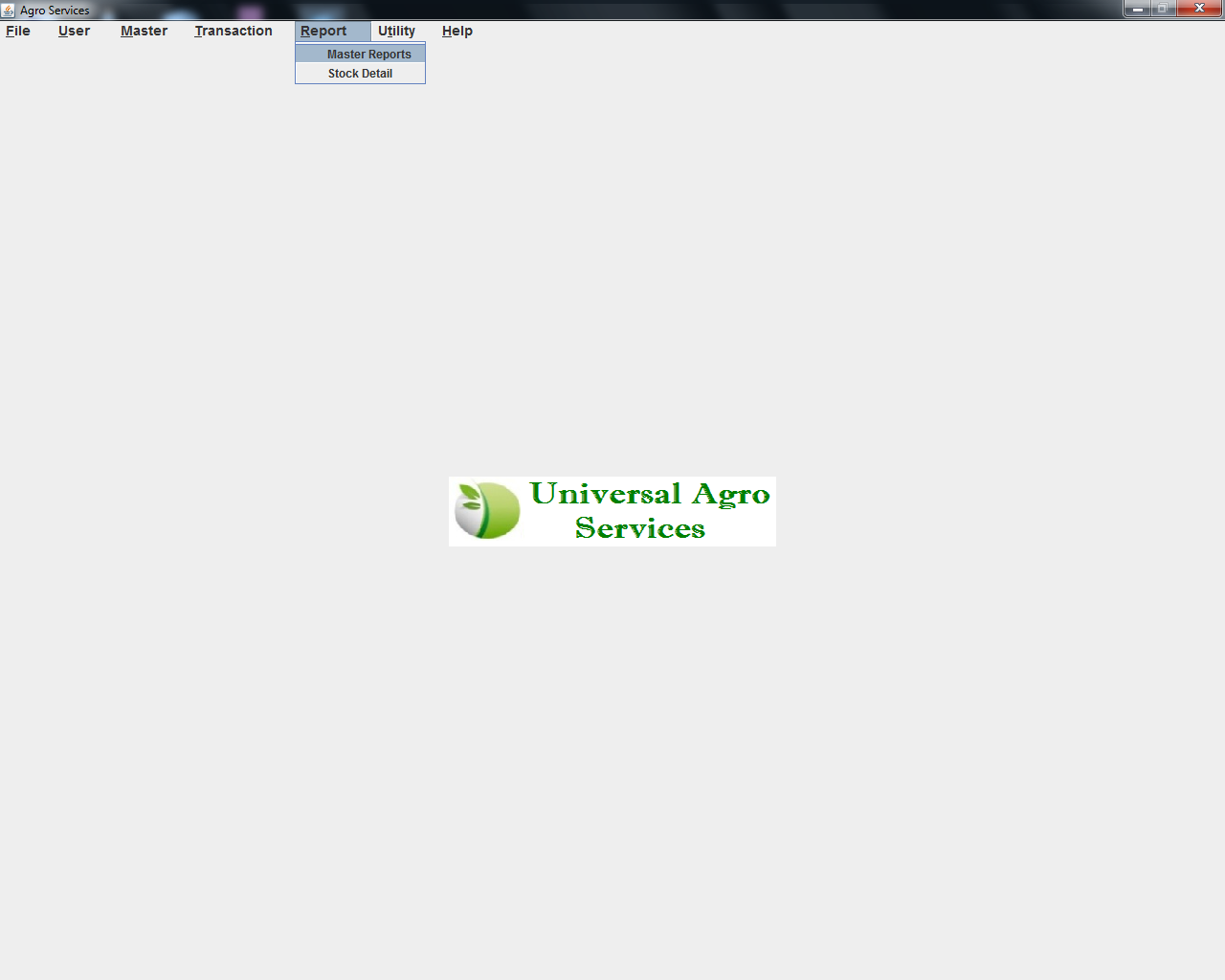
**SALE BILL**

****

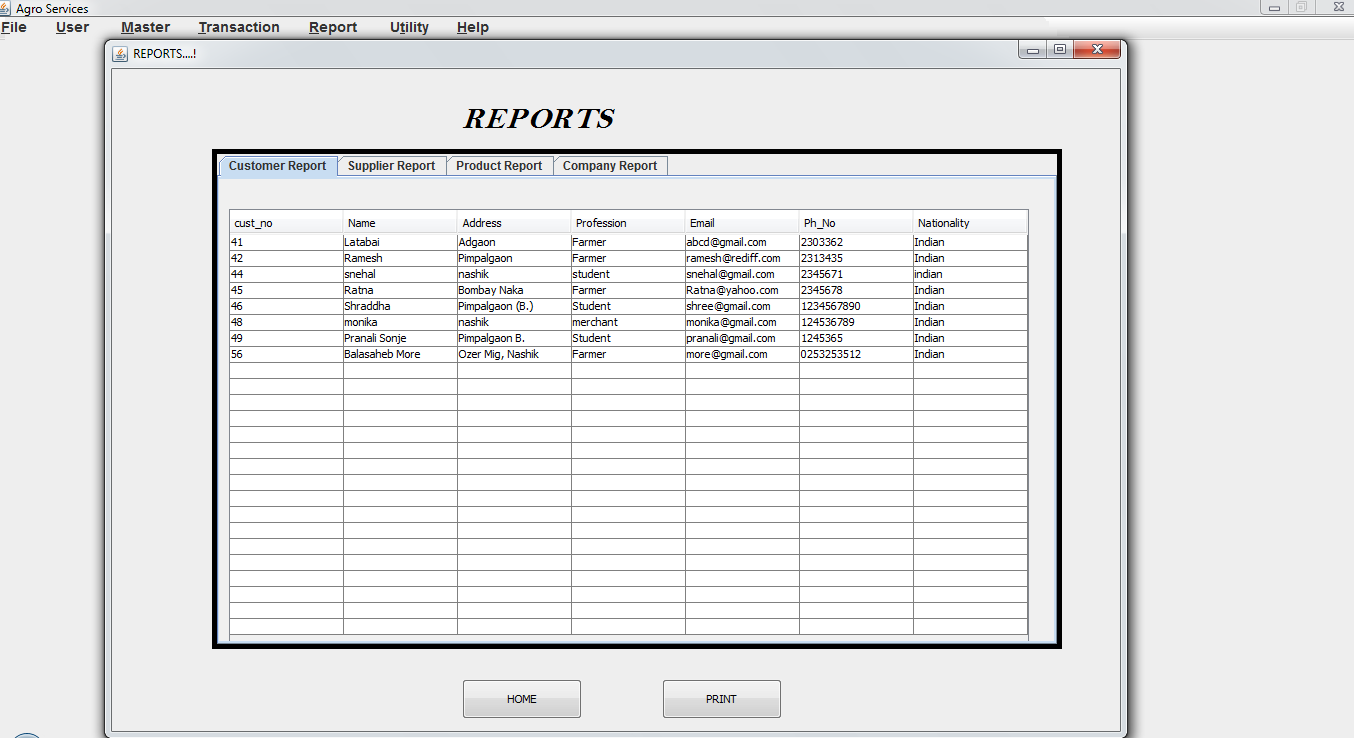
**SALE BILL ENTRY**



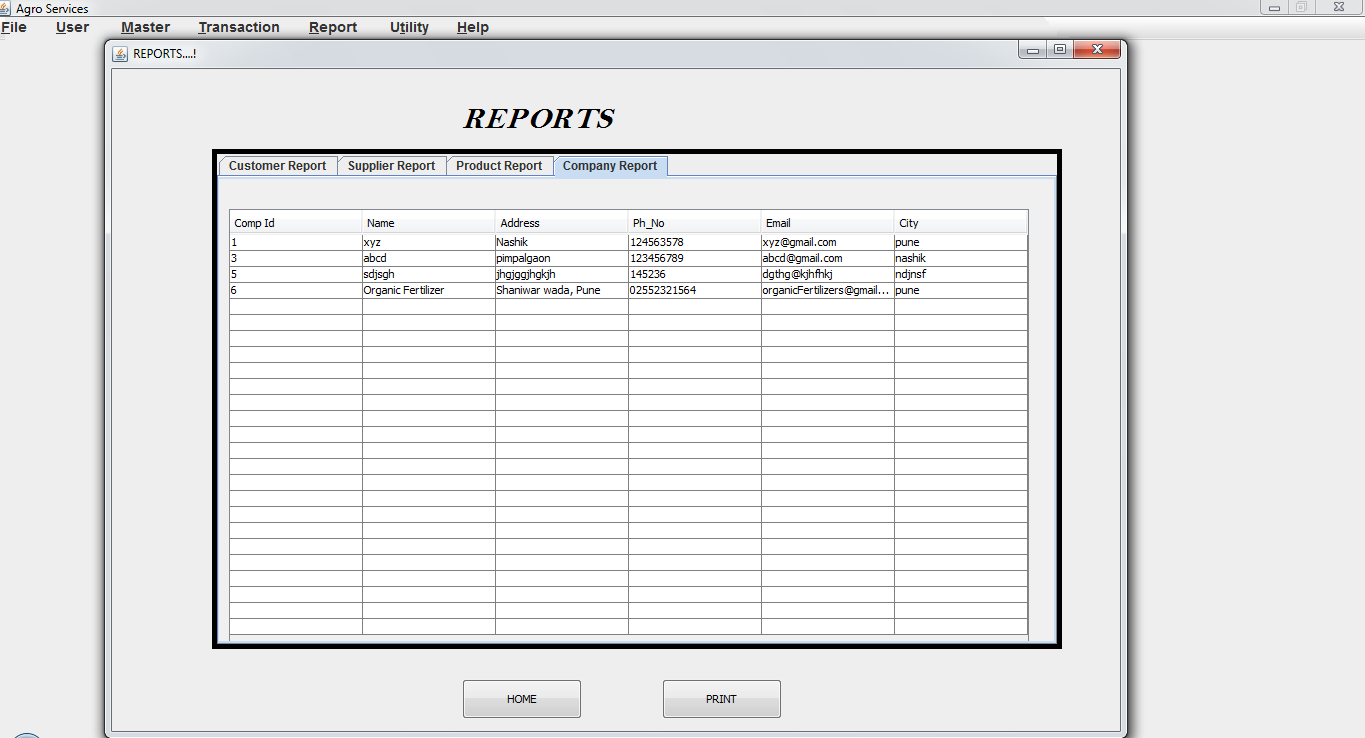
**MASTER REPORTS**



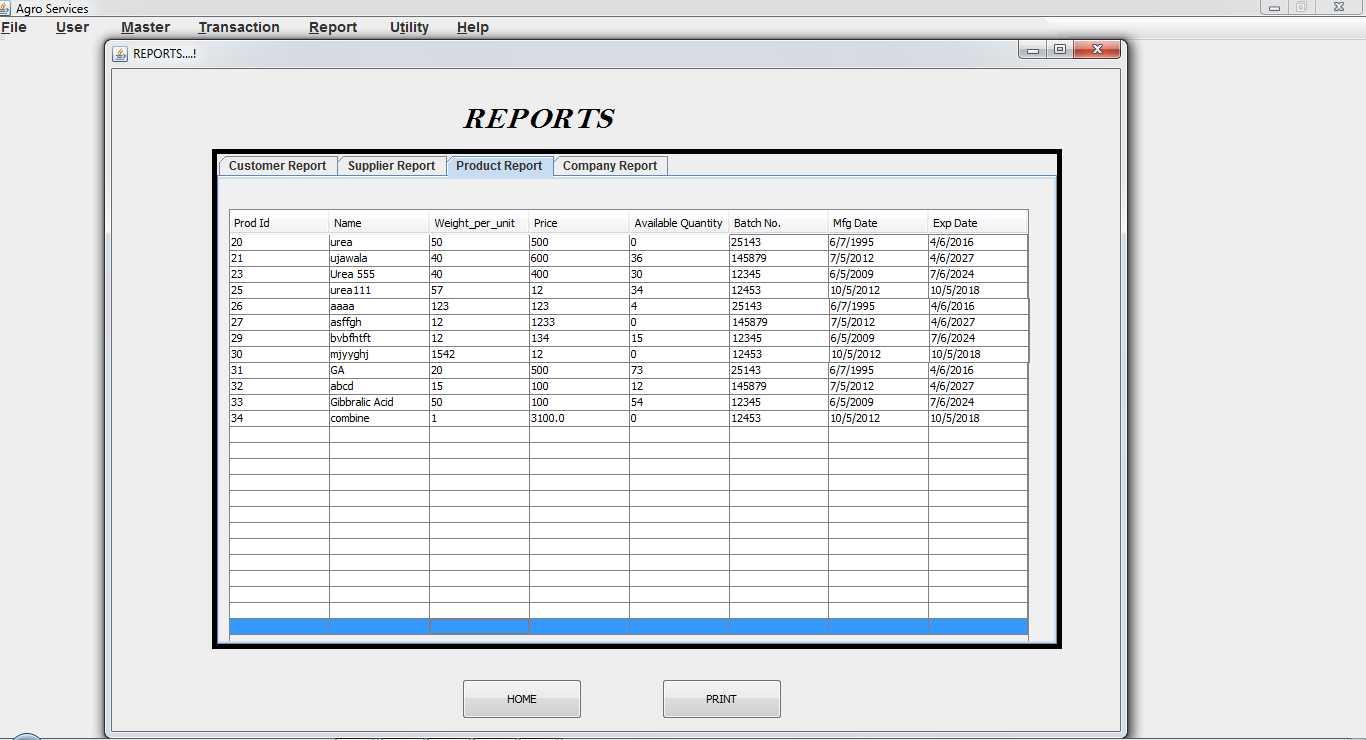
**CUSTOMER REPORT**

****

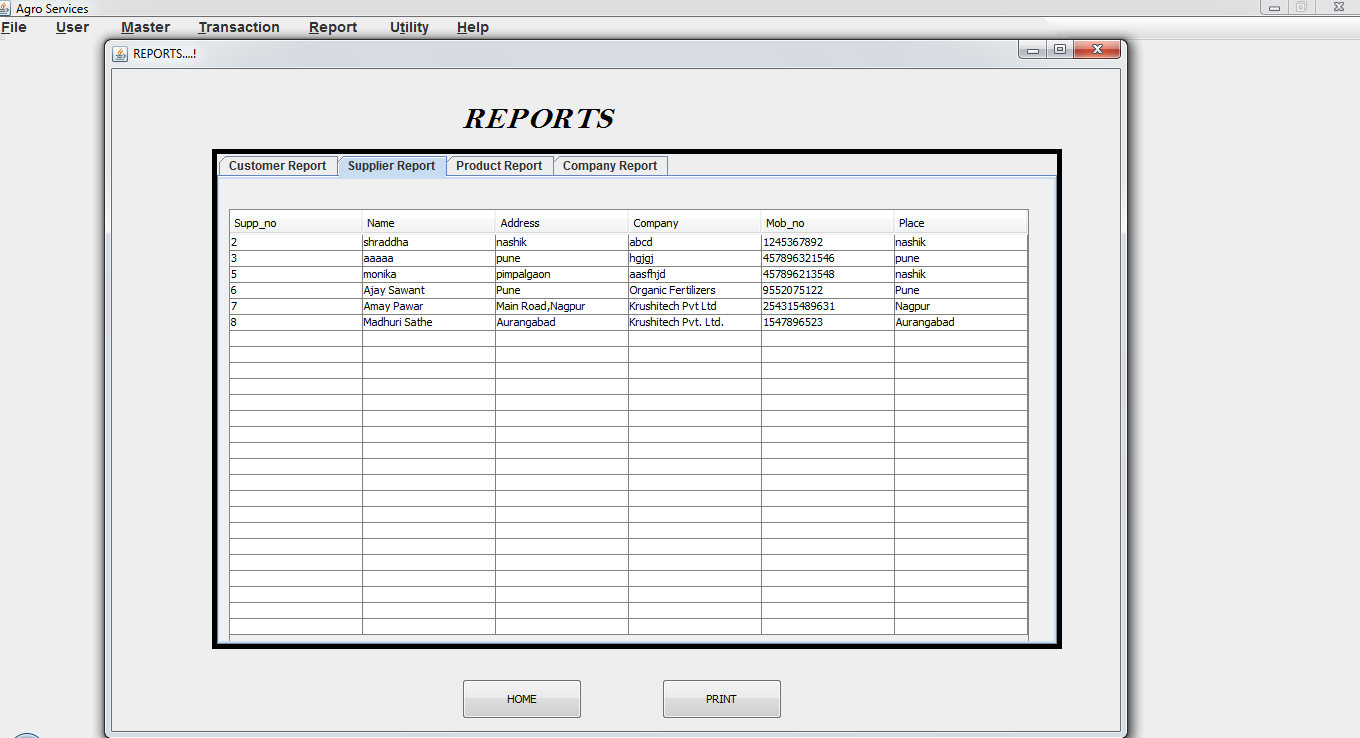
**COMPANY REPORT**

****

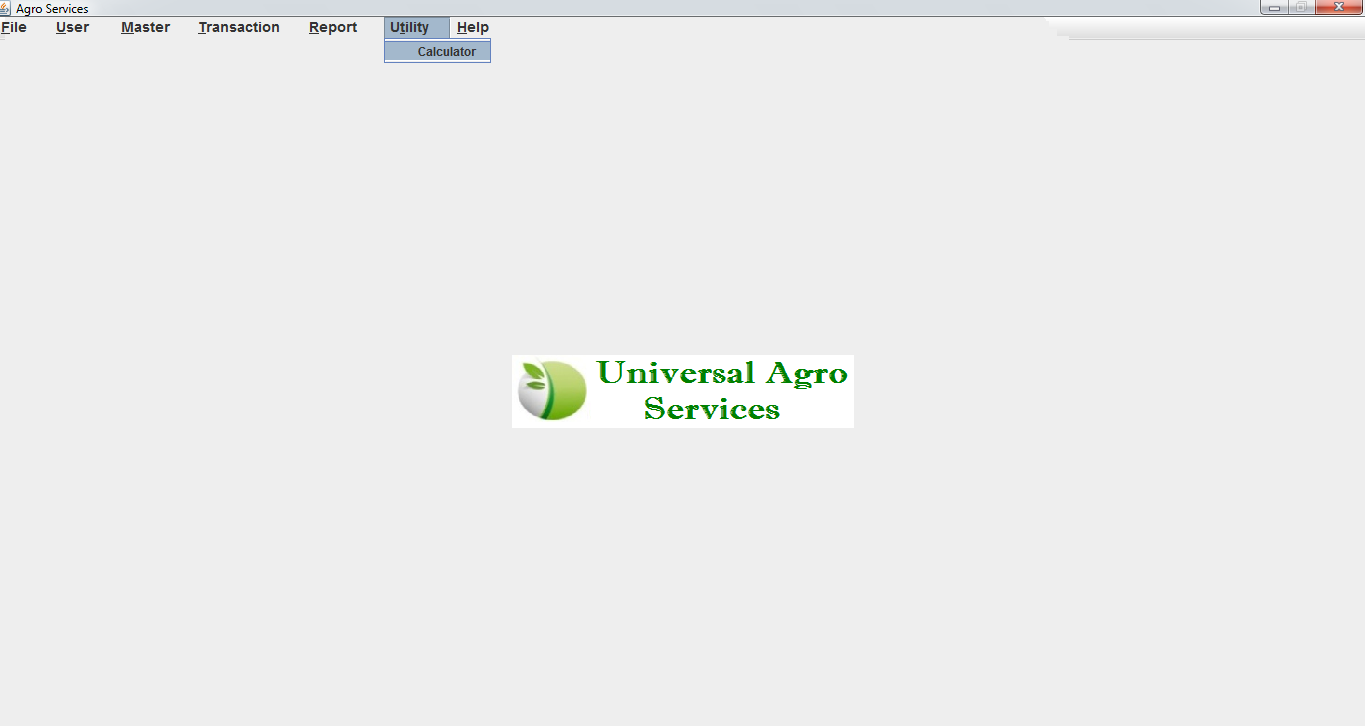
**PRODUCT REPORT**

****

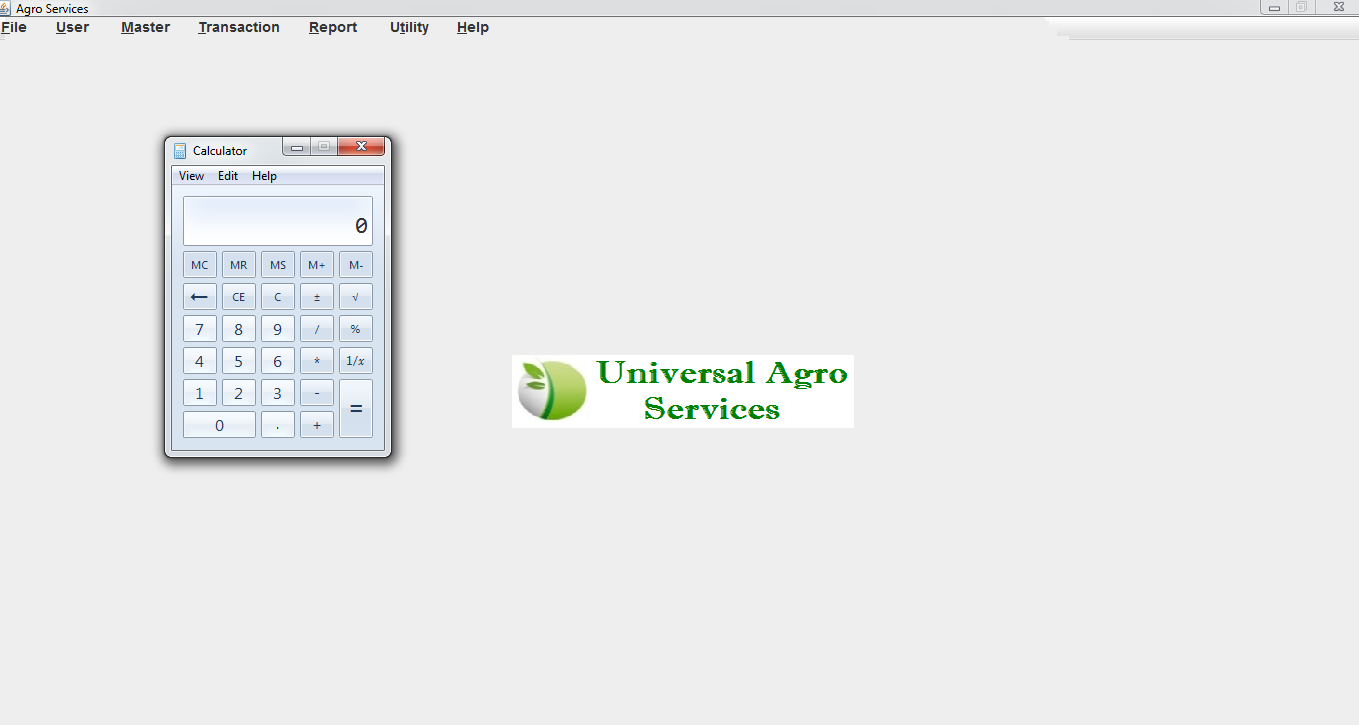
**SUPPLIER REPORT**



**UTILITY MASTER**

****

**UTILITY**

****

**CONCLUSION**

It is user friendly interface which will help the organization to contribute to perform the task more easily.

In this system for the records of many clients, we have also provided Validation to all fields in form based on their data types in database.

We are providing many buttons like Insert, Update, Remove and Search that help user to retrieve data. All records are correctly save in database so there is no chance of ambiguity. Relationships of tables are well maintained.

**LIMITATIONS AND FUTURE ENHANCEMENTS**

**LIMITATIONS**

1. The Look of the project might be better than this.
2. While creating the PO or sale bill the we can enter only 4 record.
3. The user authorities are not given.

**FUTURE ENHANCEMENTS**

1. The Look and fill will be enhanced more.
2. The users will be given the authorities and the screens will be displayed as per the authority.
3. The PO and Sale Orders may be done for more no of records.

**BIBLIOGRAPHY**

Books:

* The complete reference of java

Seventh addition

By

Herbert Schildt

* Programming in java-II

-by Neelima Padmawar & Alok Pawar.

Websites:-

<http://java.sun.com>