



UNIT ID: H7E1 04

UNIT NAME: Software Application Development

Project Document

NAME : N.M Dilanka Sadaruwan
SCN: 197353074
NIC : 973664182v
Phone :0710164681
BRANCH : Colombo

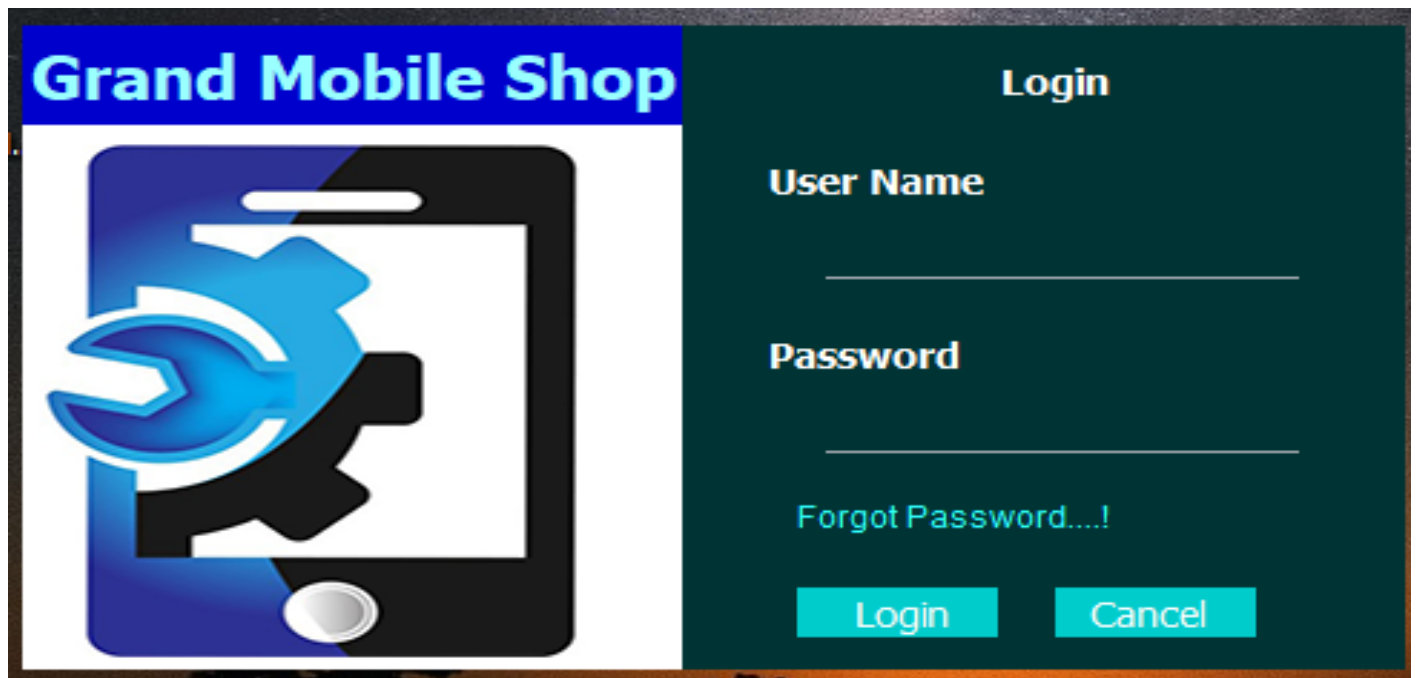


1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 01

3. Title of the Use case : login

4.GUI Name :login



01.Brief Description of the GUI:

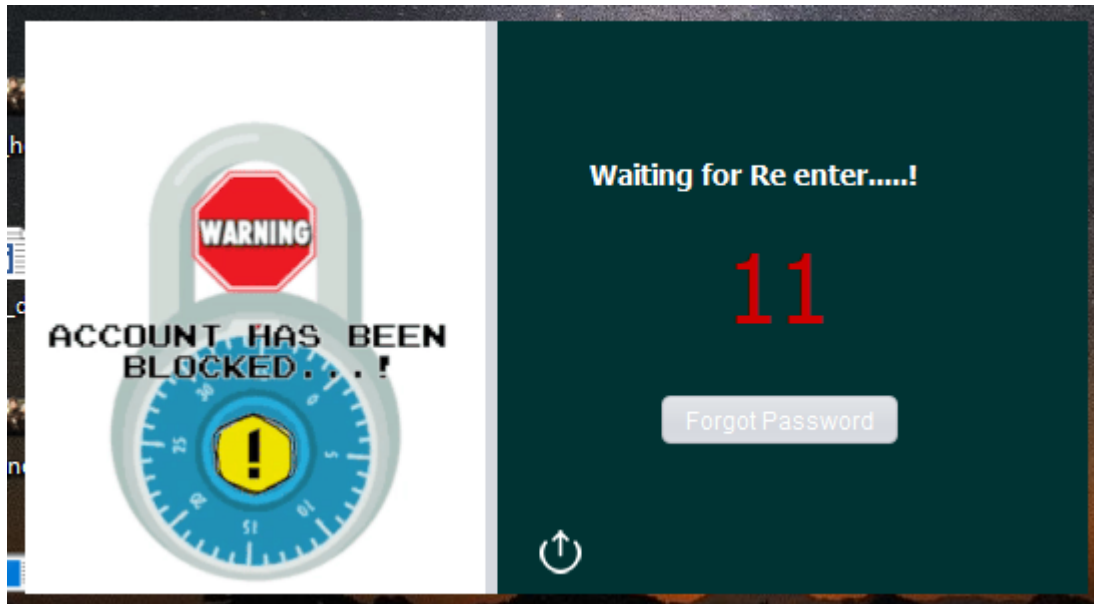
When we run the system open this GUI. We use this for Login to System. This GUI have Two Text field, first one use type user name and second use type password. And this GUI This GUI have a two Button, first one is a Login button. It use to login a system after we are type username and password.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 02

3. Title of the Use case : Login

4.GUI Name :Block State



01.Brief Description of the GUI:

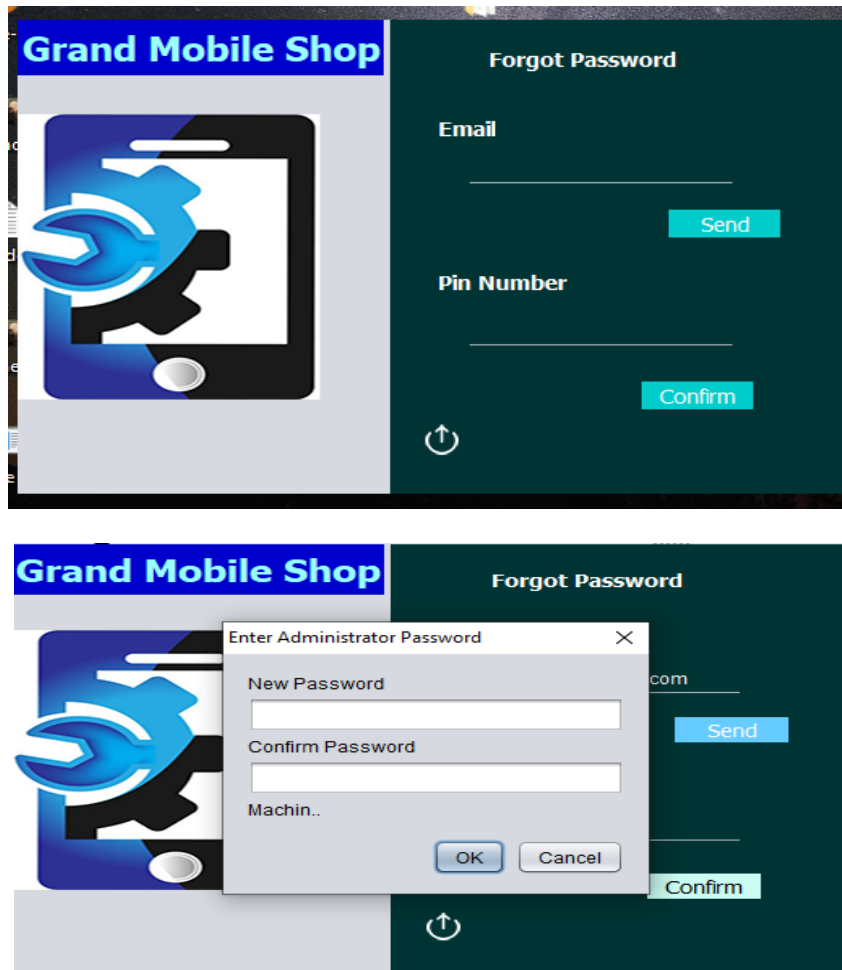
when we login to system we are type a wrong password or user name three time and this GUI open. This GUI have a 15 sec timer. When zero a time and enable a Forgot Password button. This GUI has some kind of button on bottom in GUI. It used to be Exit on System.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 03

3. Title of the Use case : Login

4.GUI Name :Forgot Password



01.Brief Description of the GUI:

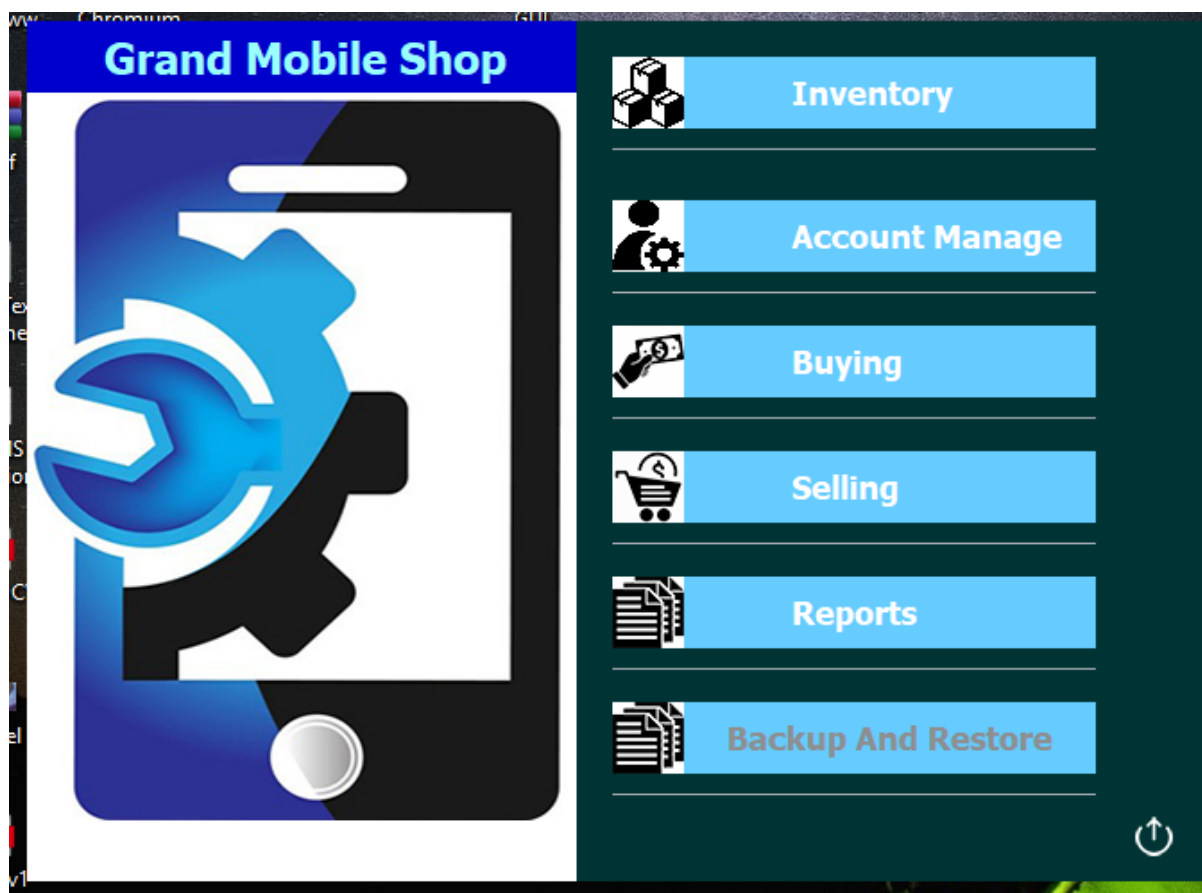
When user can't remember password he can update a password using this GUI. This GUI have two text field and two button. First text field use to be type email and click "**send**" button. Then send a OTP code typed email from system. And user can copy past his OTP code in second field. And click "**Confirm**" and Enter New password.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 04

3. Title of the Use case :

4.GUI Name :Main Menu



01.Brief Description of the GUI:

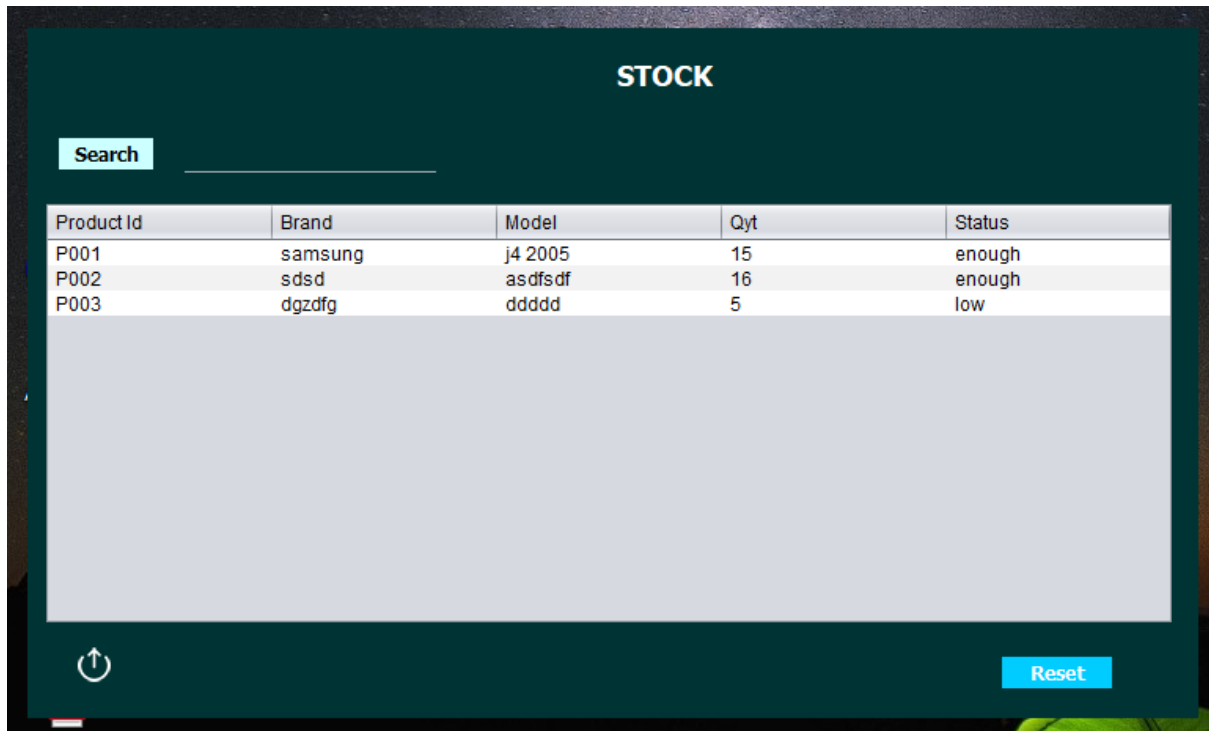
This GUI is a main part of System. It links to all process in System. This GUI have five main button (Inventory, Account Manage, Buying, Selling, Reports, Backup and Restore). "Inventory" button open Stock GUI, "Account" button use open user account setting GUI, "Buying" button Open GRN GUI, "Selling" button use open Invoice GUI, "Report" Button use open summery reports GUI and "Backup and Restore" button use open System backup and restore GUI. And this GUI has Logout button on bottom of right side in GUI.it use to Logout and open Login GUI.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 05

3. Title of the Use case : Manage Stock

4.GUI Name :Stock



Product Id	Brand	Model	Qty	Status
P001	samsung	j4 2005	15	enough
P002	sdsd	asdfsdf	16	enough
P003	dgzdfg	dddddd	5	low

01.Brief Description of the GUI:

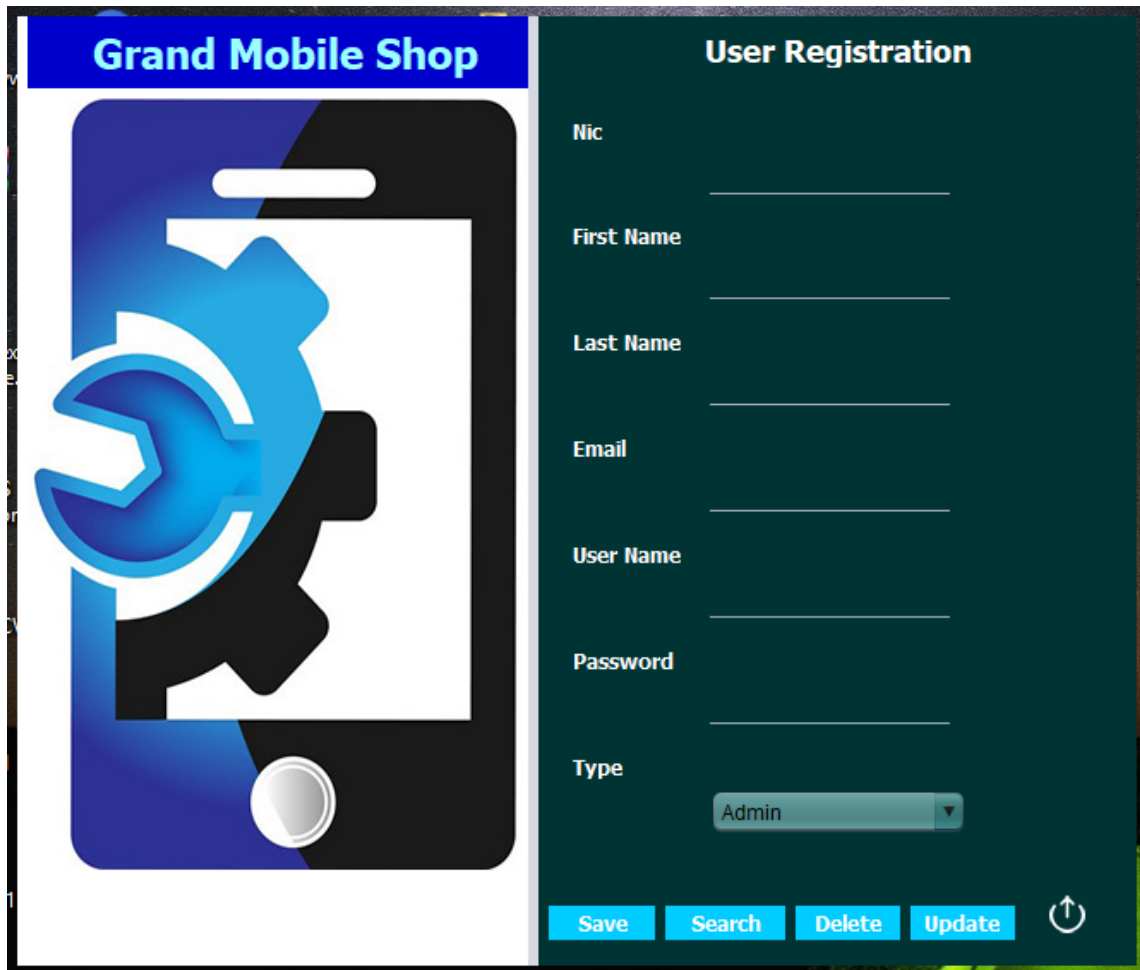
This GUI have mainly two process. Mainly we have check all products details and product quantity using this GUI. This GUI has searching aria. We can search and check detail (ex-: Availability) one product using product ID. And this GUI has “Reset” button. After we are search a product and we can reset a table and check all product in stock. And it has a back button, we can use it and going to main menu.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 06

3. Title of the Use case : User Manage

4.GUI Name :Account Manage



01.Brief Description of the GUI:

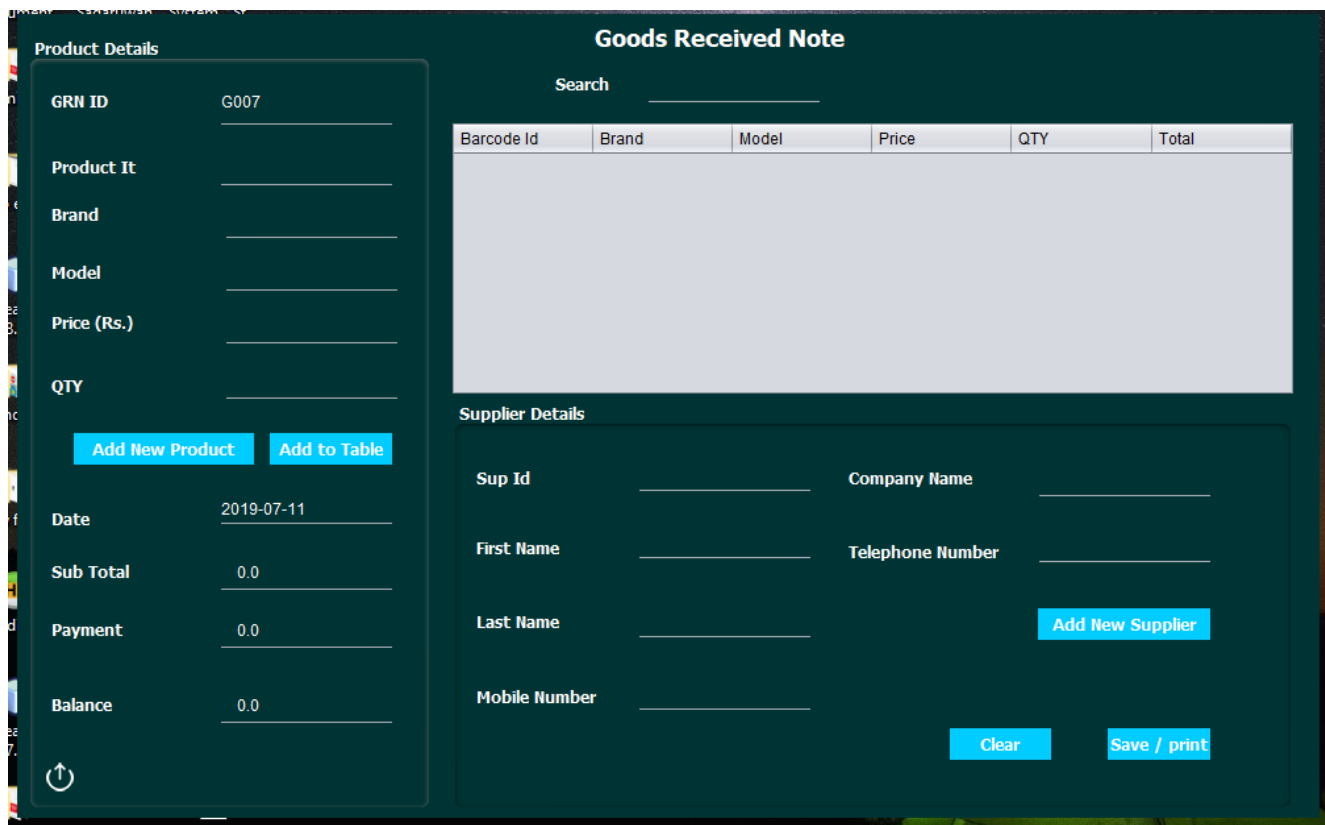
Using this GUI, we can register new user and search user details and update user and delete user. When we enter details in this text fields and we can register ne user using “save” button. When we enter NIC and we can check user details using “search” button. And we can delete searched user using “delete” button. And we can update user details using “update” button;

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 07

3. Title of the Use case : Manage Stock

4.GUI Name :Goods Received Note



The screenshot shows a web application titled "Goods Received Note". It features a dark green sidebar on the left with a "Product Details" section containing input fields for GRN ID (G007), Product It, Brand, Model, Price (Rs.), QTY, Date (2019-07-11), Sub Total (0.0), Payment (0.0), and Balance (0.0). There are buttons for "Add New Product" and "Add to Table". The main area has a "Search" bar and a table with columns: Barcode Id, Brand, Model, Price, QTY, and Total. Below the table is a "Supplier Details" section with input fields for Sup Id, Company Name, First Name, Telephone Number, Last Name, and Mobile Number, along with an "Add New Supplier" button. At the bottom right are "Clear" and "Save / print" buttons.

01.Brief Description of the GUI:

Mainly this GUI use to buy products and create a Goods Received Note. First enter product id and load a product details and enter price and quantity of buying product using text field in product details panel. And we can to table details entered product using "Add To Table" button. And we can type supplier id on first text field in supplier details panel and load a supplier details. When we add product to table update a sub total field and we can add to payment then save and print Goods Received note using "Save / Print" button.

We haven't details a product we can open register new product GUI using "Add New Product" button and register product. And we haven't supplier details we can open register new supplier GUI using "Add New Supplier" button then we can register supplier.

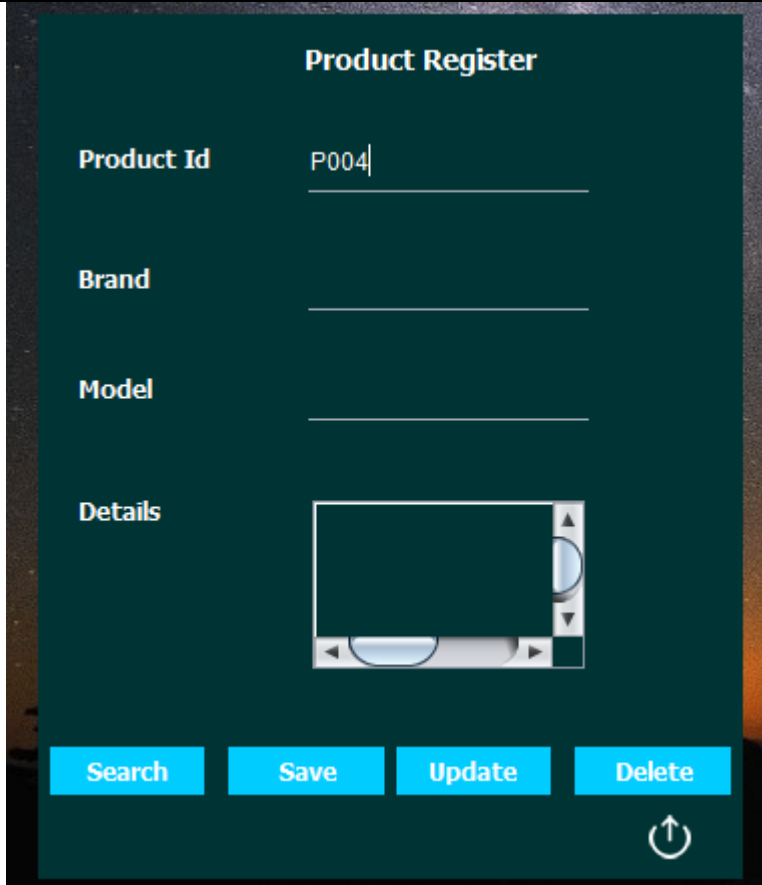
If we want to check old goods received note we can type a GRN id on search field and press enter key then load GRN details in a table.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 08

3. Title of the Use case : Stock

4.GUI Name : Product Register



01.Brief Description of the GUI:

This GUI using handle product details, it has four process. First one is when we enter new product details and we can save it using “Save” button. When type a product id and click “Search” button we can check product details. And we edit loaded details and click “Update” button we can update product details. And when we click “Delete” button we can delete loaded product. This GUI has another button bottom in GUI we can go back to Main Menu and exit this GUI using this button.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 09

3. Title of the Use case : Supplier Manage

4.GUI Name :New Supplier

New Supplier

Sup ID SUB003

First Name

Last Name

Email

Mobile Number

Company Name

Telephone Number

Clear **Save** **Search** **Delete**

⬆

01.Brief Description of the GUI:

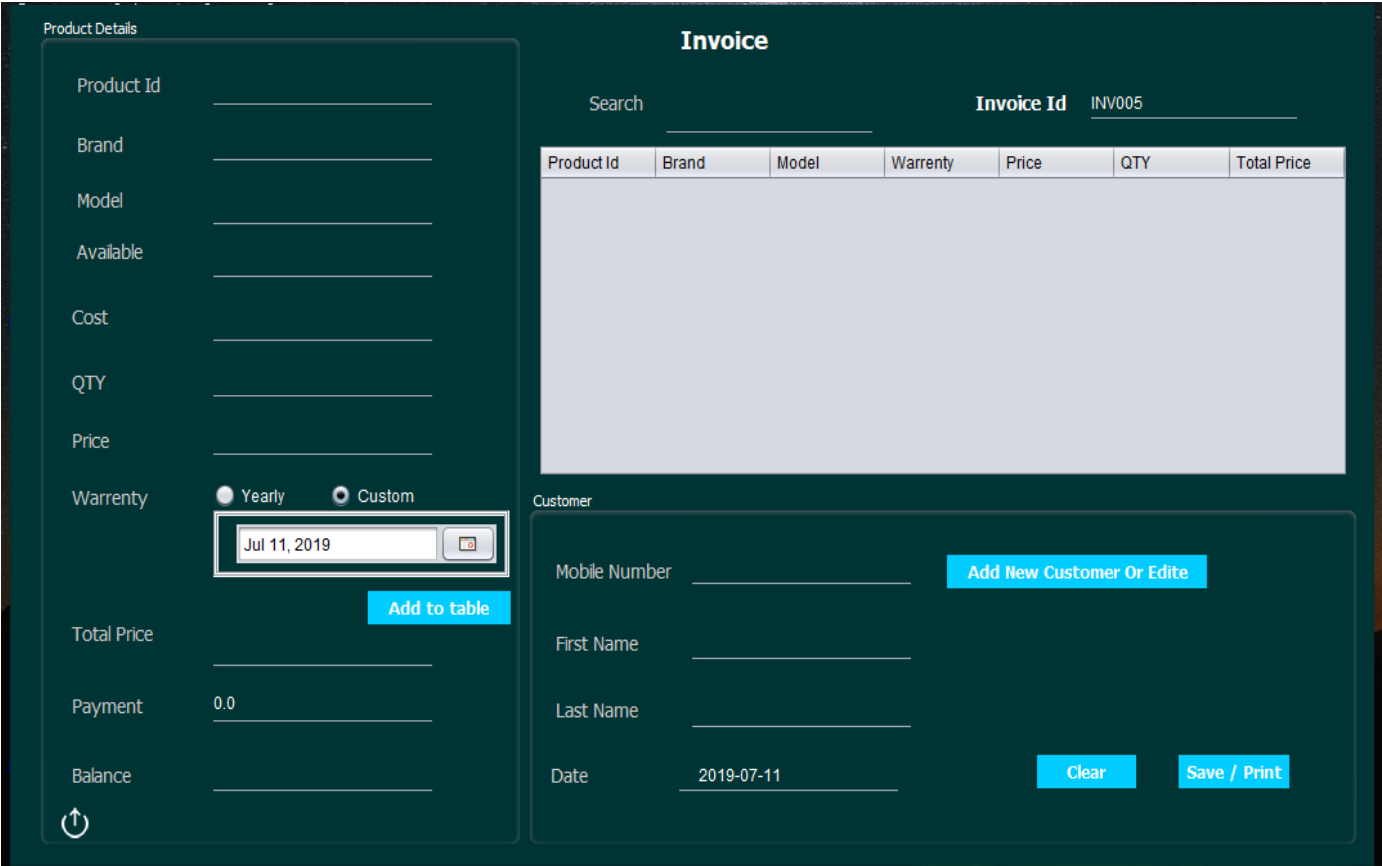
This GUI mainly using an Add New Supplier to database. We can enter supplier details and Add new supplier using "Save" button. When we type a supplier id in sup Id field and click "Search" button we can load and view supplier details. When we click "Delete" button and we can delete loaded supplier. And we can clear all fields using "Clear" button. We can exit and going to GRN GUI using other button on bottom in GUI.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 10

3. Title of the Use case : Manage Stock

4.GUI Name : Invoice



The screenshot shows a web application interface for managing stock and creating invoices. It is divided into two main sections: 'Product Details' on the left and 'Invoice' on the right.

Product Details Section:

- Fields for Product Id, Brand, Model, Available, Cost, QTY, Price, and Warrenty.
- Warrenty type selection: ☐ Yearly, ☒ Custom.
- A date picker showing 'Jul 11, 2019'.
- An 'Add to table' button.
- Summary fields: Total Price, Payment (0.0), and Balance.
- A refresh icon at the bottom left.

Invoice Section:

- A search bar and an 'Invoice Id' field with the value 'INV005'.
- A table with columns: Product Id, Brand, Model, Warrenty, Price, QTY, and Total Price.
- A 'Customer' section with fields for Mobile Number, First Name, Last Name, and Date (2019-07-11).
- Buttons: 'Add New Customer Or Edit', 'Clear', and 'Save / Print'.

01.Brief Description of the GUI:

This GUI mainly use to Selling products and create an invoice. When we enter product Id on product id field load a product details on other field. Then we can enter selling price and quantity and warranty using date chooser and we can add to table this product using "Add To Table" button. When we type customer mobile number we can load customer details on fields. When we add product to table then update total price field and we can enter payment and save and print invoice using "Save / Print" button. When we haven't customer details we can click "Add New Customer" button to open customer GUI. This GUI has clear button we can clear all fields using this button.

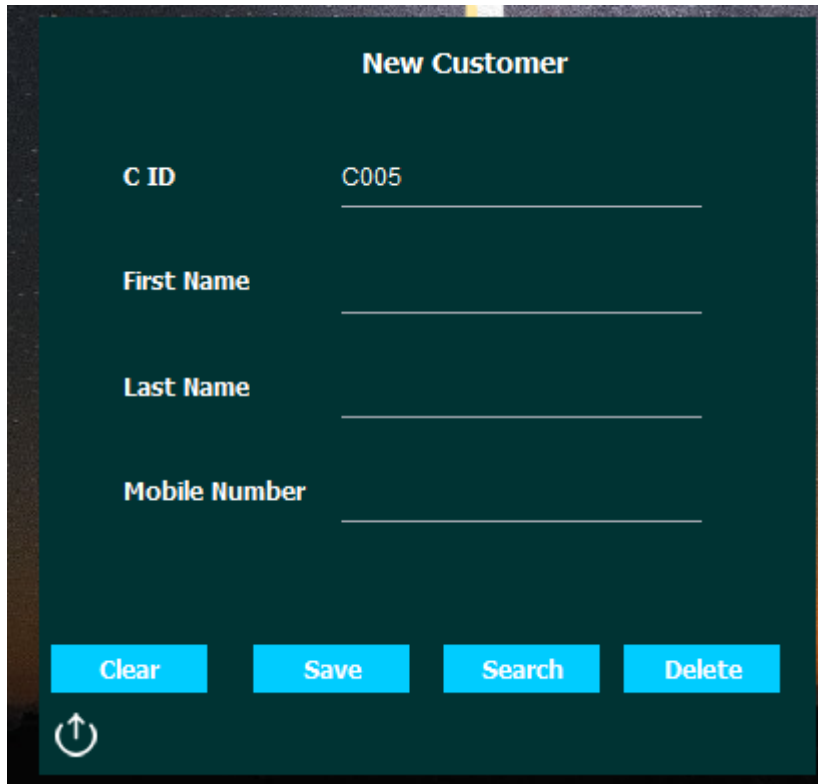
If we want check old invoice details we can type invoice id on search field then load details in table.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 11

3. Title of the Use case : Manage Customer

4.GUI Name :New Customer



01.Brief Description of the GUI:

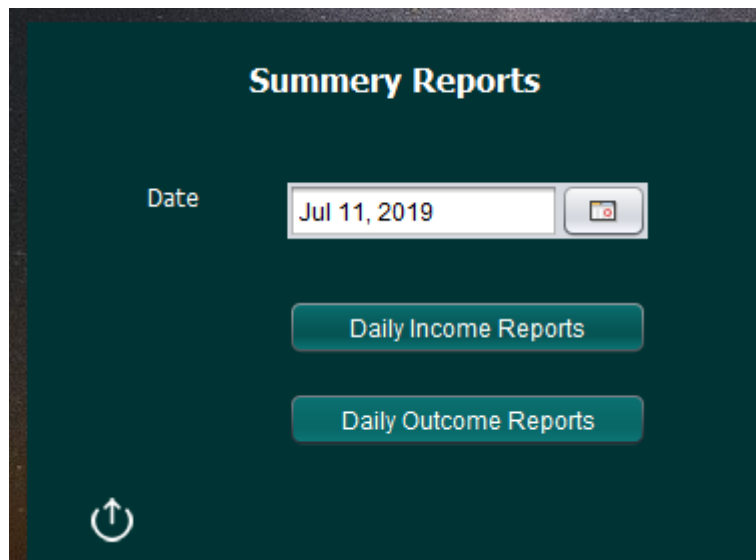
When we enter customer details on fields then we can add new customer using "Save" button. And we can check customer details using "Search" button after type customer id in C id field then load details on fields in this GUI. We can delete loaded customer using "delete" button. Using "Clear" button we can clear all fields. If we want going back to invoice GUI and exit this we can use special button on bottom in GUI.

1. Candidate Name : Dilanka Sadaruwan

2. GUI No : 12

3. Title of the Use case :

4.GUI Name :Reports



01.Brief Description of the GUI:

This GUI use to be create and view daily summery reports. We have two summery reports. First one is daily Income summery report and second one is daily outcome summery report. When we select a date using date chooser then click "Daily Income Report" button we can view daily income summer reports on selected day. And we click "Daily Outcome report" button we can view daily outcome report on selected day. If we want going back to main menu we can it using special button on bottom in GUI.

Business Report

H7E1 04/EX/03

FORM NO: H7E1 04/EX/03/FRM-28

1. Candidate Name : Dilanka Sadaruwan

2. Report No : 01

3. Title of the Report : Goods Received Note

4.KPI Name : Inventory

GRAND MOBILE SHOP																							
Goods Received Note																							
GRN ID	G008																						
Date	2019-07-11																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">product ID</th> <th style="text-align: left;">Buying Price</th> <th style="text-align: left;">QTY</th> <th style="text-align: left;">Total</th> </tr> </thead> <tbody> <tr> <td>P001</td> <td>35,000.0</td> <td>2</td> <td>70,000.0</td> </tr> <tr> <td>P002</td> <td>25,000.0</td> <td>1</td> <td>95,000.0</td> </tr> <tr> <td>P003</td> <td>15,000.0</td> <td>3</td> <td>140,000.0</td> </tr> <tr> <td>P004</td> <td>38,000.0</td> <td>5</td> <td>330,000.0</td> </tr> </tbody> </table>	product ID	Buying Price	QTY	Total	P001	35,000.0	2	70,000.0	P002	25,000.0	1	95,000.0	P003	15,000.0	3	140,000.0	P004	38,000.0	5	330,000.0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="text-align: right;">Sub Total</td> <td>635,000.0</td> </tr> </tbody> </table>	Sub Total	635,000.0
product ID	Buying Price	QTY	Total																				
P001	35,000.0	2	70,000.0																				
P002	25,000.0	1	95,000.0																				
P003	15,000.0	3	140,000.0																				
P004	38,000.0	5	330,000.0																				
Sub Total	635,000.0																						
grandmobile@gmail.com	1																						
Grand Mobile Shop																							

01.Brief Description of the Report:


This report is generated after buying products from suppliers. It used as a document to be put in their buying product details. This report is given to supplier after received products.

1. Candidate Name : Dilanka Sadaruwan

2. Report No : 02

3. Title of the Report: Invoice

4.KPI Name : Inventory

GRAND MOBILE SHOP		INVOICE				
GRN ID	INV005	Date	Jul 11, 2019			
product id	Brand	Model	Warranty End	QTY	Price	Total
P001	samsung	s4 2005	2020-07-12	2	38,000.0	76,000.0
P002	sdcd	asdfsdf	2020-07-12	3	50,000.0	166,000.0
P004	huawei	nova 2i 2018	2019-07-11	10	50,000.0	566,000.0
Sub Total					808,000.0	
<div style="display: flex; justify-content: space-between;"> grandmobile@gmail.com 1 Grand Mobile Shop </div>						

01.Brief Description of the Report:

This report is generated after selling products for customers. It used as a bill put in their details of buying products. This report is given to customer after them buying products.

1. Candidate Name : Dilanka Sadaruwan

2. Report No : 03

3. Title of the Report: Daily Income Summery

4.KPI Name :

GRAND MOBILE SHOP				
Daily Income Summery Reports				
Date		2019-07-11		
Invoice Id	Product Id	Selling Price	QTY	Total
INV004	P001	36,000.0	2	72,000.0
INV005	P001	38,000.0	2	148,000.0
INV005	P002	30,000.0	3	238,000.0
INV005	P004	40,000.0	10	538,000.0
INV006	P001	38,000.0	10	1,018,000.0
INV006	P002	30,000.0	11	1,348,000.0
INV007	P001	38,000.0	10	1,728,000.0
INV007	P002	28,000.0	3	1,812,000.0
INV007	P003	20,000.0	5	1,912,000.0
INV008	P001	38,000.0	10	2,292,000.0
INV009	P002	26,000.0	2	2,344,000.0
Sub Total				13,550,000.0
grandmobile@gmail.com		1	Grand Mobile Shop	

01.Brief Description of the Report:

This report including an all selling products details. This used as a document check a daily income summery.

1. Candidate Name : Dilanka Sadaruwan

2. Report No : 04

3. Title of the Report : Daily Outcome Summery

4.KPI Name :

GRAND MOBILE SHOP

Daily Outcome Summery Reports

Date 2019-07-11

GRN Id	Product Id	Buying Price	QTY	Total
G002	P001	35,000.0	10	350,000.0
G002	P002	25,000.0	8	550,000.0
G002	P003	15,000.0	5	625,000.0
G003	P001	35,000.0	3	730,000.0
G004	P002	25,000.0	5	855,000.0
G005	P001	35,000.0	2	925,000.0
G006	P002	25,000.0	3	1,000,000.0
G007	P001	35,000.0	5	1,175,000.0
G007	P004	38,000.0	5	1,365,000.0
G007	P002	25,000.0	5	1,490,000.0
G007	P003	15,000.0	7	1,595,000.0
G008	P001	35,000.0	2	1,665,000.0
G008	P002	25,000.0	1	1,690,000.0
G008	P003	15,000.0	3	1,735,000.0
G008	P004	38,000.0	5	1,925,000.0

Sub Total	17,675,000.0
------------------	---------------------

grandmobile@gmail.com

1

Grand Mobile Shop

01.Brief Description of the Report:

This report including an all Buying products details in a day. This used as a document check a daily outcome summery.

Login Interface Code

```
package GUI;

import java.awt.Color;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

public class Loginn extends javax.swing.JFrame {

    String un;

    String pw;

    String type1;

    static String type;

    int val;

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0, 204, 204);

    Color cc4 = new Color(255, 255, 255);

    public Loginn() {

        initComponents();

        unname.setText(null);

        pw_txt.setText(null);

    }

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

        pw_txt.grabFocus();

    }

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

        Loginbtn.grabFocus();

    }

    private void cancelbtnMouseClicked(java.awt.event.MouseEvent evt) {

        System.exit(0);

    }

}
```



```

        if (un.equals(decriptun) && pw.equals(decriptpw)) {

            logger.Logger.writeLogger(decriptun, "undefind", "Succes", attempt_count);

            System.out.println("0222");

            Main_Menu mn = new Main_Menu();

            mn.setVisible(true);

            this.dispose();

        } else {

            System.out.println("0111");

            JOptionPane.showMessageDialog(this, "Your Password or Username is Incorrect...!", "Incorrect Entry",
JOptionPane.ERROR_MESSAGE);

            unname.setText(null);

            pw_txt.setText(null);

            attempt_count++;

            logger.Logger.writeLogger("undefind", "undefind", "Fail", attempt_count);

            if (attempt_count > 3) {

                new Lock_state().setVisible(true);

                this.dispose();

                attempt_count = 0;

            }

        }

    } else {

        JOptionPane.showMessageDialog(this, "Your Password or Username is Incorrect...!", "Incorrect Entry",
JOptionPane.ERROR_MESSAGE);

    }

    } catch (Exception e) {

        e.printStackTrace();

    }

    } else {

        JOptionPane.showMessageDialog(this, "Please Enter Your User Name Or Password Maximum 15 characters
...!", "check Character Lenght", JOptionPane.ERROR_MESSAGE);

        unname.setText(null);

        pw_txt.setText(null);

    }

    } else {

        JOptionPane.showMessageDialog(this, "Please Enter Your User Name Or Password minimum 4 characters...!
", "check Character Lenght", JOptionPane.ERROR_MESSAGE);

        unname.setText(null);

        pw_txt.setText(null);

    }

}

```

```

    }
}

private void unameMouseEntered(java.awt.event.MouseEvent evt) {
    uname.setToolTipText("Please Enter Your UserName");
}

private void pw_txtMouseEntered(java.awt.event.MouseEvent evt) {
    pw_txt.setToolTipText("Please Enter Your Password");
}

private void jLabel5MouseClicked(java.awt.event.MouseEvent evt) {
    Forgot_password frg = new Forgot_password();
    frg.setVisible(true);
    this.dispose();
}

private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
}

private void jLabel3MouseEntered(java.awt.event.MouseEvent evt) {
}

private void jLabel15MouseEntered(java.awt.event.MouseEvent evt) {
    Color cc1 = new Color(0, 0, 0);
    Color cc2 = new Color(255, 255, 255);
    jLabel15.setBackground(cc1);
    jLabel15.setForeground(cc2);
}

private void jLabel15MouseExited(java.awt.event.MouseEvent evt) {
    Color cc3 = new Color(0, 0, 204);
    Color cc4 = new Color(153, 255, 255);
    jLabel15.setBackground(cc3);
    jLabel15.setForeground(cc4);
}

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    setTimer();
}

private void LoginbtnMouseEntered(java.awt.event.MouseEvent evt) {
    Loginbtn.setBackground(cc1);

```

```

        Loginbtn.setForeground(cc2);
    }

    private void LoginbtnMouseExited(java.awt.event.MouseEvent evt) {
        Loginbtn.setBackground(cc3);
        Loginbtn.setForeground(cc4);
    }

    private void cancelbtnMouseEntered(java.awt.event.MouseEvent evt) {
        cancelbtn.setBackground(cc1);
        cancelbtn.setForeground(cc2);
    }

    private void cancelbtnMouseExited(java.awt.event.MouseEvent evt) {
        cancelbtn.setBackground(cc3);
        cancelbtn.setForeground(cc4);
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Loginn.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Loginn.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Loginn.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Loginn.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Loginn().setVisible(true);
            }
        });
    }
}
// Variables declaration - do not modify

```

```

private javax.swing.JLabel Loginbtn;

private javax.swing.JLabel cancelbtn;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JPasswordField pw_txt;

private javax.swing.JTextField uname;

// End of variables declaration

void setTimer() {

    try {

        File file = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details\\countdn.dat");

        if (!(file.exists())) {

            file.mkdir();

        }

        BufferedReader br = new BufferedReader(new FileReader(file));

        String st = br.readLine();

        int st1 = Integer.parseInt(st);

        if (st1 == 15) {

        } else {

            System.out.println(st1);

            Lock_state lkl = new Lock_state();

            lkl.setVisible(true);

            this.dispose();

        }

        br.close();

    } catch (Exception e) {

        e.printStackTrace();

    } }}

```

Block State

```
package GUI;
```

```
import java.io.BufferedReader;
```

```
import java.io.BufferedWriter;
```

```
import java.io.File;
```

```
import java.io.FileReader;
```

```
import java.io.FileWriter;
```

```
public class Lock_state extends javax.swing.JFrame {
```

```
    int val;
```

```
    public Lock_state() {
```

```
        initComponents();
```

```
        setCountDown();
```

```
        countDown();
```

```
        fp_btn.setEnabled(false);
```

```
    }
```

```
    private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
```

```
        this.dispose();
```

```
    }
```

```
    private void formWindowActivated(java.awt.event.WindowEvent evt) {
```

```
    }
```

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

```
        new Forgot_password().setVisible(true);
```

```
        this.dispose();
```

```
    }
```

```
    public static void main(String args[]) {
```

```
        try {
```

```
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
                if ("Nimbus".equals(info.getName())) {
```

```
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
```

```
                    break;
```

```
                }
```

```
            }
```

```
        } catch (ClassNotFoundException ex) {
```

```
            java.util.logging.Logger.getLogger(Lock_state.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
        } catch (InstantiationException ex) {
```

```
            java.util.logging.Logger.getLogger(Lock_state.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```



```

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Lock_state.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Lock_state.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

//</editor-fold>

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Lock_state().setVisible(true);

    }

});

}

```

```

// Variables declaration - do not modify

private javax.swing.JButton fp_btn;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JLabel time;

// End of variables declaration

private void countDown() {

    Thread thread;

    thread = new Thread() {

        @Override

        public void run() {

            int tim = Integer.parseInt(time.getText());

            for (int i = 0; i < tim; i++) {

                try {

                    Thread.sleep(1000);

                } catch (Exception e) {

                    e.printStackTrace();

                }

                time.setText(Integer.parseInt(time.getText()) - 1 + "");

                writeCurrentTimeToFile();

            }

            disp();

```

```

    }

    private void writeCurrentTimeToFile() {

        try {

            File folder = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile");

            File folder2 = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details");

            File file = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details\\countdn.dat");

            if (!(folder.exists())) {

                folder.mkdir();

            }

            if (!(folder2.exists())) {

                folder2.mkdir();

            }

            file.setWritable(true);

            if (file.exists()) {

                file.delete();

            }

            FileWriter fr = new FileWriter(file);

            BufferedWriter br = new BufferedWriter(fr);

            br.write(String.valueOf(time.getText()));

            file.setWritable(false);

            br.close();

            fr.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

};

thread.start();

}

private void setCountDown() {

    try {

        File file = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details\\countdn.dat");

        BufferedReader br = new BufferedReader(new FileReader(file));

        String st = br.readLine();

        time.setText(st);

        br.close();

    } catch (Exception e) {

```

```

    }

}

private void disp() {

    writeToCountDownFile();

    writeToAttemptFile();

fp_btn.setEnabled(true);

}

private void writeToCountDownFile() {

    if (val == 0) {

        try {

            File folder = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile");

            File folder2 = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details");

            File file = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details\\countdn.dat");

            if (!folder.exists()) {

                folder.mkdir();

            }

            if (!folder2.exists()) {

                folder2.mkdir();

            }

            file.setWritable(true);

            if (file.exists()) {

                file.delete();

            }

            FileWriter fr = new FileWriter(file);

            BufferedWriter br = new BufferedWriter(fr);

            br.write(String.valueOf(15));

            file.setWritable(false);

            br.close();

            fr.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

private void writeToAttemptFile() {

    try {

        File folder = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile");

```

```
File folder2 = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details");

File file = new File(System.getenv("PROGRAMDATA") + "\\Login System Mobile\\Login Details\\atmpt.dat");

if (!(folder.exists())) {

    folder.mkdir();

}

if (!(folder2.exists())) {

    folder2.mkdir();

}

file.setWritable(true);

if (file.exists()) {

    file.delete();

}

FileWriter fr = new FileWriter(file);

BufferedWriter br = new BufferedWriter(fr);

br.write(String.valueOf(1));

file.setWritable(false);

br.close();

fr.close();

} catch (Exception e) {

    e.printStackTrace();

}

}

}
```

Forgot Password

```
package GUI;

import Cryptography.Cryption;

import Model.Email;

import java.awt.Color;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import java.util.Random;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JLabel;

import javax.swing.JPasswordField;

public class Forgot_password extends javax.swing.JFrame {

    StringBuilder sb = new StringBuilder();

    String charSet = "abcdefghijklmnopqrstuvwxyz245879lmnopqrst";

    Random rnd = new Random();

    String OTP;

    String pw2;

    public Forgot_password() {

        initComponents();
    }

    private void email_fMouseEntered(java.awt.event.MouseEvent evt) {

    }

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

    }

    private void send_btnMouseClicked(java.awt.event.MouseEvent evt) {

        if (!email_f.getText().isEmpty()) {

            try {

                String emailu = email_f.getText();

                ResultSet rs = database.db.search("SELECT * FROM users WHERE users.email = '" + emailu + "'");

                if (rs.next()) {

                    String eml = email_f.getText().trim();

                    while (sb.length() < 6) {

                        int index = (int) (rnd.nextFloat() * charSet.length());

                        sb.append(charSet.charAt(index));

                    }

                }

            } catch (SQLException ex) {

                Logger.getLogger(Forgot_password.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    }

}
```

```

    }

    OTP = sb.toString();

    status.setText("Sending Code...");

    Email.sendMail(eml, " Grand Mobile Phone Shop ", "Your System Admin Verification Code : " + OTP);

    } else {

        JOptionPane.showMessageDialog(this, "Incorrect Email", "Invalide", JOptionPane.ERROR_MESSAGE);

    }

    } catch (Exception e) {

        e.printStackTrace();

    }

    } else {

        JOptionPane.showMessageDialog(this, "Pleace check Field", "Empty Field", JOptionPane.ERROR_MESSAGE);

    }

}

private void send_btnMouseEntered(java.awt.event.MouseEvent evt) {

    Color cc1 = new Color(204, 253, 244);

    Color cc0 = new Color(0, 0, 0);

    send_btn.setBackground(cc1);

    send_btn.setForeground(cc0);

}

private void send_btnMouseExited(java.awt.event.MouseEvent evt) {

    Color cc = new Color(102, 204, 255);

    Color cc2 = new Color(255, 255, 255);

    send_btn.setBackground(cc);

    send_btn.setForeground(cc2);

}

private void confr_btnMouseClicked(java.awt.event.MouseEvent evt) {

    if (pin_f.getText().equals(OTP)) {

        JPasswordField jp1 = new JPasswordField();

        JPasswordField jpc1 = new JPasswordField();

        JLabel jl1 = new JLabel();

        String d = "Machin..";

        jl1.setText(d);

        Object[] jpw = {"New Password", jp1, "Confirm Password", jpc1, jl1};

        int okcan = JOptionPane.showConfirmDialog(this, jpw, "Enter New Password", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);

```

```

String pw1 = jp1.getText();

pw2 = jpc1.getText();

if (okcan == JOptionPane.OK_OPTION) {

    if (pw1.equals(pw2)) {

        try {

            database.db.iud("UPDATE users SET users.password = '" + Cryption.encrypt(pw2) + "' WHERE users.email = '" +
email_f.getText() + "'");

            JOptionPane.showMessageDialog(this, "Updated");

        } catch (Exception e) {

            e.printStackTrace();

        }

        new Main_Menu().setVisible(true);

        this.dispose();

    } else {

        JOptionPane.showMessageDialog(this, "different");

    }

}

if (okcan == JOptionPane.CANCEL_OPTION || okcan == JOptionPane.CLOSED_OPTION) {

    email_f.setText(null);

    pin_f.setText(null);

}

} else {

    JOptionPane.showMessageDialog(this, "Incorrect OTP code..!", "Incorrect", JOptionPane.ERROR_MESSAGE);

    pin_f.setText(null);

}

}

private void confr_btnMouseEntered(java.awt.event.MouseEvent evt) {

    Color cc1 = new Color(204, 253, 244);

    Color cc0 = new Color(0, 0, 0);

    confr_btn.setBackground(cc1);

    confr_btn.setForeground(cc0);

}

private void confr_btnMouseExited(java.awt.event.MouseEvent evt) {

    Color cc = new Color(102, 204, 255);

    Color cc2 = new Color(255, 255, 255);

    confr_btn.setBackground(cc);

    confr_btn.setForeground(cc2);

}

```

```

}

private void pin_fMouseEntered(java.awt.event.MouseEvent evt) {
}

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

private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
}

private void jLabel15MouseEntered(java.awt.event.MouseEvent evt) {
    Color cc1 = new Color(0, 0, 0);
    Color cc2 = new Color(255, 255, 255);
    jLabel15.setBackground(cc1);
    jLabel15.setForeground(cc2);
}

private void jLabel15MouseExited(java.awt.event.MouseEvent evt) {
    Color cc3 = new Color(0, 0, 204);
    Color cc4 = new Color(153, 255, 255);
    jLabel15.setBackground(cc3);
    jLabel15.setForeground(cc4);
}

public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Forgot_password.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Forgot_password.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Forgot_password.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Forgot_password.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```



```
//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Forgot_password().setVisible(true);

    }

});

}

// Variables declaration - do not modify

private javax.swing.JLabel confr_btn;

private javax.swing.JTextField email_f;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JTextField pin_f;

private javax.swing.JLabel send_btn;

private javax.swing.JLabel status;

// End of variables declaration

}
```

Mail Menu

```
package GUI;

import java.awt.Color;

public class Main_Menu extends javax.swing.JFrame {

    public static Main_Menu o;

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(102, 204, 255);

    Color cc4 = new Color(255, 255, 255);

    public Main_Menu() {

        initComponents();

        jLabel12.setEnabled(false);

    }

    private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {

        Loginn lg = new Loginn();

        lg.setVisible(true);

        this.dispose();

    }

    private void jLabel4MouseEntered(java.awt.event.MouseEvent evt) {

        jLabel4.setBackground(cc1);

        jLabel4.setForeground(cc2);

    }

    private void jLabel4MouseExited(java.awt.event.MouseEvent evt) {

        jLabel4.setBackground(cc3);

        jLabel4.setForeground(cc4);

    }

    private void jLabel6MouseEntered(java.awt.event.MouseEvent evt) {

        jLabel6.setBackground(cc1);

        jLabel6.setForeground(cc2);

    }

    private void jLabel6MouseExited(java.awt.event.MouseEvent evt) {

        jLabel6.setBackground(cc3);

        jLabel6.setForeground(cc4);

    }

    private void jLabel8MouseEntered(java.awt.event.MouseEvent evt) {

        jLabel8.setBackground(cc1);
```

```

        jLabel8.setForeground(cc2);
    }

    private void jLabel8MouseExited(java.awt.event.MouseEvent evt) {
        jLabel8.setBackground(cc3);
        jLabel8.setForeground(cc4);
    }

    private void jLabel10MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel10.setBackground(cc1);
        jLabel10.setForeground(cc2);
    }

    private void jLabel10MouseExited(java.awt.event.MouseEvent evt) {
        jLabel10.setBackground(cc3);
        jLabel10.setForeground(cc4); }

    private void jLabel3MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel3.setBackground(cc1);
        jLabel3.setForeground(cc2);}

    private void jLabel3MouseExited(java.awt.event.MouseEvent evt) {
        jLabel3.setBackground(cc3);
        jLabel3.setForeground(cc4); }

    private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {
        Inventory intv = new Inventory();
        intv.setVisible(true);
        this.dispose();
    }

    private void jLabel6MouseClicked(java.awt.event.MouseEvent evt) {
        Sign_In accm = new Sign_In();
        accm.setVisible(true);
        this.dispose(); }

    private void jLabel8MouseClicked(java.awt.event.MouseEvent evt) {
        GRN gn1 = new GRN();
        gn1.setVisible(true);
        this.dispose();
    }

    private void jLabel10MouseClicked(java.awt.event.MouseEvent evt) {
        Invoice inv = new Invoice();
        inv.setVisible(true);
        this.dispose();
    }

```

```

    }

    private void jLabel12MouseEntered(java.awt.event.MouseEvent evt) {

        jLabel12.setBackground(cc1);

        jLabel12.setForeground(cc2);

    }


    private void jLabel12MouseExited(java.awt.event.MouseEvent evt) {

        jLabel12.setBackground(cc3);

        jLabel12.setForeground(cc4);

    }

    private void jLabel15MouseEntered(java.awt.event.MouseEvent evt) {

        Color cc1 = new Color(0, 0, 0);

        Color cc2 = new Color(255, 255, 255);

        jLabel15.setBackground(cc1);

        jLabel15.setForeground(cc2);

    }

    private void jLabel15MouseExited(java.awt.event.MouseEvent evt) {

        Color cc3 = new Color(0, 0, 204);

        Color cc4 = new Color(153, 255, 255);

        jLabel15.setBackground(cc3);

        jLabel15.setForeground(cc4);

    }

    private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {

        new Reports().setVisible(true);

        this.dispose();

    }

    public static void main(String args[]) {

        try {

            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                } }

        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Main_Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Main_Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }
    }

```

```

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Main_Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Main_Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new Main_Menu().setVisible(true);    }

    }); }

// Variables declaration - do not modify

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel3;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JSeparator jSeparator5;

private javax.swing.JSeparator jSeparator6;

// End of variables declaration

}

```

Inventory

```
package GUI;
```

```
import java.awt.Color;
```

```
import java.sql.ResultSet;
```

```
import java.util.Vector;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import source1.SuggestionList;
```

```
public class Inventory extends javax.swing.JFrame {
```

```
    String pid1;
```

```
    String brnd;
```

```
    String modl1;
```

```
    String bprice1;
```

```
    String slng1;
```

```
    String qtyt1;
```

```
    String statust;
```

```
    Color cc1 = new Color(204, 253, 244);
```

```
    Color cc2 = new Color(0, 0, 0);
```

```
    Color cc3 = new Color(0, 204, 255);
```

```
    Color cc4 = new Color(255, 255, 255);
```

```
    public Inventory() {
```

```
        initComponents();
```

```
        tbled();
```

```
    }
```

```
    private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
```

```
        Main_Menu mn = new Main_Menu();
```

```
        mn.setVisible(true);
```

```
        this.dispose();
```

```
    }
```

```
    private void inv_search_comPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {
```

```
        SuggestionList.setSelectedItem(invt_serarch, inv_search_com);
```

```
    }
```

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

```
        try {
```

```
            DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
```

```
            String intp = invt_serarch.getText();
```

```

        ResultSet rs = database.db.search("SELECT stock.qty,stock.prduct_p_id,product.brand,product.model FROM stock
INNER JOIN product ON stock.prduct_p_id=product.p_id WHERE product.p_id='" + intp + "'");

        while (rs.next()) {

            qtyt1 = rs.getString("stock.qty");

            pid1 = rs.getString("stock.prduct_p_id");

            brnd = rs.getString("product.brand");

            modl1 = rs.getString("product.model");

            int qtyin = Integer.parseInt(qtyt1);

            if (qtyin <= 10) {

                statust = "low";

            } else {

                statust = "enough";

            }

            Vector v = new Vector();

            v.add(pid1);

            v.add(brnd);

            v.add(modl1);

            v.add(qtyt1);

            v.add(statust);

            dtm.setRowCount(0);
//            Object[] row1 = {pid1, brnd, modl1, qtyt1, statust};

            dtm.addRow(v);

        }

        jTable1.setModel(dtm);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

private void invt_serarchKeyReleased(java.awt.event.KeyEvent evt) {

    SuggestionList.showList(invt_serarch, inv_search_com, "SELECT product.p_id FROM product WHERE product.p_id LIKE
" + invt_serarch.getText() + "%'", "product.p_id", evt, this.getClass().getName());

}

private void invt_serarchKeyTyped(java.awt.event.KeyEvent evt) {

}

private void invt_serarchKeyPressed(java.awt.event.KeyEvent evt) {

}

private void jLabel9MouseClicked(java.awt.event.MouseEvent evt) {

if(jTable1.getRowCount()>1){

```

```

}else{

    tbled();
}

}

private void jLabel9MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel9.setBackground(cc1);

    jLabel9.setForeground(cc2);

}

private void jLabel9MouseExited(java.awt.event.MouseEvent evt) {

    jLabel9.setBackground(cc3);

    jLabel9.setForeground(cc4);

}

public static void main(String args[]) {

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Inventory.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Inventory.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Inventory.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Inventory.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

}

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Inventory().setVisible(true);

    }

});

}

```



```

// Variables declaration - do not modify

private javax.swing.JComboBox<String> inv_search_com;

private javax.swing.JTextField invt_serarch;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JSeparator jSeparator11;

private javax.swing.JTable jTable1;

void tbled() {

    DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

    try {

        ResultSet rs = database.db.search("SELECT stock.prduct_p_id, stock.qty,product.brand,product.model FROM stock
INNER JOIN product ON stock.prduct_p_id = product.p_id;");

        while (rs.next()) {

            pid1 = rs.getString("stock.prduct_p_id");

            brnd = rs.getString("product.brand");

            modl1 = rs.getString("product.model");

            qtyt1 = rs.getString("stock.qty");

            int qtyin = Integer.parseInt(qtyt1);

            if (qtyin <= 10) {

                statust = "low";

            } else {

                statust = "enough";

            }

            Object[] row1 = {pid1, brnd, modl1, qtyt1, statust};

            dtm.addRow(row1);

        }

        jTable1.setModel(dtm);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

}

```

Account Manage

```
package GUI;

import Cryptography.Cryption;

import java.awt.Color;

import java.sql.ResultSet;

import java.util.logging.Level;

import java.util.logging.Logger;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.swing.JOptionPane;

import javax.swing.JPasswordField;

public class Sign_In extends javax.swing.JFrame {

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0, 204, 255);

    Color cc4 = new Color(255, 255, 255);

    public Sign_In() {

        initComponents();

    }

    String uid1;

    String fname;

    String lname;

    String email1;

    String username1;

    String password1;

    String type;

    String adpw;

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

    }

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

        l_name.grabFocus();

    }

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

        email.grabFocus();

    }

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

```

        user_name.grabFocus();
    }

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

        pw_txt.grabFocus();
    }

    private void jLabel8MouseClicked(java.awt.event.MouseEvent evt) {

        uid1 = uid_txt.getText();

        fname = f_name.getText();

        lname = l_name.getText();

        email1 = email.getText();

        username1 = user_name.getText();

        password1 = pw_txt.getText();

        type = String.valueOf(jComboBox1.getSelectedItem());

        try {

            ResultSet rs = database.db.search("SELECT * FROM users WHERE u_id= '" + uid1 + "'");

            if (rs.next()) {

                f_name.setText(rs.getString("f_name"));

                l_name.setText(rs.getString("l_name"));

                email.setText(rs.getString("email"));

                String deun = Cryptography.Cryption.decrypt(rs.getString("username"));

                String depw = Cryptography.Cryption.decrypt(rs.getString("password"));

                System.out.println(deun);

                System.out.println(depw);

                user_name.setText(deun);

                pw_txt.setText(depw);

                jComboBox1.setSelectedItem(rs.getString("type"));

            }

        } catch (Exception ex) {

            ex.printStackTrace();

        }

    }

    private void jLabel9MouseClicked(java.awt.event.MouseEvent evt) {

        uid1 = uid_txt.getText();

        fname = f_name.getText();

        lname = l_name.getText();

        email1 = email.getText();

        username1 = user_name.getText();
    }

```

```

password1 = pw_txt.getText();

type = String.valueOf(jComboBox1.getSelectedItem());

if (!(fname.isEmpty() || lname.isEmpty() || email1.isEmpty() || username1.isEmpty() || password1.isEmpty())) {

    boolean nic1 = validateNic(uid1);

    if (nic1) {

        boolean eme = validateEmail(email1);

        if (eme) {

            try {

                JPasswordField jp = new JPasswordField();

                jp.grabFocus();

                int okcan = JOptionPane.showConfirmDialog(this, jp, "Enter Administrator Password",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

                if (okcan == JOptionPane.OK_OPTION) {

                    try {

                        adpw = Cryption.encrypt(jp.getText());

                        System.out.println(adpw + 1);

                        System.out.println(adpw + 2);

                        String typ = "Admin";

                        ResultSet rs = database.db.search("SELECT * FROM users WHERE users.`type`='" + typ + "' AND
users.password = '" + adpw + "';");

                        if (rs.next()) {

                            database.db.iud("INSERT INTO users (u_id,f_name,l_name,email,username,password,type) VALUES ('"
+ uid1 + "','" + fname + "','" + lname + "','" + email1 + "','" + Cryption.encrypt(user_name.getText()) + "','" +
Cryption.encrypt(pw_txt.getText()) + "','" + type + "')");

                            JOptionPane.showMessageDialog(this, "Saved..!");

                            uid_txt.grabFocus();

                            reset();

                        } else {

                            JOptionPane.showMessageDialog(this, "Password you entered is invalid or not an administrator",
"invalid password", JOptionPane.ERROR_MESSAGE);

                        }

                    } catch (Exception e) {

                        e.printStackTrace();

                    }

                }

            }

            if (okcan == JOptionPane.CANCEL_OPTION || okcan == JOptionPane.CLOSED_OPTION) {

                reset();

                uid_txt.grabFocus();

            }

}

```

```

        } catch (Exception ex) {

            JOptionPane.showMessageDialog(this, "Already existe");

            reset();

            uid_txt.grabFocus();

        }

    } else {

        JOptionPane.showMessageDialog(this, "Invalide Email...!");

        email.setText(null);

        email.grabFocus();

    }

} else {

    System.out.println("falce");

    JOptionPane.showMessageDialog(this, "invalide Nic");

    uid_txt.grabFocus();

    uid_txt.setText(null);

}

} else {

    JOptionPane.showMessageDialog(this, "Empty");

}

}

private void jLabel9MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel9.setBackground(cc1);

    jLabel9.setForeground(cc2);

}

private void jLabel9MouseExited(java.awt.event.MouseEvent evt) {

    jLabel9.setBackground(cc3);

    jLabel9.setForeground(cc4);

}

private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {

    new Main_Menu().setVisible(true);

    this.dispose();

}

private void jLabel12MouseClicked(java.awt.event.MouseEvent evt) {

}

private void jLabel12MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel12.setBackground(cc1);

```

```

        jLabel12.setForeground(cc2);
    }

    private void jLabel12MouseExited(java.awt.event.MouseEvent evt) {
        jLabel12.setBackground(cc3);
        jLabel12.setForeground(cc4);
    }

    private void f_nameKeyTyped(java.awt.event.KeyEvent evt) {
        if (Character.isDigit(evt.getKeyChar())) {
            evt.consume();
        }
    }

    private void l_nameKeyTyped(java.awt.event.KeyEvent evt) {
        if (Character.isDigit(evt.getKeyChar())) {
            evt.consume();
        }
    }

    private void uid_txtActionPerformed(java.awt.event.ActionEvent evt) {
        f_name.grabFocus();
    }

    private void uid_txtKeyTyped(java.awt.event.KeyEvent evt) {
    }

    private void jLabel15MouseClicked(java.awt.event.MouseEvent evt) {
        uid1 = uid_txt.getText();
        fname = f_name.getText();
        lname = l_name.getText();
        email1 = email.getText();
        username1 = user_name.getText();
        password1 = pw_txt.getText();
        type = String.valueOf(jComboBox1.getSelectedItem());

        if (!(uid1.isEmpty() || fname.isEmpty() || lname.isEmpty() || email1.isEmpty() || username1.isEmpty() ||
            password1.isEmpty())) {
            JPasswordField jp = new JPasswordField();
            jp.grabFocus();

            int okcan = JOptionPane.showConfirmDialog(this, jp, "Enter Administrator Password",
                JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

            if (okcan == JOptionPane.OK_OPTION) {
                try {
                    adpw = Cryption.encrypt(jp.getText());
                }
            }
        }
    }

```

```

String typ = "Admin";

ResultSet rs = database.db.search("SELECT * FROM users WHERE users.type='" + typ + "' AND users.password =
'" + adpw + "';");

if (rs.next()) {

    database.db.iud("UPDATE users SET users.f_name = '" + fname + "', users.l_name = '" + lname + "',
users.email = '" + email1 + "', users.username = '" + Crypton.encrypt(username1) + "',users.password = '" +
Crypton.encrypt(password1) + "',users.type = '" + type + "' WHERE users.u_id = '" + uid1 + "'");

    JOptionPane.showMessageDialog(this, "Updated...!");

    reset();

} else {

    JOptionPane.showMessageDialog(this, "Password you entered is invalid or not an administrator", "invalid
password", JOptionPane.ERROR_MESSAGE);

}

} catch (Exception e) {

    e.printStackTrace();

}

}

if (okcan == JOptionPane.CANCEL_OPTION || okcan == JOptionPane.CLOSED_OPTION) {

    reset();

    uid_txt.grabFocus();

}

} else {

    JOptionPane.showMessageDialog(this, "Can't update with empty field, please check..!");

}

}

private void jLabel15MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel15.setBackground(cc1);

    jLabel15.setForeground(cc2);

}

private void jLabel15MouseExited(java.awt.event.MouseEvent evt) {

    jLabel15.setBackground(cc3);

    jLabel15.setForeground(cc4);

}

private void jLabel8MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel8.setBackground(cc1);

    jLabel8.setForeground(cc2);

}

private void jLabel8MouseExited(java.awt.event.MouseEvent evt) {

    jLabel8.setBackground(cc3);

```

```

        jLabel8.setForeground(cc4);
    }

    private void jLabel16MouseEntered(java.awt.event.MouseEvent evt) {
        Color cc1 = new Color(0, 0, 0);

        Color cc2 = new Color(255, 255, 255);

        jLabel16.setBackground(cc1);

        jLabel16.setForeground(cc2);
    }

    private void jLabel16MouseExited(java.awt.event.MouseEvent evt) {
        Color cc3 = new Color(0, 0, 204);

        Color cc4 = new Color(153, 255, 255);

        jLabel16.setBackground(cc3);

        jLabel16.setForeground(cc4);
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Sign_In.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Sign_In.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Sign_In.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Sign_In.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Sign_In().setVisible(true);
            }
        });
    }
}

```



```
private javax.swing.JTextField email;

private javax.swing.JTextField f_name;

private javax.swing.JComboBox<String> jComboBox1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JSeparator jSeparator5;

private javax.swing.JSeparator jSeparator6;

private javax.swing.JSeparator jSeparator7;

private javax.swing.JTextField l_name;

private javax.swing.JPasswordField pw_txt;

private javax.swing.JTextField uid_txt;

private javax.swing.JTextField user_name;

void reset() {

    uid_txt.setText(null);

    f_name.setText(null);

    l_name.setText(null);

    email.setText(null);

    user_name.setText(null);

    pw_txt.setText(null);
```

```

        jComboBox1.setSelectedItem("Admin");
    }

    boolean validateEmail(String email) {

        boolean status;

        Pattern ptn = Pattern.compile("^[_A-Za-z0-9-\\+](\\.[_A-Za-z0-9-]+)*@[A-Za-z0-9-](\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$");

        Matcher matcher = ptn.matcher(email);

        if (matcher.matches()) {

            status = true;

        } else {

            status = false;

        }

        return status;
    }

    boolean MobileValidate(String in) {

        boolean Mobile_status = false;

        if (in.length() == 10) {

            if (in.charAt(0) == '0' && in.charAt(1) == '7') {

                if (in.charAt(2) == '1' || in.charAt(2) == '2' || in.charAt(2) == '5' || in.charAt(2) == '6' || in.charAt(2) == '7' || in.charAt(2) == '8') {

                    Mobile_status = true;

                }

            }

        } else {

            Mobile_status = false;

        }

        return Mobile_status;
    }

    boolean validateNic(String nic) {

        boolean nic_state = true;

        if (nic.length() == 10) {

            if (nic.contains("v") || nic.contains("x") || nic.contains("V") || nic.contains("X")) {

                try {

                    String number = nic.substring(0, 9);

                    Integer.parseInt(number);

                    int days = Integer.parseInt(nic.substring(2, 5));

                    if (366 < days) {

```

```

        nic_state = days > 500;
    }

    if (days > 866) {
        nic_state = false;
    }

    } catch (Exception e) {

        nic_state = false;

    }

    } else {

        nic_state = false;

    }

    } else {

        if (nic.length() == 12) {

            try {

                Long.parseLong(nic);

                int days = Integer.parseInt(nic.substring(4, 7));

                if (366 < days) {

                    nic_state = days > 500;

                }

                if (days > 866) {

                    nic_state = false;

                }

            } catch (Exception e) {

                nic_state = false;

            }

        }

    } else {

        nic_state = false;

    }

    }

    return nic_state;

}

```

GRN Code

```
package GUI;

import source1.SuggestionList;

import java.awt.Color;

import java.io.InputStream;

import java.sql.ResultSet;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import java.util.Vector;

import java.util.function.Supplier;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.event.TableModelEvent;

import javax.swing.event.TableModelListener;

import javax.swing.table.DefaultTableModel;

import net.sf.jasperreports.engine.JREmptyDataSource;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.view.JasperViewer;

public class GRN extends javax.swing.JFrame {

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0, 204, 255);

    Color cc4 = new Color(255, 255, 255);

    String barcode;

    String brand1;

    String model;

    String price;

    int qty1;

    String supid;

    String fname;

    String lname;

    String mobile1;

    String company_name;
```

```

String telep;

int i = 0;

DefaultTableModel dtm;

Date d1 = new Date();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

public GRN() {

    initComponents();

    idgen();

    barcd_txt.grabFocus();

    date_txt.setText(sdf.format(d1));

    dtm = (DefaultTableModel) jTable1.getModel();

    dtm.addTableModelListener(

        new TableModelListener() {

            @Override

            public void tableChanged(TableModelEvent e

            ){

                if (e.getType() == TableModelEvent.INSERT || e.getType() == TableModelEvent.DELETE || e.getType() ==
TableModelEvent.UPDATE) {

                    Double tot = 0d;

                    for (int i = 0; i < jTable1.getRowCount(); i++) {

                        tot = tot + Double.parseDouble(jTable1.getValueAt(i, 5).toString());

                    }

                    sub_total_txt.setText(String.valueOf(tot));

                }

            }

        });

}

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

}

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

}

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

}

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

}

private void jLabel14MouseClicked(java.awt.event.MouseEvent evt) {

    New_Supplier nsp = new New_Supplier();

```

```

        nsp.setVisible(true);

        this.dispose();
    }

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

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

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

        qty_txt.grabFocus();
    }

    private void jLabel17MouseClicked(java.awt.event.MouseEvent evt) {

        try {

            DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

            String grnid1 = grn_id.getText();

            supid = sup_search_txt.getText();

            String dd = date_txt.getText();

            database.db.iud("INSERT INTO grn(grn_id,date,supplier_sup_id) VALUES('"+ grnid1 + "','" + dd + "','" + supid + "')");

            DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

            for (int j = 0; j < dt.getRowCount(); j++) {

                String barcode1 = String.valueOf(dt.getValueAt(j, 0));

                String brandd1 = String.valueOf(dt.getValueAt(j, 1));

                String modell1 = String.valueOf(dt.getValueAt(j, 2));

                String pricee1 = String.valueOf(dt.getValueAt(j, 3));

                String qtyy1 = String.valueOf(dt.getValueAt(j, 4));

                int qtyy2 = Integer.parseInt(qtyy1);

                database.db.iud("INSERT INTO grn_item
(grn_item.QTY,grn_item.buying_price,grn_item.grn_grn_id,grn_item.product_p_id) VALUES ('" + qtyy2 + "','" + pricee1 +
'",'" + grn_id.getText() + "','" + barcode1 + "')");

                ResultSet rs3 = database.db.search("SELECT stock.prduct_p_id,stock.qty FROM stock WHERE stock.prduct_p_id='"
+ barcode1 + "'");

                if (rs3.next()) {

                    int qty1n = Integer.parseInt(rs3.getString("stock.qty"));

                    int neqty = qty1n + qtyy2;

                    database.db.iud("UPDATE stock SET stock.qty = '" + neqty + "' WHERE stock.prduct_p_id = '" + barcode1 + "'");

                } else {

                    System.out.println("2222");

                    database.db.iud("INSERT INTO stock (stock.prduct_p_id, stock.qty) VALUES ('" + barcode1 + "','" + qtyy2 + "')");

                }
            }
        }
    }

```

```

    }

    ResultSet rs4 = database.db.search("SELECT * FROM grn_item WHERE grn_grn_id = '" + grn_id.getText() + "'");

    if (rs4.next()) {

        grnreport(rs4.getString("grn_grn_id"));

    } else {

        System.out.println("2525");

    }

    JOptionPane.showMessageDialog(this, "Saved Succsusfully...!");


    dtm.setRowCount(0);

    sub_total_txt.setText("0");

    reset2();

    idgen();

} catch (Exception ex) {

    ex.printStackTrace();

}

barcd_txt.grabFocus();

}

private void jLabel14MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel14.setBackground(cc1);

    jLabel14.setForeground(cc2);

}

private void jLabel14MouseExited(java.awt.event.MouseEvent evt) {

    jLabel14.setBackground(cc3);

    jLabel14.setForeground(cc4);

}

private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {

    New_Product npp = new New_Product();

    npp.setVisible(true);

    this.dispose();

}

private void jLabel18MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc1);

    jLabel18.setForeground(cc2);

}

private void jLabel18MouseExited(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc3);

```

```

        jLabel18.setForeground(cc4);
    }

    private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {
        reset2();
        reset1();
    }

    private void jLabel22MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel22.setBackground(cc1);
        jLabel22.setForeground(cc2);
    }

    private void jLabel22MouseExited(java.awt.event.MouseEvent evt) {
        jLabel22.setBackground(cc3);
        jLabel22.setForeground(cc4);
    }

    private void qty_txtActionPerformed(java.awt.event.ActionEvent evt) {
        jLabel24.grabFocus();
    }

    private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {
        Main_Menu mn = new Main_Menu();
        mn.setVisible(true);
        this.dispose();
    }

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

    private void jLabel17MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel17.setBackground(cc1);
        jLabel17.setForeground(cc2);
    }

    private void jLabel17MouseExited(java.awt.event.MouseEvent evt) {
        jLabel17.setBackground(cc3);
        jLabel17.setForeground(cc4);
    }

    private void search_g_txtKeyReleased(java.awt.event.KeyEvent evt) {
        SuggestionList.showList(search_g_txt, gsearch, "SELECT grn_id FROM grn WHERE grn_id LIKE '" +
        search_g_txt.getText() + "%", "grn_id", evt, this.getClass().getName());
    }

    private void jLabel24MouseClicked(java.awt.event.MouseEvent evt) {

```



```

try {

    barcode = barcd_txt.getText();

    brand1 = brand_txt.getText();

    model = model_txt.getText();

    price = price_txt.getText();

    qty1 = Integer.parseInt(qty_txt.getText());

    int total1 = Integer.parseInt(price) * qty1;

    DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

    String brd12 = null;

    String pric12 = null;

    String qrt12 = null;

    for (int j = 0; j < dtm.getRowCount(); j++) {

        brd12 = dtm.getValueAt(j, 0).toString();

    }

    if (barcode == brd12) {

        reset1();

    }

    Object[] row = {barcode, brand1, model, price, qty1, total1};

    dtm.addRow(row);

    reset1();

    barcd_txt.grabFocus();

} catch (Exception e) {

}

}

private void jLabel24MouseEntered(java.awt.event.MouseEvent evt) {

}

private void jLabel24MouseExited(java.awt.event.MouseEvent evt) {

}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

    if (evt.getClickCount() == 2) {

        DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

        int tr = JOptionPane.showConfirmDialog(this, "Are you sure remove this row..", "confirm",
JOptionPane.YES_NO_OPTION);

        if (tr == JOptionPane.YES_OPTION) {

            dtm.removeRow(jTable1.getSelectedRow());

        }

    }

}

```

```

}

private void sup_search_txtKeyReleased(java.awt.event.KeyEvent evt) {

    SuggestionList.showList(sup_search_txt, sup_search, "SELECT sup_id FROM supplier WHERE sup_id LIKE '" +
sup_search_txt.getText() + "%'", "sup_id", evt, this.getClass().getName());

}

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

    supid = sup_search_txt.getText();

    fname = fname_c_txt.getText();

    lname = lname_c_txt.getText();

    mobile1 = model_txt.getText();

    company_name = company_c_txt.getText();

    telep = telep_txt.getText();

    try {

        ResultSet rs1 = database.db.search("SELECT supplier.f_name, supplier.l_name,
supplier.mobile,company.name,company.telephone FROM supplier INNER JOIN company ON supplier.company_id =
company.id WHERE supplier.sup_id='" + supid + "'");

        if (rs1.next()) {

            fname_c_txt.setText(rs1.getString("supplier.f_name"));

            lname_c_txt.setText(rs1.getString("supplier.l_name"));

            mobile_c_txt.setText(rs1.getString("supplier.mobile"));

            company_c_txt.setText(rs1.getString("company.name"));

            telep_txt.setText(rs1.getString("company.telephone"));

        }

    } catch (Exception e) {

    }

}

private void search_g_txtKeyPressed(java.awt.event.KeyEvent evt) {

}

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

    DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

    dtm.setRowCount(0);

    try {

        ResultSet rs1 = database.db.search("SELECT grn_item.product_p_id, grn_item.buying_price,grn_item.QTY,
product.brand,
product.model,supplier.sup_id,supplier.f_name,supplier.l_name,supplier.mobile,company.name,company.telephone
FROM grn_item INNER JOIN product ON grn_item.product_p_id = product.p_id JOIN grn ON grn_item.grn_grn_id =
grn.grn_id JOIN supplier ON grn.supplier_sup_id = supplier.sup_id JOIN company ON supplier.company_id = company.id
WHERE grn_item.grn_grn_id = '" + search_g_txt.getText() + "'");

        while (rs1.next()) {

            int qrty1 = rs1.getInt("grn_item.QTY");

            int price1 = Integer.parseInt(rs1.getString("grn_item.buying_price"));

```

```

        int total = qrty1 * price1;

        Object[] row1 = {rs1.getString("grn_item.product_p_id"), rs1.getString("product.brand"),
rs1.getString("product.model"), rs1.getString("grn_item.buying_price"), rs1.getString("grn_item.QTY"), total};

        dtm.addRow(row1);

    }

    } catch (Exception e) {

        e.printStackTrace();

    }

}

private void search_g_txtPropertyChange(java.beans.PropertyChangeEvent evt) {

}

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

}

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

}

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

}

private void barcd_txtKeyReleased(java.awt.event.KeyEvent evt) {

    SuggestionList.showList(barcd_txt, barcodecom, "SELECT p_id FROM product WHERE p_id LIKE '" + barcd_txt.getText()
+ "%'", "p_id", evt, this.getClass().getName());

}

private void price_txtKeyTyped(java.awt.event.KeyEvent evt) {

    if (!Character.isDigit(evt.getKeyChar())) {

        evt.consume();

    }

}

private void qty_txtKeyTyped(java.awt.event.KeyEvent evt) {

    if (!Character.isDigit(evt.getKeyChar())) {

        evt.consume();

    }

}

private void barcd_txtKeyTyped(java.awt.event.KeyEvent evt) {

    barcode = barcd_txt.getText();

    dtm = (DefaultTableModel) jTable1.getModel();

    String tbar1 = null;

    for (int j = 0; j < dtm.getRowCount(); j++) {

        tbar1 = dtm.getValueAt(j, 0).toString();

    }

```

```

if (barcode.equals(tbar1)) {

    barcd_txt.setText(null);

    barcd_txt.grabFocus();

} else {

    try {

        ResultSet rs1 = database.db.search("SELECT * FROM product WHERE p_id = '" + barcode + "'");

        if (rs1.next()) {

            brand_txt.setText(rs1.getString("brand"));

            model_txt.setText(rs1.getString("model"));

            price_txt.grabFocus();

        }

        ResultSet rs2 = database.db.search("SELECT grn_item.buying_price FROM grn_item WHERE
grn_item.product_p_id = '"+barcode+"'");

        if(rs2.next()){

            price_txt.setText(rs2.getString("grn_item.buying_price"));

        }

        if(price_txt.getText().isEmpty()){

            price_txt.setEditable(true);

        }else{

            price_txt.setEditable(false);

        }

    } catch (Exception ex) {

        ex.printStackTrace();

    }

}

}

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

}

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

}

private void payment_txtKeyReleased(java.awt.event.KeyEvent evt) {

    double subltte = Double.parseDouble(sub_total_txt.getText());

    double pamnt1 = Double.parseDouble(payment_txt.getText());

    double blnce1 = subltte - pamnt1;

    String blnce2 = String.valueOf(blnce1);

    balance_txt.setText(blnce2);

}

```

```

private void barcodecomPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {

    SuggestionList.setSelectedItem(barcd_txt, barcodecom);

}

private void sup_searchPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {

    SuggestionList.setSelectedItem(sup_search_txt, sup_search);

}

private void barcodecomPopupMenuCanceled(javax.swing.event.PopupMenuEvent evt) {

}

public static void main(String args[]) {

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(GRN.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(GRN.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(GRN.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(GRN.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new GRN().setVisible(true);

        }

    });

}

private javax.swing.JTextField balance_txt;

private javax.swing.JTextField barcd_txt;

private javax.swing.JComboBox<String> barcodecom;

private javax.swing.JTextField brand_txt;

private javax.swing.JTextField company_c_txt;

private javax.swing.JTextField date_txt;

```

```
private javax.swing.JTextField fname_c_txt;

private javax.swing.JTextField grn_id;

private javax.swing.JComboBox<String> gsearch;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel17;

private javax.swing.JLabel jLabel18;

private javax.swing.JLabel jLabel19;

private javax.swing.JLabel jLabel20;

private javax.swing.JLabel jLabel21;

private javax.swing.JLabel jLabel22;

private javax.swing.JLabel jLabel23;

private javax.swing.JLabel jLabel24;

private javax.swing.JLabel jLabel25;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JSeparator jSeparator10;

private javax.swing.JSeparator jSeparator11;

private javax.swing.JSeparator jSeparator12;

private javax.swing.JSeparator jSeparator13;

private javax.swing.JSeparator jSeparator14;

private javax.swing.JSeparator jSeparator15;
```

```
private javax.swing.JSeparator jSeparator16;

private javax.swing.JSeparator jSeparator17;

private javax.swing.JSeparator jSeparator18;

private javax.swing.JSeparator jSeparator19;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JSeparator jSeparator5;

private javax.swing.JSeparator jSeparator6;

private javax.swing.JSeparator jSeparator7;

private javax.swing.JSeparator jSeparator8;

private javax.swing.JSeparator jSeparator9;

private javax.swing.JTable jTable1;

private javax.swing.JTextField lname_c_txt;

private javax.swing.JTextField mobile_c_txt;

private javax.swing.JTextField model_txt;

private javax.swing.JTextField payment_txt;

private javax.swing.JTextField price_txt;

private javax.swing.JTextField qty_txt;

private javax.swing.JTextField search_g_txt;

private javax.swing.JTextField sub_total_txt;

private javax.swing.JComboBox<String> sup_search;

private javax.swing.JTextField sup_search_txt;

private javax.swing.JTextField telep_txt;

void reset1() {

    barcd_txt.setText(null);

    brand_txt.setText(null);

    model_txt.setText(null);

    price_txt.setText(null);

    qty_txt.setText(null);

}

void reset2() {

    dtm = (DefaultTableModel) jTable1.getModel();

    dtm.setRowCount(0);

    sup_search_txt.setText(null);

    fname_c_txt.setText(null);

    lname_c_txt.setText(null);

    mobile_c_txt.setText(null);
```

```

        company_c_txt.setText(null);

        telep_txt.setText(null);

        search_g_txt.setText(null);
    }

    void idgen() {

        try {

            ResultSet rs1 = database.db.search("SELECT COUNT(grn_id) AS x FROM grn");

            if (rs1.next()) {

                int x = Integer.parseInt(rs1.getString("x"));

                x++;

                grn_id.setText("G00" + x);

            }

        } catch (Exception e) {

            e.printStackTrace();

        } }

    void idgenpd() {

        try {

            ResultSet rs1 = database.db.search("SELECT COUNT(p_id) AS x FROM product");

            if (rs1.next()) {

                int x = Integer.parseInt(rs1.getString("x"));

                x++;

                barcd_txt.setText("P00" + x);

            }

        } catch (Exception e) {

            e.printStackTrace();

        } }

    void grnreport(String grnn) {

        try {

            Map<String, Object> parameter = new HashMap<>();

            parameter.put("GRNID", grnn);

            InputStream jr = GRN.class.getResourceAsStream("/BILL1/grn_mobile.jasper");

            JasperPrint jp = JasperFillManager.fillReport(jr, parameter, database.db.getConnection());

            JasperViewer.viewReport(jp, false);

        } catch (Exception e) {

            e.printStackTrace();

        } }

}

```


Product Registration

```
package GUI;

import java.awt.Color;

import java.sql.ResultSet;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class New_Product extends javax.swing.JFrame {

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0,204,255);

    Color cc4 = new Color(255, 255, 255);

    String barcd;

    String brnd;

    String mdl;

    String dtls;

    public New_Product() {

        initComponents();

        idgenpd();

    }

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

    }

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

    }

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

    }

    private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {

        barcd= barcode_txt.getText();

        try {

            ResultSet rs = database.db.search("SELECT * FROM product WHERE product.p_id = '"+barcd+"'");

            if(rs.next()){

                brand_txt.setText(rs.getString("brand"));

                model_txt.setText(rs.getString("model"));

                details_txt.setText(rs.getString("details"));

            }

        } catch (Exception e) {

        }

    }

}
```

```

}

private void jLabel22MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel22.setBackground(cc1);

    jLabel22.setForeground(cc2);

}

private void jLabel22MouseExited(java.awt.event.MouseEvent evt) {

    jLabel22.setBackground(cc3);

    jLabel22.setForeground(cc4);

}

private void jLabel17MouseClicked(java.awt.event.MouseEvent evt) {

    if (!(barcode_txt.getText().isEmpty())) {

        if (!(brand_txt.getText().isEmpty())) {

            if (!(model_txt.getText().isEmpty())) {

                if (!(details_txt.getText().isEmpty())) {

                    try {

                        barcd = barcode_txt.getText();

                        brnd = brand_txt.getText();

                        mdl = model_txt.getText();

                        dtls = details_txt.getText();

                        database.db.iud("INSERT INTO product (p_id,brand,model,details) VALUES ('" + barcd + "','" + brnd + "','" +
mdl + "','" + dtls + "')");

                        JOptionPane.showMessageDialog(this, "Saved..!");

                        reset();

                        idgenpd();

                    } catch (Exception e) {

                        e.printStackTrace();

                    }

                } else {

                    details_txt.grabFocus();

                    JOptionPane.showMessageDialog(this, "Fill Empty Field...!");

                }

            } else {

                model_txt.grabFocus();

                JOptionPane.showMessageDialog(this, "Fill Empty Field...!");

            }

        } else {

            brand_txt.grabFocus();

```

```

        JOptionPane.showMessageDialog(this, "Fill Empty Field...!");
    }
} else {
    JOptionPane.showMessageDialog(this, "Fill Empty Field...!");
    barcode_txt.requestFocus();
}
}

private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {
    GRN grn1 = new GRN();
    grn1.setVisible(true);
    this.dispose();
}

private void jLabel17MouseEntered(java.awt.event.MouseEvent evt) {
    jLabel17.setBackground(cc1);
    jLabel17.setForeground(cc2);
}

private void jLabel17MouseExited(java.awt.event.MouseEvent evt) {
    jLabel17.setBackground(cc3);
    jLabel17.setForeground(cc4);
}

private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {
    barcd = barcode_txt.getText();

    try {
        ResultSet rs = database.db.search("SELECT grn_item.grn_item_id, grn_item.grn_grn_id, grn_item.product_p_id,
grn.stock_st_id FROM grn_item INNER JOIN grn ON grn_item.grn_grn_id = grn.grn_id WHERE grn_item.product_p_id
='"+barcd+"'");

        if(rs.next()){

            String std = rs.getString("grn.stock_st_id");

            System.out.println(std);

            ResultSet rs1 = database.db.search("SELECT invoice.inv_id FROM invoice WHERE invoice.stock_st_idi ='"+std+"'");
            if(rs1.next()){

                JOptionPane.showMessageDialog(this, "Cant Delete this Item");
            }else{

                database.db.iud("DELETE FROM product WHERE p_id = '"+barcd+"'");

                JOptionPane.showMessageDialog(this, "Delete Succusfully..!");
            }
        }
    } catch (Exception e) {

```

```

    }

}

private void jLabel18MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc1);

    jLabel18.setForeground(cc2);

}

private void jLabel18MouseExited(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc3);

    jLabel18.setForeground(cc4);

}

private void barcode_txtKeyTyped(java.awt.event.KeyEvent evt) {

    try {

        barcd = barcode_txt.getText();

        brnd = brand_txt.getText();

        mdl = model_txt.getText();

        dtls = details_txt.getText();

        ResultSet rs1 = database.db.search("SELECT * FROM product WHERE p_id = '" + barcode_txt.getText() + "'");

        if (rs1.next()) {

            evt.consume();

            brand_txt.setText(rs1.getString("brand"));

            model_txt.setText(rs1.getString("model"));

            details_txt.setText(rs1.getString("details"));

        }

    } catch (Exception ex) {

        Logger.getLogger(New_Product.class.getName()).log(Level.SEVERE, null, ex);

    }

}

private void jLabel23MouseExited(java.awt.event.MouseEvent evt) {

    jLabel23.setBackground(cc3);

    jLabel23.setForeground(cc4);

}

private void jLabel23MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel23.setBackground(cc1);

    jLabel23.setForeground(cc2);

}

private void jLabel23MouseClicked(java.awt.event.MouseEvent evt) {

    barcd = barcode_txt.getText();

```

```

        brnd = brand_txt.getText();

        mdl = model_txt.getText();

        dtls = details_txt.getText();

        try {

            database.db.iud("UPDATE product SET product.brand = '"+brnd+"', product.model='"+mdl+"', product.details = '"+dtls+"' WHERE product.p_id='"+barcd+"'");

            System.out.println("115");

            System.out.println("225");

            JOptionPane.showMessageDialog(this, "Updated...!");

            reset();

            idgenpd();

        } catch (Exception e) {

        }

    }

    public static void main(String args[]) {

        try {

            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                }

            }

        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(New_Product.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(New_Product.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(New_Product.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(New_Product.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

                new New_Product().setVisible(true);

            }

        });

    }

```

```

private javax.swing.JTextField barcode_txt;

private javax.swing.JTextField brand_txt;

private javax.swing.JTextArea details_txt;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel17;

private javax.swing.JLabel jLabel18;

private javax.swing.JLabel jLabel20;

private javax.swing.JLabel jLabel22;

private javax.swing.JLabel jLabel23;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JTextField model_txt;

void reset() {

    barcode_txt.setText(null);

    brand_txt.setText(null);

    model_txt.setText(null);

    details_txt.setText(null);

}

void idgenpd() {

    try {

        ResultSet rs1 = database.db.search("SELECT COUNT(p_id) AS x FROM product");

        if (rs1.next()) {

            int x = Integer.parseInt(rs1.getString("x"));

            x++;

            barcode_txt.setText("P00" + x);  }

        } catch (Exception e) {

            e.printStackTrace();  }

    }

}

```

Add new Supplier

```
package GUI;

import database.db;

import java.awt.Color;

import java.awt.Component;

import java.sql.ResultSet;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.swing.JOptionPane;

import validation.Validation;

public class New_Supplier extends javax.swing.JFrame {

    final static String number_digit = "0125678";

    String supid;

    String fname;

    String lname;

    String email;

    String mobile;

    String cname;

    String ctele1;

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0,204,255);

    Color cc4 = new Color(255, 255, 255);

    public New_Supplier() {

        initComponents();

        idgen();

    }

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

        l_name_txt.grabFocus();

    }

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

        email_txt.grabFocus();

    }

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

        Mobile_txt.grabFocus();

    }

    private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {
```

```

        reset();
    }

    private void jLabel22MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel22.setBackground(cc1);
        jLabel22.setForeground(cc2);
    }

    private void jLabel22MouseExited(java.awt.event.MouseEvent evt) {
        jLabel22.setBackground(cc3);
        jLabel22.setForeground(cc4);
    }

    private void jLabel17MouseClicked(java.awt.event.MouseEvent evt) {
        supid = sup_id_txt.getText();
        fname = f_name_txt.getText();
        lname = l_name_txt.getText();
        email = email_txt.getText();
        mobile = Mobile_txt.getText();
        cname = c_name_txt.getText();
        ctele1 = telep_txt.getText();

        if (!(fname.isEmpty() || lname.isEmpty() || email.isEmpty() || mobile.isEmpty() || cname.isEmpty() ||
        ctele1.isEmpty())) {
            boolean eee = validateEmail(email);

            if (eee) {
                boolean mmm = MobileValidate(mobile);

                if (mmm) {
                    System.out.println("1");

                    boolean tmm = MobileValidate(ctele1);

                    if (tmm) {
                        System.out.println("2");

                        try {
                            int comId = 0;

                            String qry2 = "INSERT INTO company (name,telephone) VALUES('" + cname + "','" + ctele1 + "')";

                            database.db.iud(qry2);

                            System.out.println("CC2");

                            ResultSet r = db.search("SELECT MAX(id) FROM company");

                            while (r.next()) {
                                comId = Integer.parseInt(r.getString("MAX(id)"));
                            }
                        }
                    }
                }
            }
        }
    }

```



```

        String qry1 = "INSERT INTO supplier (sup_id,f_name,l_name,email,mobile,company_id) VALUES('" + supid +
        "','" + fname + "','" + lname + "','" + email + "','" + mobile + "','" + comId + "')";

        database.db.iud(qry1);

        JOptionPane.showMessageDialog(this, "Saved..!");

        reset();

    } catch (Exception e) {

        e.printStackTrace();

    }

    } else {

        JOptionPane.showMessageDialog(this, "Invalive Mobile");

        telep_txt.setText(null);

        telep_txt.grabFocus();

    }

    } else {

        JOptionPane.showMessageDialog(this, "Invalive Mobile");

        Mobile_txt.setText(null);

        Mobile_txt.grabFocus();

    }

    } else {

        JOptionPane.showMessageDialog(this, "Invalide Email");

        email_txt.setText(null);

        email_txt.grabFocus();

    }

    } else {

        JOptionPane.showMessageDialog(this, "assd");

    }

}

private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {

    GRN grn1 = new GRN();

    grn1.setVisible(true);

    this.dispose();

}

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

    c_name_txt.grabFocus();

}

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

}

```

```

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

    telep_txt.grabFocus();

}

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

}

private void f_name_txtKeyTyped(java.awt.event.KeyEvent evt) {

    if (Character.isDigit(evt.getKeyChar())) {

        evt.consume();

    }

}

private void l_name_txtKeyTyped(java.awt.event.KeyEvent evt) {

    if (Character.isDigit(evt.getKeyChar())) {

        evt.consume();

    }

}

private void email_txtKeyTyped(java.awt.event.KeyEvent evt) {

}

private void f_name_txtFocusLost(java.awt.event.FocusEvent evt) {

}

private void c_name_txtKeyTyped(java.awt.event.KeyEvent evt) {

}

private void jLabel17MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel17.setBackground(cc1);

    jLabel17.setForeground(cc2);

}

private void jLabel17MouseExited(java.awt.event.MouseEvent evt) {

    jLabel17.setBackground(cc3);

    jLabel17.setForeground(cc4);

}

private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {

    supid = sup_id_txt.getText();

    fname = f_name_txt.getText();

    lname = l_name_txt.getText();

    email = email_txt.getText();

    mobile = Mobile_txt.getText();

    cname = c_name_txt.getText();

    ctele1 = telep_txt.getText();

```

```

try {

    ResultSet rs = database.db.search("SELECT supplier.f_name, supplier.l_name,supplier.email,supplier.mobile,
company.name,company.telephone FROM supplier INNER JOIN company ON supplier.company_id = company.id WHERE
supplier.sup_id='"+ supid + "'");

    if (rs.next()) {

        f_name_txt.setText(rs.getString("supplier.f_name"));

        l_name_txt.setText(rs.getString("supplier.l_name"));

        email_txt.setText(rs.getString("supplier.email"));

        Mobile_txt.setText(rs.getString("supplier.mobile"));

        c_name_txt.setText(rs.getString("company.name"));

        telep_txt.setText(rs.getString("company.telephone"));

        JOptionPane.showMessageDialog(this, "Search");

    }

} catch (Exception e) {

}

}

private void jLabel18MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc1);

    jLabel18.setForeground(cc2);

}

private void jLabel18MouseExited(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc3);

    jLabel18.setForeground(cc4);

}

private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {

    supid = sup_id_txt.getText();

    fname = f_name_txt.getText();

    lname = l_name_txt.getText();

    email = email_txt.getText();

    mobile = Mobile_txt.getText();

    cname = c_name_txt.getText();

    ctele1 = telep_txt.getText();

    try {

        database.db.iud("DELETE FROM supplier WHERE sup_id = '"+supid+"'");

        reset();

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, "Cant Delete this Supplier");

    }

}

```

```

}

private void jLabel19MouseEntered(java.awt.event.MouseEvent evt) {
    jLabel19.setBackground(cc1);
    jLabel19.setForeground(cc2);
}

private void jLabel19MouseExited(java.awt.event.MouseEvent evt) {
    jLabel19.setBackground(cc3);
    jLabel19.setForeground(cc4);
}

public static void main(String args[])
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(New_Supplier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(New_Supplier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(New_Supplier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(New_Supplier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new New_Supplier().setVisible(true);
        }
    });
}

private javax.swing.JTextField Mobile_txt;
private javax.swing.JTextField c_name_txt;
private javax.swing.JTextField email_txt;
private javax.swing.JTextField f_name_txt;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel17;

private javax.swing.JLabel jLabel18;

private javax.swing.JLabel jLabel19;

private javax.swing.JLabel jLabel20;

private javax.swing.JLabel jLabel22;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JSeparator jSeparator10;

private javax.swing.JSeparator jSeparator11;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JSeparator jSeparator8;

private javax.swing.JSeparator jSeparator9;

private javax.swing.JTextField l_name_txt;

private javax.swing.JTextField sup_id_txt;

private javax.swing.JTextField telep_txt;

boolean validateEmail(String email) {

    boolean status;

    Pattern ptn = Pattern.compile("^[_A-Za-z0-9-\\w]+(\\.[_A-Za-z0-9-\\w]+)*@[A-Za-z0-9-\\w]+(\\.[A-Za-z0-9-\\w]+)*\\.([A-Za-z]{2,})$");

    Matcher matcher = ptn.matcher(email);

    if (matcher.matches()) {

        status = true;

    } else {

        status = false;

    }

    return status;

}

void idgen() {

    try {

```

```

        ResultSet rs1 = database.db.search("SELECT COUNT(sup_id) AS x FROM supplier");

        if (rs1.next()) {

            int x = Integer.parseInt(rs1.getString("x"));

            x++;

            sup_id_txt.setText("SUB00" + x);

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}

boolean MobileValidate(String in) {

    boolean Mobile_status = false;

    if (in.length() == 10) {

        System.out.println("1");

        if (in.charAt(0) == '0' && in.charAt(1) == '7') {

            System.out.println("2");

            if (in.charAt(2) == '1' || in.charAt(2) == '2' || in.charAt(2) == '5' || in.charAt(2) == '6' || in.charAt(2) == '7' ||
in.charAt(2) == '8') {

                System.out.println("3");

                Mobile_status = true;

                System.out.println("valide");

            }

        }

    } else {

        Mobile_status = false;

    }

    return Mobile_status;}

void reset() {

    idgen();

    f_name_txt.setText(null);

    l_name_txt.setText(null);

    email_txt.setText(null);

    Mobile_txt.setText(null);

    c_name_txt.setText(null);

    telep_txt.setText(null);

}

}

```

Invoice code

```
package GUI;

import source1.SuggestionList;

import java.awt.Color;

import java.io.InputStream;

import java.sql.ResultSet;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.event.TableModelEvent;

import javax.swing.event.TableModelListener;

import javax.swing.table.DefaultTableModel;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.view.JasperViewer;

public class Invoice extends javax.swing.JFrame {

    String barcd1;

    String brand1;

    String model1;

    int qty1;

    String price1;

    String warranty1;

    int cct1;

    int pricec;

    int avqt;

    int ccqt;

    Date d1 = new Date();

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    DefaultTableModel dtm;

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0, 204, 255);

    Color cc4 = new Color(255, 255, 255);

    public Invoice() {
```

```

initComponents();

idgen();

date_w.setDate(d1);

date1.setText(sdf.format(d1));

dtm = (DefaultTableModel) ino_tb1.getModel();

dtm.addTableModelListener(
    new TableModelListener() {
        @Override
        public void tableChanged(TableModelEvent e
    ) {
            if (e.getType() == TableModelEvent.INSERT || e.getType() == TableModelEvent.DELETE || e.getType() ==
TableModelEvent.UPDATE) {
                Double tot = 0d;

                for (int i = 0; i < ino_tb1.getRowCount(); i++) {
                    tot = tot + Double.parseDouble(ino_tb1.getValueAt(i, 6).toString());
                }

                Sub_total_txt.setText(String.valueOf(tot));
            }
        }
    }
);
}

private void yearlyActionPerformed(java.awt.event.ActionEvent evt) {
    jPanel3.removeAll();
    jPanel3.add(jPanel4);
    jPanel3.repaint();
    jPanel3.revalidate();
}

private void customActionPerformed(java.awt.event.ActionEvent evt) {
    jPanel3.removeAll();
    jPanel3.add(jPanel5);
    jPanel3.repaint();
    jPanel3.revalidate();
}

private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {
    Main_Menu mn = new Main_Menu();
    mn.setVisible(true);
}

```



```

        this.dispose();
    }

    private void jLabel24MouseClicked(java.awt.event.MouseEvent evt) {
        New_Customer ncc = new New_Customer();
        ncc.setVisible(true);
        this.dispose();
    }

    private void jLabel24MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel24.setBackground(cc1);
        jLabel24.setForeground(cc2);
    }

    private void jLabel24MouseExited(java.awt.event.MouseEvent evt) {
        jLabel24.setBackground(cc3);
        jLabel24.setForeground(cc4);
    }

    private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {
        reset2();
        reset1();
    }

    private void jLabel22MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel22.setBackground(cc1);
        jLabel22.setForeground(cc2);
    }

    private void jLabel22MouseExited(java.awt.event.MouseEvent evt) {
        jLabel22.setBackground(cc3);
        jLabel22.setForeground(cc4);
    }

    private void jLabel17MouseClicked(java.awt.event.MouseEvent evt) {
        if (!(mobile_c_txt.getText().isEmpty() || fname_c_txt.getText().isEmpty() || lname_c_txt.getText().isEmpty())) {
            double ttlep = Double.parseDouble(Sub_total_txt.getText());
            double pymntp = Double.parseDouble(payment_txt.getText());
            if (pymntp >= ttlep) {
                try {
                    dtm = (DefaultTableModel) ino_tb1.getModel();

                    String prdti = null;

                    String warrnt1 = null;

                    String prict1 = null;

```

```

String qtyt1 = null;

String wwid = null;

String invid1 = inv_id.getText();

String ddd1 = date1.getText();

String ccci = null;

ResultSet rs2 = database.db.search("SELECT c_id FROM customer WHERE customer.mobile = '" +
mobile_c_txt.getText() + "';");

if (rs2.next()) {

    ccci = rs2.getString("c_id");

}

database.db.iud("INSERT INTO invoice (inv_id,date,customer_c_id) VALUES('" + invid1 + "','" + ddd1 + "','" +
ccci + "')");

for (int p = 0; p < dtm.getRowCount(); p++) {

    prdti = dtm.getValueAt(p, 0).toString();

    warrnt1 = dtm.getValueAt(p, 3).toString();

    prict1 = dtm.getValueAt(p, 4).toString();

    qtyt1 = dtm.getValueAt(p, 5).toString();

    database.db.iud("INSERT INTO warranty (start_date,end_date) VALUES('" + date1.getText() + "','" + warrnt1 +
"')");

    ResultSet rs1 = database.db.search("SELECT MAX(warranty.w_id) AS x FROM warranty ");

    if (rs1.next()) {

        wwid = rs1.getString("x");

    }

    database.db.iud("INSERT INTO invoice_item
(selling_price,QTY,invoice_inv_id,customer_c_id,warranty_id,product_p_id) VALUES('" + prict1 + "','" + qtyt1 + "','" + invid1
+ "','" + ccci + "','" + wwid + "','" + prdti + "')");

}

invcereport(invid1);

ResultSet rs3 = database.db.search("SELECT stock.prduct_p_id,stock.qty FROM stock WHERE
stock.prduct_p_id='" + prdti + "'");

if (rs3.next()) {

    System.out.println("01");

    int nqty = Integer.parseInt(qtyt1);

    int gqty = Integer.parseInt(rs3.getString("stock.qty"));

    if (nqty <= gqty) {

        int sntqty = gqty - nqty;

        System.out.println(prict1);

        System.out.println(sntqty);

        database.db.iud("UPDATE stock SET stock.qty = '" + sntqty + "' WHERE stock.prduct_p_id = '" + prdti +
"');

```

```

        System.out.println("03");

        JOptionPane.showMessageDialog(this, "Saved....!");

    } else {

        JOptionPane.showMessageDialog(this, "Not enough this Item");

    }

}

reset2();

idgen();

//        dtm.setRowCount(0);

    } catch (Exception e) {

        e.printStackTrace();

    }

} else {

    JOptionPane.showMessageDialog(this, "Please Enter Correct Payment");

    payment_txt.requestFocus();

}

} else {

    JOptionPane.showMessageDialog(this, "Empty Field...!");

}

}

private void jLabel25MouseClicked(java.awt.event.MouseEvent evt) {

    if (!(barcode_txt.getText().isEmpty() || brand_txt.getText().isEmpty() || model_txt.getText().isEmpty() ||
    qty_txt.getText().isEmpty() || price_txt.getText().isEmpty())) {

        barcd1 = barcode_txt.getText();

        brand1 = brand_txt.getText();

        model1 = model_txt.getText();

        qty1 = Integer.parseInt(qty_txt.getText());

        price1 = price_txt.getText();

        pricec = Integer.parseInt(price1);

        int totalc = qty1 * pricec;

        warranty1 = warrant();

        String cccc = cost_txt.getText();

        cct1 = Integer.parseInt(cccc);

        avqt = Integer.parseInt(ava_txt.getText());

        ccqt = Integer.parseInt(qty_txt.getText());

        if (avqt < ccqt) {

            JOptionPane.showMessageDialog(this, "check QTY");

```

```

        qty_txt.grabFocus();
    } else {
        if (pricec < cct1) {
            System.out.println("");
            JOptionPane.showMessageDialog(this, "check price");
            price_txt.grabFocus();
        } else {
            DefaultTableModel dtm = (DefaultTableModel) ino_tb1.getModel();
            Object[] row = {barcd1, brand1, model1, warranty1, price1, qty1, totalc};
            dtm.addRow(row);
            reset1();
            barcode_txt.grabFocus();
        }
    }
} else {
    JOptionPane.showMessageDialog(this, "Empty Field... Please Check Befor Add to Table....!");
}
}

private void jLabel25MouseEntered(java.awt.event.MouseEvent evt) {
    jLabel25.setBackground(cc1);
    jLabel25.setForeground(cc2);
}

private void jLabel25MouseExited(java.awt.event.MouseEvent evt) {
    jLabel25.setBackground(cc3);
    jLabel25.setForeground(cc4);
}

private void barcode_txtKeyReleased(java.awt.event.KeyEvent evt) {
    SuggestionList.showList(barcode_txt, bardoce_com, "SELECT p_id FROM product WHERE p_id LIKE '" +
barcode_txt.getText() + "%'", "p_id", evt, this.getClass().getName());
}

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

}

private void ino_tb1MouseClicked(java.awt.event.MouseEvent evt) {
    if (evt.getClickCount() == 2) {

```

```

DefaultTableModel dtm = (DefaultTableModel) ino_tb1.getModel();

int tr = JOptionPane.showConfirmDialog(this, "Are you sure remove this row..", "confirm",
JOptionPane.YES_NO_OPTION);

if (tr == JOptionPane.YES_OPTION) {

    dtm.removeRow(ino_tb1.getSelectedRow());

}

}

}

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

    price_txt.grabFocus();

}

private void barcode_txtKeyTyped(java.awt.event.KeyEvent evt) {

    barcd1 = barcode_txt.getText();

    dtm = (DefaultTableModel) ino_tb1.getModel();

    String tbar1 = null;

    for (int j = 0; j < dtm.getRowCount(); j++) {

        tbar1 = dtm.getValueAt(j, 0).toString();

    }

    if (barcd1.equals(tbar1)) {

        barcode_txt.setText(null);

        barcode_txt.grabFocus();

    } else {

        try {

            barcd1 = barcode_txt.getText();

            ResultSet rs = database.db.search("SELECT * FROM product WHERE p_id = '" + barcd1 + "'");

            if (rs.next()) {

                brand_txt.setText(rs.getString("brand"));

                model_txt.setText(rs.getString("model"));

                qty_txt.grabFocus();

            }

            ResultSet rs0 = database.db.search("SELECT stock.qty FROM stock WHERE stock.prduct_p_id = '" + barcd1 + "'");

            if (rs0.next()) {

                ava_txt.setText(rs0.getString("stock.qty"));

            }

            ResultSet rs2 = database.db.search("SELECT grn_item.buying_price FROM grn_item WHERE
grn_item.product_p_id = '" + barcd1 + "'");

            if (rs2.next()) {

                cost_txt.setText(rs2.getString("grn_item.buying_price"));

```

```

    }

    } catch (Exception e) {

    }

}

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

}

private void mobile_c_txtKeyReleased(java.awt.event.KeyEvent evt) {

    SuggestionList.showList(mobile_c_txt, mobile_c_com, "SELECT customer.mobile FROM customer WHERE
customer.mobile LIKE '" + mobile_c_txt.getText() + "%'", "customer.mobile", evt, this.getClass().getName());    // TODO
add your handling code here:

}

private void mobile_c_txtKeyTyped(java.awt.event.KeyEvent evt) {

    try {

        String mobilec = mobile_c_txt.getText();

        ResultSet rs = database.db.search("SELECT * FROM customer WHERE customer.mobile = '" + mobilec + "'");

        if (rs.next()) {

            fname_c_txt.setText(rs.getString("f_name"));

            lname_c_txt.setText(rs.getString("l_name"));

        }

    } catch (Exception e) {

    }

}

private void jLabel17MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel17.setBackground(cc1);

    jLabel17.setForeground(cc2);

}

private void jLabel17MouseExited(java.awt.event.MouseEvent evt) {

    jLabel17.setBackground(cc3);

    jLabel17.setForeground(cc4);

}

private void qty_txtKeyTyped(java.awt.event.KeyEvent evt) {

    if (!Character.isDigit(evt.getKeyChar())) {

        evt.consume();

    }

}

private void price_txtKeyTyped(java.awt.event.KeyEvent evt) {

    if (!Character.isDigit(evt.getKeyChar())) {

```

```

        evt.consume();

    }

}

private void inv_search_txtKeyReleased(java.awt.event.KeyEvent evt) {

    SuggestionList.showList(inv_search_txt, inv_search_com, "SELECT invoice_item.invoice_inv_id FROM invoice_item
WHERE invoice_item.invoice_inv_id LIKE '" + inv_search_txt.getText() + "%'", "invoice_item.invoice_inv_id", evt,
this.getClass().getName());

}

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

}

private void inv_search_txtKeyTyped(java.awt.event.KeyEvent evt) {

    try {

        String invId2 = inv_search_txt.getText();

        ResultSet rs = database.db.search("SELECT invoice_item.selling_price,invoice_item.QTY,product.p_id,
product.brand,product.model,warranty.end_date FROM invoice_item INNER JOIN product ON invoice_item.product_p_id
= product.p_id JOIN warranty ON invoice_item.warranty_id = warranty.w_id WHERE invoice_item.invoice_inv_id = '" +
invId2 + "'");

        String pidt = null;

        String brdnt = null;

        String mdlIt = null;

        String wrmnt = null;

        int prcet = 0;

        int qtyt = 0;

        while (rs.next()) {

            pidt = rs.getString("product.p_id");

            brdnt = rs.getString("product.brand");

            mdlIt = rs.getString("product.model");

            wrmnt = rs.getString("warranty.end_date");

            prcet = Integer.parseInt(rs.getString("invoice_item.selling_price"));

            qtyt = Integer.parseInt(rs.getString("invoice_item.QTY"));

            int totalt = prcet * qtyt;

            DefaultTableModel dtm = (DefaultTableModel) ino_tb1.getModel();

            Object[] row1 = {pidt, brdnt, mdlIt, wrmnt, prcet, qtyt, totalt};

            dtm.setRowCount(0);

            dtm.addRow(row1);

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}

```

```

    }

    private void payment_txtKeyTyped(java.awt.event.KeyEvent evt) {

        double subtt = Double.parseDouble(Sub_total_txt.getText());

        double pymt = Double.parseDouble(payment_txt.getText());

        double blncet = subtt - pymt;

        String blncs = String.valueOf(blncet);

        balance_txt.setText(blncs);

    }

    private void inv_search_comPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {

        SuggestionList.setSelectedItem(inv_search_txt, inv_search_com);

    }

    private void mobile_c_comPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {

        SuggestionList.setSelectedItem(mobile_c_txt, mobile_c_com);

    }

    private void bardoce_comPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {

        SuggestionList.setSelectedItem(barcode_txt, bardoce_com);

    }

    public static void main(String args[]) {

        try {

            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

                if ("Nimbus".equals(info.getName())) {

                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

                    break;

                }

            }

        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

                new Invoice().setVisible(true);

            }

        });
    }

```



```
    }  
    });  
}  
  
private javax.swing.JTextField Sub_total_txt;  
  
private javax.swing.JTextField ava_txt;  
  
private javax.swing.JTextField balance_txt;  
  
private javax.swing.JTextField barcode_txt;  
  
private javax.swing.JComboBox<String> bardoce_com;  
  
private javax.swing.JTextField brand_txt;  
  
private javax.swing.ButtonGroup buttonGroup1;  
  
private javax.swing.ButtonGroup buttonGroup2;  
  
private javax.swing.JTextField cost_txt;  
  
private javax.swing.JLabel jLabel9;  
  
private javax.swing.JPanel jPanel1;  
  
private javax.swing.JPanel jPanel2;  
  
private javax.swing.JPanel jPanel3;  
  
private javax.swing.JPanel jPanel4;  
  
private javax.swing.JPanel jPanel5;  
  
private javax.swing.JPanel jPanel6;  
  
private javax.swing.JScrollPane jScrollPane1;  
  
private javax.swing.JSeparator jSeparator10;  
  
private javax.swing.JSeparator jSeparator11;  
  
private javax.swing.JSeparator jSeparator12;  
  
private javax.swing.JSeparator jSeparator13;  
  
private javax.swing.JSeparator jSeparator14;  
  
private javax.swing.JSeparator jSeparator15;  
  
private javax.swing.JSeparator jSeparator16;  
  
private javax.swing.JSeparator jSeparator17;  
  
private javax.swing.JSeparator jSeparator2;  
  
private javax.swing.JSeparator jSeparator3;  
  
private javax.swing.JSeparator jSeparator4;  
  
private javax.swing.JSeparator jSeparator5;  
  
private javax.swing.JSeparator jSeparator6;  
  
private javax.swing.JSeparator jSeparator7;  
  
private javax.swing.JSeparator jSeparator8;  
  
private javax.swing.JSeparator jSeparator9;  
  
private javax.swing.JTextField lname_c_txt;
```

```

private javax.swing.JComboBox<String> mobile_c_com;

private javax.swing.JTextField mobile_c_txt;

private javax.swing.JTextField model_txt;

private javax.swing.JRadioButton one_r;

private javax.swing.JTextField payment_txt;

private javax.swing.JTextField price_txt;

private javax.swing.JTextField qty_txt;

private javax.swing.JRadioButton three_r;

private javax.swing.JRadioButton two_r;

private javax.swing.JRadioButton yearly;

String warrant() {
    String timep = "";
    if (yearly.isSelected()) {
        if (one_r.isSelected()) {
            timep = "One Year";
        } else if (two_r.isSelected()) {
            timep = "Two Years";
        } else {
            timep = "Tree Years";
        }
    } else {
        String yd1 = sdf.format(date_w.getDate());
        timep = yd1;
    }
    return timep;
}

void reset1() {
    barcode_txt.setText(null);
    brand_txt.setText(null);
    model_txt.setText(null);
    qty_txt.setText(null);
    price_txt.setText(null);
    cost_txt.setText(null);
    ava_txt.setText(null);
}

void reset2() {
    mobile_c_txt.setText(null);

```

```

        fname_c_txt.setText(null);

        lname_c_txt.setText(null);

        Sub_total_txt.setText(null);

        payment_txt.setText("0.0");

        dtm = (DefaultTableModel) ino_tb1.getModel();

        dtm.setRowCount(0);
    }

    void idgen() {

        try {

            ResultSet rs1 = database.db.search("SELECT COUNT(inv_id) AS x FROM invoice");

            if (rs1.next()) {

                int x = Integer.parseInt(rs1.getString("x"));

                x++;

                inv_id.setText("INV00" + x);

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

    void invcereport(String invdd) {

        try {

            Map<String, Object> parameter = new HashMap<>();

            parameter.put("INVID", invdd);

            InputStream jr = GRN.class.getResourceAsStream("/BILL1/invoice_mobile.jasper");

            JasperPrint jp = JasperFillManager.fillReport(jr, parameter, database.db.getConnection());

            JasperViewer.viewReport(jp, false);

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

```

Add New Customer

```
package GUI;

import java.awt.Color;

import java.sql.ResultSet;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class New_Customer extends javax.swing.JFrame {

    String cid = null;

    String fname;

    String lname;

    String mobile;

    Color cc1 = new Color(204, 253, 244);

    Color cc2 = new Color(0, 0, 0);

    Color cc3 = new Color(0, 204, 255);

    Color cc4 = new Color(255, 255, 255);

    public New_Customer() {

        initComponents();

        idgen();

    }

    private void jLabel22MouseEntered(java.awt.event.MouseEvent evt) {

        jLabel22.setBackground(cc1);

        jLabel22.setForeground(cc2);

    }

    private void jLabel22MouseExited(java.awt.event.MouseEvent evt) {

        jLabel22.setBackground(cc3);

        jLabel22.setForeground(cc4);

    }

    private void jLabel17MouseClicked(java.awt.event.MouseEvent evt) {

        if (!(c_id_txt.getText().isEmpty() || f_name.getText().isEmpty() || l_name.getText().isEmpty() ||
        mobile_txt.getText().isEmpty())) {

            try {

                cid = c_id_txt.getText();

                fname = f_name.getText();

                lname = l_name.getText();

                mobile = mobile_txt.getText();

            }

        }

    }

}
```

```

        database.db.iud("INSERT INTO customer (c_id,f_name,l_name,mobile) VALUES('" + cid + "','" + fname + "','" + lname + "','" + mobile + "')");

        JOptionPane.showMessageDialog(this, "Saved..!");

        reset();

    } catch (Exception ex) {

        ex.printStackTrace();

    }

    } else {

        JOptionPane.showMessageDialog(this, "Please empty field...!!");

    }

}

private void jLabel20MouseClicked(java.awt.event.MouseEvent evt) {

    Invoice inv1 = new Invoice();

    inv1.setVisible(true);

    this.dispose();

}

private void jLabel23MouseClicked(java.awt.event.MouseEvent evt) {

    cid = c_id_txt.getText();

    fname = f_name.getText();

    lname = l_name.getText();

    mobile = mobile_txt.getText()

}

private void jLabel23MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel23.setBackground(cc1);

    jLabel23.setForeground(cc2);

}

private void jLabel23MouseExited(java.awt.event.MouseEvent evt) {

    jLabel23.setBackground(cc3);

    jLabel23.setForeground(cc4);

}

private void jLabel18MouseEntered(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc1);

    jLabel18.setForeground(cc2);

}

private void jLabel18MouseExited(java.awt.event.MouseEvent evt) {

    jLabel18.setBackground(cc3);

    jLabel18.setForeground(cc4);

}

```

```

    }

    private void jLabel17MouseEntered(java.awt.event.MouseEvent evt) {
        jLabel17.setBackground(cc1);
        jLabel17.setForeground(cc2);
    }

    private void jLabel17MouseExited(java.awt.event.MouseEvent evt) {
        jLabel17.setBackground(cc3);
        jLabel17.setForeground(cc4);
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(New_Customer.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(New_Customer.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(New_Customer.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(New_Customer.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new New_Customer().setVisible(true);
            }
        });
    }

    private javax.swing.JTextField c_id_txt;

    private javax.swing.JTextField f_name;

    private javax.swing.JLabel jLabel1;

    private javax.swing.JLabel jLabel17;

    private javax.swing.JLabel jLabel18;

```

```

private javax.swing.JLabel jLabel20;

private javax.swing.JLabel jLabel22;

private javax.swing.JLabel jLabel23;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JSeparator jSeparator5;

private javax.swing.JTextField l_name;

private javax.swing.JTextField mobile_txt;

void reset() {

    idgen();

    f_name.setText(null);

    l_name.setText(null);

    mobile_txt.setText(null);

}

void idgen() {

    try {

        ResultSet rs1 = database.db.search("SELECT Count(c_id) AS i FROM customer");

        if (rs1.next()) {

            int i = Integer.parseInt(rs1.getString("i"));

            i++;

            c_id_txt.setText("C00" + i);

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}

}

```

Report GUI Code

```
package GUI;

import java.io.InputStream;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import javax.swing.JOptionPane;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.view.JasperViewer;

public class Reports extends javax.swing.JFrame {

    boolean action;

    String DateToSearch;

    public Reports() {

        initComponents();

        Date d1 = new Date();

        r_date_ch.setDate(d1);

    }

    private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {

        Main_Menu mn = new Main_Menu();

        mn.setVisible(true);

        this.dispose();

    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        Date dd = new Date();

        Date ccdate = r_date_ch.getDate();

        if (ccdate.after(dd)) {

            JOptionPane.showMessageDialog(this, "Invalide Date");

            r_date_ch.setDate(dd);

        } else {

            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

            String cdc = sdf.format(r_date_ch.getDate());

            dailyIncomeReport(cdc);

        }

    }

}
```



```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    Date dd = new Date();

    Date ccdate = r_date_ch.getDate();

    if (ccdate.after(dd)) {

        JOptionPane.showMessageDialog(this, "Invalide Date");

        r_date_ch.setDate(dd);

    } else {

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

        String cdc = sdf.format(r_date_ch.getDate());

        dailyOutcomeReport(cdc);

    }

}

public static void main(String args[]) {

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Reports.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Reports.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Reports.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Reports.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new Reports().setVisible(true);

        }

    });

}

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

```

```

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private com.toedter.calendar.JDateChooser r_date_ch;

void dailyIncomeReport(String cdate) {

    try {

        Map<String, Object> parameter = new HashMap<>();

        parameter.put("DATE", cdate);

        InputStream jr = GRN.class.getResourceAsStream("/BILL1/daily_income.jasper");

        JasperPrint jp = JasperFillManager.fillReport(jr, parameter, database.db.getConnection());

        JasperViewer.viewReport(jp, false);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

void dailyOutcomeReport(String cdate1){

    try {

        Map<String, Object> parameter = new HashMap<>();

        parameter.put("DATE", cdate1);

        InputStream jr = GRN.class.getResourceAsStream("/BILL1/daily_outcome.jasper");

        JasperPrint jp = JasperFillManager.fillReport(jr, parameter, database.db.getConnection());

        JasperViewer.viewReport(jp, false);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

}

```

Email Code

```
package Model

import java.util.Properties;

import javax.mail.Message;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage

public class Email {

    public static void sendMail(String address, String subject, String message) throws Exception {

        String from = "serviceishara@gmail.com";

        String pass = "ishara123";

        String[] to = {address};

        String host = "smtp.gmail.com";

        Properties prop = System.getProperties();

        prop.put("mail.smtp.starttls.enable", "true");

        prop.put("mail.smtp.host", host);

        prop.put("mail.smtp.user", from);

        prop.put("mail.smtp.password", pass);

        prop.put("mail.smtp.port", "587");

        prop.put("mail.smtp.auth", "true");

        Session session = Session.getDefaultInstance(prop);

        MimeMessage msg = new MimeMessage(session);

        msg.setFrom(new InternetAddress(from));

        InternetAddress[] toaddress = new InternetAddress[to.length];

        for (int i = 0; i < to.length; i++) {

            toaddress[i] = new InternetAddress(to[i]);

        }

        for (int i = 0; i < toaddress.length; i++) {

            msg.setRecipient(Message.RecipientType.TO, toaddress[i]);

        }

        msg.setSubject(subject);

        msg.setContent(message, "text/html; charset=utf-8");

        Transport transport = session.getTransport("smtp");

        transport.connect(host, from, pass);

        transport.sendMessage(msg, msg.getAllRecipients());

        transport.close();    }}
```

Suggestion List class

```
package source1;

import database.db;

import java.awt.event.KeyEvent;

import java.sql.ResultSet;

import java.util.Vector;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JComboBox;

import javax.swing.JTextField;

import sqllogger.SQLLog;

public class SuggestionList {

    static int i = 0;

    static boolean key = true;

    static SQLLog writeToSql = new SQLLog();

    public static synchronized void showList(JTextField txtField, JComboBox suggestBox, String sqlQuery, String
columnName, KeyEvent evt, String classTag) {

        key = true;

        if (txtField.getText().isEmpty()) {

            suggestBox.hidePopup();

        } else {

            String s = txtField.getText().substring(0, txtField.getSelectionStart());

            if (evt.getKeyCode() == 38 || evt.getKeyCode() == 40) {

                txtField.setText(s);

            }

            try {

                if (Character.isDefined(evt.getKeyChar()) && evt.getKeyCode() != 10) {

                    ResultSet r = database.db.search(sqlQuery);

                    writeToSql.writeToFileSearch(sqlQuery, "Successfull", null, classTag);

                    Vector v = new Vector();

                    while (r.next()) {

                        v.add(r.getString(columnName));

                    }

                    DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);

                    suggestBox.setModel(dcm);

                    if (!(suggestBox.getItemCount() == 0)) {

                        suggestBox.showPopup();

                        if (evt.getKeyCode() == 8 || evt.getKeyCode() == 127) {
```

```

        txtField.setText(s);
    } else {
        int x = txtField.getText().length();
        txtField.setText(suggestBox.getSelectedItem().toString());
        txtField.setSelectionStart(x);
    }
}

}

} catch (Exception e) {
    writeToSql.writeToFileSearch(sqlQuery, "Successfull", e.toString(), classTag);
    e.printStackTrace();
}

if (suggestBox.isPopupVisible()) {
    int count = suggestBox.getItemCount();
    switch (evt.getKeyCode()) {
        case 38:
            txtField.setText(s);
            i--;
            if (i == -1) {
                i = count - 1;
            }
            suggestBox.setSelectedIndex(i);
            break;
        case 40:
            txtField.setText(s);
            i++;
            if (i == count) {
                i = 0;
            }
            suggestBox.setSelectedIndex(i);
            break;
        case 10:
            suggestBox.setSelectedIndex(i);
            txtField.setText(suggestBox.getSelectedItem().toString());
            suggestBox.hidePopup();
            i = 0;
            break;
    }
}

```

```

        default:

            i = 0;

            break;

        }

    }

}

key = false;

}

public static synchronized void setSelectedItem(JTextField txtField, JComboBox suggestBox) {

    if (key == false) {

        txtField.setText(suggestBox.getSelectedItem().toString());

    }

}

}

```

DB class

```

package database;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

public class db {

    private static Connection c;

    private static synchronized void createConnection() throws SQLException, ClassNotFoundException {

        Class.forName("com.mysql.jdbc.Driver");

        c = DriverManager.getConnection("jdbc:mysql://localhost:3307/grand_mobile", "root", "456789");

    }

    public static Connection getConnection() throws SQLException, ClassNotFoundException, Exception {

        if (c == null) {

            createConnection();

        }

        return c;

    }

    public static void iud(String qry) throws SQLException, ClassNotFoundException{

        if (c == null) {

```

```

        createConnection();
    }

    c.createStatement().executeUpdate(qry);
}

public static ResultSet search(String qry) throws SQLException, ClassNotFoundException{

    if (c == null) {

        createConnection();

    }

    return c.createStatement().executeQuery(qry);

}
}

```

Logger Class

```

package logger;

import Cryptography.Cryption;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Logger {

    private static String username;

    private static String date;

    private static String time;

    private static String type;

    private static String status;

    private static String attempt;

    public static void writeLogger(String uname, String usertype,String logstatus,int att){

        Date d = new Date();

        username = uname;

        time = new SimpleDateFormat("hh:mm:ss a").format(d);

        date = new SimpleDateFormat("yyyy-MM-dd").format(d);

        type = usertype;

        type = Cryption.encrypt(type);

        status = logstatus;

        attempt = String.valueOf(att);
    }
}

```

```

        writeFile();
    }

    private static void writeFile(){
        try {
            File folder = new File(System.getenv("PROGRAMDATA") + "\\Grand Mobile");

            File folder2 = new File(System.getenv("PROGRAMDATA") + "\\Grand Mobile\\Compleate Log");

            File file = new File(System.getenv("PROGRAMDATA") + "\\Grand Mobile\\Compleate Log\\user.log");

            if (!folder.exists()) {
                folder.mkdir();
            }

            if (!folder2.exists()) {
                folder2.mkdir();
            }

            if (!file.exists()) {
                file.createNewFile();
            }

            file.setWritable(true);

            FileWriter fr = new FileWriter(file, true);

            BufferedWriter br = new BufferedWriter(fr);

            br.write(username + "," + date + "," + time + "," + type + "," + status + "," + attempt + "\n");

            file.setWritable(false);

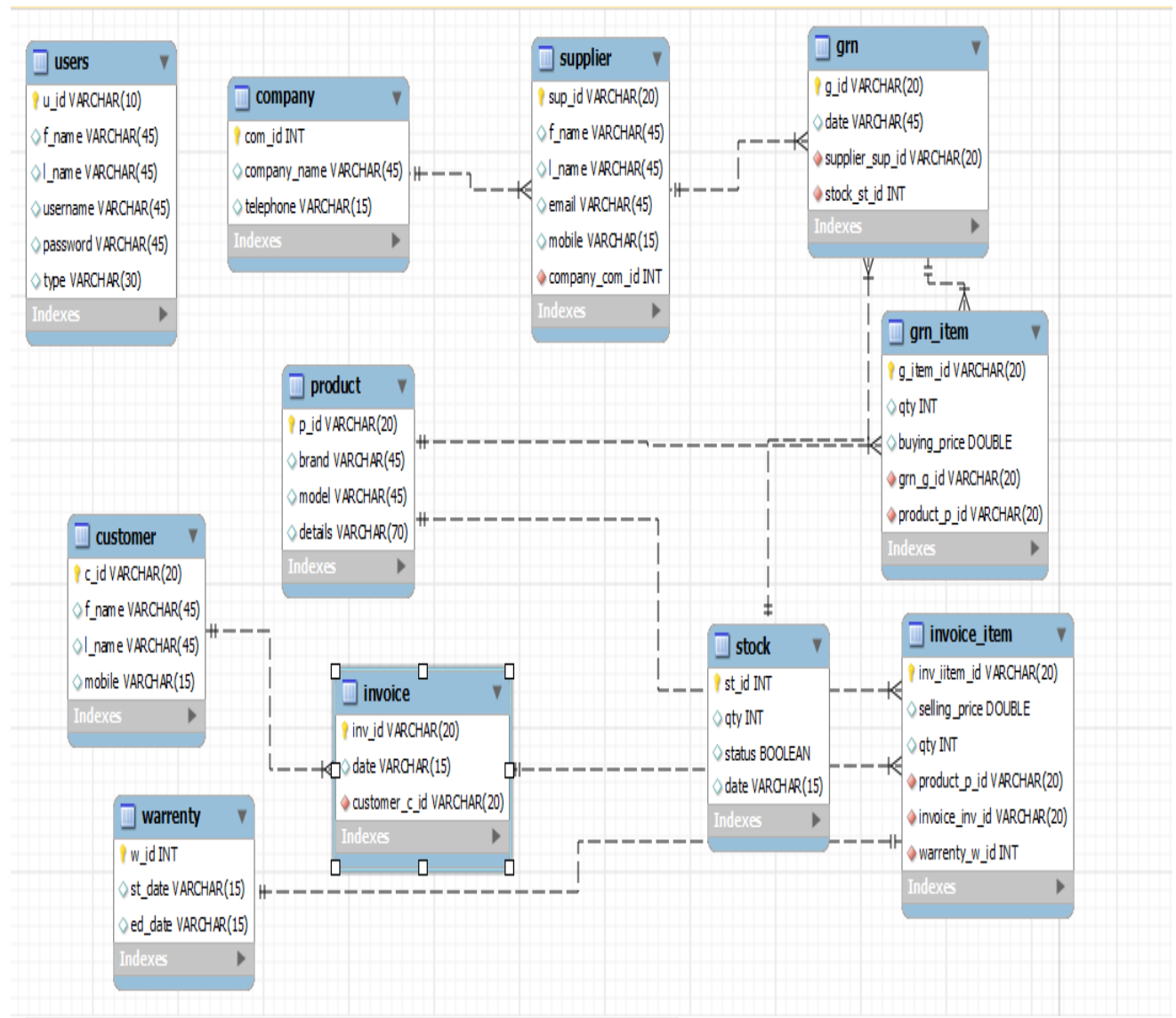
            br.close();

            fr.close();

        } catch (Exception e) {
        }
    }
}

```


ER Diagram

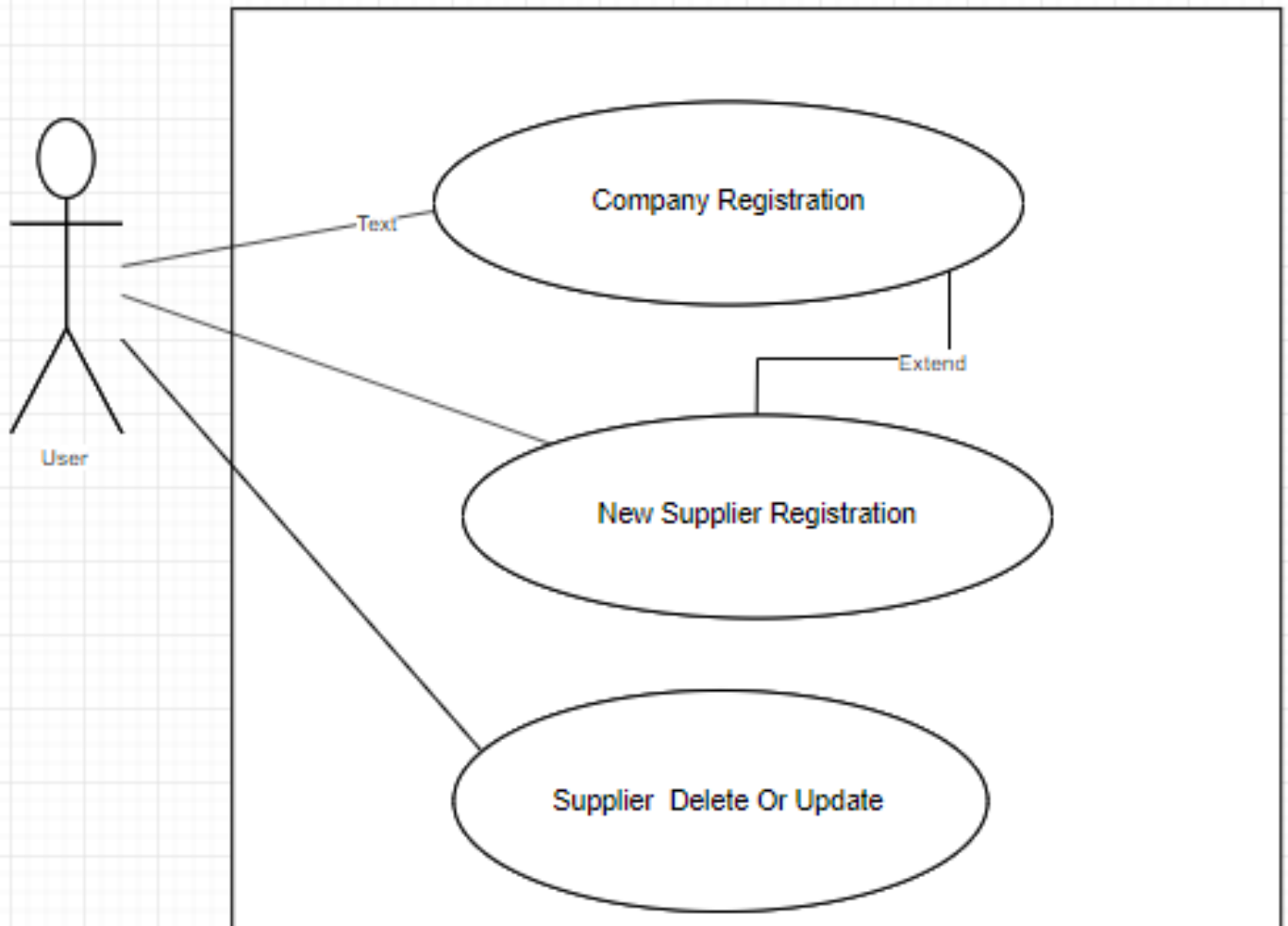


1. Candidate Name : : Dilanka Sadaruwan

2. Use case No : 01

3. Title of the Use case : Supplier Manage

4. Actor : User



01. Brief Description of the Use case: In this part the user can register Company and delete or update. And he can register Supplier delete them and update them. As a this use case Supplier depend on Company.

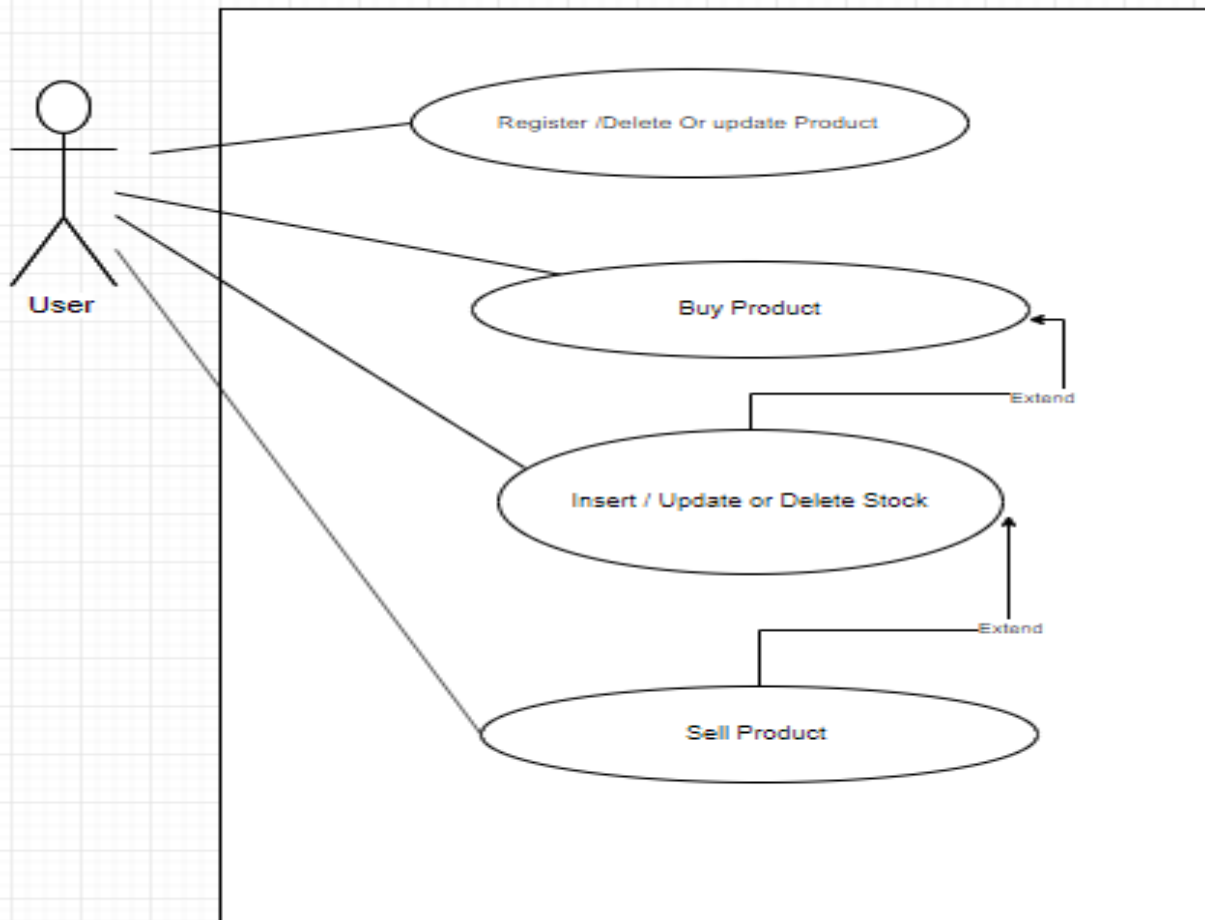
02. Assumptions: User mean Admin Or Cashier or any one user use this System.

1. Candidate Name : Dilanka Sadaruwan

2. Use case No : 02

3. Title of the Use case : Stock Manage

4. Actor :User



01. Brief Description of the Use case: This case mainly manage a stock. In this case user register delete or update products details. And he can Buy Product from supplier and Update Stock. And he can sell Product for customer and update stock. In this case user can delete stock .

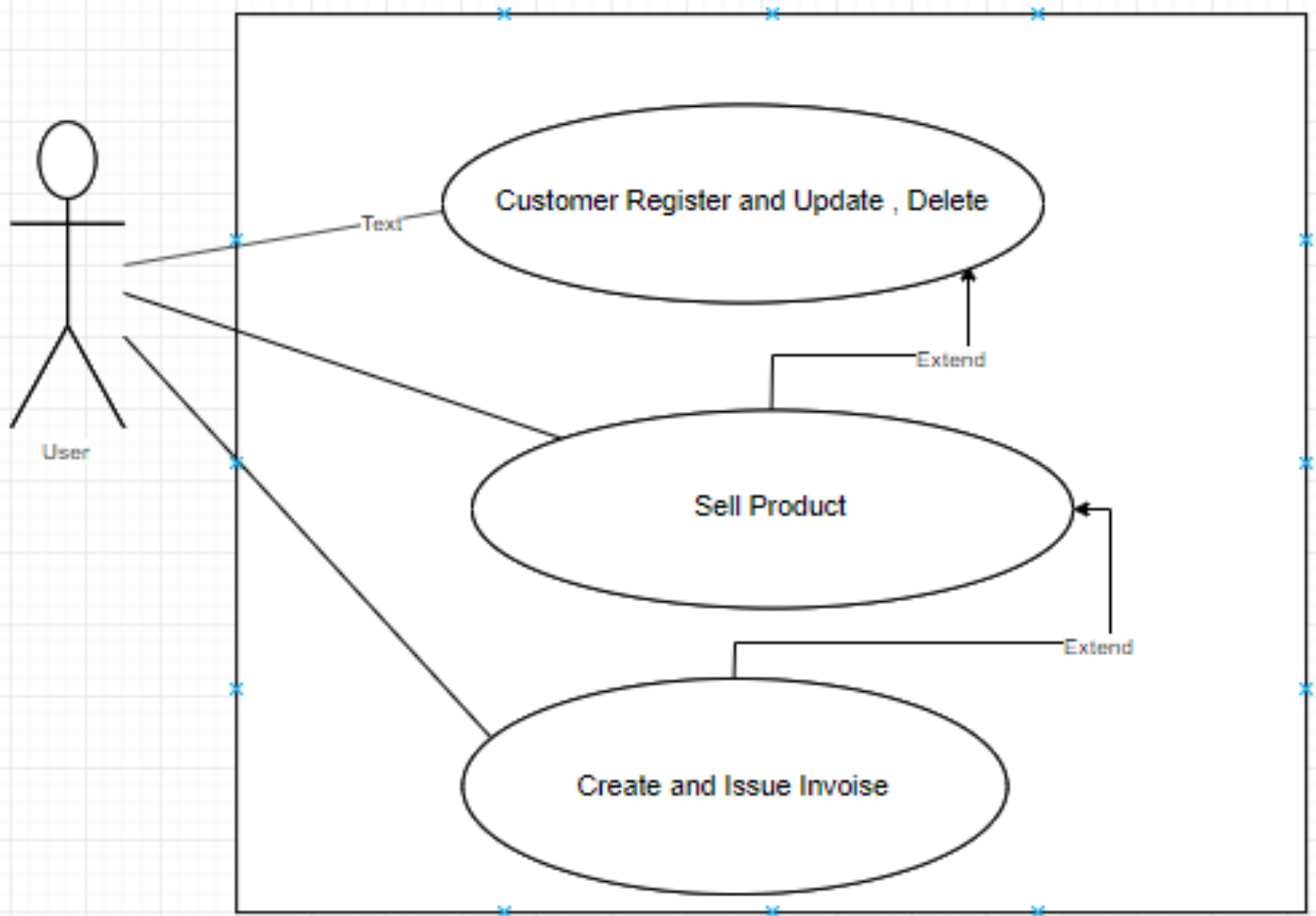
02. Assumptions : think supplier and customer in this case and user can be admin or any other user

1. Candidate Name : Dilanka Sadaruwan

2. Use case No : 03

3. Title of the Use case : Customer Manage

4. Actor :User



01.Brief Description of the Use case: In this case user can be register and delete update Customer. And he can sell Product and create an Invoice Include a Sell product. Creating invoice depend Selling products.

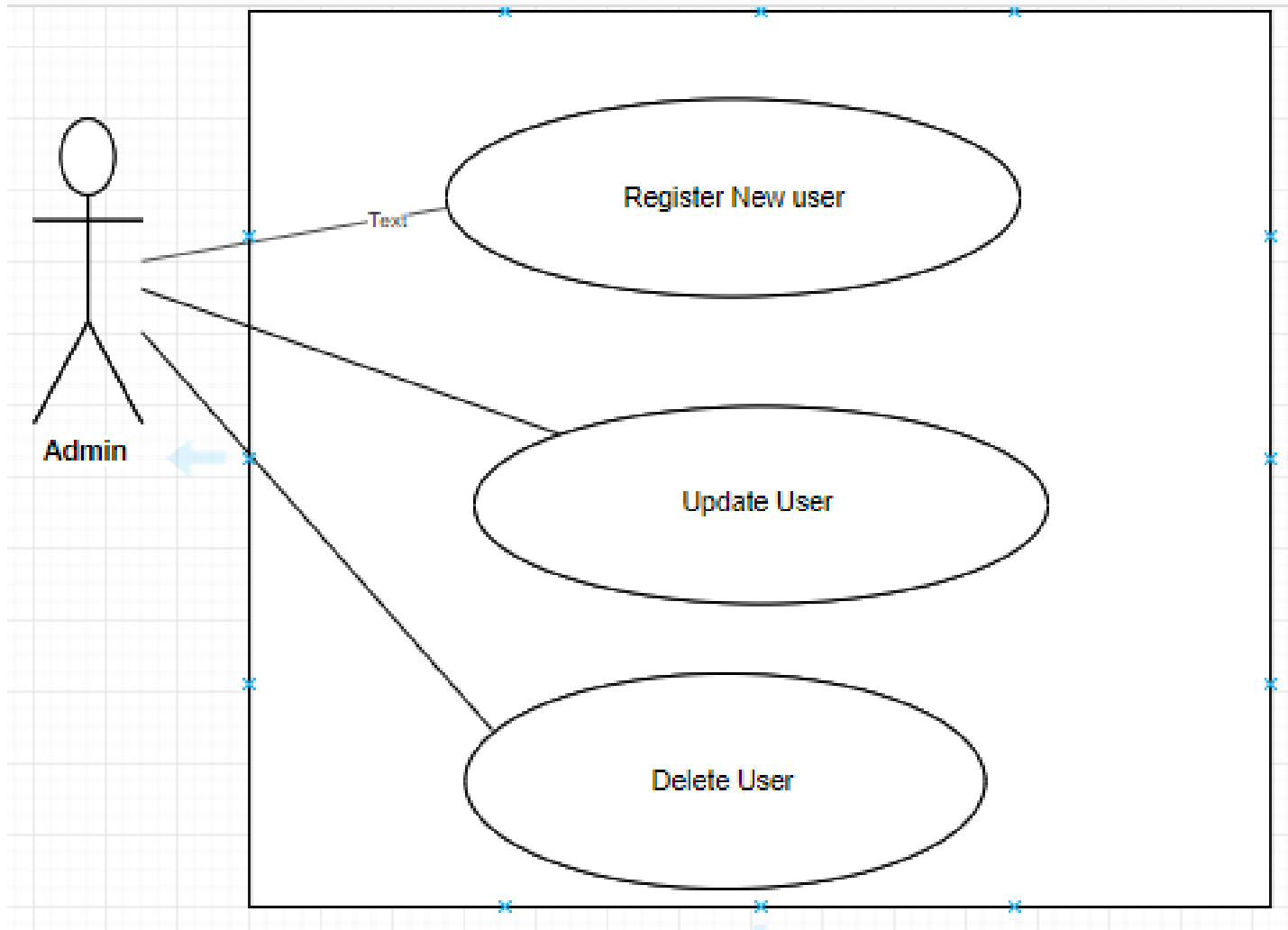
02. Assumptions : ignore Selling Part Including a Stock And Product

1. Candidate Name : Dilanka Sadaruwan

2. Use case No : 04

3. Title of the Use case : User Manage

4. Actor :Admin



01. Brief Description of the Use case: In this case Main Process is a Manage users. In this case Admin can register new user And he can update and delete users.

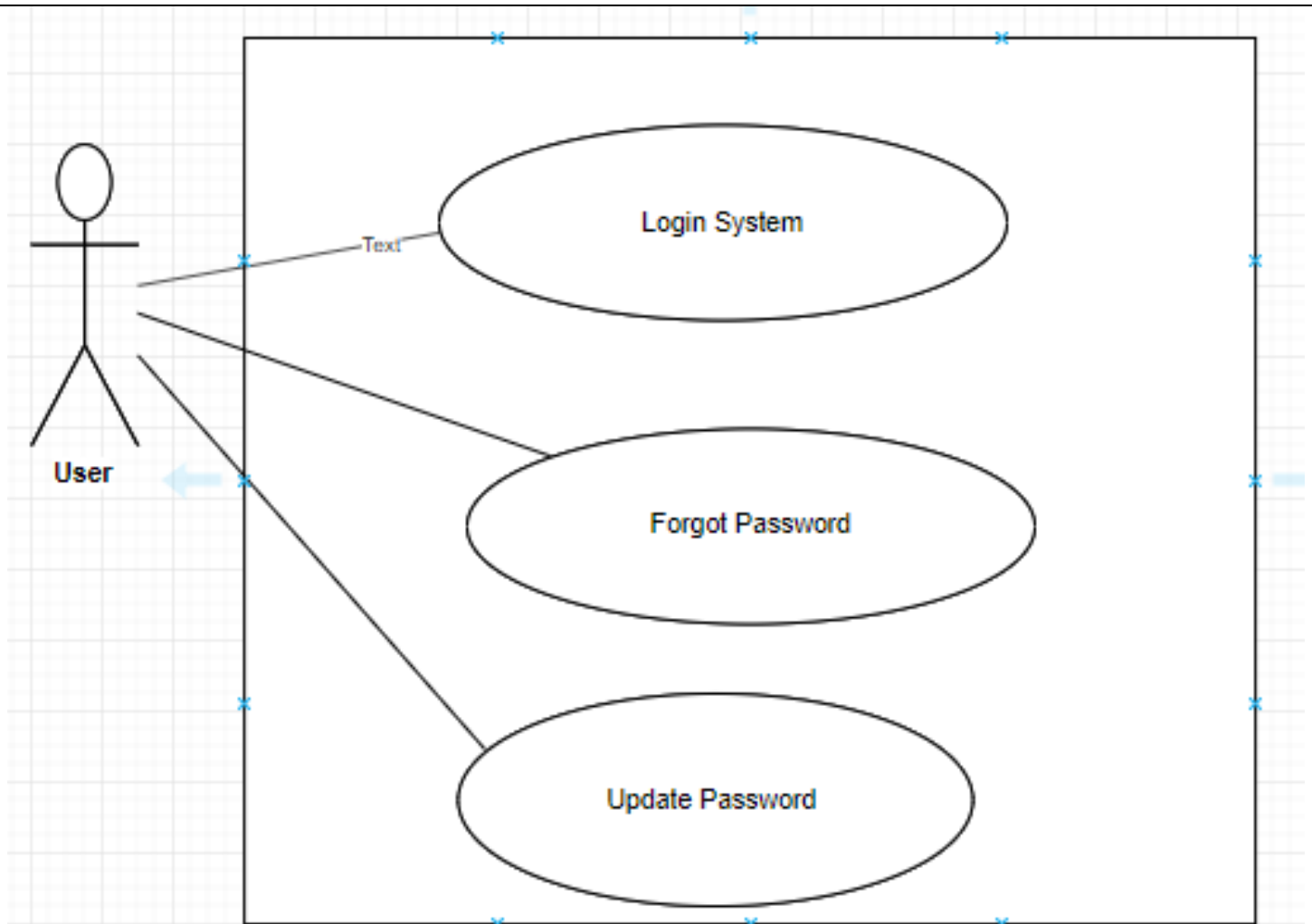
02. Assumptions : user manage can be Only Admin

1. Candidate Name : Dilanka Sadaruwan

2. Use case No : 05

3. Title of the Use case : Login

4. Actor :User



01. Brief Description of the Use case:

In this case processed by Login. User can be Login to System And when he can't remember username or password he can forgot a password and enter the new password.

02. Assumptions :