

Project Report on

# Electricity Billing System Using Java GUI

Submitted

by

Mohana Ganapathy.V

# TABLE OF CONTENTS

SI.NO	Table of Contents	Page NO
1	Abstract	3
2	Introduction	4
3	Electricity Billing System	5
4	Problem Statement	6
5	Literature Survey	7
6	Features in a project	8
7	Basic steps in constructing a Java GUI Project -Net Beans Installation -Create a Project File -Drag and Drop GUI'S -Button Configuration -Program for Calculating the bills -Run the File	9
8	Output	16
9	Testing	17
10	Future Scope	17
11	Conclusion	17

## **ABSTRACT**

The purpose of Electricity Billing System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Electricity Billing System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation the same. Basically, the project describes how to manage for good performance and better services for the clients.

## **INTRODUCTION**

The "Electricity Billing System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Electricity Billing System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the informations of Unit of Energy, Electricity, Store Record, Connections, Electricity Board. Every Electricity Billing System has different Electricity needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

## **ELECTRICITY BILLING SYSTEM**

The purpose of the Electricity Billing System is to automate the existing manual system with the help of computerized equipment and fullfledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a long period with easy access and manipulation of the same.

The required software and hardware are easily available and easy to work with. Electricity Billing System, as described above, can lead to an errorfree, secure, reliable and fast management system.

It can assist the user to concentrate on their other activities rather than concentrate on record keeping. Thus, it will help the organization's better utilization of resources. The organization can maintain computerized records without redundant entries.

That means that one need not be distracted by information that is not relevant while being able to reach the information.

## **PROBLEM STATEMENT**

The conventional system of electricity billing is not so effective; one staff has to visit each customer's house to note the meter readings and collect the data. Then, another staff has to compute the consumed units and calculate the money to be paid. Again, the bills prepared are to be delivered to customers. Finally, individual customer has to go to electricity office to pay their dues.

Hence, the conventional electricity billing system is uneconomical, requires many staffs to do simple jobs and is a lengthy process overall. In order to solve this lengthy process of billing, a web based computerized system is essential. This proposed electricity billing system project overcomes all these drawbacks with the features aforementioned. It is beneficial to both consumers and the company which provides electricity.

With the new system, there is reduction in the number of staffs to be employed by the company. The working speed and performance of the software is faster with high performance which saves time. Furthermore, there is very little chance of miscalculation and being corrupted by the staffs.

## **LITERATURE SURVEY**

- Electricity gives a wide variety of well-known electrical effects, such as lighting, static electricity, electromagnetic induction and the flow of electrical current in an electrical wire (IEEE, 2008).

The power sector of Nigeria has been struggling with the provision and maintenance of stable electricity in the country (Abubakar, 2009).

Online bill payment is an electronic payment service that allows you to setup a secure online account to make one time or recurring bill payments

- All bills and transactions are accessed on the bill pay website (ShelleyElmblad).
- Users may receive their bills late, or users may receive the wrong bill Lack of latest storage methods and facilities Human errors.

## **FEATURE'S IN A PROJECT:**

### **Text field:-**

In this project there are three main text field. In first one is Customer Id in which we have to fill Customer Id in second one is for Customer Name in this we have to fill customer Name and in the third one is for Unit which is consumed by the customer

### **Cal Button:-**

Cal Button calculate the gross amount of the unit which consumed by the customer and using java code it calculate the total bill amount. In this project for different- different unit range there is different rate of charge according to our given data

### **Print Button:-**

In print button it print the all detailed which was shown by the cal button like Customer Id, Customer Name Unit, and Amount etc.



## **Basic steps in constructing a Electricity Billing System in Java GUI Project:**

- Net Beans Installation
- Create a Project File
- Drag and Drop GUI'S
- Button Configuration
- Program for Calculating the bills
- Run the File.

### **Net Beans Installation:**

NetBeans IDE is a Free open-Source, Cross-platform Integrated Development Environment (IDE) with built-in support for the JAVA Programming Language. It can run any machine which consists of the Java Virtual Machine (JVM). It consists of many features for application development as follows:

- Drag & Drop Graphical User Interface (GUI) Creation.
- Excellent editing (advanced source code editor)
- Web-service
- Excellent debugging
- Wizards, code generation and management tools, and many more.

Step 1: You need to have a setup file of the NetBeans JAVA into your setup.

Step 2: If you didn't have the setup you can download from the following link:

**[https://netbeans.org/images\\_www/v6/download/community/8.2](https://netbeans.org/images_www/v6/download/community/8.2)**

step 3: You can download any type of setup as per your requirements from the above mention webpage.

Step 4. Right-click on the setup or you can Double-Click on the setup by using the

mouse.

Step 5. Click on the next option

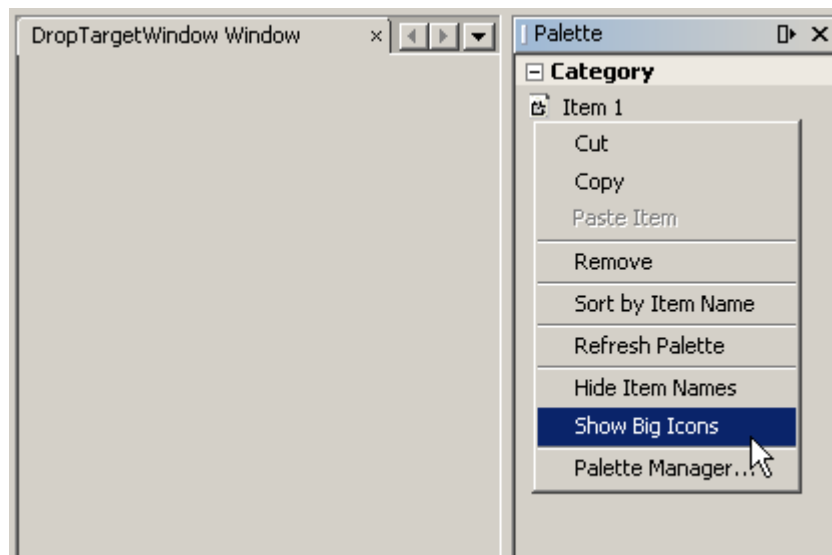
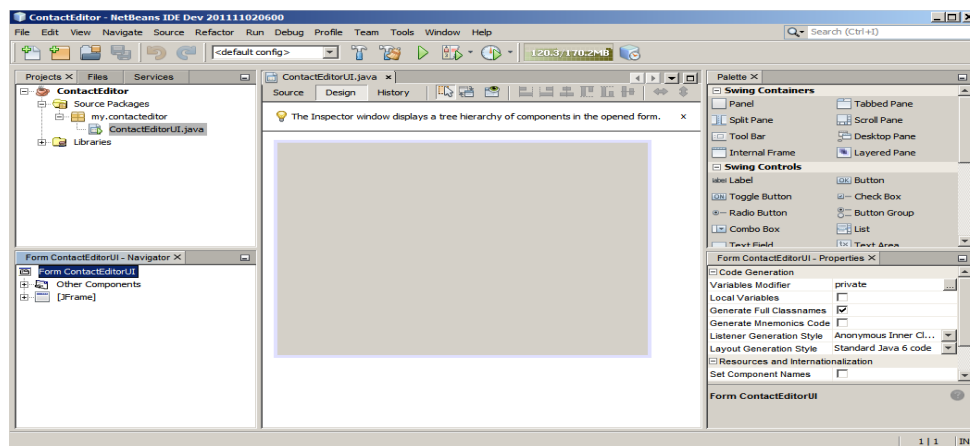
## **Create a Project File:**

### Launch the NetBeans IDE

1. In the NetBeans IDE, choose File | New Project....
2. In the New Project wizard, expand the Java category and select Java Application
3. In the Name and Location page of the wizard, do the following
4. Click Finish.

## **Drag and Drop GUI'S:**

In the Projects window, right-click the DragAndDropSample project, and choose New > File/Folder. In the New File wizard, under Categories, select NetBeans Module Development. Under File Types, select Window Component. Click Next.



### **Button Configuration:**

In this project I used only three button configurations:

#### **Button 1: Text field**

In this project there are three main text field. In first one is Customer Id in which we have to fill Customer Id in second one is for Customer Name in this we have to fill customer Name and in the third one is for Unit which is consumed by the customer

#### **Button 2 : Cal Button**

Cal Button calculate the gross amount of the unit which consumed by the customer and using java code it calculate the total bill amount. In this project for different- different unit range there is different rate of charge according to our given data

```
cid = txtcid.getText();
conu = Integer.parseInt(txtunit.getText());

if(conu < 500)
{
    chg = 1.00;
}
else if(conu < 500 && conu < 600)
{
    chg = 1.80;
}
else if(conu < 600 && conu < 800)
{
    chg = 2.80;
}
else
{
    chg = 3.00;
}

gramt = conu * chg;

print();
```

#### **Button 3: Print Button**

In print button it print the all detailed which was shown by the cal button like Customer Id, Customer Name Unit, and Amount etc.

## **Program for Calculating the bills:**

```
import java.awt.print.PrinterException;
```

```
/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to  
change this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this  
template
```

```
*/
```

```
/**
```

```
 *
```

```
 * @author Administrator
```

```
*/
```

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

```
    /**
```

```
     * Creates new form ebill
```

```
     */
```

```
    public ebill() {
```

```
        initComponents();
```

```
    }
```

```
    /**
```

```
     * This method is called from within the constructor to initialize the form.
```

```
     * WARNING: Do NOT modify this code. The content of this method is always
```

```
     * regenerated by the Form Editor.
```

```
     */
```

```
    @SuppressWarnings("unchecked")
```

```
    double chg,gramt;
```

```
    int conu;
```

```
    String cid;
```

```
    public void print()
```

```

{
    String billid = txtcid.getText();
    String customer = txtcname.getText();
    conu = Integer.parseInt(txtunit.getText());

    txtprint.setText(txtprint.getText() + "\n");
    txtprint.setText( txtprint.getText() + "Electricity Billing System" + "\n");
    txtprint.setText( txtprint.getText() + "\n");
    txtprint.setText( txtprint.getText() + "Bill No : - " + billid + "\n");
    txtprint.setText( txtprint.getText() + "Customer : - " + customer + "\n");
    txtprint.setText( txtprint.getText() + "Unit : - " + conu + "\n");
    txtprint.setText( txtprint.getText() + "Amount : - " + gramt + "\n");
    txtprint.setText(txtprint.getText() + "Thank you Come Again \n");
}
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    txtcid = new javax.swing.JTextField();
    txtcname = new javax.swing.JTextField();
    txtunit = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    txtprint = new javax.swing.JTextArea();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
    jLabel1.setText("Electricity Billing System");

```

```
jPanel1.setBorder(new  
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
```

```
jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N  
jLabel2.setText("Customer ID");
```

```
jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N  
jLabel3.setText("Customer Name");
```

```
jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N  
jLabel4.setText("Unit");
```

```
txtcid.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        txtcidActionPerformed(evt);  
    }  
});
```

```
txtunit.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        txtunitActionPerformed(evt);  
    }  
});
```

```
jButton1.setText("calculate");  
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
});
```

```
txtprint.setColumns(20);  
txtprint.setRows(5);  
jScrollPane1.setViewportView(txtprint);
```

[illegible]

```

        .addComponent(txtcid,
javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(txtunit,
javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(72, 72, 72)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 222,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(21, 21, 21))
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(82, 82, 82))))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(29, 29, 29)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(txtcid,
javax.swing.GroupLayout.PREFERRED_SIZE,

```



```
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(56, 56, 56)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA  
SELINE)  
        .addComponent(jLabel3)  
        .addComponent(txtcname,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(70, 70, 70)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA  
SELINE)  
        .addComponent(jLabel4)  
        .addComponent(txtunit,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)))  
        .addComponent(jScrollPane1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 207,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA  
SELINE)  
        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,  
59, javax.swing.GroupLayout.PREFERRED_SIZE)  
        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,  
59, javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addContainerGap(67, Short.MAX_VALUE))  
    );
```

```
        javax.swing.GroupLayout layout = new  
        javax.swing.GroupLayout(getContentPane());  
        getContentPane().setLayout(layout);  
        layout.setHorizontalGroup(
```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(205, 205, 205)
            .addComponent(jLabel1)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(25, 25, 25)
            .addComponent(jLabel1)
            .addGap(18, 18, 18)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(15, 15, 15))
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

```

```

private void txtcidActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void txtunitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handli

    cid = txtcid.getText();
    conu = Integer.parseInt(txtunit.getText());

    if(conu < 500)
    {
        chg = 1.00;
    }
    else if(conu < 500 && conu< 600)
    {
        chg = 1.80;
    }
    else if(conu < 600 && conu< 800)
    {
        chg = 2.80;
    }
    else
    {
        chg = 3.00;
    }

    gramt = conu * chg;

    print();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        txtprint.print();
    } catch (PrinterException ex) {
        ex.printStackTrace();
    }
}

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    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(ebill.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ebill.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ebill.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

```

```
}
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

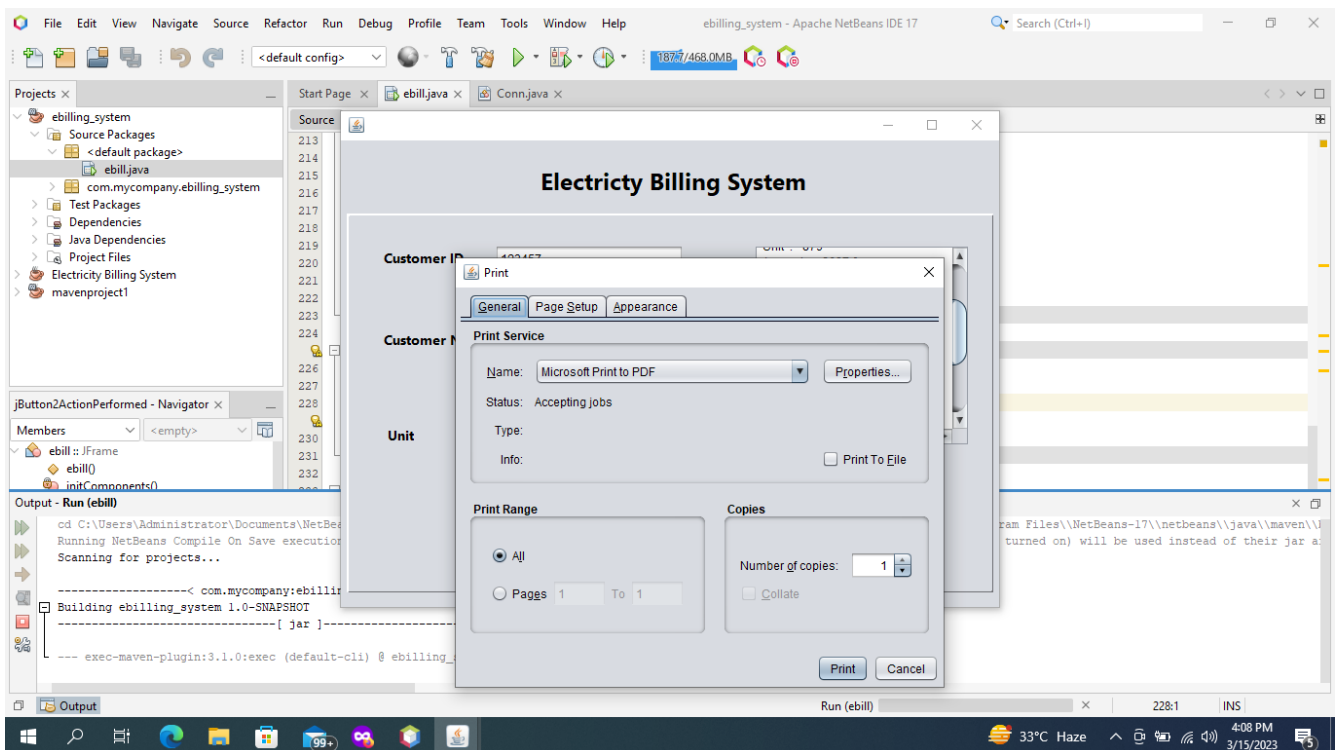
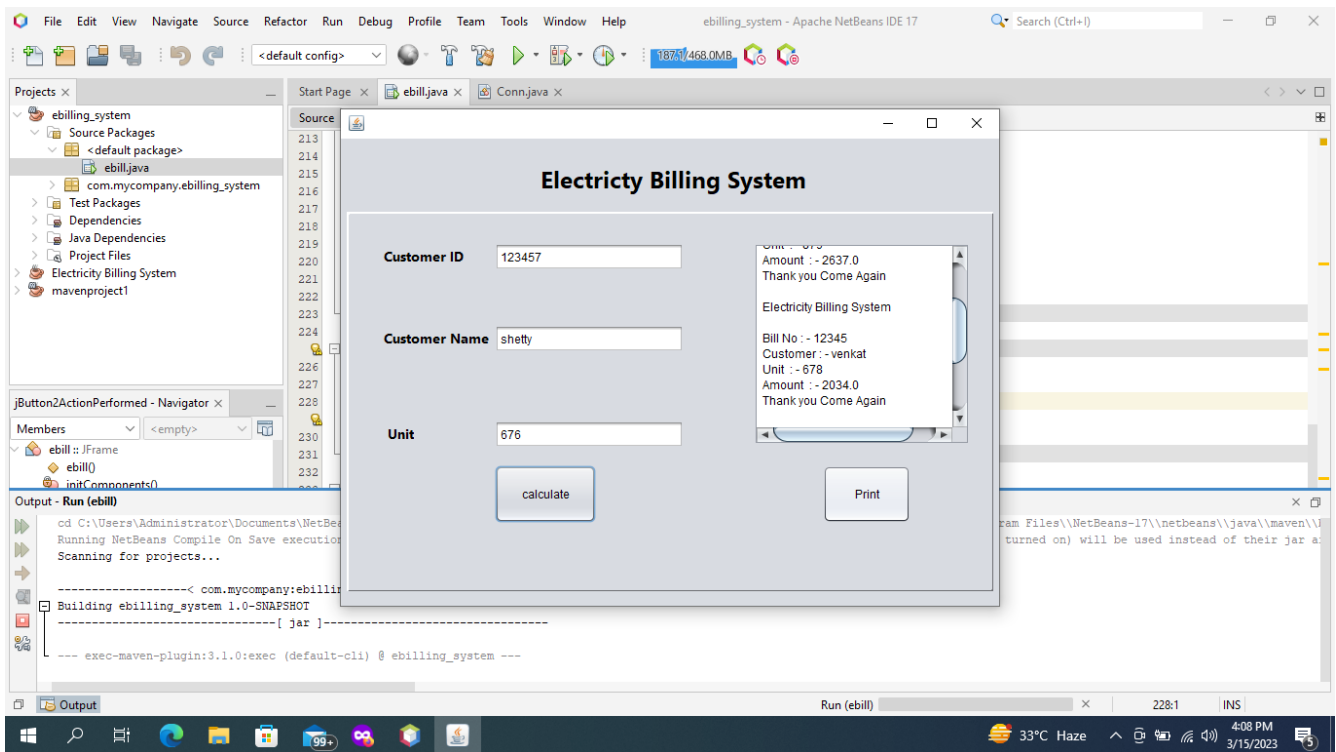
```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new ebill().setVisible(true);  
    }  
});  
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton jButton1;  
private javax.swing.JButton jButton2;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTextField txtcid;  
private javax.swing.JTextField txtcname;  
private javax.swing.JTextArea txtprint;  
private javax.swing.JTextField txtunit;  
// End of variables declaration
```

```
}
```

# OUTPUT:



## **TESTING**

The e-billing system should be thoroughly tested to ensure that it works as expected. This can be done by creating test cases for various scenarios, such as adding a new bill, updating the stock of existing accounts, and generating reports as pdf.

The test cases should verify that the data entered into the system is stored. Additionally, the system should be tested for performance and scalability, as e-billings typically deal with large amounts of data and transactions.

This can be done by simulating real-world scenarios and measuring the system's response time.

## **FUTURE SCOPE:**

I have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

I hope that the project will serve its purpose for which it is develop there by underlining success of process.

## **CONCLUSION**

In conclusion, an e-billing system using Java GUI and NetBeans can provide an efficient and user-friendly to find an easy way of billing their own electricity bill. The system can be easily customized to meet the specific requirements of the e-billing. With the help of Java GUI and NetBeans, customers can keep track of their EB bills, manage their billing, and make informed decisions about their billings.