**HINDUSTAN COLLEGE OF**

**SCIENCE & TECHNOLOGY**

**FARAH**

**MATHURA**



#### A PROJECT REPORT ON

**ELECTRICITY BILL CALCULATION**

Submitted By:

##### TISHA ARORA( CS , II YEAR )

##### 

Towards The Partial Fulfillment of the

##### BACHELOR OF TECHNOLOGY (CS)

##### [2021-2025]

**CERTIFICATE**

This is to certify that the project

**“ELECTRICITY BILL**

**CALCULATION”**

Has been satisfactorily completed by

##### TISHA ARORA

##### 

##### 

Towards the partial fulfillment of the ‘BTech CS’ for the academic year [2021-2025] at HINDUSTAN COLLEGE OF SCIENCE & TECHNOLOGY , FARAH, MATHURA is approved.

### 

**ACKNOWLEDGEMENT**

With immense pleasure we are presenting **“Electricity Bill Calculation**” Project report as part of the curriculum of ‘Bachelor of Technology’. We wish to thank all the people who gave us unending support.

We express our profound thanks to our project guide and project incharge Prof. “Yash Sir” and all those who have indirectly guided and helped us in preparation of this project.

# TABLE OF CONTENTS

|  |
| --- |
| 1. INTRODUCTION   2. SYSTEM ANALYSIS   1. FEASIBILITY STUDY   4. LITERATURE SURVEY  5. ABOUT TECHNOLOGY  6. SOFTWARE ENGINEERING APPROACH  7. FUTURE SCOPE AND LIMITATIONS  8. SOURCE CODE & EXECUTION  9. EXECUTION SNAPSHOTS  10. CONCLUSION  11. BIBLIOGRAPHY |

## Chapter 1

**INTRODUCTION**

**1.1:PROJECT DEFINITION**

Electricity Billing System is a software-based application.

The software provides facility of data sharing, it does not require any staff as in the conventional system. Once it is installed on the system only the meter readings are to be given by the admin where customer can view all details, it has the provision of security restriction.

The system excludes the need of maintaining paper electricity bill, administrator does not have to keep a manual track of the users, users can pay the amount without visiting the office. Thus, it saves human efforts and resources.

The electricity billing software calculates the units consumed by the customer and makes bills, it requires small storage for installation and functioning. There is provision for debugging if any problem is encountered in the system.

**1.2: OBJECTIVES**

The objectives of our project are as follows:

##### To keep the information of consuming unit energy of current month.

* + - To keep the information of Customer.
    - To keep the information of consuming unit energy of previous month.
    - To calculate the units consumed every month regularly.
    - To generate the bills adding penalty and rent.
    - To save the time by implementing payment process online**.**
* This project aims to serve the electricity department by computerizing the building system .
* It mainly focuses on the calculation of units consumed during the specified time and the money to be charged by the electricity officers.
* This computer a system will make the overall billing system easily accessible, comfortable and effective for consumers.

**1.3: SCOPE OF THE PROJECT**

The main objective of the electricity billing system is to manage the details of electricity bill connections store record customer it manages all the information about electricity electricity board customer electricity the project is totally built at administrative and and thus only the administrator is guaranteed the access the purpose of the project is to build an application program to reduce the manual work for managing the electricity will electricity board and connections it tracks all the details about the connections is to record in customer

## 1.4: PLATFORM SPECIFICATION

##### 1.4.1:Hardware Requirements:

* + - Hardware Specification: -Processor Intel Pentium III6.30GHz or higher
    - RAM: -128MB
    - Monitor: -LCD Monitor
    - Keyboard: -Standard keyboard
    - Mouse: -Compatible mouse
    - Storage:-20GB

##### 1.4.2:Software Requirements:

* + - Operating System: -Windows 7
    - Software: -JRE
    - Any Browser :-Chrome , MS edge
    - Java JDK1.7

.

## CHAPTER 2

**SYSTEM ANALYSIS**

* **Cost:**This project system excludes the need of maintaining paper electricity bill as all the electricity bill records are managed electronically.
* **Information Processing** :Administrator doesn't have to keep a manual track of the users. The system automatically calculates the fine.Users don't have to visit to the office for bill payment.
* **Travel:**There is no need of delivery boy for delivery bills to user's place.
* **Effect:**Thus, it saves human efforts and resources.To recover this cost of supply and sustain its operations.
* **Rebate:**To encourage consumers to conserve energy efficiently and to promote renewable energy initiatives and usage.It is allowed by DISCOMs ,if solar water heaters are used.
* **Arrears:**Any pending payments relating to the previous month are added in the current month bill.
* **Incentives:**Bills are paid through the electronic clearing system and consumers pay INR 1000 in advance towards the next month bill.
* **Other Charges:**Apart from the charges it is binding on the electricity company to mention the datails of all such charges in the consumer bill.If the consumer needs clarification about any of the charges mentioned in the bill,they can request the same from the electricity company to provide such clarification to the consumer.

## CHAPTER 3

**FEASIBILITY STUDY**

Feasibility study is the face in which the analyst check that the candidate system is feasible for the organisation or not this entails identification description and evaluation of the system feasibility studies done to select the best system that meet the performance requirements.

if the feasibility study is to serve as a decision document it must answer key questions.

1. Is there a new and better way to do the job that will benefit the user ?
2. What are the cost and saving of alternative?
3. What is recommended ?

The most successful system project are not necessarily the biggest aur most visible in the business but rather truly meet users expectations.

**3.1:FEASIBILITY CONSIDERATIONS**

Three key considerations are involved in the feasibility study.

They are as follows:

**3.2:ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system we analyse the candidate system is feasible as then the manual system because it saves the money, time and manpower. It also feasible according to the cost benefit analysis .

**3.3:TECHNICAL FEASIBILITY**

Technical feasibility centres around the technology used it means the candidate system is technically feasible that is it don't have any technical fault and work for properly in the given environment our system is technically feasible it is providing a required output.

**3.4:BEHAVIORAL FEASIBILITY**

Behavioral feasibility is the analysis of behaviour of the candidate system in this we analyze that the candidate system is working properly or not if working then it communicating proper with the environment or not all

these metals and analyzed. and a good candidate system is prepared due to the change of system what is the change in behaviour of the users these factors are all analyzed.

**CHAPTER – 4**

**Literature Survey**

**4.1:Implementation of Operations**

❖

Adding Customer: Here admin can add new customer to the customer list who started using electricity bill system.

❖

Searching Deposit Details: Here admin can search according to meter number and month to view deposit details.

❖

Viewing Details: Here admin and user can view customer details and about details.

❖

Adding Tax: Here admin can add tax details.

❖

Updating Customer: Here customer can update his/her details by using meter number of the customer.

❖

Delete Customer: Here admin can delete details based on meter number.

**4.2: Visual Studio online:-**

Visual Studio Code, also commonly referred to as VS Code,[10] is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.[11] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer.



When you develop web projects in Visual Studio, you need a web server to test or run them. Visual Studio lets you test with different web servers, including IIS Express, Internet Information Services (IIS), External Hosts, or Custom Web

## 

## Servers. You can use any of these web servers with a file-based web application project. For a file-based web site project, you can use IIS Express. This topic describes each of the web servers and how to select a web server for testing.

IIS Express includes many of the features of IIS 7 and IIS 8 web servers without requiring you to install and manage IIS on your computer. Because IIS Express is similar to IIS 7 and IIS 8, it lets you transition web sites to IIS more smoothly than using other Web servers.

IIS Express offers the following features:

- It supports and enables the same extensibility model and Web.config file settings as IIS 7 and IIS 8.

- It does not require changes in your web application code.

- It can be installed side-by-side with the full version of IIS and other web servers. You can choose a different web server for each project.

**4.3:Apache NetBeans IDE 16**

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on Windows, macOS, Linux and Solaris.

Modularity:

All the functions of the IDE are provided by modules. Each module provides a well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules.

NetBeans IDE offers first-class tools for Java web, enterprise, desktop, and mobile application development. It is consistently the first IDE to support the latest versions of the JDK, Java EE, and JavaFX. It provides smart overviews to help you understand and manage your applications, including ouf-of-the-box support for popular technologies such as Maven.

With its end-to-end application development features, constantly improving Java Editor, and continual speed and performance enhancements, NetBeans IDE sets the standard for application development with cutting edge technologies out of the box.



**IDE Bundle for PHP:**

NetBeans supports PHP since version 5.6. The bundle for PHP includes:

* syntax highlighting, code completion, occurrence highlighting, error highlighting, CVS version control
* semantic analysis with highlighting of parameters and unused local variables.

**CHAPTER - 5**

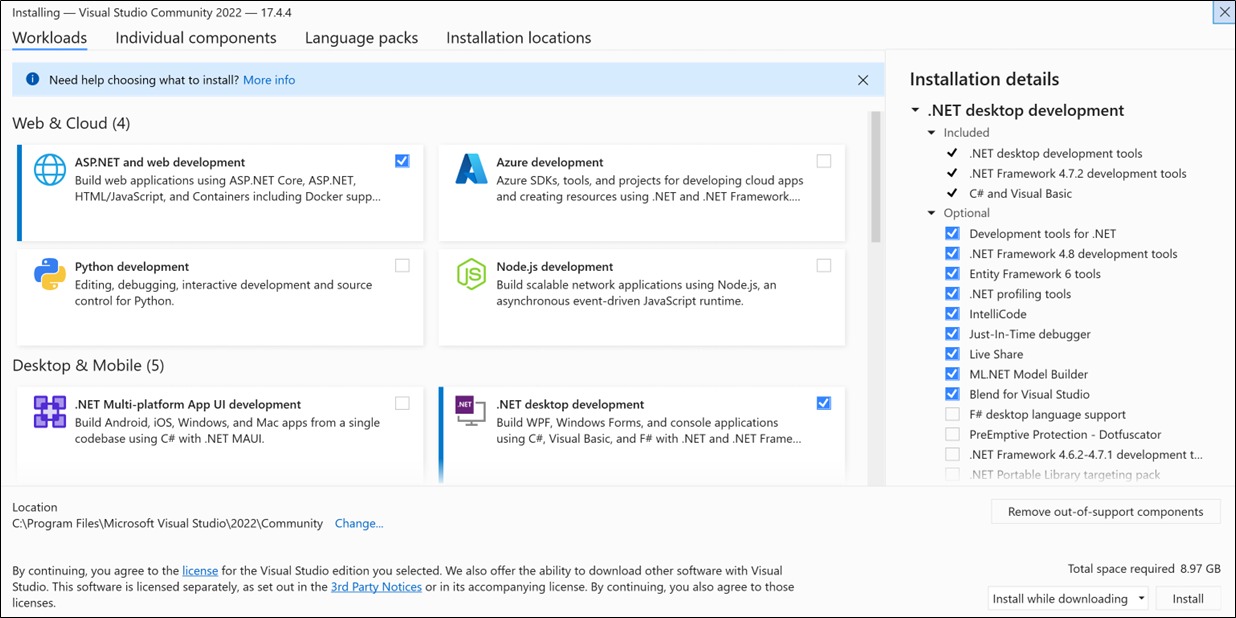
**5.1:About Technology**

**VS code**

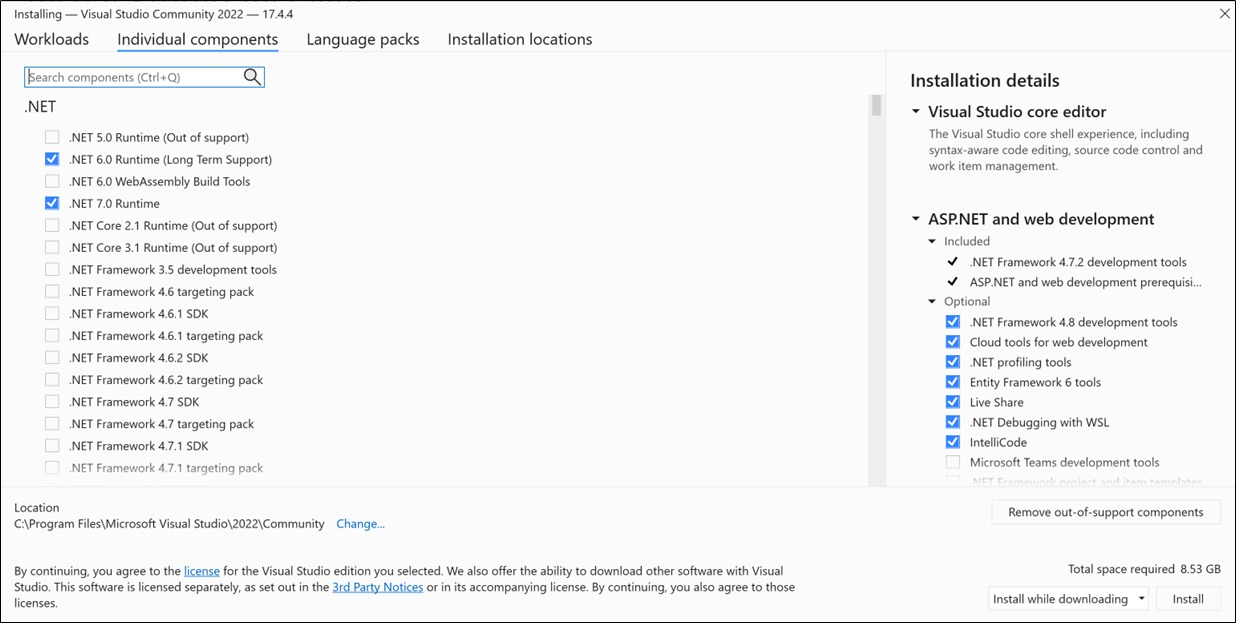
* This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets.
* Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse
* Unwanted files and folders can be excluded from the project tree via the settings.
* A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, time travel debuggers, perform static code analysis, and add code linters using the Language Server Protocol.
* This allows users to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.
* Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development
* This allows it to be used on any platform, in any locale, and for any given programming language.
* Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled

**5.2:Download Software**

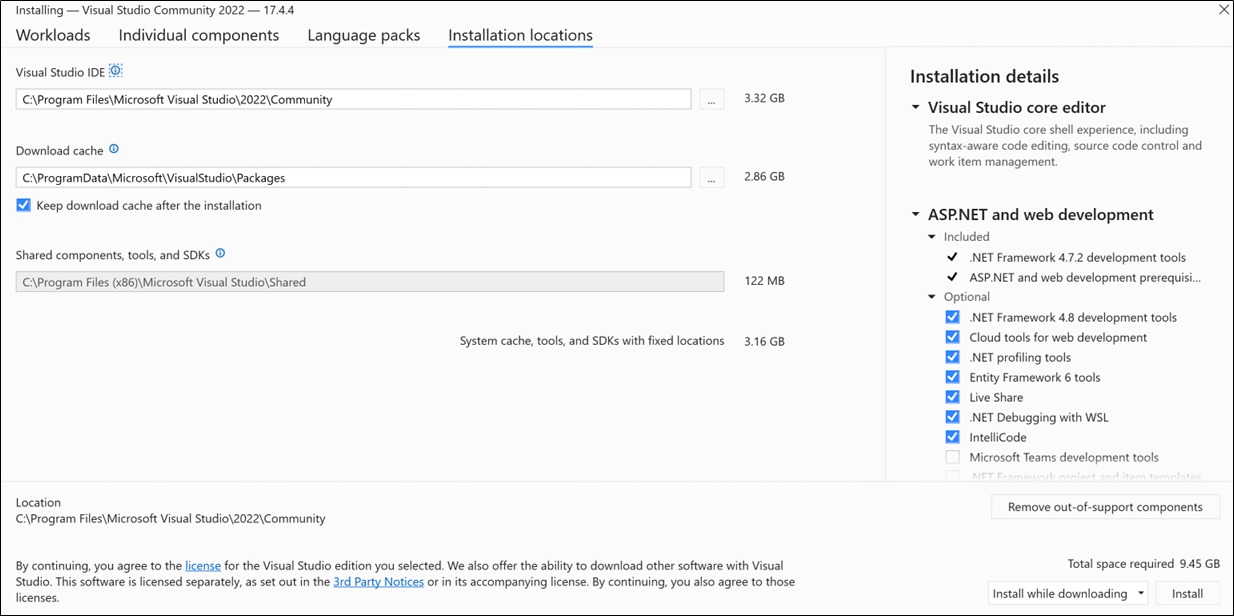
* Check the system requirements. These requirements help you know whether your computer supports Visual Studio 2022.
* Apply the latest Windows updates. These updates ensure that your computer has both the latest security updates and the required system components for Visual Studio.
* Reboot. The reboot ensures that any pending installs or updates don't hinder your Visual Studio install.
* Free up space. Remove unneeded files and applications from your system drive by, for example, running the Disk Cleanup app.
* Select the workload you want in the Visual Studio Installer.



* By installing or adding individual components from the Individual components tab. Choose what you want, and then follow the prompts.



* You can reduce the installation footprint of Visual Studio on your system drive. For more information, see Select installation locations.



**CHAPTER 6**

**SOFTWARE ENGINEERING APPROACH**

**6.1:REQIREMENT ANALYSIS**

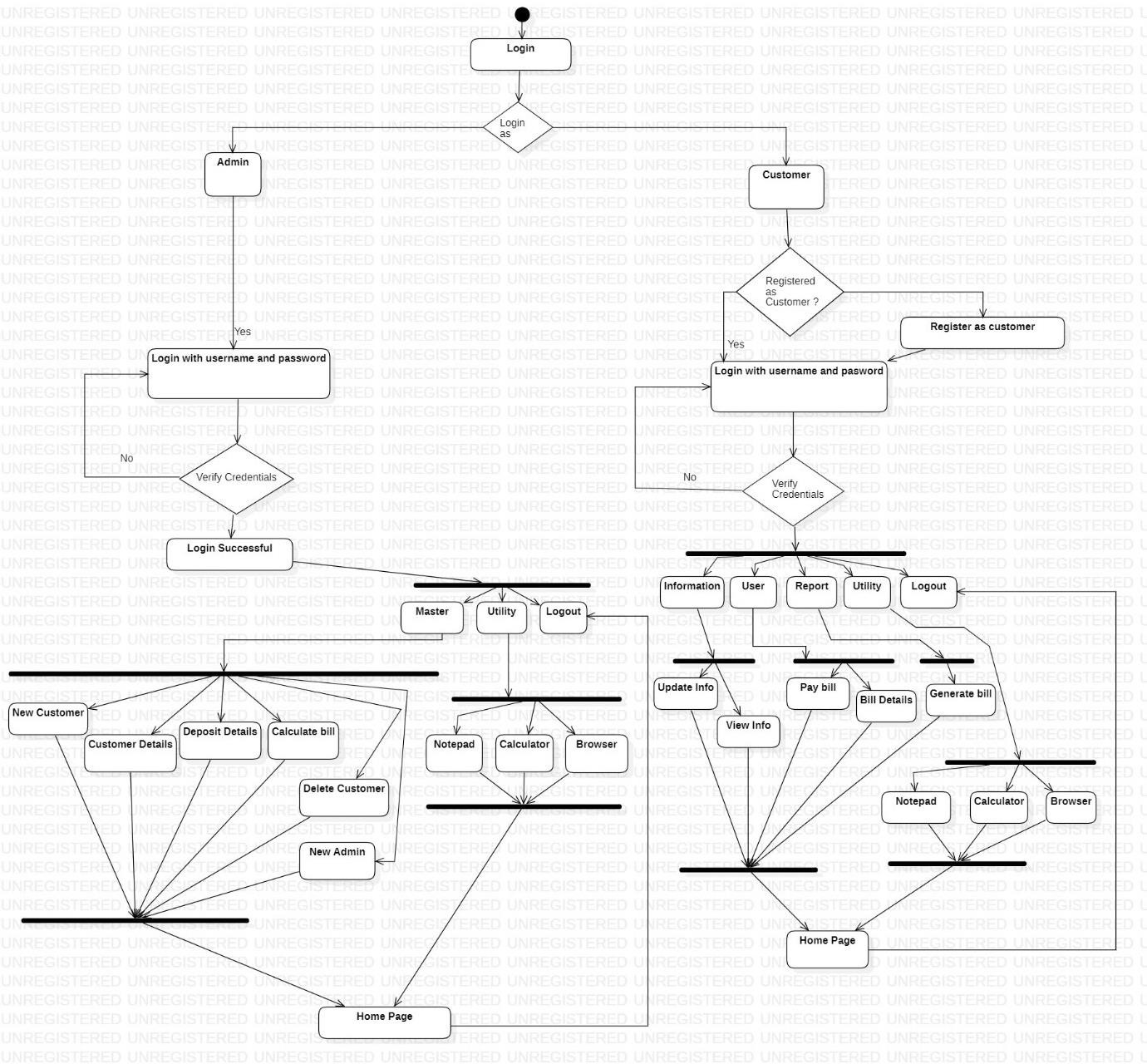
The purpose of the document is to collect and analyze all assorted ideas that have come up to the define the system its requirements with respect to consumers also we shall predict and sought out how we hope this product will be used in order to gain a better understanding of the project outline concept that may be developed later and document ideas that are being considered but may be discarded as the product develops in short the purpose of this SRS document is to provide a detailed overview of a software product its parameters and goals this document describes the project's target audience and its user interface hardware and software requirements it defines how our client team and audience with the product and its functionality nonetheless it helps any designer and developer to assist in software delivery lifecycle processes.

**6.2: Model Use**

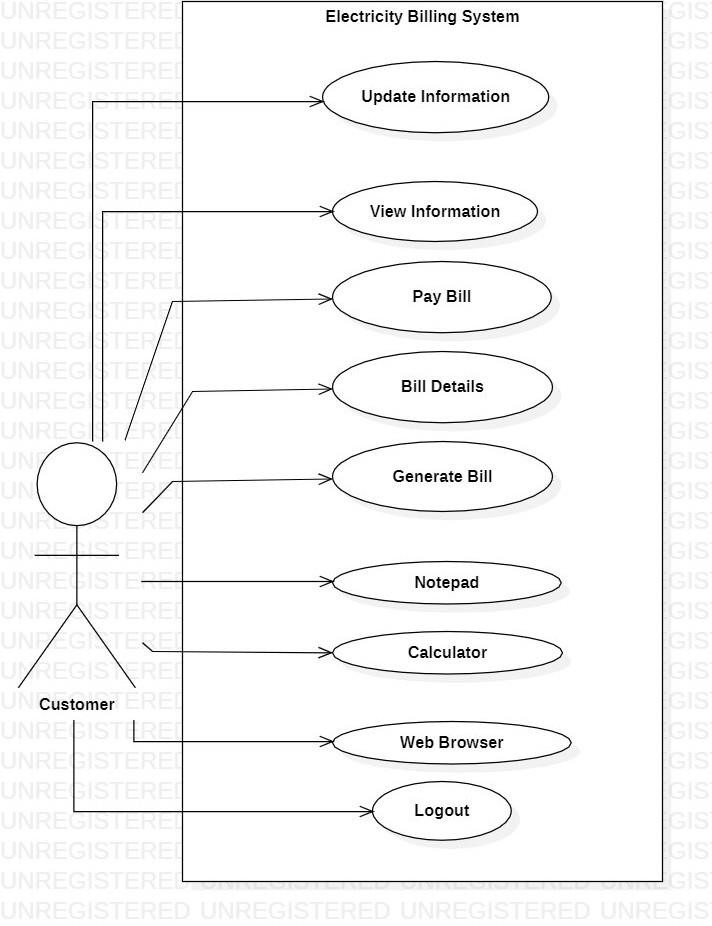
**ER Diagram**

The ER diagram of Electricity Billing System.

It has 5 entities namely login, customer, tax, bill, and meter info. The entities have attributes which are primary , foreign and attributes.

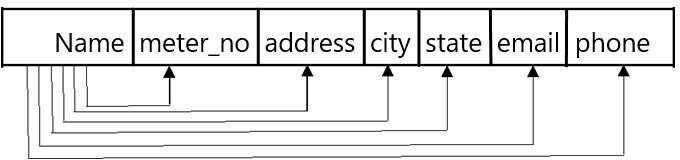


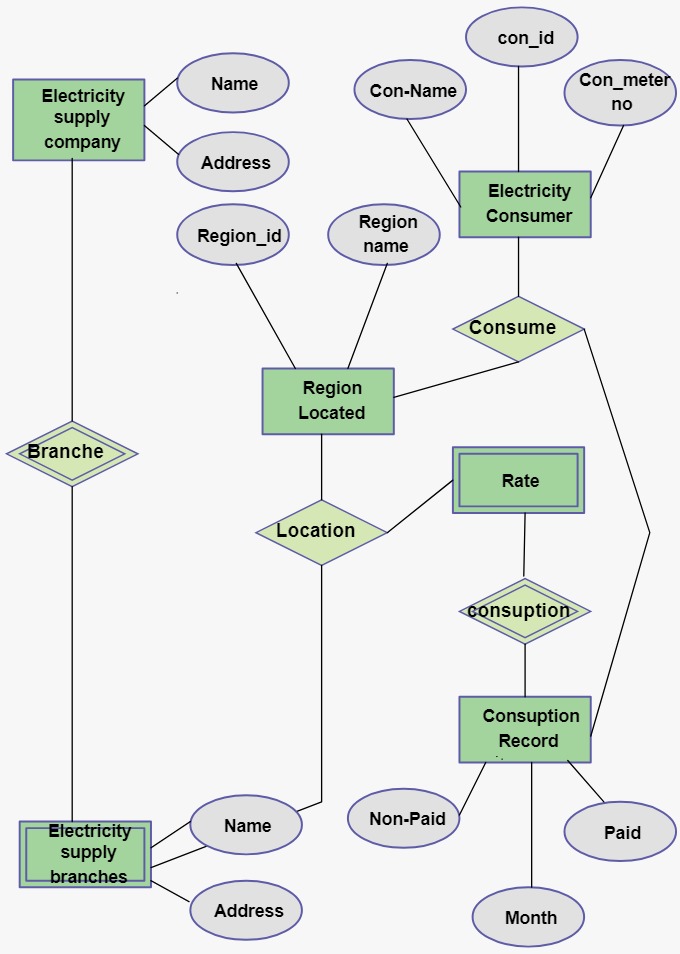
**Case Diagram**



**Activity Diagram**

Customer





**CHAPTER 7**

**FUTURE SCOPE AND LIMITATIONS**

**SOFTWARE SCOPE:**

* **Extensibility**: This software is extendable in ways that its original developers may not expect. The following principles enhances extensibility like hide data structure, avoid traversing multiple

Links or methods avoid case statements on object type and distinguish public and private operations.

* **Reusability**: Reusability is possible as and when require in this application. We can update it next version. Reusable software reduces design, coding and testing cost by amortizing effort Over several designs. Reducing the amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both types of reusability:

Sharing of newly written code within a project and reuse of previously written code on new projects.

* **Understand ability:** A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method, which small and coherent helps to accomplish this.
* **Cost-effectiveness:** Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement.

Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

##### LIMITATIONS:

**This application cannot be accessed remotely.**

* + **This application requires knowledgeable person to use this application.**
  + **This application does not have journals.**

**Source code & Its Execution**

import java.awt.print.PrinterException;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*

\* 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 Sachin Arora

\*/

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")

// <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, 18)); // NOI18N

jLabel1.setText("Electricity Billing Counter");

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");

jButton1.setText("Cal.");

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);

jButton2.setText("Print");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(45, 45, 45)

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

.addGroup(jPanel1Layout.createSequentialGroup()

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

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel4))

.addGap(26, 26, 26)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(txtcID, javax.swing.GroupLayout.DEFAULT\_SIZE, 99, Short.MAX\_VALUE)

.addComponent(txtcName)

.addComponent(txtUnit))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(16, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2)

.addGap(87, 87, 87))))

);

jPanel1Layout.setVerticalGroup(

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

.addGroup(jPanel1Layout.c reateSequentialGroup()

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(47, 47, 47)

.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(38, 38, 38)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(txtcName, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(36, 36, 36)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(txtUnit, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(27, 27, 27)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 176, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(23, 23, 23))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 43, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(14, 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(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1)

.addGap(153, 153, 153))

.addGroup(layout.createSequentialGroup()

.addGap(27, 27, 27)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(13, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(32, 32, 32)

.addComponent(jLabel1)

.addGap(18, 18, 18)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

String cid;

double charge,gramount;

int count;

public void print()

{

String cid = txtcID.getText();

String cname = txtcName.getText();

String unit = txtUnit.getText();

txtPrint.setText(txtPrint.getText() + "\n");

txtPrint.setText(txtPrint.getText() + "Electricity Billing System" + "\n");

txtPrint.setText(txtPrint.getText() + "\n");

txtPrint.setText(txtPrint.getText() + "Customer ID : = "+ cid + "\n");

txtPrint.setText(txtPrint.getText() + "\n");

txtPrint.setText(txtPrint.getText() + "Customer Name : = " + cname +"\n");

txtPrint.setText(txtPrint.getText() + "Units: = " + unit + "\n");

txtPrint.setText(txtPrint.getText() + "Amount: = " + gramount + "\n");

txtPrint.setText(txtPrint.getText() + "Thank you Come Again..!");

}

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

// TODO add your handling code here:

cid = txtcID.getText();

count = Integer.parseInt(txtUnit.getText());

if(count < 500)

{

charge = 1.00;

}

else if(count < 500 && count < 600)

{

charge = 1.80;

}

else if(count < 600 && count < 800)

{

charge = 2.80;

}

else

{

charge = 3.00;

}

gramount = charge \* count;

print();

}

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

try {

// TODO add your handling code here:

txtPrint.print();

} catch (PrinterException ex) {

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

}

}

/\*\*

\* @param args the command line argum ents

\*/

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.JTextArea txtPrint;

private javax.swing.JTextField txtUnit;

private javax.swing.JTextField txtcID;

private javax.swing.JTextField txtcName;

// End of variables declaration

}

String cid;

double charge,gramount;

int count;

public void print()

{

String cid = txtcID.getText();

String cname = txtcName.getText();

String unit = txtUnit.getText();

txtPrint.setText(txtPrint.getText() + "\n");

txtPrint.setText(txtPrint.getText() + "Electricity Billing System" + "\n");

txtPrint.setText(txtPrint.getText() + "\n");

txtPrint.setText(txtPrint.getText() + "Customer ID : = "+ cid + "\n");

txtPrint.setText(txtPrint.getText() + "\n");

txtPrint.setText(txtPrint.getText() + "Customer Name : = " + cname +"\n");

txtPrint.setText(txtPrint.getText() + "Units: = " + unit + "\n");

txtPrint.setText(txtPrint.getText() + "Amount: = " + gramount + "\n");

txtPrint.setText(txtPrint.getText() + "Thank you Come Again..!");

        }

cid = txtcID.getText();

count = Integer.parseInt(txtUnit.getText());

if(count < 500)

{

charge = 1.00;

}

else if(count < 500 && count < 600)

{

charge = 1.80;

}

else if(count < 600 && count < 800)

{

charge = 2.80;

}

else

{

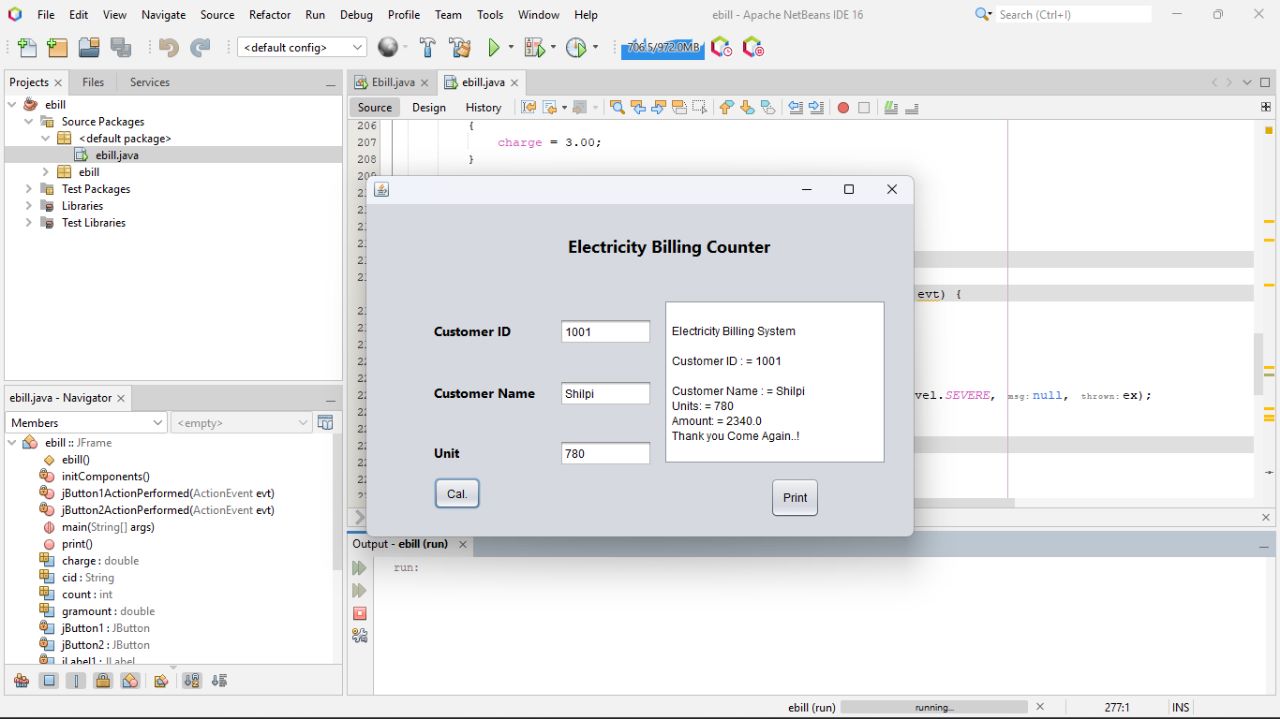
charge = 3.00;

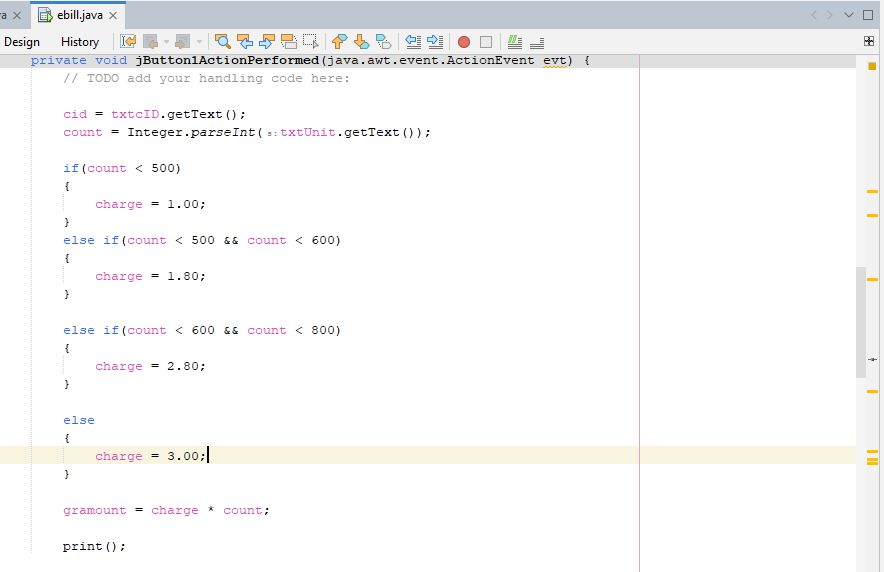
}

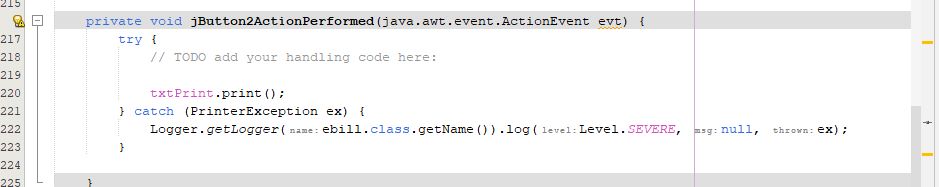
gramount = charge \* count;

        print();

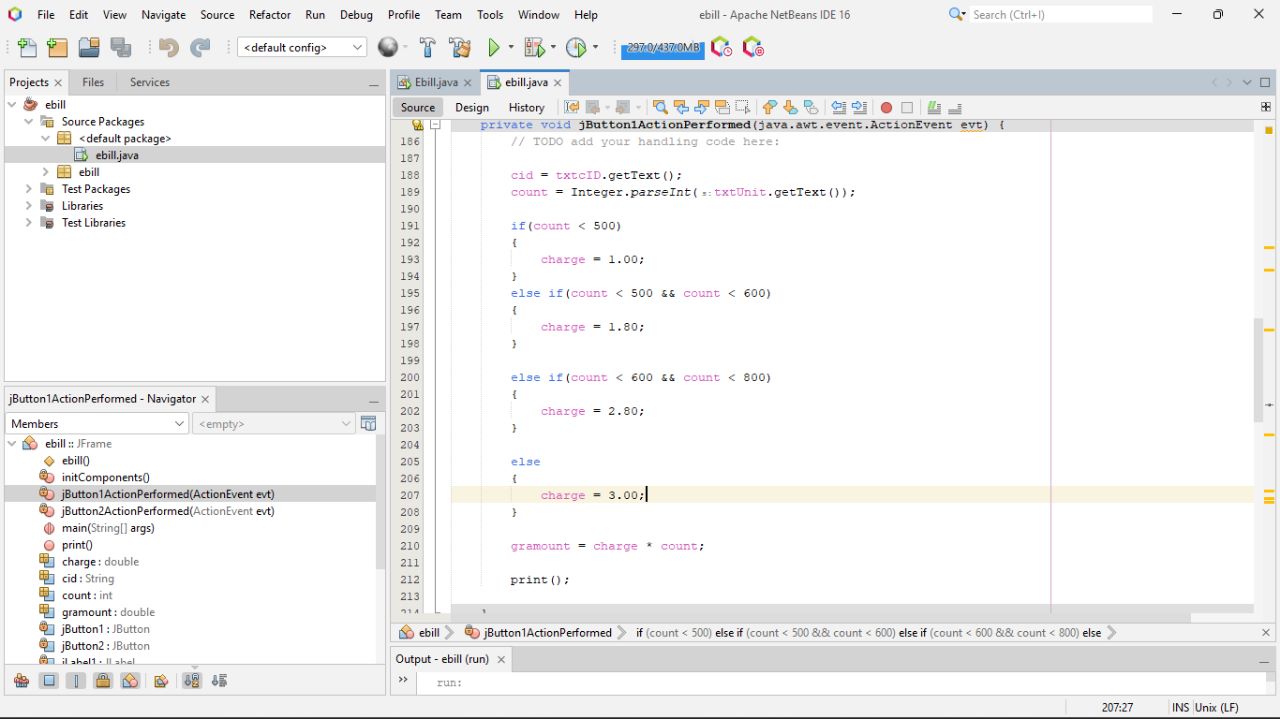
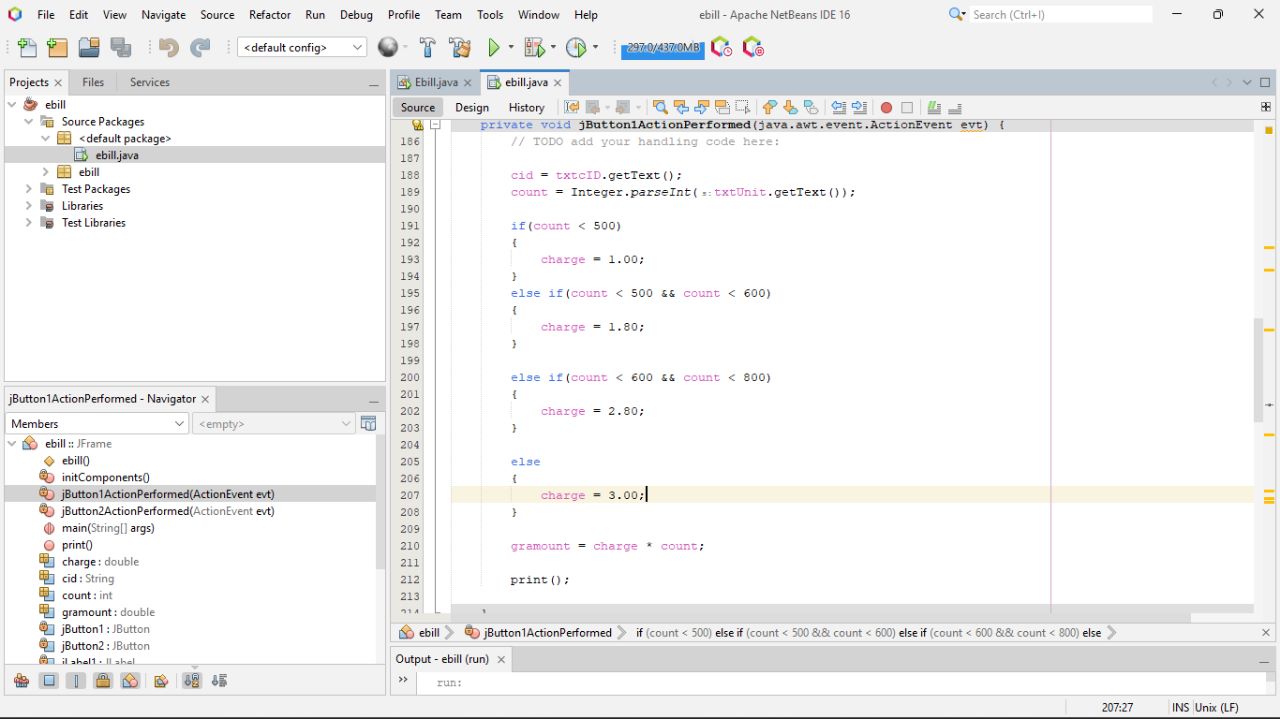
**Execution Snapshots**

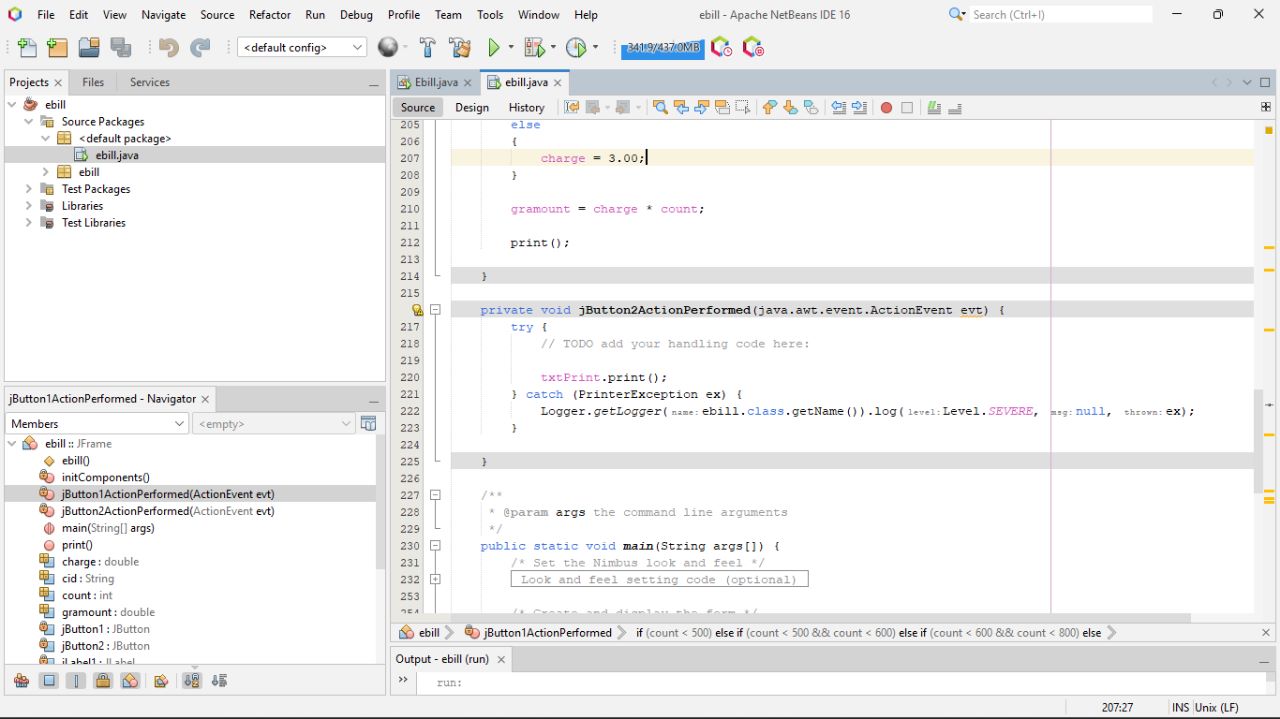


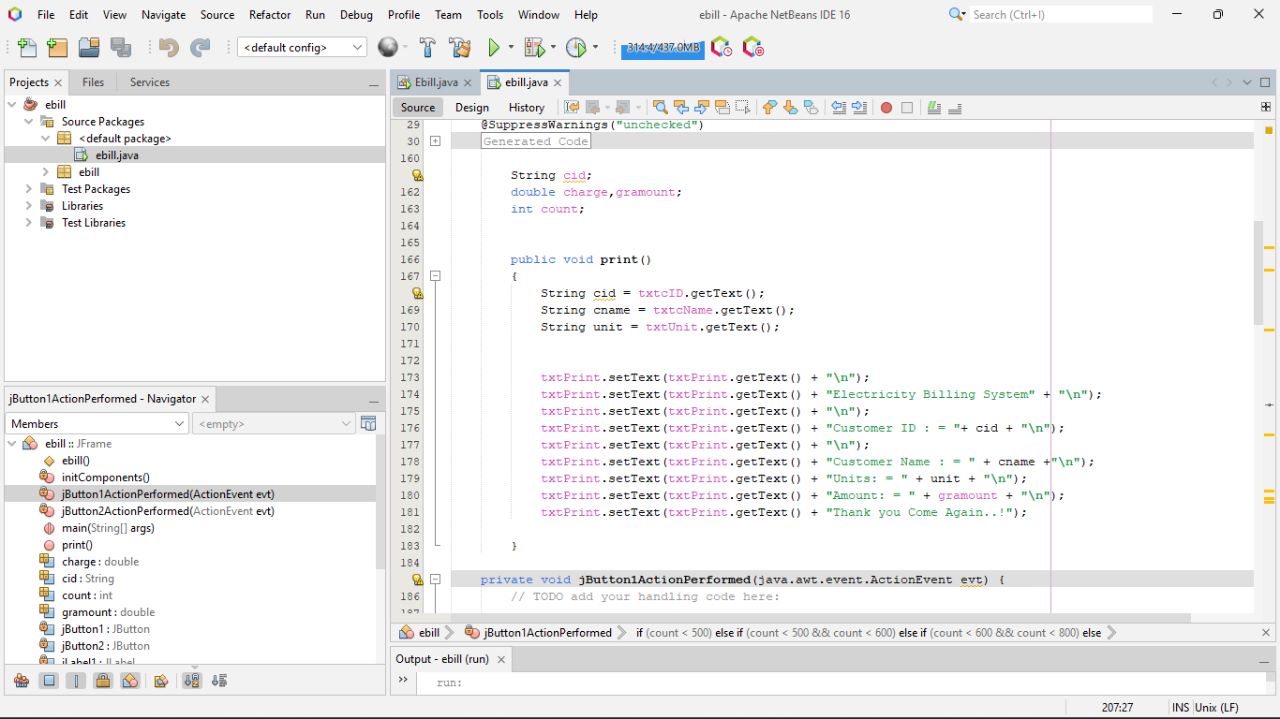










**Conclusion**

After all the hard work is done for electricity bill management system is here. It is a software which helps the user to work with the billing cycles, paying bills, managing different DETAILS under which are working etc.

This software reduces the amount of manual data entry and gives greater efficiency. The User Interface of it is very friendly and can be easily used by anyone.

It also decreases the amount of time taken to write details and other modules.

# BIBLIOGRAPHY

##### REFERENCES

##### Websites

* + - <https://www.youtube.com/watch?v=iWitVuW2D1o&t=4s>
    - [http://www.github.com](http://www.github.com/)
    - [www.google.com](http://www.google.com/)
    - <http://www.javatpoint.com/>