



ELECTRICITY BILLING SYSTEM



A PROJECT REPORT

Submitted by

SUDHAKARAN M (2303811724321111)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by
AICTE, New Delhi)

SAMAYAPURAM – 621 112

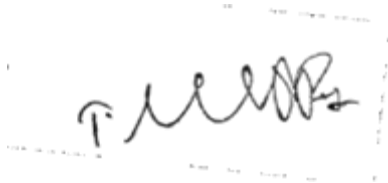
DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “ **ELECTRICITY BILLING SYSTEM**” is the bonafide work of **SUDHAKARAN M (2303811724321111)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

Dr. T. AVUDAIAPPAN M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,

K. Ramakrishnan College of Engineering,

Samayapuram, Trichy -621 112.



Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24



INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “ **ELECTRICITY BILLING SYSTEM**” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING**.



Signature

SUDHAKARAN M

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”**, for providing us with the opportunity to do this project.

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.**, Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E.**, Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The TNEB Electricity Billing System is a Java-based application designed to manage electricity billing and payment processes for consumers. Using the Swing framework, the application provides a user-friendly graphical interface that allows easy addition of consumer details, including name, meter number, contact number, and address, with validation to ensure the contact number is exactly 10 digits. The system enables updating of meter readings, ensuring that new readings cannot be lower than previous ones. It calculates the electricity bill based on consumed units using tiered pricing, with the total bill formatted to two decimal places for precision. The system also tracks the payment status of bills, allowing bills to be marked as paid or pending. The application features a clean and intuitive design with well-styled buttons and text areas, offering an efficient way to manage consumer accounts and billing processes. The TNEB Electricity Billing System is an ideal solution for utility companies like TNEB, helping them streamline their operations while ensuring accurate and user-friendly management of electricity billing and payment.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	VIII
1	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 OBJECTIVE	1
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 BLOCK DIAGRAM	4
3	JAVA PROGRAMMING CONCEPTS	5
	3.1 CLASSES AND OBJECT	5
	3.2 METHODS	5
	3.3 ENCAPSULATION	5
	3.4 INHERITANCE	5
	3.5 POLYMORPHISM	5
	3.6 ABSTRACTION	5
	3.7 CONTROL STRUCTURES	5
4	MODULE DESCRIPTION	6
	4.1 CUSTOMER MANAGEMENT MODULE	6
	4.2 METER READING MODULE	6
	4.3 BILLING MODULE	6
	4.4 PAYMENT TRACKING MODULE	6
	4.5 USER INTERFACE (GUI) MODULE	6
5	CONCLUSION	7
	REFERENCES	8
	APPENDICES	9
	APPENDIX A – SOURCE CODE	9
	APPENDIX B – SCREEN SHOTS	19

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The TNEB Electricity Billing System is a Java-based application designed to simplify electricity billing and payment management. Using the Swing framework, it provides an easy-to-use interface for managing consumer details, meter readings, and billing processes. The system allows utility companies like TNEB to efficiently add new customers, update meter readings, calculate bills based on consumption, and track payment statuses. With built-in validation for phone numbers and meter readings, as well as a tiered pricing model for bill calculation, this application offers an accurate and streamlined solution for electricity billing.

1.2 OBJECTIVE

The objective of the TNEB Electricity Billing System is to provide a streamlined, efficient, and accurate solution for managing electricity billing and payment processes. The key objectives are:

- **Consumer Management:** To facilitate the addition and management of consumer information, including personal details, meter numbers, and contact information, with proper validation.
- **Meter Reading Updates:** To allow users to update meter readings while ensuring that new readings are valid and greater than or equal to previous readings.
- **Bill Generation:** To calculate electricity bills based on consumption, applying a tiered pricing structure, and displaying the bill with precision to two decimal

places.

- **Payment Tracking:** To enable the tracking of bill payment status, allowing users to mark bills as paid or pending.
- **User-Friendly Interface:** To create an intuitive, visually appealing interface that makes managing and accessing billing data easy for both administrators and consumers.

This system aims to improve the operational efficiency of electricity billing and provide accurate, transparent billing information to both utility companies and consumers.

CHAPTER 2

PROJECT METHODOLOGY

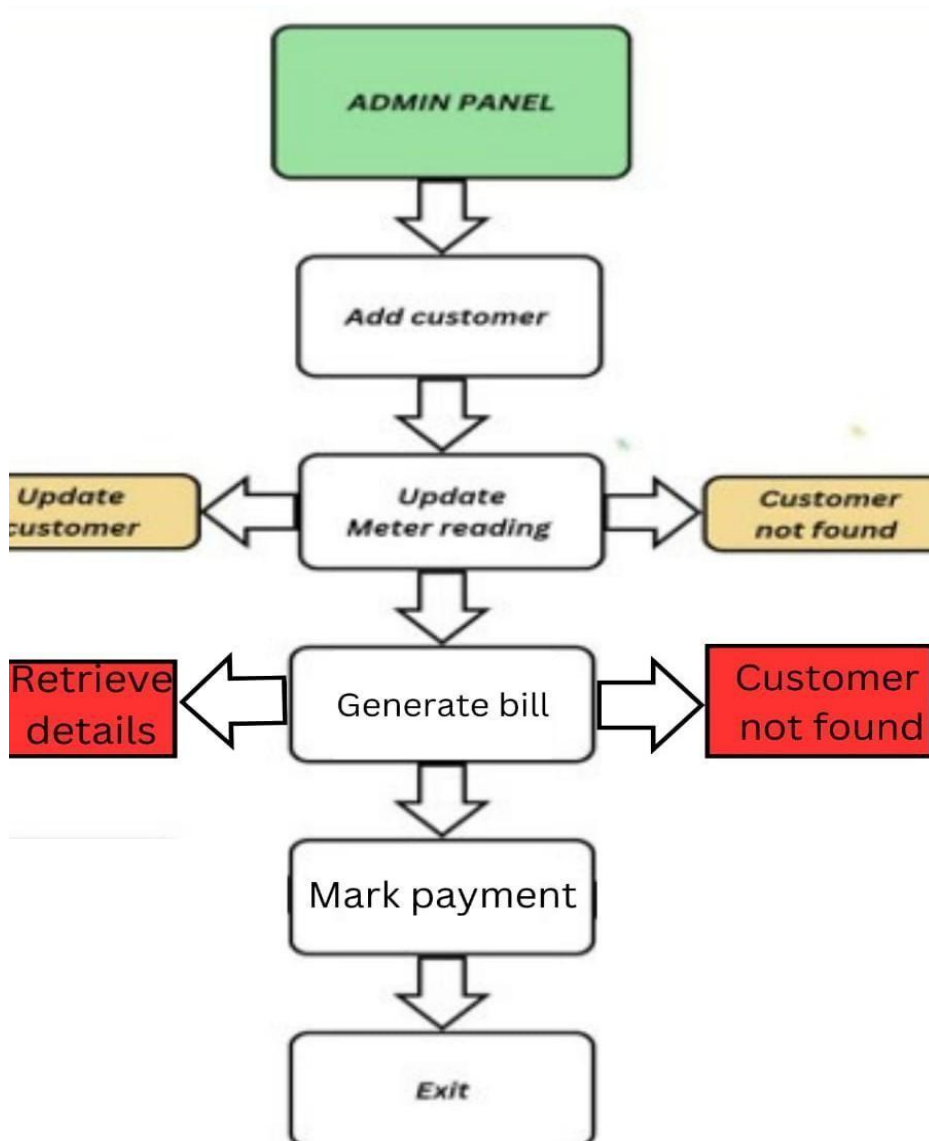
2.1 PROPOSED WORK

The TNEB Electricity Billing System will focus on the following key features:

- **Consumer Registration:** Allow users to add consumer details such as name, meter number, contact number (validated to 10 digits), and address.
- **Meter Reading Management:** Enable updating of meter readings, ensuring the new reading is not less than the previous one.
- **Bill Generation:** Automatically calculate electricity bills using a tiered pricing structure and display the bill with two decimal precision.
- **Payment Tracking:** Track and update the payment status of each bill, marking them as "Paid" or "Pending."
- **User Interface:** Create an intuitive GUI for easy navigation and interaction.

This system aims to simplify and automate the billing process for utility companies, ensuring accuracy and efficiency.

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

□ 3.1 Classes and Objects:

- Customer and AdminPanelGUI are classes. Customer stores consumer details, while AdminPanelGUI manages the user interface.

□ 3.2 Methods:

- Methods like addCustomer(), updateMeterReading(), and generateBill() define the logic for adding customers, updating readings, and generating bills.

□ 3.3 Encapsulation:

- Fields in Customer are private, with public methods to access or modify them, ensuring data protection.

□ 3.4 Inheritance:

- AdminPanelGUI extends JFrame, inheriting methods to manage the window and layout.

□ 3.5 Polymorphism:

- Used in the handling of different buttons, all responding to addActionListener() with different actions.

□ 3.6 Abstraction:

- Methods like calculateBill() hide the complex logic behind simple interfaces for the user.

□ 3.7 Control Structures:

- Conditional statements (if, else) and exception handling (try-catch) manage input validation and bill calculation.

CHAPTER 4

MODULE DESCRIPTION

3.1 Customer Management Module

Adds and manages customer details (name, meter number, phone,address).

3.2 Meter Reading Module

Manages meter readings, ensuring the current reading is valid and calculates consumed units.

3.3 Billing Module

Calculates electricity bills based on consumption and tiered rates.

3.4 Payment Tracking Module

Tracks and updates bill payment status (Paid or Pending).

3.5 User Interface (GUI) Module

Provides a graphical interface for interacting with the system (add customers, update readings, generate bills).

CHAPTER 5

CONCLUSION

The TNEB Electricity Billing System efficiently manages customer details, meter readings, bill calculations, and payment tracking through a structured and user- friendly interface. By utilizing key Java concepts such as classes, methods, encapsulation, and exception handling, the system ensures smooth operation, data protection, and accurate billing. This modular approach enhances scalability and maintainability, making it an effective solution for electricity billing management. The system provides administrators with an intuitive interface for adding customers, updating readings, and generating bills, ensuring streamlined operations and user satisfaction.

REFERENCES:

- Oracle Java Documentation: <https://docs.oracle.com/en/java/>
- "Java Swing Tutorial" by Tutorialspoint:
https://www.tutorialspoint.com/java/java_gui.htm
- "Mastering Java" by Ravi Kant Soni
- Java Programming Concepts - Event Handling:
<https://www.geeksforgeeks.org/event-handling-in-java/>
- "Effective Java" by Joshua Bloch

APPENDICES

APPENDIX A – SOURCE CODE

```
import java.awt.*;
import java.util.HashMap;
import javax.swing.*;

class Customer
{
    String name;
    int meterNumber;
    String contactNumber;
    String address;
    double previousReading;
    double currentReading;
    boolean paymentComplete;

    public Customer(String name, int meterNumber, String contactNumber,
String address) {
        this.name = name;
        this.meterNumber = meterNumber;
        this.contactNumber = contactNumber;
        this.address = address;
        this.previousReading = 0;
        this.currentReading = 0;
        this.paymentComplete = false;
    }

    public double getConsumedUnits() {
```

```
        return currentReading >= previousReading ? currentReading -  
previousReading : currentReading;  
    }
```

```
public void updateMeterReading(double newReading)  
    {this.previousReading = this.currentReading;  
    this.currentReading = newReading;  
    }
```

```
public double calculateBill() {  
    double consumedUnits = getConsumedUnits();  
    double ratePerUnit;  
  
    if (consumedUnits <= 100)  
        {ratePerUnit = 10;  
    } else if (consumedUnits <= 500)  
        {ratePerUnit = 50;  
    } else {  
        ratePerUnit = 100;  
    }  
  
    return consumedUnits * ratePerUnit;  
}
```

```
public void markPaymentComplete()  
    {this.paymentComplete = true;  
    }
```

```
public boolean isPaymentComplete() {
```

```

        return this.paymentComplete;
    }
}

```

```

class TNEBBillingGUI extends JFrame {
    private HashMap<Integer, Customer> customers;
    private JTextArea outputArea;

    public TNEBBillingGUI()
    {
        customers = new HashMap<>();

        setTitle(" TNEB Electricity Billing System ");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        setLocationRelativeTo(null); // Center the window
        setResizable(false);

        JPanel buttonPanel = new JPanel(new GridLayout(5, 1, 10, 10));
        buttonPanel.setBackground(new Color(34, 45, 65));

        outputArea = new JTextArea();
        outputArea.setFont(new Font("Monospaced", Font.BOLD, 14));
        outputArea.setEditable(false);
        outputArea.setBackground(new Color(245, 245, 245));
        JScrollPane scrollPane = new JScrollPane(outputArea);

        JButton addCustomerButton = createStyledButton("+ Add Consumer",

```

```

new Color(72, 201, 176));

    JButton updateReadingButton = createStyledButton("⏮ Update
Reading", new Color(241, 196, 15));

    JButton generateBillButton = createStyledButton("➤ Generate Bill",
new Color(231, 76, 60));

    JButton markPaymentButton = createStyledButton("➡ Mark Payment",
new Color(46, 204, 113));

    JButton exitButton = createStyledButton("+ Exit", new Color(46, 134,
222));


    buttonPanel.add(addCustomerButton);
    buttonPanel.add(updateReadingButton);
    buttonPanel.add(generateBillButton);
    buttonPanel.add(markPaymentButton);
    buttonPanel.add(exitButton);


    addCustomerButton.addActionListener(e -> addCustomer());
    updateReadingButton.addActionListener(e -> updateMeterReading());
    generateBillButton.addActionListener(e -> generateBill());
    markPaymentButton.addActionListener(e -> markPaymentComplete());
    exitButton.addActionListener(e -> System.exit(0));


    JLabel title = new JLabel(" ⚡ TNEB Electricity Billing System ⚡",
JLabel.CENTER);

    title.setFont(new Font("Serif", Font.BOLD, 24));
    title.setOpaque(true);
    title.setBackground(new Color(123, 36, 28));
    title.setForeground(Color.WHITE);

```

```

        title.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        add(title, BorderLayout.NORTH);
        add(buttonPanel, BorderLayout.WEST);
        add(scrollPane, BorderLayout.CENTER);
    }

    private JButton createStyledButton(String text, Color backgroundColor)
    {
        JButton button = new JButton(text);
        button.setFont(new Font("SansSerif", Font.BOLD, 16));
        button.setForeground(Color.WHITE);
        button.setBackground(backgroundColor);
        button.setFocusPainted(false);
        button.setBorder(BorderFactory.createCompoundBorder(
            BorderFactory.createLineBorder(Color.DARK_GRAY, 2),
            BorderFactory.createEmptyBorder(5, 15, 5, 15)
        ));
        button.setRolloverEnabled(true);
        button.addMouseListener(new java.awt.event.MouseAdapter()
        {
            public void mouseEntered(java.awt.event.MouseEvent evt)
            {
                button.setBackground(button.getBackground().darker());
            }

            public void mouseExited(java.awt.event.MouseEvent evt)
            {
                button.setBackground(backgroundColor);
            }
        });
        return button;
    }
}

```

```

private void addCustomer() {
    String name = JOptionPane.showInputDialog(this, "Enter Consumer
Name:");
    if (name == null || name.isEmpty())
        { showMessage("+ Name cannot be
empty!");return;
    }

    String meterInput = JOptionPane.showInputDialog(this, "Enter Meter
Number:");
    try {
        int meterNumber = Integer.parseInt(meterInput);
        if (customers.containsKey(meterNumber)) {
            showMessage("+ Meter number already exists!");
            return;
        }

        String contactNumber = JOptionPane.showInputDialog(this, "Enter
Contact Number:");
        if (contactNumber == null || !contactNumber.matches("\\d{10}"))
            { showMessage("+ Contact number must be exactly 10
digits!");return;
        }


        String address = JOptionPane.showInputDialog(this, "Enter
Address:");
        if (address == null || address.isEmpty()) {

```

```

        showMessage("+ Address cannot be empty!");
        return;
    }

    customers.put(meterNumber, new Customer(name, meterNumber,
contactNumber, address));

    showMessage("  Consumer added successfully:\nName: " + name
+ "\nMeter Number: " + meterNumber);
    } catch (NumberFormatException e)
    {
        showMessage("+ Invalid meter number!");
    }
}

private void updateMeterReading() {
    String meterInput = JOptionPane.showInputDialog(this, "Enter Meter
Number:");
    try {
        int meterNumber = Integer.parseInt(meterInput);
        if (!customers.containsKey(meterNumber)) {
            showMessage("+ Consumer not found!");
            return;
        }


        String readingInput = JOptionPane.showInputDialog(this, "Enter
New Meter Reading:");
        double newReading = Double.parseDouble(readingInput);
        Customer customer = customers.get(meterNumber);
        if (newReading < customer.previousReading) {

```



```
        showMessage("+ New reading cannot be less than the previous  
reading!");
```

```
    }
```

```
        customer.updateMeterReading(newReading);  
        showMessage("  Meter reading updated for meter number: " +  
meterNumber);
```

```
    } catch (NumberFormatException e)  
    {  
        showMessage("+ Invalid  
input!");  
    }  
}
```

```
private void generateBill() {  
    String meterInput = JOptionPane.showInputDialog(this, "Enter Meter  
Number:");
```

```
    try {  
        int meterNumber = Integer.parseInt(meterInput);  
        if (!customers.containsKey(meterNumber)) {  
            showMessage("+ Consumer not found!");  
            return;  
        }  
    }
```

```
    Customer customer = customers.get(meterNumber);  
    double consumedUnits = customer.getConsumedUnits();  
    double totalBill = customer.calculateBill();
```

```
    String formattedBill = String.format("%.2f", totalBill);
```

```

        outputArea.append("\n TNEB Electricity Bill \n");
        outputArea.append("Consumer Name: " + customer.name + "\n");
        outputArea.append("Meter Number: " + customer.meterNumber +
"\n");
        outputArea.append("Contact Number: " + customer.contactNumber +
"\n");
        outputArea.append("Address: " + customer.address + "\n");
        outputArea.append("Previous Reading: " +
customer.previousReading + " kWh\n");
        outputArea.append("Current Reading: " + customer.currentReading +
" kWh\n");
        outputArea.append("Consumed Units: " + consumedUnits + "
kWh\n");
        outputArea.append("Total Bill: ₹" + formattedBill + "\n");
        outputArea.append("Payment Status: " +
(customer.isPaymentComplete() ? " Paid" : " Pending")
+" \n"); outputArea.append("----- \n");
    } catch (NumberFormatException e)
    {
        showMessage(" Invalid
input!");
    }
}

```


```

private void markPaymentComplete() {
    String meterInput = JOptionPane.showInputDialog(this, "Enter Meter
Number:");
    try {
        int meterNumber = Integer.parseInt(meterInput);
        if (!customers.containsKey(meterNumber)) {

```

```

        showMessage("+ Consumer not found!");
        return;
    }

    Customer customer = customers.get(meterNumber);
    customer.markPaymentComplete();
    showMessage("  Payment marked as complete for meter number: "
+ meterNumber);
    } catch (NumberFormatException e)
    {
        showMessage("+ Invalid
        input!");
    }
}

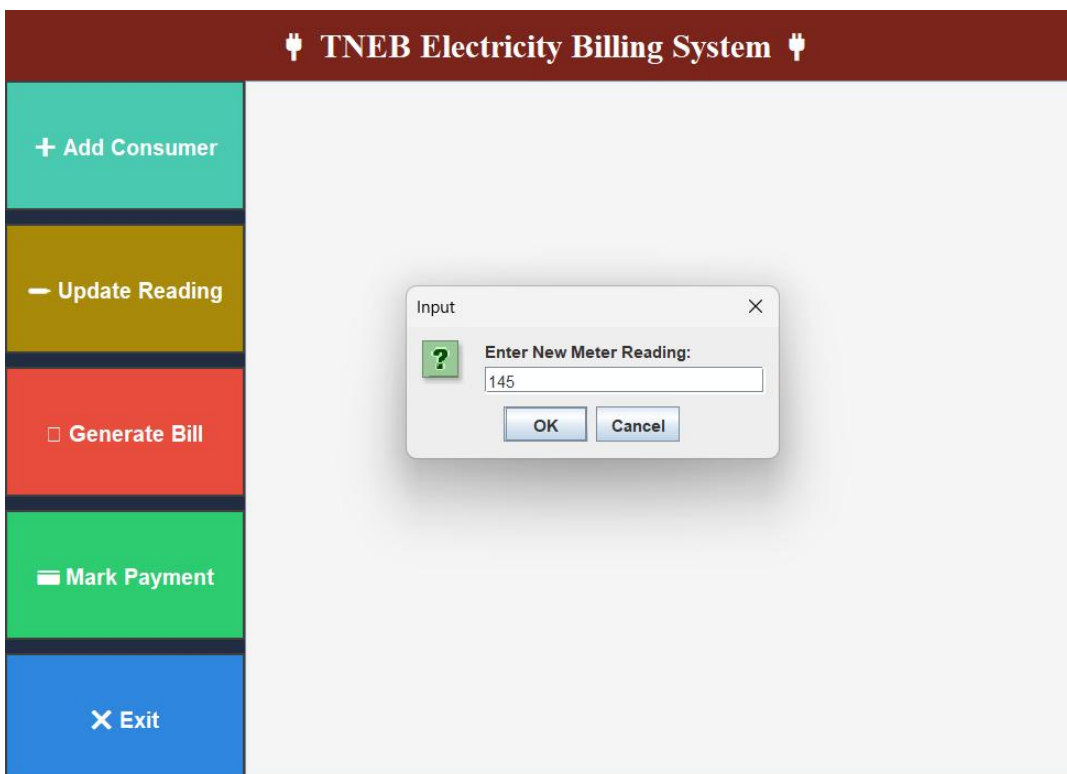
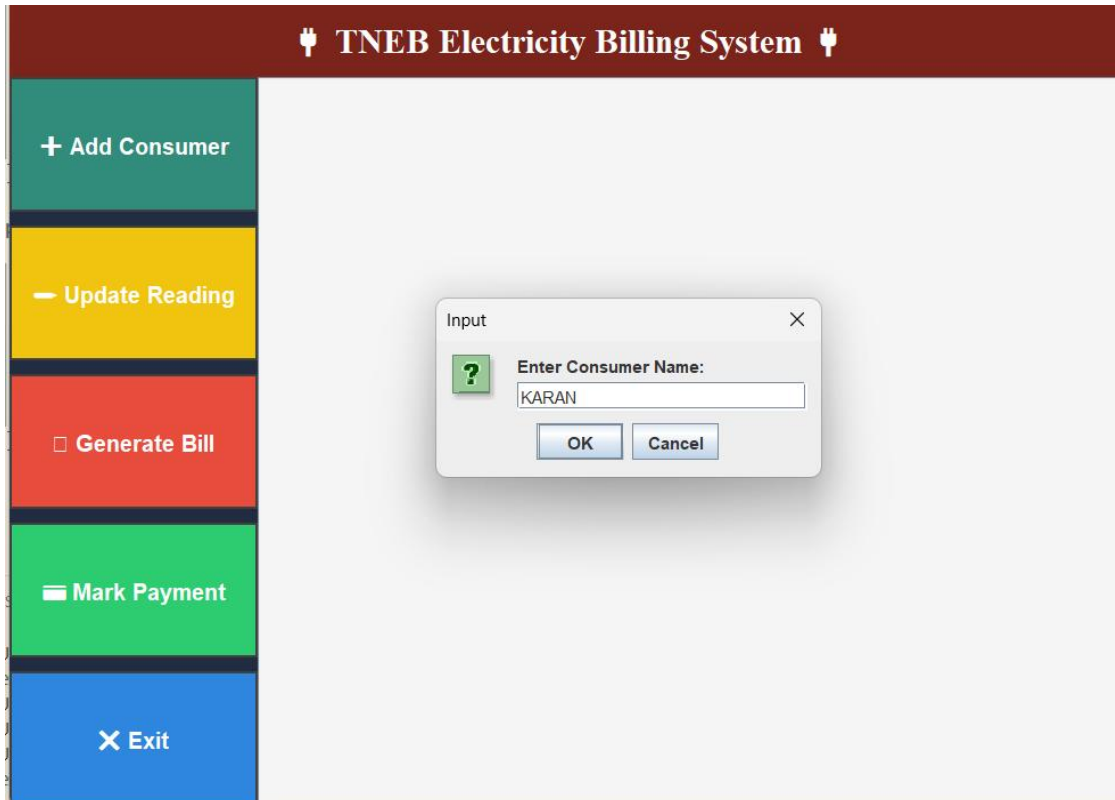
private void showMessage(String message)
{
    JOptionPane.showMessageDialog(this, message);
}

}

public class TNEB {
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() -> {
            TNEBBillingGUI gui = new TNEBBillingGUI();
            gui.setVisible(true);
        });
    }
}

```

APPENDIX B - SCREENSHOTS



⚡ TNEB Electricity Billing System ⚡	
+ Add Consumer	⚡ TNEB Electricity Bill ⚡ Consumer Name: KARAN Meter Number: 7 Contact Number: 1234567890 Address: TRICHY
— Update Reading	Previous Reading: 0.0 kWh Current Reading: 145.0 kWh Consumed Units: 145.0 kWh Total Bill: ₹7250.00 Payment Status: ✗ Pending
□ Generate Bill	-----
☑ Mark Payment	
✗ Exit	

⚡ TNEB Electricity Billing System ⚡	
+ Add Consumer	⚡ TNEB Electricity Bill ⚡ Consumer Name: KARAN Meter Number: 7 Contact Number: 1234567890 Address: TRICHY
— Update Reading	Previous Reading: 0.0 kWh Current Reading: 145.0 kWh Consumed Units: 145.0 kWh Total Bill: ₹7250.00 Payment Status: ✗ Pending
□ Generate Bill	⚡ TNEB Electricity Bill ⚡ Consumer Name: KARAN Meter Number: 7 Contact Number: 1234567890 Address: TRICHY
☑ Mark Payment	Previous Reading: 0.0 kWh Current Reading: 145.0 kWh Consumed Units: 145.0 kWh Total Bill: ₹7250.00 Payment Status: ✔ Paid
✗ Exit	-----