



University of Information Technology and Sciences
Department of CSE

Project Report
of
Electricity Billing System

By Team D3F4UL7

Project Title: Electricity Billing System

Course: Computer Science and Engineering

Instructor: **Propa Punam**
Lecturer
Department of CSE (UITs)

Author:

1. Md Raiyan Rahim Fahad
Project Leader & Lead Designer
ID: 0432410005101036

2. Md. Siam Hosen
System Strategist & Communication Lead
ID: 0432410005101034

3. Asma Akter Akhi
Data Specialist & Logic Analyst
ID: 0432410005101043

4. Seazan Ahmed Khan
Operation & Management Lead
ID: 0432410005101038

Submission Date: 6 December 2024

Abstract

The Electricity Billing System is a Java-based application designed to manage customer electricity billing efficiently. The system includes functionalities such as user authentication (login and signup), bill calculation, and bill generation. It uses Java Swing for the graphical user interface (GUI) and MySQL for the backend database. This system enables accurate and automated calculation of electricity bills, integrating features such as dynamic user input, customizable bill generation, and printable invoices.

Objectives

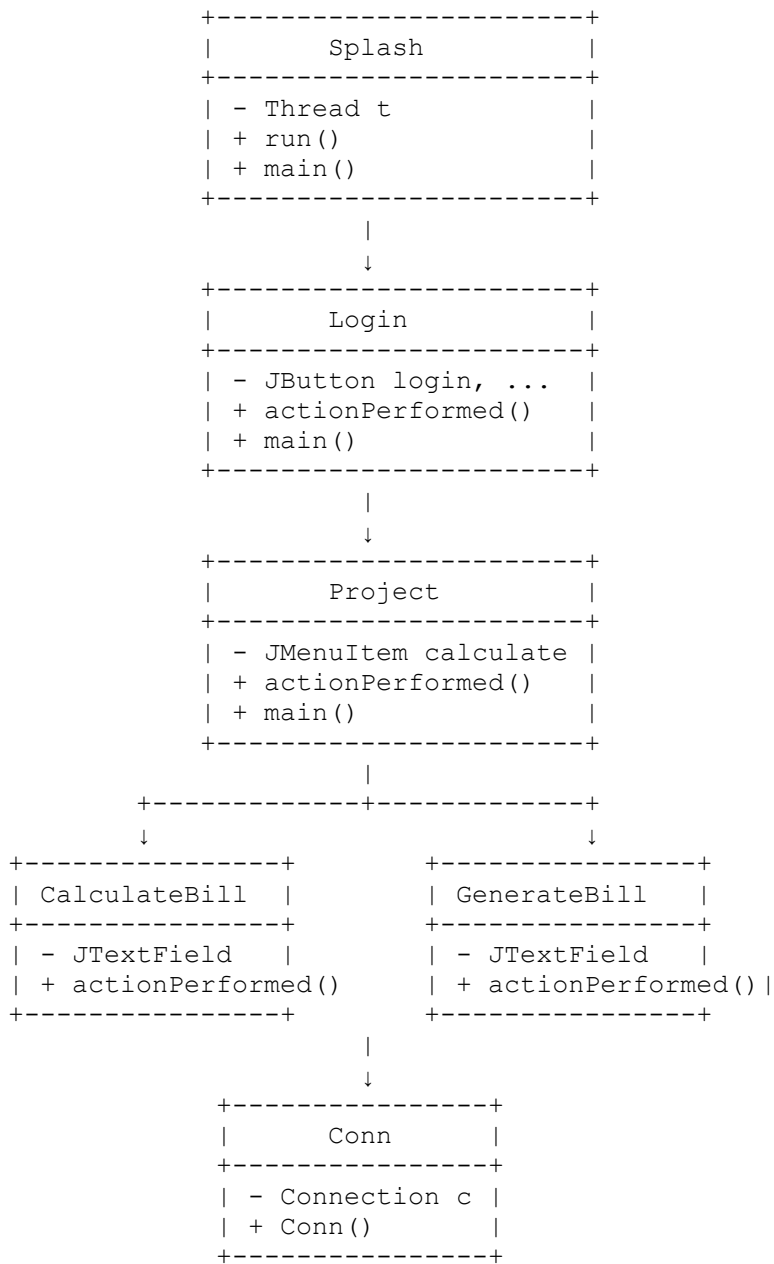
1. To create a user-friendly interface for managing electricity billing.
 2. To implement secure user authentication (login and signup).
 3. To calculate electricity bills dynamically based on consumption units.
 4. To generate and print detailed billing invoices.
 5. To connect the application with a MySQL database for data storage and retrieval.
-

Equipment and Components

- **Hardware:**
 - Personal Computer (PC)
 - Internet Connection (for database access)
 - **Software:**
 - Java Development Kit (JDK)
 - NetBeans or Eclipse IDE
 - MySQL Database
 - Java Swing Library
 - JDBC Driver for MySQL
-

Class Diagram

Here is the Class Diagram Illustration



Relationships:

- **Inheritance:** There is the concept of inheritance implemented.
- **Dependencies:**
 - Splash depends on Login.
 - Login depends on Project, Signup, and Conn.
 - Project depends on CalculateBill and GenerateBill.
 - CalculateBill and GenerateBill depend on Conn for database interaction.

Theory

The Electricity Billing System is a Java desktop application utilizing the Model-View-Controller (MVC) pattern. It leverages:

- **Java Swing:** For building the GUI components.
- **JDBC (Java Database Connectivity):** For communicating with the MySQL database.
- **Threading:** To introduce a splash screen delay.
- **Event Handling:** For capturing user actions like button clicks.
- **SQL Queries:** For managing user authentication and bill records.

Methodology

1. **Design GUI Interfaces:**
 - Developed Splash, Login, Signup, Project, CalculateBill, and GenerateBill classes using Java Swing.
 2. **Database Connectivity:**
 - Implemented the Conn class to connect to the MySQL database.
 3. **Authentication System:**
 - Created login and signup interfaces for user management.
 4. **Bill Calculation Logic:**
 - Used conditional statements to calculate the bill based on units consumed.
 5. **Invoice Generation:**
 - Designed printable invoices for billing details.
-

Code

Here is the source code for this programme:

Splash Class

```
public class Splash extends JFrame implements Runnable {
    Thread t;
    Splash() {
        ImageIcon i1 = new
ImageIcon(ClassLoader.getResource("icon/elec.jpg"));
        JLabel image = new JLabel(i1);
        add(image);
        setSize(1066, 600);
        setLocation(200, 100);
        setVisible(true);
        t = new Thread(this);
        t.start();
    }
    public void run() {
        try {
            Thread.sleep(3000);
            setVisible(false);
            new Login();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        new Splash();
    }
}
```

Login Class

```
public class Login extends JFrame implements ActionListener {
    JButton login, cancel, signup;
    JTextField namefield, passfield;
    Login() {
        super("Login Page");
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
        // Labels and fields for username and password
    }
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == login) {
            String susername = namefield.getText();
            String spassword = passfield.getText();
            try {
                Conn c = new Conn();
            }
        }
    }
}
```

```

        String query = "select * from login1 where username =
        '"+susername+"' and password = '"+spassword+"'";
        ResultSet rs = c.s.executeQuery(query);
        if(rs.next()){
            setVisible(false);
            new Project();
        }
    } catch(Exception e){
        e.printStackTrace();
    }
}
}

public static void main(String[] args){
    new Login();
}
}

```

Signup Class

```

public class Signup extends JFrame implements ActionListener {

    JButton create, cancel;
    Choice accountType;
    JTextField id, user, nm, pass;

    Signup() {
        setSize(700, 400);
        setLocation(450, 250);
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JPanel panel = new JPanel();
        panel.setBounds(30, 30, 650, 300);
        panel.setBorder(new TitledBorder(
            new LineBorder(new Color(100, 216, 230), 3),
            "Create Account",
            TitledBorder.LEADING,
            TitledBorder.TOP,
            null,
            new Color(100, 216, 230)
        ));
        panel.setLayout(null);
        panel.setBackground(Color.WHITE);
        add(panel);

        // Labels

        JLabel meterNo = new JLabel("ID Number :");
        meterNo.setBounds(130, 50, 140, 25);
        meterNo.setForeground(Color.BLACK);
        meterNo.setFont(new Font("Tahoma", Font.BOLD, 14));
        panel.add(meterNo);

        JLabel username = new JLabel("Username :");
        username.setBounds(130, 90, 140, 25);
        username.setForeground(Color.BLACK);
        username.setFont(new Font("Tahoma", Font.BOLD, 14));
        panel.add(username);

        JLabel name = new JLabel("Name :");
    }
}

```

```

name.setBounds(130, 135, 140, 25);
name.setForeground(Color.BLACK);
name.setFont(new Font("Tahoma", Font.BOLD, 14));
panel.add(name);

JLabel password = new JLabel("Password :");
password.setBounds(130, 180, 140, 25);
password.setForeground(Color.BLACK);
password.setFont(new Font("Tahoma", Font.BOLD, 14));
panel.add(password);

// Text Fields

id = new JTextField();
id.setBounds(330, 50, 200, 23);
panel.add(id);

user = new JTextField();
user.setBounds(330, 90, 200, 23);
panel.add(user);

nm = new JTextField();
nm.setBounds(330, 135, 200, 23);
panel.add(nm);

pass = new JTextField();
pass.setBounds(330, 180, 200, 23);
panel.add(pass);

// Buttons

create = new JButton("Create");
create.setBackground(new Color(100, 216, 230));
create.setForeground(Color.BLACK);
create.setBounds(200, 240, 100, 25);
create.addActionListener(this);
panel.add(create);

cancel = new JButton("Cancel");
cancel.setBackground(new Color(100, 216, 230));
cancel.setForeground(Color.BLACK);
cancel.setBounds(370, 240, 100, 25);
cancel.addActionListener(this);
panel.add(cancel);

setVisible(true);
}

@Override
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == create) {
        String suser = user.getText();
        String snm = nm.getText();
        String spass = pass.getText();
        String sid_no = id.getText();

        try {
            Conn c = new Conn();
            String query = "insert into login1 values('"+sid_no+"',
'"+suser+"', '"+snm+"', '"+spass+"')";

            c.s.executeUpdate(query);

```



```

        JOptionPane.showMessageDialog(null, "Account Created
Successfully");
        setVisible(false);
        new Login();

    } catch (Exception e){
        e.printStackTrace();
    }

    } else if (ae.getSource() == cancel) {
        setVisible(false);

        new Login();
    }
}

public static void main(String[] args) {
    new Signup();
}
}

```

Connection Class

```

public class Conn {

    Connection c;
    Statement s;

    Conn(){
        try {
            c = DriverManager.getConnection("jdbc:mysql://localhost:3306/ebilling",
"root", "@iri^niarobi");
            s = c.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Project Class

```

public class Project extends JFrame implements ActionListener {

    // Declare instance variables for menu items that need action listeners
    JMenuItem calculatebill, newcustomer, customerdetails, depositdetails,
updateinfo, viewinfo, paybill, billdetails, generatebill, notepad, calculator,
exit;

    Project() {
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getResource("icon/elect1.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1550, 850,
Image.SCALE_DEFAULT);

```

```

        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        add(image);

        JMenuBar mb = new JMenuBar();
        setJMenuBar(mb);

        // First Menu
        JMenu master = new JMenu("      Master      ");
        master.setForeground(Color.RED);
        mb.add(master);

        // First Menu Dropdowns
        newcustomer = createMenuItem(" New Customer", "icon/icon1.png", 'N',
        KeyEvent.VK_N, master);
        customerdetails = createMenuItem(" Customer Details", "icon/icon2.png",
        'D', KeyEvent.VK_D, master);
        depositdetails = createMenuItem(" Deposit Details", "icon/icon3.png",
        'M', KeyEvent.VK_M, master);
        calculatebill = createMenuItem(" Calculate Bill", "icon/icon4.png", 'B',
        KeyEvent.VK_B, master);

        // Second Menu
        JMenu info = new JMenu("      Information      ");
        info.setForeground(Color.RED);
        mb.add(info);

        // Second Menu Dropdowns
        updateinfo = createMenuItem(" Update Information", "icon/icon5.png", 'U',
        KeyEvent.VK_U, info);
        viewinfo = createMenuItem(" View Information", "icon/icon6.png", 'V',
        KeyEvent.VK_V, info);

        // Third Menu
        JMenu user = new JMenu("      User      ");
        user.setForeground(Color.RED);
        mb.add(user);

        // Third Menu Dropdowns
        paybill = createMenuItem(" Pay Bill", "icon/icon7.png", 'P',
        KeyEvent.VK_P, user);
        billdetails = createMenuItem(" Bill Details", "icon/icon8.png", 'T',
        KeyEvent.VK_T, user);

        // Fourth Menu
        JMenu report = new JMenu("      Report      ");
        report.setForeground(Color.RED);
        mb.add(report);

        // Fourth Menu Dropdowns
        generatebill = createMenuItem(" Generate Bill", "icon/icon9.png", 'G',
        KeyEvent.VK_G, report);

        // Fifth Menu
        JMenu utility = new JMenu("      Utility      ");
        utility.setForeground(Color.RED);
        mb.add(utility);

        // Fifth Menu Dropdowns
        notepad = createMenuItem(" Notepad", "icon/icon12.png", 'P',
        KeyEvent.VK_P, utility);
        calculator = createMenuItem(" Calculator", "icon/icon9.png", 'C',
        KeyEvent.VK_C, utility);

```

```

        // Sixth Menu
        JMenu mexit = new JMenu("      Exit      ");
        mexit.setForeground(Color.RED);
        mb.add(mexit);

        // Sixth Menu Dropdowns
        exit = createMenuItem(" Exit", "icon/icon11.png", 'E', KeyEvent.VK_E,
mexit);

        // Add action listeners
        calculatebill.addActionListener(this);
        generatebill.addActionListener(this);
        exit.addActionListener(this);

        setVisible(true);
    }

    private JMenuItem createMenuItem(String text, String iconPath, char mnemonic,
int acceleratorKey, JMenu menu) {
        JMenuItem item = new JMenuItem(text);
        item.setFont(new Font("monospaced", Font.BOLD, 12));
        item.setBackground(Color.WHITE);
        ImageIcon icon = new ImageIcon(ClassLoader.getResource(iconPath));
        Image image = icon.getImage().getScaledInstance(20, 20,
Image.SCALE_DEFAULT);
        item.setIcon(new ImageIcon(image));
        item.setMnemonic(mnemonic);
        item.setAccelerator(KeyStroke.getKeyStroke(acceleratorKey,
ActionEvent.CTRL_MASK));
        menu.add(item);
        return item;
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == calculatebill) {
            // Example call to another class
            new CalculateBill();
        } else if (ae.getSource() == generatebill) {
            new GenerateBill();
        } else if (ae.getSource() == exit) {
            setVisible(false);
            dispose(); // Properly close the application
        }
    }

    public static void main(String[] args) {
        new Project();
    }
}

```

Calculate Bill Class

```

public class CalculateBill extends JFrame implements ActionListener {

    JTextField tfmeterno, tfunits, tfdueunits;
    JButton submit, cancel;
    JLabel lblmetertype, lblmeterno, lblunits, lbldueunits;
    Choice metertype;

```

```

public CalculateBill() {
    setSize(500, 340);
    setLocation(500, 300);

    JPanel p = new JPanel();
    p.setLayout(null);
    p.setBackground(new Color(235, 235, 235));
    add(p);

    JLabel heading = new JLabel("Calculate Electricity Bill");
    heading.setBounds(100, 20, 300, 25);
    heading.setFont(new Font("Tahoma", Font.BOLD, 24));
    p.add(heading);

    // Meter Type

    lblmetertype = new JLabel("Meter Type:");
    lblmetertype.setBounds(50, 70, 150, 25);
    lblmetertype.setFont(new Font("Tahoma", Font.BOLD, 14));
    p.add(lblmetertype);

    metertype = new Choice();
    metertype.setBounds(290, 70, 150, 25);
    metertype.add("Regular");
    metertype.add("Industrial");
    p.add(metertype);

    // Meter Number

    lblmeterno = new JLabel("Meter Number:");
    lblmeterno.setBounds(50, 110, 150, 25);
    lblmeterno.setFont(new Font("Tahoma", Font.BOLD, 14));
    p.add(lblmeterno);

    tfmeterno = new JTextField();
    tfmeterno.setBounds(290, 110, 150, 25);
    p.add(tfmeterno);

    // Units Consumed

    lblunits = new JLabel("Units Consumed:");
    lblunits.setBounds(50, 150, 150, 25);
    lblunits.setFont(new Font("Tahoma", Font.BOLD, 14));
    p.add(lblunits);

    tfunits = new JTextField();
    tfunits.setBounds(290, 150, 150, 25);
    p.add(tfunits);

    // Due Units

    lbldueunits = new JLabel("Due Units:");
    lbldueunits.setBounds(50, 190, 150, 25);
    lbldueunits.setFont(new Font("Tahoma", Font.BOLD, 14));
    p.add(lbldueunits);

    tfdueunits = new JTextField();
    tfdueunits.setBounds(290, 190, 150, 25);
    p.add(tfdueunits);

    // Submit Button

```

```

        submit = new JButton("Submit");
        submit.setBounds(100, 250, 100, 25);
        submit.setBackground(Color.BLACK);
        submit.setForeground(Color.WHITE);
        submit.addActionListener(this);
        p.add(submit);

        // Cancel Button

        cancel = new JButton("Cancel");
        cancel.setBounds(290, 250, 100, 25);
        cancel.setBackground(Color.BLACK);
        cancel.setForeground(Color.WHITE);
        cancel.addActionListener(this);
        p.add(cancel);

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == submit) {
            try {
                float units = Float.parseFloat(tfunits.getText());
                float dueUnits = Float.parseFloat(tfdueunits.getText());

                float currentBill = calculateBill(units);
                float dueBill = calculateBill(dueUnits);
                float totalPayable = currentBill + dueBill;

                showResult(currentBill, dueBill, totalPayable);
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(this, "Please enter valid numeric
values for units.");
            }
        } else if (ae.getSource() == cancel) {
            setVisible(false);
            dispose();
        }
    }

    private float calculateBill(float units) {
        if (units >= 0 && units <= 50) {
            return units * 4.63f;
        } else if (units > 50 && units <= 75) {
            return units * 5.26f;
        } else if (units > 75 && units <= 200) {
            return units * 7.20f;
        } else if (units > 200 && units <= 300) {
            return units * 7.59f;
        } else if (units > 300 && units <= 400) {
            return units * 8.02f;
        } else if (units > 400 && units <= 600) {
            return units * 12.67f;
        } else {
            return units * 14.61f;
        }
    }

    private void showResult(float currentBill, float dueBill, float totalPayable)
    {
        JFrame resultFrame = new JFrame("Bill Summary");
        resultFrame.setSize(400, 300);
    }

```

```

        resultFrame.setLocation(560, 320);
        resultFrame.getContentPane().setBackground(new Color(235, 235, 235));
        resultFrame.setLayout(null);

        JLabel heading = new JLabel("Bill Summary");
        heading.setBounds(130, 20, 200, 25);
        heading.setFont(new Font("Tahoma", Font.BOLD, 18));
        resultFrame.add(heading);

        JLabel currentLabel = new JLabel("Current Bill: Taka " +
String.format("%.2f", currentBill));
        currentLabel.setBounds(100, 70, 300, 25);
        currentLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));
        resultFrame.add(currentLabel);

        JLabel dueLabel = new JLabel("Due Bill: Taka " + String.format("%.2f",
dueBill));
        dueLabel.setBounds(110, 110, 300, 25);
        dueLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));
        resultFrame.add(dueLabel);

        JLabel totalLabel = new JLabel("Total Payable (Without VAT): Taka " +
String.format("%.2f", totalPayable));
        totalLabel.setBounds(40, 150, 350, 25);
        totalLabel.setFont(new Font("Tahoma", Font.BOLD, 14));
        resultFrame.add(totalLabel);

        JButton closeButton = new JButton("Close");
        closeButton.setBounds(130, 200, 100, 30);
        closeButton.setBackground(Color.BLACK);
        closeButton.setForeground(Color.WHITE);
        closeButton.addActionListener(e -> resultFrame.dispose());
        resultFrame.add(closeButton);

        resultFrame.setVisible(true);
    }

    public static void main(String[] args) {
        new CalculateBill();
    }
}

```

Generate Bill Class

```

public class GenerateBill extends JFrame implements ActionListener {

    JTextField tfcname, tfunits, tfdueunits, tfmeterno, tfcaddress, tfcemail,
    tfcnumber;
    JButton gbill, cancel;
    JLabel cmetertype, cmonth;
    Choice metertype, month;

    public GenerateBill() {
        setSize(500, 700);
        setLocation(500, 75);

        JPanel p = new JPanel();
        p.setLayout(null);
        p.setBackground(new Color(235, 235, 235));
        add(p);
    }
}

```

```

JLabel heading = new JLabel("Dhaka Power Distribution Company");
heading.setBounds(25, 20, 500, 50);
heading.setFont(new Font("Tahoma", Font.BOLD, 24));
p.add(heading);

JLabel dpdc = new JLabel("(DPDC)");
dpdc.setBounds(190, 70, 150, 25);
dpdc.setFont(new Font("Tahoma", Font.BOLD, 24));
p.add(dpdc);

JLabel ministry = new JLabel("Ministry of Power, Energy & Mineral
Resources");
ministry.setBounds(50, 105, 500, 25);
ministry.setFont(new Font("Tahoma", Font.BOLD, 16));
p.add(ministry);

JLabel dash = new JLabel("_____");
dash.setBounds(25, 135, 500, 25);
dash.setFont(new Font("Tahoma", Font.BOLD, 16));
p.add(dash);

// Customer Details
addLabelAndField(p, "Customer Name:", 190, tfcname = new JTextField());
addLabelAndField(p, "Customer Address:", 230, tfcaddress = new
JTextField());
addLabelAndField(p, "Customer Email:", 270, tfcemail = new JTextField());
addLabelAndField(p, "Contact Number:", 310, tfcnumber = new
JTextField());
addLabelAndField(p, "Meter Number:", 350, tfmeterno = new JTextField());

// Meter Type
cmeterType = new JLabel("Meter Type:");
cmeterType.setBounds(35, 390, 150, 25);
cmeterType.setFont(new Font("Tahoma", Font.BOLD, 14));
p.add(cmeterType);

meterType = new Choice();
meterType.setBounds(220, 390, 230, 25);
meterType.add("Regular");
meterType.add("Industrial");
p.add(meterType);

// Current Month
cmonth = new JLabel("Current Month:");
cmonth.setBounds(35, 430, 150, 25);
cmonth.setFont(new Font("Tahoma", Font.BOLD, 14));
p.add(cmonth);

month = new Choice();
month.setBounds(220, 430, 230, 25);
month.add("January");
month.add("February");
month.add("March");
month.add("April");
month.add("May");
month.add("June");
month.add("July");
month.add("August");
month.add("September");
month.add("October");
month.add("November");
month.add("December");

```

```

        p.add(month);

        // Units and Due Units
        addLabelAndField(p, "Units Consumed:", 470, tfunits = new JTextField());
        addLabelAndField(p, "Due Units:", 510, tfdueunits = new JTextField());

        // Generate Bill Button
        gbill = new JButton("Generate Bill");
        gbill.setBounds(65, 585, 150, 25);
        gbill.setBackground(Color.BLACK);
        gbill.setForeground(Color.WHITE);
        gbill.addActionListener(this);
        p.add(gbill);

        // Cancel Button
        cancel = new JButton("Cancel");
        cancel.setBounds(270, 585, 150, 25);
        cancel.setBackground(Color.BLACK);
        cancel.setForeground(Color.WHITE);
        cancel.addActionListener(this);
        p.add(cancel);

        setVisible(true);
    }

    private void addLabelAndField(JPanel p, String label, int y, JTextField
field) {
        JLabel l = new JLabel(label);
        l.setBounds(35, y, 150, 25);
        l.setFont(new Font("Tahoma", Font.BOLD, 14));
        p.add(l);

        field.setBounds(220, y, 230, 25);
        p.add(field);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == gbill) {
            try {
                float units = Float.parseFloat(tfunits.getText());
                float costPerUnit = getCostPerUnit(units);
                float currentBill = units * costPerUnit;
                float vat = currentBill * 0.05f;
                float meterCharge = 150f;
                float totalPayable = currentBill + vat + meterCharge;

                showResult(currentBill, vat, meterCharge, totalPayable);
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(this, "Please enter valid numeric
values for units.");
            }
        } else if (ae.getSource() == cancel) {
            setVisible(false);
            dispose();
        }
    }

    private float getCostPerUnit(float units) {
        if (units > 0 && units <= 50) return 4.63f;
        else if (units > 50 && units <= 75) return 5.26f;
        else if (units > 75 && units <= 200) return 7.20f;
        else if (units > 200 && units <= 300) return 7.59f;
    }

```



```

        else if (units > 300 && units <= 400) return 8.02f;
        else if (units > 400 && units <= 600) return 12.67f;
        else return 14.61f;
    }

private void showResult(float currentBill, float vat, float meterCharge, float
totalPayable) {
    float dueUnits = Float.parseFloat(tfdueunits.getText());
    float dueAmount = dueUnits * getCostPerUnit(dueUnits);

    JFrame display = new JFrame("Bill Summary");
    display.setSize(305, 450);
    display.setLocation(620, 210);
    display.setLayout(new BorderLayout());

    JPanel displayPanel = new JPanel();
    displayPanel.setLayout(new BoxLayout(displayPanel, BoxLayout.Y_AXIS));
    displayPanel.add(new JLabel("

displayPanel.add(new JLabel(" "));
displayPanel.add(new JLabel(" Dhaka Power Distribution Company (DPDC)"));
displayPanel.add(new JLabel(" Ministry of Power, Energy & Mineral
Resources"));
displayPanel.add(new JLabel(" Electricity Department"));
displayPanel.add(new JLabel(" Payment Invoice"));
displayPanel.add(new JLabel("

displayPanel.add(new JLabel(" "));
displayPanel.add(new JLabel(" 01. Customer Name: " + tfcname.getText()));
displayPanel.add(new JLabel(" 02. Customer Address: " +
tfcaddress.getText()));
displayPanel.add(new JLabel(" 03. Customer Division: Dhaka"));
displayPanel.add(new JLabel(" 04. Customer Phone Number: " +
tfcnumber.getText()));
displayPanel.add(new JLabel(" 05. Customer Email: " + tfcemail.getText()));
displayPanel.add(new JLabel(" 06. Customer Meter Number: " +
tfmeterno.getText()));
displayPanel.add(new JLabel(" 07. Customer Meter Type: " +
metertype.getSelectedItem()));
displayPanel.add(new JLabel(" 08. Current Month: " +
month.getSelectedItem()));
displayPanel.add(new JLabel(" 09. Cost per unit: Taka " +
getCostPerUnit(Float.parseFloat(tfunits.getText()))));
displayPanel.add(new JLabel(" 10. Meter charge: Taka " +
String.format("%.2f", meterCharge)));
displayPanel.add(new JLabel(" 11. VAT: Taka " + String.format("%.2f", vat)));
displayPanel.add(new JLabel(" 12. Unit consumed: " + tfunits.getText()));
displayPanel.add(new JLabel(" 13. Total charge: Taka " +
String.format("%.2f", currentBill)));
displayPanel.add(new JLabel(" 14. Previous Due Amount: Taka " +
String.format("%.2f", dueAmount)));
displayPanel.add(new JLabel(" 15. Total payable: Taka " +
String.format("%.2f", totalPayable)));

    JButton printButton = new JButton("Print");
    printButton.addActionListener(e -> {
        PrinterJob printerJob = PrinterJob.getPrinterJob();
        printerJob.setJobName("Electricity Bill");

        // Define what to print
        printerJob.setPrintable((graphics, pageFormat, pageIndex) -> {
            if (pageIndex > 0) {
                return Printable.NO_SUCH_PAGE;

```

```
        }
        // Print the content of the display frame
        Graphics2D g2d = (Graphics2D) graphics;
        g2d.translate(pageFormat.getImageableX(),
pageFormat.getImageableY());
        displayPanel.paint(g2d);
        return Printable.PAGE_EXISTS;
    });

    // Display print dialog
    if (printerJob.printDialog()) {
        try {
            printerJob.print();
        } catch (PrinterException ex) {
            JOptionPane.showMessageDialog(display, "Print error: " +
ex.getMessage());
        }
    }
});

JButton closeButton = new JButton("Close");
closeButton.addActionListener(e -> display.dispose());

JPanel buttonPanel = new JPanel();
buttonPanel.add(printButton);
buttonPanel.add(closeButton);

display.add(displayPanel, BorderLayout.CENTER);
display.add(buttonPanel, BorderLayout.SOUTH);

display.setVisible(true);
}

public static void main(String[] args) {
    new GenerateBill();
}
}
```

Observations

- The GUI is intuitive and responsive.
- The database integration effectively manages user data and billing records.
- Bill calculation logic works accurately based on predefined rates.

Results

- Successful user login and signup.
- Accurate bill calculations based on user inputs.
- Printable bill generation with detailed breakdowns.

Discussion and Analysis

- **Strengths:** The system effectively automates the electricity billing process and provides a clear user interface.
- **Weaknesses:**
 - The login fields lack input validation (e.g., password masking).
 - All of the menus and dropdowns are not functional right now.
 - Only and only controllable by admins.
 - Depends on MySQL server.
 - Lack of the option of logging in as a user.
- **Improvements:**
 - Implement password encryption.
 - Additional error handling.
 - Implementation of online bill paying system for users.
 - Activation of all menus and dropdowns

Conclusion

The Electricity Billing System successfully demonstrates a functional Java application for managing electricity billing. The integration of Swing, JDBC, and MySQL provides a comprehensive solution for user authentication, bill calculation, and invoice generation.

References

1. Oracle Java Documentation: <https://docs.oracle.com/javase/>
 2. MySQL Documentation: <https://dev.mysql.com/doc/>
 3. Java Swing Tutorials: <https://docs.oracle.com/javase/tutorial/uiswing/>
-

Thank you for Looking on to our work

From

Team D3F4UL7