**C PROJECT**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### C. V. Raman Global University, Odisha, Bhubaneswar

PIN -752054, India.

**Topic** **:Electric Bill Generator System**

**SUBMITTED BY:**

|  |  |
| --- | --- |
| **Members** | **Cranes Registration no** |
| Debjeet Barik | CL2025010601963078 |
| Ch Jagan Subodhi | CL20250106018923105 |
| Subham Kumar Sahu | CL20250106019399103 |
| Birupa Nanda Nayak | CL20250106018793100 |
| Pruthviswar Mahapatra | CL2025010601890486 |

**Introduction:**

The **Electric Bill Generation System** is a comprehensive, terminal-based application designed to simplify and automate the process of calculating and generating electricity bills. It provides an intuitive platform where consumers can monitor their energy consumption, receive real-time billing details, compare previous bills, track payment history, and get notified about unusual usage patterns. The system caters to both consumers and utility providers, ensuring accurate billing, secure transactions, and an efficient way to manage electricity payments and usage analytics.

The primary objective of the **Electric Bill Generation System** is to digitize and simplify the traditional electricity billing process, reducing manual workload and ensuring accurate bill calculations. The system enhances accessibility by allowing users to track their energy consumption, receive billing alerts, and make online payments without visiting utility offices. Registered users can perform tasks such as viewing and comparing their bills, checking payment history, and updating their account information, while administrators can manage customer data, generate reports, and oversee the billing and payment processes to ensure the smooth operation of the system.

This project has been implemented using the **C programming language** The user interface is built using **C with Console-Based Input/Output**, providing a structured and interactive text-based system. For data processing, the system employs **C file handling** to store and retrieve customer details, billing records, and payment history using text or binary files. Additionally, **Standard C Libraries** such as **stdio.h, stdlib.h, string.h, and math.h** are utilized for core application logic, bill calculations, and efficient data management. These technologies work together to ensure a fast, lightweight, and standalone electricity billing system with accurate calculations and secure data handling.

By leveraging the capabilities of these efficient development tools and technologies, the **Electric Bill Generation System** aims to enhance the user experience, ensure accurate bill calculations, and promote the adoption of digital solutions for electricity billing and payment management. This system reduces manual errors, improves efficiency, and provides consumers with a seamless way to track their energy usage and payments.

**General Overview:**

* Provide an easy-to-use interface for users to generate and manage electricity bills.
* Allow users to input electricity consumption details for bill calculation.
* Enable users to view current and past bills, along with payment history.
* Facilitate secure login/logout functionalities for both consumers and administrators.
* Generate alerts for unusual energy consumption patterns.
* Allow administrators to manage customer details, track payments, and generate monthly reports.

**Features of the Electric Bill Generation System**

* **Add New Customer** – Allows administrators to register new customers by storing their personal details and electricity usage records.
* **View Customer Details** – Enables users to access stored customer information, including account details and consumption history.
* **Generate New Bill** – Calculates the monthly electricity bill based on usage and predefined tariff rates.
* **View Latest Bill** – Displays the most recent electricity bill for the customer.
* **Record Payment** – Updates the system with payment transactions and adjusts outstanding balances accordingly.
* **View Payment History** – Provides a log of past payments made by the customer for reference and tracking.
* **Compare with Previous Bill** – Allows users to compare their current bill with previous months to monitor usage trends.
* **Project Next Month's Bill** – Estimates the upcoming bill based on past consumption patterns and trends.
* **Generate Energy Usage Alert** – Notifies users when their energy consumption is unusually high or exceeds predefined limits.
* **Update Customer Information** – Enables modification of customer details, such as contact information and address.
* **Show All Customers** – Displays a list of all registered customers in the system for administrative management.
* **Search Customer** – Allows quick lookup of customer details using search parameters such as name or account number.
* **Generate Monthly Report** – Creates a summarized report of electricity consumption, billing, and payments for a given period.
* **Exit** – Provides an option to close the application safely.

**Implementation**

**Customer Management in Electric Bill Generation System**

* addCustomer() function:
  + Adds a new customer with a unique Customer ID (starting from 1001).
  + Collects and stores name, address, phone, email, customer type, and meter number.
  + Records the connection date and sets the customer as active.
  + Saves data after successful customer addition.
* findCustomerByMeterNumber() function:
  + Searches for a customer using their meter number.
  + Returns the index if found; otherwise, returns -1.

**Customer Display and Bill Calculation in Electric Bill Generation System**

* **displayCustomer(int index)** function:
  + Displays complete customer details, including **ID, name, address, phone, email, meter number, and customer type**.
  + Shows **connection date, active status, and the number of bills** issued.
* **calculateBillAmount(CustomerType type, float usage, TimeOfUseUsage tou\_usage)** function:
  + Determines the **appropriate rate structure** based on customer type (Residential, Commercial, or Industrial).
  + Calculates the bill using **tiered pricing** (separate rates for different usage levels).
  + Incorporates **time-of-use billing** by adding peak and off-peak hour charges.
  + Applies **taxes** and returns the final bill amount.

**Bill Generation in Electric Bill Generation System**

**Key Functionalities:**

* **Manages Billing History:**
  + If the customer reaches the maximum bill history limit, older records are shifted to make room for new ones.
* **Bill Creation:**
  + Assigns a **unique Bill ID** and records the **bill date and due date** (15 days from the bill date).
  + Stores **previous and current meter readings** to calculate total electricity usage.
* **Time-of-Use (TOU) Tracking:**
  + Takes input for **peak-hour and off-peak-hour** usage.
* **Bill Calculation:**
  + Uses the **calculateBillAmount()** function to compute the total bill amount based on **tiered pricing and TOU rates**.
* **Finalization and Storage:**
  + Updates the **customer’s billing history** and saves the data for future reference.
  + Calls **displayBill()** to show the generated bill.

This function ensures **accurate billing, efficient usage tracking, and proper data management** for customers.

**Bill Display in Electric Bill Generation System**

**Key Functionalities:**

* **Bill Information:**
  + Displays **Bill ID, bill date, due date, and customer details** (ID, name, address, and meter number).
* **Usage Details:**
  + Shows **previous and current meter readings** and calculates total consumption.
  + Displays **peak-hour and off-peak-hour usage** separately.
* **Billing Summary:**
  + Shows the **total amount due** and the **payment status** (Paid/Unpaid).
  + If paid, includes **payment date and method**.

This function ensures **clear and structured bill presentation**, making it easy for customers to understand their **electricity usage and payment status**.

**Payment Recording in Electric Bill Generation System**

**Key Functionalities:**

* **Checks Payment Status:** 
  + Prevents duplicate payments by verifying if the bill is already paid.
* **Records Payment Details:** 
  + Marks the bill as paid and stores the payment date.
  + Accepts the payment method (Cash, Credit Card, or Bank Transfer).
* **Data Storage:** 
  + Saves the updated information for future reference.

This function ensures secure payment tracking and prevents duplicate transactions, improving billing accuracy.

**Payment History in Electric Bill Generation System**

**Key Functionalities:**

* **Checks for Payment Records:**
  + If no payments exist, notifies the user.
* **Displays Billing Details:**
  + Lists **Bill ID, bill date, amount, and payment status** (Paid/Unpaid).
* **Shows Payment Information (if paid):**
  + Displays **payment date and method** (e.g., Cash, Credit Card, Bank Transfer).

This function helps in **tracking past payments** and ensures **transparency in billing records**.

**Bill Comparison in Electric Bill Generation System**.

**Key Functionalities:**

* **Checks for sufficient billing history** (at least two bills required).
* **Compares total usage and bill amount** between the last two bills.
* **Calculates and displays differences** in usage and cost (both in absolute and percentage values).
* **Alerts users if consumption increases by more than 20%**, providing energy-saving tips.

This feature helps users track energy consumption trends, identify unusual spikes, and manage electricity usage efficiently.

**Projected Bill Calculation in Electric Bill Generation System**

The **projectNextBill(int customer\_index)** function estimates the next month’s electricity bill based on previous usage trends.

**Key Functionalities:**

* **Checks for past bills** to make an accurate projection.
* **Calculates the average increase in usage** from previous bills.
* **Estimates total consumption** using past trends and time-of-use ratios.
* **Predicts the next bill amount** using standard billing calculations.
* **Provides energy-saving tips** to help users reduce future costs.

This feature helps users **anticipate electricity expenses, plan their budget, and adopt energy-efficient habits**.

**Energy Usage Alert in Electric Bill Generation System**.

**Key Functionalities:**

* **Compares last month’s usage** with the customer’s average monthly consumption.
* **Detects significant increases** (above 20%) and issues alerts with **possible causes**.
* **Suggests energy-saving actions** like scheduling an energy audit and checking for faulty appliances.
* **Identifies reduced consumption** and acknowledges energy-efficient behavior.
* **Analyzes peak hour usage** and provides cost-saving recommendations.

This feature helps users monitor electricity usage, identify unusualconsumption trends, and optimize energy costs effectively.

**Customer Information Update in Electric Billing System**

**Key Functionalities:**

* **Search by meter number** if the customer index is unknown.
* **Update fields** such as name, address, phone, email, and customer type.
* **Change active status** (Active/Inactive) for account management.
* **Validation and confirmation prompts** before modifying data.
* **Ensures data persistence** by saving changes after updates.

This feature enhances data accuracy, customer record management, and user flexibility in maintaining account information.

**Displaying All Customers in Electric Billing System**

**Key Features:**

* **Checks for customer records** and alerts if none are found.
* **Displays customer details** including ID, name, meter number, type, and status.
* **Formats data neatly** in a table for easy readability.
* **Shows total customer count** at the end.

This function helps in **quickly reviewing customer records**, making it useful for administrators and customer support teams.

**Customer Search Function Overview**

**Key Features:**

* **Search Options:**
  + By **Name** (partial matching supported).
  + By **Meter Number**.
  + By **Customer ID** (exact match).
  + By **Phone Number**.
* **Formatted Output:**
  + Displays **Customer ID, Name, Meter Number, Type, and Status** in a tabular format.
* **Total Results Count:** Shows the number of matches found.
* **Option to View Full Details:** If results are found, the user can **view detailed information** by entering a customer ID.

**Usefulness:**

This function **enhances customer management** by making it easy to **locate specific customers quickly** based on multiple search parameters. It is especially useful for billing support teams and customer service representatives.

**Electric Billing System Report Generation Function**

**Key Features:**

1. **File Creation:**
   * Saves the report as **report\_DD\_MM\_YYYY.txt** using the current date.
2. **Customer Summary:**
   * Total customers and their active/inactive status.
   * Breakdown by customer type (Residential, Commercial, Industrial).
3. **Billing Summary:**
   * Total **bills generated, paid, and unpaid**.
   * **Total revenue collected & outstanding** amounts.
   * Overall energy usage (units consumed).
4. **Usage by Customer Type:**
   * Segments electricity consumption and charges by Residential, Commercial, and Industrial users.
5. **Time of Use Analysis:**
   * Separates **Peak Hours (2 PM - 8 PM)** and **Off-Peak Hours (8 PM - 2 PM)** energy usage.
6. **Top 5 Consumers:**
   * Identifies customers with the highest monthly energy usage and billing amount.

**Conclusion:**

The Bill Generation System effectively demonstrates the practical application of enterprise-level Java technologies to develop a secure, scalable, and efficient billing solution for various business domains.

The system provides comprehensive functionality for both end users and administrators. For users, it enables features such as creating invoices, tracking payment status, viewing billing history, and generating reports. For administrators, the platform offers tools to manage customer records, monitor transactions, configure tax settings, and maintain data security through role-based access control mechanisms.

One of the key highlights of this project is its modular and extensible architecture, allowing for the easy addition of new features and future improvements. For example, the system can be enhanced by integrating an online payment gateway to enable secure, real-time transactions. Additionally, implementing email and SMS notifications can improve user experience by providing instant updates on invoice generation, due dates, and payment confirmations. Furthermore, ensuring mobile responsiveness through modern UI frameworks or developing a mobile application can enhance accessibility, particularly for business users who need on-the-go billing management.

From a learning perspective, this project serves as an excellent example for students and developers to understand how enterprise applications are built using Java technologies. It covers key concepts such as session management, database connectivity, MVC architecture, and web server deployment using Apache Tomcat. Additionally, it promotes best practices in secure transaction handling, data validation, and report generation.

In conclusion, the Bill Generation System is not only a powerful tool for automating and managing the billing process efficiently but also serves as a strong foundation for developing more advanced and feature-rich financial applications in various industries.