**ABSTRACT**

Electricity consumers are often faced with the problem of inaccuracy and delay in monthly billing due to some drawbacks. Thus, it is essential to have an efficient system for such purposes via electronic platform with consideration to proximity. The proposed system automates the conventional process of paying electricity bill by visiting the Electricity Board which is tiresome and time consuming. It is also designed to automate the electricity bill calculation and payment for user convenience. The system is developed with JSP SERVLET as the base programming language which can be used to develop websites, web applications and web services. The MYSQL is a relational database management system based on Structured Query Language (SQL) which is used for the purpose of web database. The system would be having two logins: the administrative and user login. Theadministrator can view the user's account details and can add the customer's information of consuming units of energy of the current month in their account. The Admin must feed the system with the electricity usage data into respective user’s account. The system then calculates the electricity bill for every user and updates the information into their account every month. Users can then view their electricity bill and pay before the month end.

**INTRODUCTION**

Electricity Billing System is a software-based application. i. This project aims at serving the department of electricity by computerizing the billing system. ii. It mainly focuses on the calculation of units consumed during the specified time and the money to be charged by the electricity offices. iii. This computerized system will make the overall billing system easy, accessible, comfortable, and effective for consumers. To design the billing system more service oriented and simple, the following features have been implemented in the project. The application has high speed of performance with accuracy and efficiency. 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 electricity billing software calculates the units consumed by the customerand makes bills, it requires small storage for installation and functioning. There is provision for debugging if any problem is encountered in the system. The system excludes the need of maintaining paper electricity bill, administrator does not have to keep amanual track of the users, users can pay the amount without visiting the office. Thus, it saves human efforts and resources.

**Advantages of Electronic billing:**

1. The first advantage is that it offers a paperless mode of transaction which is also environmentally friendly and clutter free for both the receiver and sender of the electronic billing.
2. The online billing services are one of the least expensive forms of billing when compared to the traditional billing.
3. The electronic billing system is absolutely hassle free as one can protect the bill with the help of password and can be opened only by the recipient.
4. The electronic billing services are both customer friendly and also beneficial for the bill generators as there is focus on the process rather than on the mode of bill dispatch.
5. The electronic billing also provides a great advantage of saving time and effort that are normally lost in a traditional billing system. Besides there is no loss of bill when making use of the electronic mode of billing.

**Disadvantages of electronic billing –**

1. Most of the online billings systems are outsourced. As a result of the outsource billing there is a fee involved when making use of this system of billing.
2. With online billing system there is always fear of safety and security to the personal information due to the increased spywares and malwares being rampant on the internet.
3. When the electronic payment system is opted the customer account gets instantly debited for the funds that need to be paid while in a traditional billing system the customer gets sometime between the billing receipt and actual payment.
4. There is likelihood of many disputes that may arise due to the downtime of internet on the website that support electronic billing system. This can lead to many users being dissatisfied with the entire system and reduction in its use.

**Algorithm:**

1. Display Electricity Billing System Menu

2. Prompt user for choice (1, 2, 3, 4)

3. Read user choice

4. If choice == 1,

- Add new customer:

- Prompt for customer details

- Write customer details to file

If choice == 2,

- Generate electricity bill:

- Prompt for customer ID

- Read customer data from file

- Calculate bill amount

- Display bill information

If choice == 3,

- Remove customer record:

- Prompt for customer ID

- Read customer data from file

- Remove customer record from file

If choice == 4,

- Exit the program

5. Loop back to step 1 if choice != 4

6.End

**Flowchart:**

Display choices

1. Add new customer
2. Generate Electricity bill
3. Remove customer record
4. Exit

Declare variables, Define all required classes.

Read choice

Switch choice

Add new customer

Case 1:

Read customer Data

Case 2:

Display Bill

Generate Electricity bill

Case 3:

Remove customer record

Incorrect choice

Case 4:

Case 4:

Exit

**Code:**

import java.io.\*;

importjava.util.Scanner;

public class ElectricityBillingSystem {

static class Customer {

privateintcustomerId;

private String name;

private String address;

private double previousMeterReading;

public Customer(intcustomerId, String name, String address, double previousMeterReading) {

this.customerId = customerId;

this.name = name;

this.address = address;

this.previousMeterReading = previousMeterReading;

}

// to access customer record

publicintgetCustomerId() {

returncustomerId;

}

public String getName() {

return name;

}

public String getAddress() {

return address;

}

public double getPreviousMeterReading() {

returnpreviousMeterReading;

}

}

private static final double RATE\_PER\_UNIT = 10; // Rate per unit in rs

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("========================================");

System.out.println("Electricity Billing System Menu:");

System.out.println("========================================");

System.out.println("1. Add new customer");

System.out.println("2. Generate electricity bill");

System.out.println("3. Remove customer record");

System.out.println("4. Exit");

System.out.println("========================================");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

System.out.println("========================================");

switch (choice) {

case 1:

addNewCustomer(scanner);

break;

case 2:

generateElectricityBill(scanner);

break;

case 3:

removeCustomerRecord(scanner);

break;

case 4:

System.out.println("Exiting program. Goodbye!");

break;

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

} while (choice != 4);

scanner.close();

}

private static void addNewCustomer(Scanner scanner) {

try {

BufferedWriter writer = new BufferedWriter(new FileWriter("customers.txt", true));

System.out.println("Enter customer details:");

System.out.println("========================================");

System.out.print("Customer ID: ");

intcustomerId = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Name: ");

String name = scanner.nextLine();

System.out.print("Address: ");

String address = scanner.nextLine();

System.out.print("Previous Meter Reading: ");

doublepreviousMeterReading = scanner.nextDouble();

writer.write(customerId + "," + name + "," + address + "," + previousMeterReading);

writer.newLine();

writer.close();

System.out.println("========================================");

System.out.println("Customer added successfully!");

} catch (IOException e) {

System.out.println("Error writing to file: " + e.getMessage());

}

}

private static void generateElectricityBill(Scanner scanner) {

try {

System.out.print("Enter customer ID to generate bill: ");

intcustomerId = scanner.nextInt();

System.out.println("========================================");

BufferedReader reader = new BufferedReader(new FileReader("customers.txt"));

String line;

booleancustomerFound = false;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

int id = Integer.parseInt(parts[0]);

if (id == customerId) {

System.out.println("\nCustomer found:\n Costomer Name : " + parts[1] + " \n Address :" + parts[2]);

doublepreviousReading = Double.parseDouble(parts[3]);

System.out.println("========================================");

System.out.print("Enter current meter reading: ");

doublecurrentReading = scanner.nextDouble();

System.out.println("========================================");

doubleunitsConsumed = currentReading - previousReading;

doublebillAmount = unitsConsumed \* RATE\_PER\_UNIT;

System.out.println();

System.out.println("=========== Electricity Bill ===========");

System.out.println("Costomer Name : " + parts[1]);

System.out.println("========================================");

System.out.println("Address :" + parts[2]);

System.out.println("========================================");

System.out.println("Units consumed: " + unitsConsumed);

System.out.println("========================================");

System.out.println("Bill amount: Rs." + billAmount);

System.out.println("========================================");

System.out.println();

customerFound = true;

break;

}

}

reader.close();

if (!customerFound) {

System.out.println("Customer with ID " + customerId + " not found.");

}

} catch (IOException e) {

System.out.println("Error reading from file: " + e.getMessage());

}

}

private static void removeCustomerRecord(Scanner scanner) {

try {

System.out.print("Enter customer ID to remove: ");

intcustomerId = scanner.nextInt();

File inputFile = new File("customers.txt");

File tempFile = new File("temp.txt");

BufferedReader reader = new BufferedReader(new FileReader(inputFile));

BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));

String line;

booleancustomerFound = false;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

int id = Integer.parseInt(parts[0]);

if (id != customerId) {

writer.write(line);

writer.newLine();

} else {

customerFound = true;

}

}

reader.close();

writer.close();

if (customerFound) {

if (!inputFile.delete()) {

System.out.println("Error deleting file");

return;

}

if (!tempFile.renameTo(inputFile)) {

System.out.println("Error renaming file");

} else {

System.out.println("========================================");

System.out.println("Customer record removed successfully.");

System.out.println("========================================");

System.out.println();

}

} else {

System.out.println("Customer with ID " + customerId + " not found.");

}

} catch (IOException e) {

System.out.println("Error processing file: " + e.getMessage());

System.out.println();

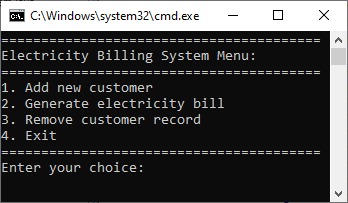
}

}

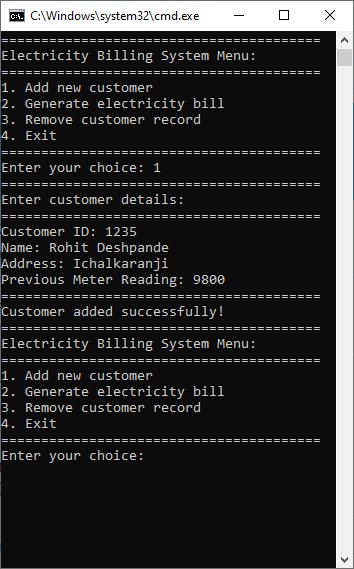
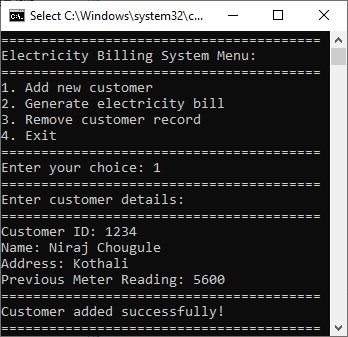
}

Output:

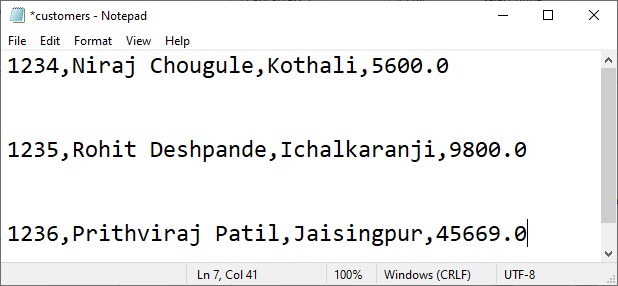
**System Menu:**



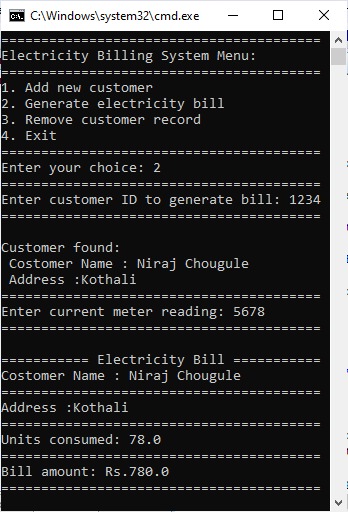
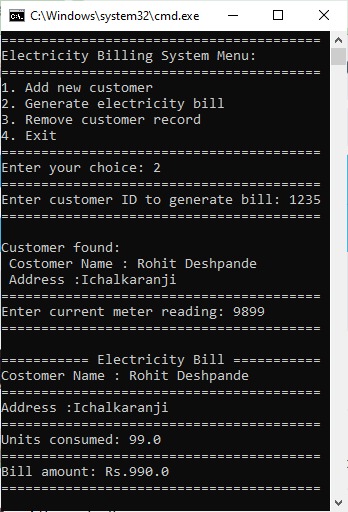
**Add new customer:**



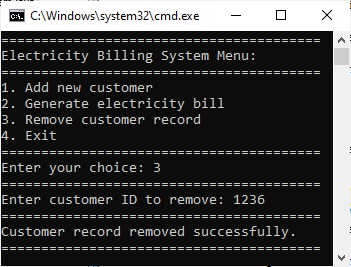
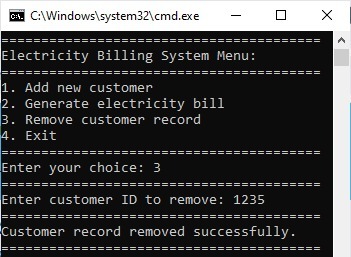
**File after adding customer:**



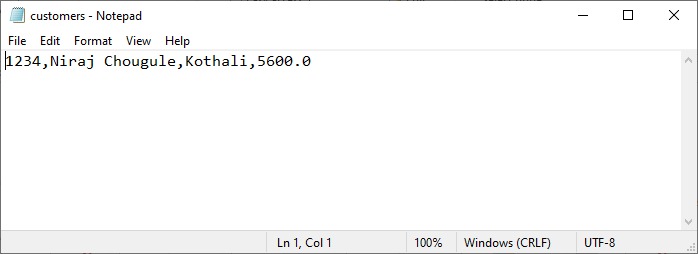
**Generate electricity bill:**

** **

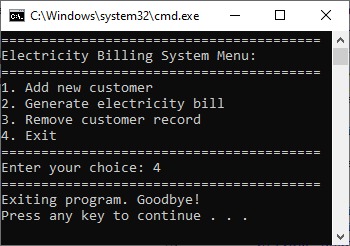
**Remove customer record:**

**File after removing customer record:**

****

**Exit from the program:**



**Conclusion**

It is software that helps the user to work with the billing cycles, paying bills, managing different departments under which employees are working, etc. This software reduces the amount of manual data entry and gives greater efficiency.

**References:**

* <https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/b.sc-cs-batchno-25.pdf>
* <https://www.lovelycoding.org/electricity-bill-management-system/>