ELECTRICITY BILL GENERATION

A MINI PROJECT REPORT

Submitted by

NIKITHA.B   230701211
PRATHEBA.D   230701239
RAGAVI K   230701249

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS) THANDALAM
CHENNAI-602105

2023 - 24

BONAFIDE CERTIFICATE

Certified that this project report "**ELECTRICITY BILL GENERATION**" is the

bonafide work of **"NIKITHA .B (230701211),PRATHEBA.D(230701239) "**

who carried out the project work under my supervision.

Submitted for the Practical Examination held on _____

SIGNATURE   **SIGNATURE**

Mr.RAGU G
Professor and II Year Academic Head   Assistant Professor (SG),
Computer Science and Engineering,   Computer Science and Engineering, Rajalakshmi
Engineering College   Rajalakshmi Engineering College,
(Autonomous),   (Autonomous),
Thandalam, Chennai - 602 105   Thandalam, Chennai - 602 105

INTERNAL EXAMINER   **EXTERNAL EXAMINER**

ABSTRACT

Electricity Billing System is a desktop based application developed in Java programming language. The project aims at serving the department of electricity by computerizing the billing system. It mainly focuses on the calculation of Units consumed during the specified time and the money to be paid to electricity offices.

The interface has 2 logins for admin and the user . The applications does not limit the users to local civilians and has the industrialists in mind and can generate the electricity bills to the industries accompanied with tax inclusion, making this a choice of wider scope.

TABLE OF CONTENTS

1. INTRODUCTION

- **1.1 Introduction** ₒ The Electricity Bill Management System is designed to automate and streamline the process of electricity bill generation, tracking, and payments for electricity consumers. This system allows electricity providers to keep an organized and accurate record of each user's monthly consumption, due payments, and billing history. By eliminating manual processing, this system reduces human error, improves efficiency, and provides users with easy access to their billing details.

- **1.2 Objectives** ₒ **To automate electricity bill calculation:** This reduces human intervention, thereby reducing the chances of errors.

  - ₒ **To maintain a centralized database of users:** This includes storing user information such as account details, address, and monthly usage.

  - ₒ **To facilitate payment processing:** Users can view their outstanding dues and pay bills seamlessly.

  - ₒ **To provide reports and history tracking:** The system will keep a history of past bills and payments for easy reference.

- **1.3 Modules** ₒ **User Management Module:** Handles customer registration, account updates, and user deletion. ₒ **Consumption Tracking Module:** Monitors and records the user's electricity consumption, typically on a monthly basis.

  - ₒ **Bill Generation Module:** Automatically calculates the bill based on the user's consumption using the applicable tariffs.

  - ₒ **Payment Processing Module:** Records payment transactions and updates billing statuses.

  - ₒ **Report Generation Module:** Provides summaries of user consumption, outstanding balances, and payment history.

2. SURVEY OF TECHNOLOGIES

- **2.1 Software Description** ○ The project is developed using Java as the primary programming language for backend logic and user interface design. MySQL (or any chosen

  RDBMS) is used as the database to store and manage data. Java's JDBC (Java Database Connectivity) is utilized to connect the Java application with the MySQL database, enabling CRUD (Create, Read, Update, Delete) operations.

- **2.2 Languages** ○ **2.2.1 SQL**

  - SQL (Structured Query Language) is used to create, manage, and manipulate the database. It is responsible for creating tables for users, bills, and payments, and executing operations such as SELECT, INSERT, UPDATE, and DELETE.

  ○ **2.2.2 Java**

  - Java provides the user interface, application logic, and backend connectivity to the database. It is chosen for its portability and ability to integrate well with databases. Key Java components include JFrame for UI, JDBC for database interaction, and Swing components for the graphical interface.

3. REQUIREMENTS AND ANALYSIS

- **3.1 Requirement Specification** ○ **Functional Requirements**

  - User registration and login.

  - Monthly consumption entry by the provider.

  - Automatic bill calculation based on consumption.

  - Payment tracking and history management.

  ○ **Non-functional Requirements**

- **Reliability:** Ensure the data remains consistent and accurate.

- **Security:** User information and payment data are securely managed.

- **Usability:** User-friendly interface for easy navigation.

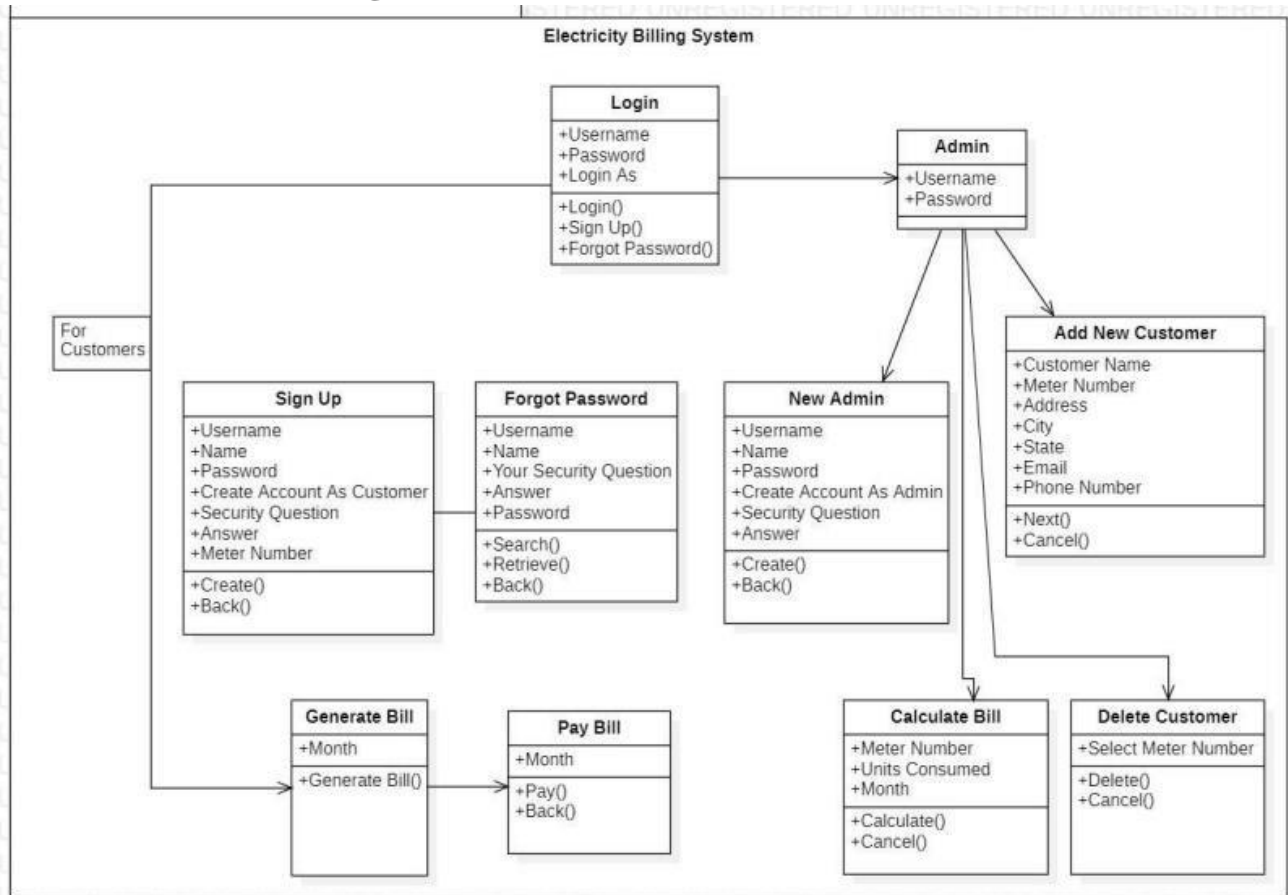- **Performance:** Capable of handling a large number of transactions.

- **3.2 Hardware and Software Requirements** ○ **Hardware Requirements**

  - Processor Intel Pentium V or higher

  - Clock Speed-1.7GHz or more

  - System Bus-64 bits
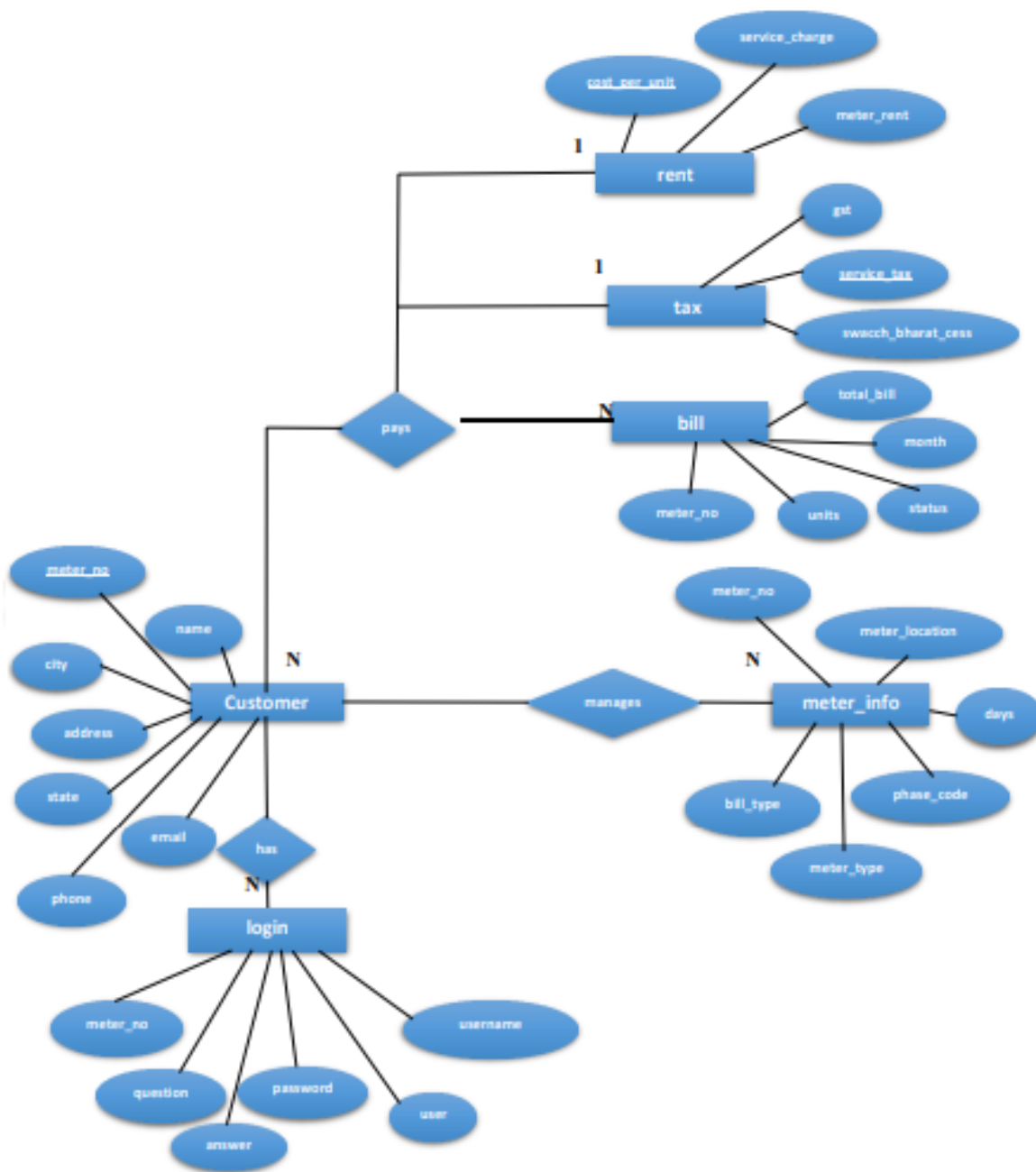
  - RAM-16GB

  - HDD-2TB ○ **Software Requirements**

  - Operating System: Windows

  - Software-Microsoft SQL Server

  - Front End-Java core/swings(NetBeans)

  - Back End-My SQL

3.3 Architecture Diagram

Electricity Billing System

Login
+Username
+Password
+Login As
+Login()
+Sign Up()
+Forgot Password()

Admin
+Username
+Password

For Customers

Sign Up
+Username
+Name
+Password
+Create Account As Customer
+Security Question
+Answer
+Meter Number
+Create()
+Back()

Forgot Password
+Username
+Name
+Your Security Question
+Answer
+Password
+Search()
+Retrieve()
+Back()

New Admin
+Username
+Name
+Password
+Create Account As Admin
+Security Question
+Answer
+Create()
+Back()

Add New Customer
+Customer Name
+Meter Number
+Address
+City
+State
+Email
+Phone Number
+Next()
+Cancel()

Generate Bill
+Month
+Generate Bill()

Pay Bill
+Month
+Pay()
+Back()

Calculate Bill
+Meter Number
+Units Consumed
+Month
+Calculate()
+Cancel()

Delete Customer
+Select Meter Number
+Delete()
+Cancel()

3.4 Entity Relationship Diagram

3.6 Normalization:
Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.
Let's discuss about anomalies first then we will discuss normal forms with examples. Anomalies in DBMS There are three types of anomalies that

occur when the database is not normalized. These are –Insertion, update and deletion anomaly.

*__First normal form(1NF)__ As per the rule of first normal form,

☐ All rows must be unique (no duplicate rows).
• Each Cell must only contain a single value (not a list).
• Each value should be non-divisible (can't be split down further) *
   __Second normal form(2NF)__ As per the rule of second normal form, ✓ Database must be in First Normal Form.

✓   Non partial dependency-All non-prime attributes should be fully functionally dependent on the candidate key

*__Third normal form(3NF)__ As per the rule of third normal form,

✓ Database must be in First and Second Normal Form.

✓   Non transitive dependency-All fields must only be determinable by the primary/composite key, not by other keys.

4 PROGRAM CODE

1 - *Create database with in mysql*

create database ebs;

2 - *Select the database you just created*

use ebs;

3 - *Create our first Table in the selected database with name login*

create table login(meter_no varchar(20), username varchar(30), name varchar(30), password varchar(20), user varchar(20));

4 - *Create the second table to store more information of user*

create table customer(name varchar(20), meter_no varchar(20), address varchar(50), city varchar(30), state varchar(30), email varchar(40), phone varchar(20));

5 - *Create the third table to store the meter information of the user*

create table meter_info(meter_no varchar(20), meter_location varchar(20), meter_type varchar(20), phase_code varchar(20), bill_type varchar(20), days varchar(20));

6 *- Create the Tax table to store tax related information*

create table tax(cost_per_unit varchar(20), meter_rent varchar(20), service_charge varchar(20), service_tax varchar(20), swacch_bharat_cess varchar(20), fixed_tax varchar(20));

7 *- Now inset values in the tax table*

insert into tax values('9','47','22','57','6','18');

8 *- Create Bill table to store electricity bill information of the user*

create table bill(meter_no varchar(20), month varchar(30), units varchar(20), totalbill varchar(20), status varchar(20));

5. RESULTS
- Results:
  - Successful implementation of the Electricity Bill Management System with the capability to:
    - Generate accurate bills based on user consumption.
    - Display real-time payment status.
    - Generate historical data reports for users and administrators.

6. CONCLUSION
- The Electricity Bill Management System efficiently automates the entire billing process, making it easier for administrators to manage large amounts of data and for customers to access their billing information. The project achieves the core objectives of reducing errors, ensuring timely bill generation, and providing secure payment processing. Future improvements could include integration with online payment gateways, mobile support, and enhanced data visualization features for usage tracking.

REFERENCES ADD LINKS ,BOOKS,REFERED TO DEVELOP THIS PROJECT

REFERENCES

1) Silberschatz, Abraham, et al. "Database System Concepts." McGraw-Hill.

2) Database Systems: https://www.geeksforgeeks.org/dbms/

3) **IEEE Xplore** – Search for related articles on **"electricity billing systems"** or **"database management in utility billing"** on IEEE Xplore for detailed case studies and system architectures.