VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI - 590018

A DBMS Mini Project Report on

"ELECTRICITY BILL MANAGEMENT SYSTEM"

Submitted in the partial fulfilment of the requirements for V Sem of the Degree of

Bachelor of Engineering in CSE (Data Science)

of Visvesvaraya Technological University, Belagavi

Submitted by:

GOKULNATH V	**1RN21CD015**
OMKAR MURTHY P	**1RN21CD034**

Under the Guidance of:

Ms. Ancy Thomas
Assistant Professor
Dept. of CSE

ESTD : 2001

An Institute with a Difference

Department of CSE (Data Science)

(Accredited by NBA 30.06.2025)

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2023-2024

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

Department of CSE (Data Science)

(Accredited by NBA 30.06.2025)

CERTIFICATE

Certified that the mini project work entitled has been successfully carried out by **GOKULNATH. V** bearing USN **1RN21CD015** and **OMKAR MURTHY. P** bearing USN **1RN21CD034** , bonafide students of **RNS Institute of Technology**, in partial fulfilment of the requirements for the 5th semester of **Bachelor of Engineering in CSE (Data Science) of Visvesvaraya Technological University, Belagavi**, during the academic year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the DBMS laboratory requirements of 5th semester B.E, CSE (Data Science).

Signature of the Guide	Signature of the HoD	Signature of the Principal
Ms. Ancy Thomas	**Dr. H S Mohan**	**Dr. Ramesh Babu H S**
Assistant Professor	Professor and HOD	Principal
Dept. of CSE	Dept. of CSE (Data Science)	

Name of the Examiners	Signature with Date
1.	
2.	

Acknowledgement

Any achievement does not depend solely on individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. Several personalities, in their capacities, have helped us to carry out this project work. We want to take this opportunity to thank them all.

We would like to profoundly thank the **Management of RNS Institute of Technology** for providing a healthy environment for the successful completion of this project work.

We are grateful to our **Director Dr. M K Venkatesha**, and **Principal Dr. Ramesh Babu H S,** RNSIT, Bangalore, for their support towards the completion of this mini project.

We would like to thank **Dr. Mohan H S**, **Professor & Head**, **Department of CSE (Data Science),** RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Ms. Ancy Thomas**, Assistant Professor, Department of CSE, RNSIT, Bangalore, for her able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of Department of CSE (Data Science), RNSIT, Bengaluru-98 for their constant support and encouragement.

Abstract

The modern world, with its increasing reliance on electrical energy for both domestic and industrial purposes, demands efficient and user-friendly systems for managing electricity consumption and billing. This project presents the development of an innovative Electricity Bill Management System (EBMS), designed to streamline the process of recording, calculating, and managing electricity bills through the utilization of Database Management Systems (DBMS). By integrating advanced DBMS technologies, the EBMS offers a comprehensive solution that enhances data integrity, security, and accessibility, while also providing a seamless interface for users and administrators.

The core objectives of the EBMS project include the creation of a centralized database to store and manage customer and billing information, the implementation of automated billing calculations based on consumption data, and the provision of real-time access to billing information for customers.

Through the application of relational database management systems (RDBMS) and the incorporation of user centric design principles, the EBMS facilitates efficient data management, reduces manual errors, and enhances the transparency of the billing process. The system's architecture is designed to support scalability, allowing for future enhancements and integration with smart grid technologies.

The implementation of the EBMS in a pilot study demonstrated notable improvements in operational efficiency, accuracy in billing calculations, and customer satisfaction levels. The system's impact extends beyond simplification of the billing process, as it also provides valuable insights into consumption patterns, aids in energy conservation efforts, and supports the overall digital transformation of utility management practices.

In conclusion, the Electricity Bill Management System represents a significant advancement in the management of electricity billing, offering a scalable, secure, and user-friendly platform that meets the needs of both utility providers and consumers. The successful deployment of this system can serve as a model for future developments in the field of utility management, highlighting the critical role of DBMS in achieving operational excellence and customer satisfaction.

Contents

Chapter - 1

Introduction

Electricity Billing System is a software-based application.

i. This project aims at serving the department of electricity by computerizing thebilling system.
ii. It mainly focuses on the calculation of units consumed during the specified timeand 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 featureshave 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 customer and 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 a manual track of the users, users can pay the amount without visiting the office. Thus, it saves human efforts and resources.

The main aim of our project is to satisfy customer by saving their time by payment process, maintaining records, and allowing the customer to view his/her records and permitting them to update their details.

The firm handles all the work manually, which is very tedious and mismatched. The objectives of our project are as follows:

- To keep the information of consuming unit energy of current month.
- To keep the information of Customer.
- To keep the information of consuming unit energy of previous month.

- To calculate the units consumed every month regularly.
- To generate the bills adding penalty and rent.
- To save the time by implementing payment process online.

1.1 Problem Statement

The manual system is suffering from a series of drawbacks. Since whole of the bills is to be maintained with hands the process of keeping and maintaining the information is very tedious and lengthy to customer. It is very time consuming and laborious process because, staff need to be visited the customers place every month to give the bills and to receive the payments. For this reason, we have provided features Present system is partially automated(computerized), existing system is quite laboriousas one must enter same information at different places.

1.2 Proposed Solution

- This project system excludes the need of maintaining paper electricity bill as allthe electricity bill records are managed electronically.
- Administrator doesn't have to keep a manual track of the users. The system automatically calculates fine.
- Users don't have to visit to the office for bill payment.
- There is no need of delivery boy for delivery bills to user's place.
- Thus, it saves human efforts and resources.

Chapter - 2

Software Requirements Specifications

The Software Requirements Specifications (SRS) outline the necessary hardware and software components, as well as the functional and non-functional requirements for developing the Electricity Bill Management System. These specifications serve as a guideline for designing, implementing, and testing the system to ensure it meets the desired objectives and user expectations.

2.1. Hardware Requirements:

- The system requires a standard computer system capable of running XAMPP and NetBeans IDE smoothly.
- A minimum of 2GB RAM is recommended for optimal performance.
- Adequate disk space is necessary to install XAMPP and NetBeans IDE, ensuring sufficient storage for project files and databases.

2.2. Software Requirements:

- **XAMPP**: This software package provides Apache, MySQL, PHP, and Perl services necessary for web development and database management.
- **NetBeans IDE**: An integrated development environment used for Java development, providing tools and features for coding, debugging, and deploying Java applications.
- **MySQL**: A powerful relational database management system utilized for storing and managing data in the Electricity Bill Management System.
- **JDBC Connectivity**: Java Database Connectivity is essential for establishing connections between Java applications and MySQL databases, enabling seamless data retrieval and manipulation.

2.3. Functional Requirements:

- **User Authentication**: Implement a secure login system where users can authenticate using their unique meter numbers and passwords.
- **Customer Management**: Provide functionalities to add, update, and delete customer information, including names, addresses, contact details, etc.
- **Meter Information**: Capture and manage details about meters such as location, type, phase code, and billing type to ensure accurate billing.
- **Tax Management**: Maintain tax information, including cost per unit, meter rent, service charges, service tax, and fixed tax to calculate accurate bills.

- **Billing**: Generate monthly bills based o meter readings, applying appropriate taxes, and calculating total charges accurately.

- **Payment Tracking**: Track the status of bills, indicating whether they are paid, unpaid, or pending, to facilitate efficient payment processing.

- **Reporting**: Generate comprehensive reports for billing history, tax calculations, customer details, etc., to provide insights and facilitate decision-making.

2.4. Non-Functional Requirements:

- **Security**: Implement robust security measures to protect sensitive data, including hashing passwords and using secure authentication mechanisms to prevent unauthorized access.

- **Performance**: Ensure that the system can handle multiple concurrent users efficiently without experiencing significant performance degradation, providing a seamless user experience.

- **Scalability**: Design the system to be scalable, capable of accommodating a growing number of users and data volume without compromising performance or functionality.

- **User Interface**: Develop an intuitive and user-friendly interface with clear navigation and interactive elements to enhance user experience and usability.

- **Reliability**: Ensure the system's reliability by implementing regular backups, error handling mechanisms, and ensuring data integrity to prevent data loss or corruption.

CHAPTER 3

PROJECT DESIGN

3.1 Preliminary Design

System design is an abstract representation of a system component and their relationship and which describe the aggregated functionally and performance of thesystem. It is also the plan or blueprint for how to obtain answer to the question being asked. The design specifies various type of approach.

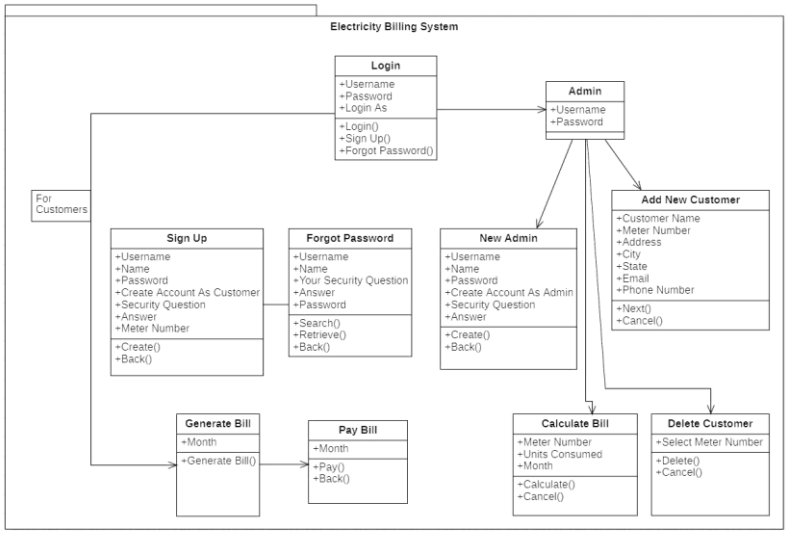


FIG 3.1: Class Diagram Of The Database

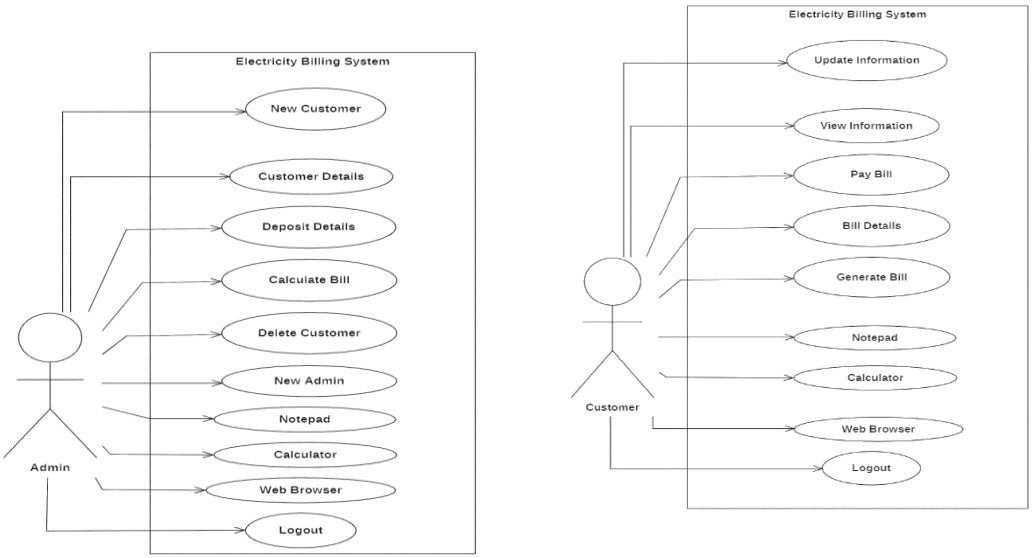


FIG 3.2: User Case Diagrams

Database design is one of the most important factors to keep in mind if you are concerned with application performance management. By designing your database to be efficient in each call it makes and to effectively create rows of data inthe database, you can reduce the amount of CPU

needed by the server to complete your request, thereby ensuring a faster application.

3.1.1 Entity-Relationship Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

There are two reasons to create a database diagram. You're either designing anew schema or you need to document our existing structure.

If you have an existing database you need to document, you create a databasediagram using data directly from your database. You can export your data base structure as a CSV file (there are some scripts on how to do this here), then have a program generate the ERD automatically.

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

- Entities, which are represented by rectangles. An entity is an object or conceptabout which you want to store information.
- A weak entity is an entity that must defined by a foreign key relationship withanother entity as it cannot be uniquely identified by its own attributes alone.
- Actions, which are represented by diamond shapes, show how two entitiesshare information in the database.
- In some cases, entities can be self-linked. For example, employees cansupervise other employees.
- Attributes, which are represented by ovals. A key attribute is the unique,distinguishing characteristic of the entity.
- A multivalued attribute can have more than one value. For example, anemployee entity can have multiple skill values.
- A derived attribute is based on another attribute. For example, an employee'smonthly salary is based on the employee's annual salary.
- Connecting lines, solid lines that connect attributes to show the relationshipsof entities in the diagram.
- Cardinality specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality.

➢ ER DIAGRAM:

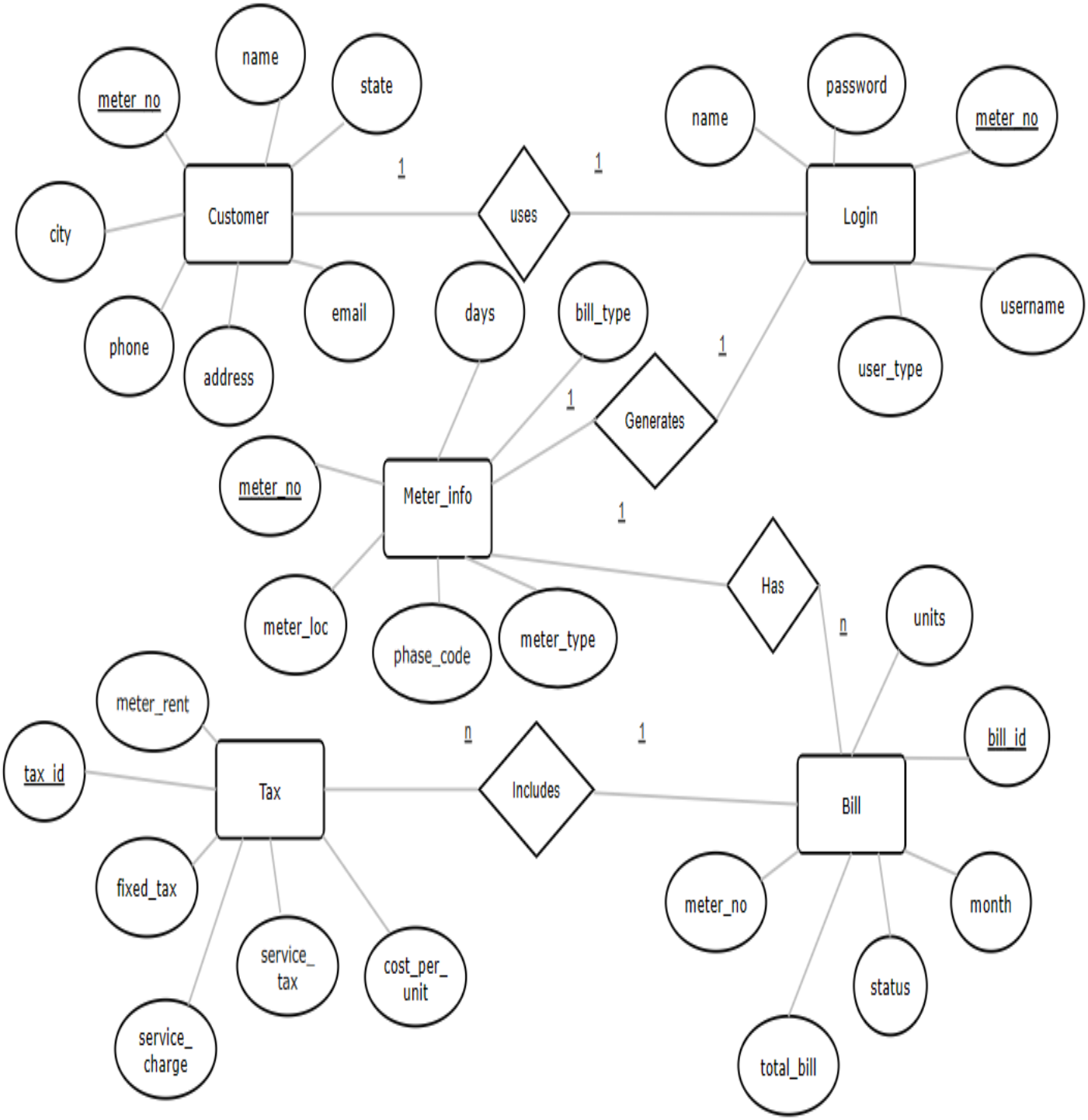


Figure 3.3: Describes the ER diagram of Electricity Billing System. It has 5 entities namely login, customer, tax, bill, and meter info. The entities have attributeswhich are primary and foreign and attributes. The primary attributes are underlined.

3.1.2 Schema Diagram

Database schema is described as database connections and constraints. It contains attributes. Every database has a state instance represent current set of databases with values. There are different types of keys in a database schema.

A primary key is a table column that can be used to uniquely identify every row of the table. Any column that has this property, these columns are called candidate key. A composite primary key is a primary key consisting of more than one column. A foreign is a column or combination of columns that contains values that are found in the primary key of some table.

All the attributes of each table are interconnected by foreign key which is primary key in another column and composite key. Primary key cannot be null. Thefact that many foreign key values repeat simply reflects the fact that its one-to-manyrelationship. In one-to-many relationship, the primary key has the one value and foreign key has many values.

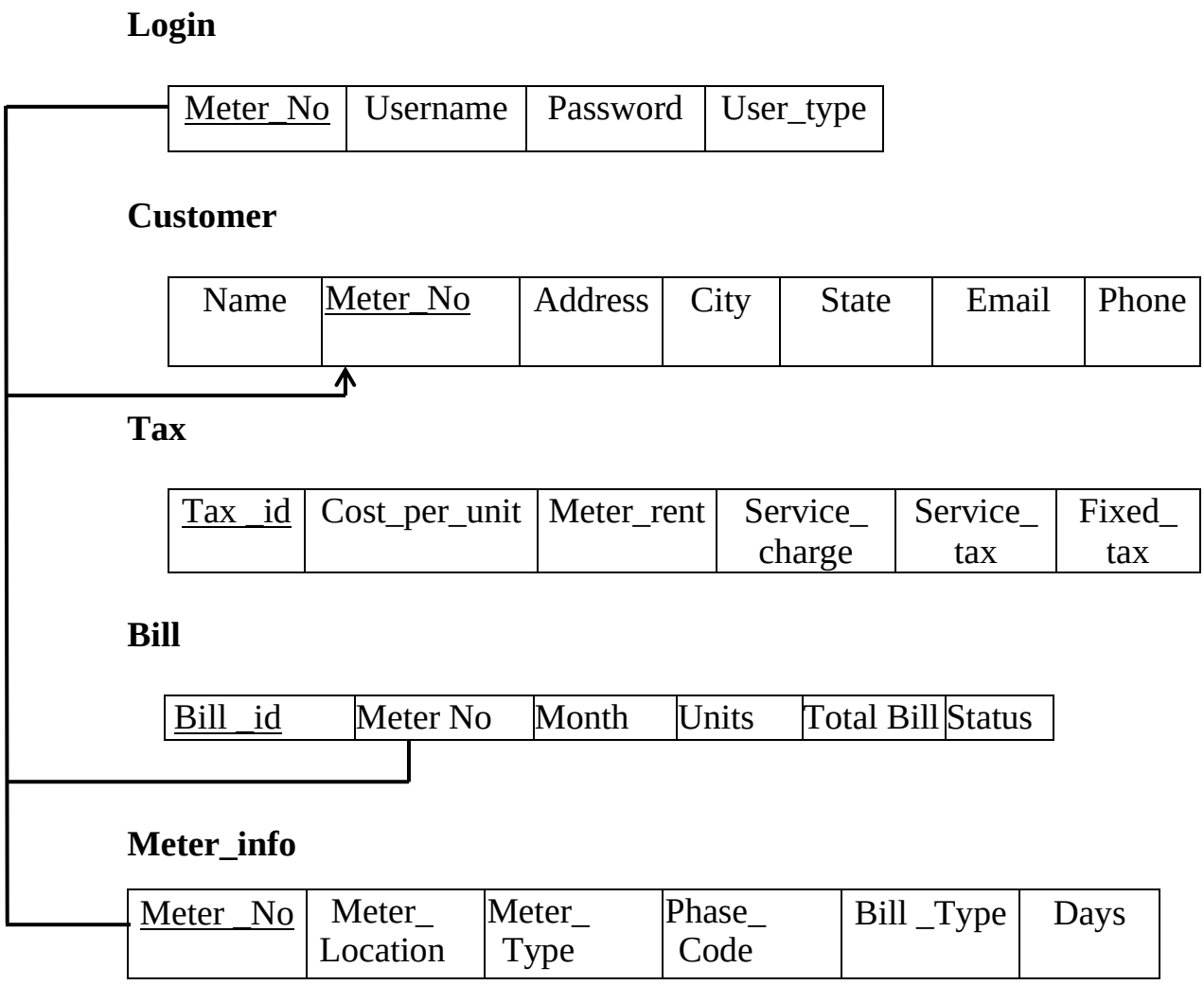


Figure 3.4: is a Schema diagram of Electricity Billing System which has sixtables i.e., login, customer, tax, rent, bill, and meter_info where each table contain attributes some with primary key, foreign key.

3.2 Normalization

Normalization is a process of organizing the data in database to avoid dataredundancy, insertion anomaly, update anomaly & deletion anomaly.

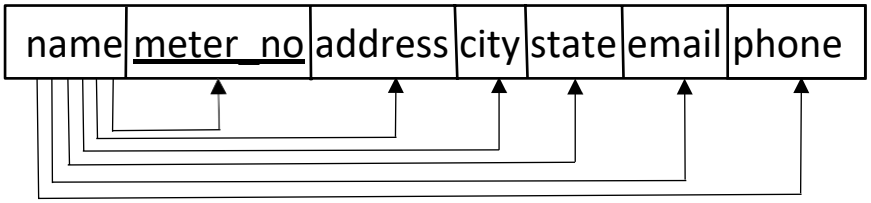
Let's discuss about anomalies first then we will discuss normal forms with examples. Anomalies in DBMS There are three types of anomalies that occur whenthe database is not normalized. These are –Insertion, update and deletion anomaly.

3.2.1 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).

Customer



3.2.2 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 functionallydependent on the candidate key.

3.2.3 Third normal form(3NF)

As per the rule of third normal form,

- ✓ Database must be in First and Second Normal Form.
- ✓ Nontransitive dependency-All fields must only be determinable by theprimary/composite key, not by other keys.

3.2.4 Normalization Analysis for Database Schema

After evaluating the structure of the database schema, we can ascertain its level of normalization, which ensures efficient data organization and minimizes redundancy. Let's break down the analysis into simpler terms suitable for reports:

1. login Table:

- Attributes: Each user's meter number (unique identifier), username, name, password, and user type.
- Each piece of information is stored distinctly, without overlapping or repetition.
- Information about each user is uniquely identified by their meter number.

2. Customer Table:

- Attributes: Customer's name, meter number (primary key), address, city, state, email, and phone number.
- Each customer's details are stored separately, avoiding any mixture of data.
- The meter number acts as a reference point, ensuring clarity and accuracy in customer identification.

3. meter_info Table:

- Attributes: Meter number (primary key), location, type, phase code, billing type, and operational days.
- Information about each meter is compartmentalized, allowing easy access and management.
- The meter number establishes a clear link to the respective customer, facilitating organized data retrieval.

4. tax Table:

- Attributes: Tax ID (unique identifier), cost per unit, meter rent, service charge, service tax, and fixed tax.
- Tax-related details are segregated into distinct fields, avoiding any overlap.
- Each tax record is uniquely identified by its Tax ID, ensuring clarity and precision in tax management.

5. bill Table:

- Attributes: Bill ID (unique identifier), meter number, month, units consumed, total bill amount, and payment status.
- Billing information is stored separately for each customer, facilitating accurate billing processes.
- The meter number serves as a reference point, ensuring seamless association with the respective customer.

- **Normalization Assessment:**

  Based on the analysis, the database schema adheres to the principles of the Third Normal Form (3NF), which is a significant milestone in ensuring data integrity and efficiency. This normalization level ensures that the database is well-structured, with minimal redundancy and optimal organization, thus supporting efficient data management and retrieval operations.

CHAPTER 4

IMPLIMENTATION

4.1 Implementation of operations

- **Adding Customer:** Here admin can add new customer to the customer listwho started using electricity bill system.
- **Searching Deposit Details:** Here admin can search according to meternumber and month to view deposit details.
- **Viewing Details**: Here admin and user can view customer details and aboutdetails.
- **Adding Tax:** Here admin can add tax details.
- **Updating Customer:** Here customer can update his/her details by using meter_no of the customer.
- **Delete Customer:** Here admin can delete details based on meter number.

4.2 Implementation of SQL statements

Insert statement:

- The INSERT INTO statement is used to insert new records in a table.
- The INSERT INTO syntax would be as follows: INSERT INTO table_nameVALUES (value1, value2, value3, ...).
- The following SQL statement insert's a new record in the “customer” table: Insert into customer VALUES (“sai”,”12345”,” btm”,” Bangalore”, “Karnataka”, “sai@gmail.com”, “9876543333”).

Update statement:

- An SQL UPDATE statement changes the data of one or more records in a table. Either all the rows can be updated, or a subset may be chosen using a condition.
- The UPDATE syntax would be as follows: UPDATE table_name SET column_name =value, column_name=value... [WHERE condition].
- The following SQL statement update's a new record in the “customer” table: UPDATE TABLE customer SET email= su@gmail.com WHERE meter_no =”12345”.

Delete statement:

- The DELETE statement is used to delete existing records in a table.

- The DELETE syntax would be as follows:

  DELETE FROM table_name WHERE condition.

- The following SQL statement delete's a record in the "customer" table: deletefrom customer where meter_no=12345.

Create statement*:*

- The CREATE TABLE Statement is used to create tables to store data.Integrity Constraints like primary key, unique key, foreign key can be definedfor the columns while creating the table.

- The syntax would be as follows: CREATETABLE table_name (column1datatype, column2datatype, column3 datatype, column N datatype,PRIMARY KEY (one or more columns)).

➢ The following SQL statement creates a table "customer" table:

```
CREATE TABLE customer(
   name VARCHAR(20),
   meter_no VARCHAR(20) PRIMARY KEY,
   address VARCHAR(50),
   city VARCHAR(30),
   state VARCHAR(30),
   email VARCHAR(40),
   phone VARCHAR(20),
   FOREIGN KEY (meter_no) REFERENCES login(meter_no)
    );
```

➢ The following SQL statement creates a table "login" table:

```
CREATE TABLE login(
   meter_no VARCHAR(20) PRIMARY KEY,
   username VARCHAR(30) UNIQUE,
   name VARCHAR(30),
   password VARCHAR(20),
   user_type VARCHAR(20)
    );
```

➢ The following SQL statement creates a table "tax" table:

```
CREATE TABLE tax(
   tax_id INT PRIMARY KEY,
   cost_per_unit VARCHAR(20),
   meter_rent VARCHAR(20),
   service_charge VARCHAR(20),
```

```
service_tax VARCHAR(20),
fixed_tax VARCHAR(20)
);
```

- The following SQL statement creates a table "bill" table:

```
CREATE TABLE bill(
bill_id INT AUTO_INCREMENT PRIMARY KEY,
meter_no VARCHAR(20),
month VARCHAR(30),
units VARCHAR(20),
totalbill VARCHAR(20),
status VARCHAR(20),
FOREIGN KEY (meter_no) REFERENCES customer(meter_no)
);
```

- The following SQL statement creates a table "meter_info" table:

```
CREATE TABLE meter_info(
meter_no VARCHAR(20) PRIMARY KEY,
meter_location VARCHAR(20),
meter_type VARCHAR(20),
phase_code VARCHAR(20),
bill_type VARCHAR(20),
days VARCHAR(20),
FOREIGN KEY (meter_no) REFERENCES customer(meter_no)

);
```

4.3 Algorithm or pseudocode of implementation

Explanation of Algorithm or pseudocode of system:

- ✓ Start system
- ✓ Enter login name and password
- ✓ On clicking the login button
- ✓ Connect to database
- ✓ Query database to know whether user credentials are correct
- ✓ If not, deny access and return login page with an error message
- ✓ If correct, check if credentials for administrator
- ✓ If yes, allow login

- ✓ Set admin session, re-direct administrator to admin login page
- ✓ If no, allow login set user session
- ✓ Re-direct user to user home page

Algorithm or pseudocode of admin:

Login:

- This program will allow the admin to enter the username and password.
- If the entered credentials are correct, then the login will be successfulotherwise need to be signup.
- If admin forgets password, it can be retrieved by giving username and answerfor security question.
- After successful login the admin will be redirected to admin portal page wherehe/she can do following activities.

New Customer:

- This program will allow the admin to enter the customer details andautomatically generates unique meter number.
- If customer name, address, city, state, email and phone number is entered,
  insert the values into customer
  else print error while next=true
  enter the meter_info details else print meter_info error
  Submit the details of customer that has been entered by clicking onto next button.
- If we need to cancel the particulars that has been entered click onto cancel option.
- If we need to submit the particulars that has been entered click onto submit option.

Customer Details:

- This program will allow the admin to view customer details.
- If we need to print the particulars that has been viewed click onto print option.

Deposit Details:

- This program will allow the admin to view bill details. If we need to sort theparticulars based on meter_no and month.
- If we need to search the particulars that has been viewed click onto searchoption.
- If we need to print the particulars that has been viewed click onto print option.

Tax Details:

- This program will allow the admin to add tax details.insert the values into tax

else print error

Submit the details of tax that has been entered by clicking onto submit button.

- If we need to cancel the particulars that has been entered click onto canceloption.

Calculate Bill:

- This program will allow the admin to calculate total_bill when units consumedare inserted where meter_no and month is selected.

  insert the values into billelse print error

  Submit the details of tax that has been entered by clicking onto submit button.
- If we need to cancel the particulars that has been entered click onto canceloption.

Algorithm or pseudocode of Customer:

Login:

- This program will allow the customer to enter the username and password. Ifthe entered credentials are correct, then the login will be successful otherwiseneed to be signup with the meter_no which is given by admin.
- If customer forgets password, it can be retrieved by giving username and answer for security question. After successful login the customer will be redirected to customer portal page where he/she can do following activities.

Update_Info:

- This program will allow the customer to update the customer details. Ifcustomer address, city, state, email and phone number is updated,

  update the values into customerelse print error

  update the details of customer that has been updated by clicking onto updatebutton.
- If we need to cancel the particulars that has been updated, click onto backoption.

View Info:

- This program will allow the customer to view his/her own details.
- If we need to go back from the particulars that has been viewed click ontoback option.

Pay Bill:

- This program will allow the customer to view bill details and redirects to pay
- the bill where status will be updated.
- If we need to cancel the particulars that has been viewed click onto backoption.

- If we need to pay the bill amount that has been viewed click onto pay option.

Bill Details:

- This program will allow the customer to view bill details.
- If we need to print the particulars that has been viewed click onto print option.

Generate Bill:

- This program will allow the customer to generate bill when meter_no andmonth is selected.
- Generate the details by clicking on generate-bill button.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Tables:

The given below table is a snapshot of backend view of the localhost and thestructures of the tables present in Electricity Billing System. The tables present are login, customer, tax, bill, meter_info.

- ✓ The login is used to store the details of login's admin and customer withmeter_no.
- ✓ The customer is used to store details of customer.
- ✓ The tax is used to store tax values.
- ✓ The bill is used to store details of bill of meter.
- ✓ The meter_info is used to store information of meter placed.

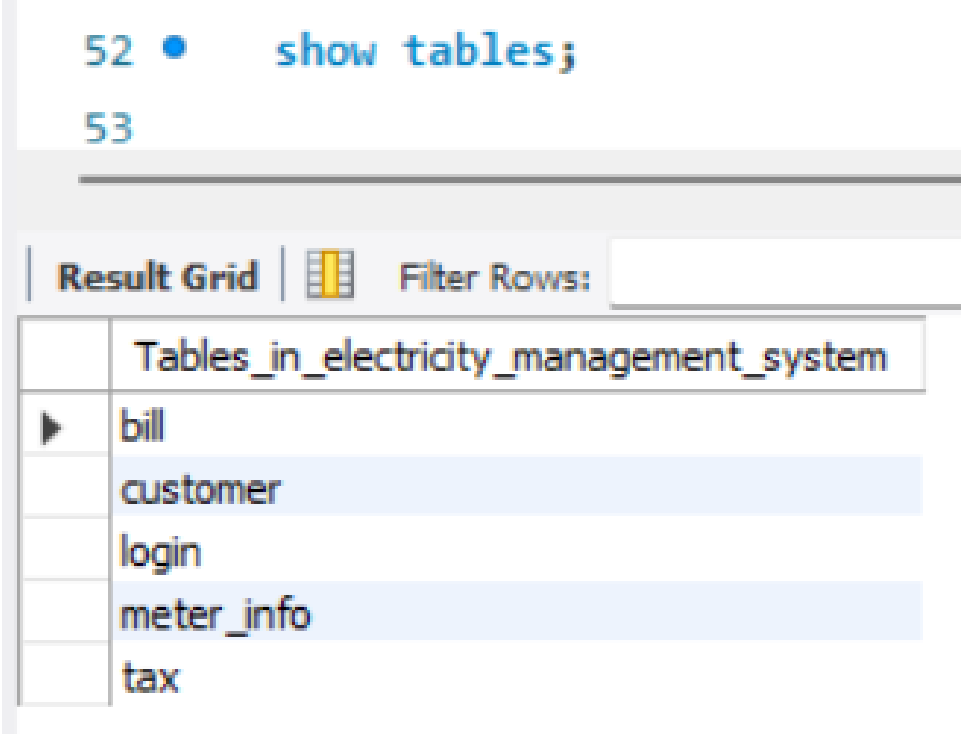


FIG 5.1:List of tables

Login Table:

```
53 •    desc login;
54
```

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
meter_no	varchar(20)	NO	PRI	NULL	
username	varchar(30)	YES	UNI	NULL	
name	varchar(30)	YES		NULL	
password	varchar(20)	YES		NULL	
user_type	varchar(20)	YES		NULL	

FIG 5.2:login table description

Customer Table:

```
54 •    desc customer;
55
```

Result Grid | Filter Rows: | Export: | Wr

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
meter_no	varchar(20)	NO	PRI	NULL	
address	varchar(50)	YES		NULL	
city	varchar(30)	YES		NULL	
state	varchar(30)	YES		NULL	
email	varchar(40)	YES		NULL	
phone	varchar(20)	YES		NULL	

FIG 5.3: customer table description

Tax Table:

```
57 •    desc tax;
58
```

Result Grid | Filter Rows: | Export: | Wrap C

Field	Type	Null	Key	Default	Extra
tax_id	int	NO	PRI	NULL	
cost_per_unit	varchar(20)	YES		NULL	
meter_rent	varchar(20)	YES		NULL	
service_charge	varchar(20)	YES		NULL	
service_tax	varchar(20)	YES		NULL	
fixed_tax	varchar(20)	YES		NULL	

FIG 5.4: tax table description

Bill Table:

```
55 •    desc bill;
56
```

Result Grid | Filter Rows: | Export: | Wrap Cell Con

Field	Type	Null	Key	Default	Extra
bill_id	int	NO	PRI	NULL	auto_increment
meter_no	varchar(20)	YES	MUL	NULL	
month	varchar(30)	YES		NULL	
units	varchar(20)	YES		NULL	
totalbill	varchar(20)	YES		NULL	
status	varchar(20)	YES		NULL	

FIG 5.5: bill table description

Meter_Info Table:

```
56 •    desc meter_info;
57
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

Field	Type	Null	Key	Default	Extra
meter_no	varchar(20)	NO	PRI	NULL	
meter_location	varchar(20)	YES		NULL	
meter_type	varchar(20)	YES		NULL	
phase_code	varchar(20)	YES		NULL	
bill_type	varchar(20)	YES		NULL	
days	varchar(20)	YES		NULL	

FIG 5.6: meter_info table description

5.2 Snapshots:

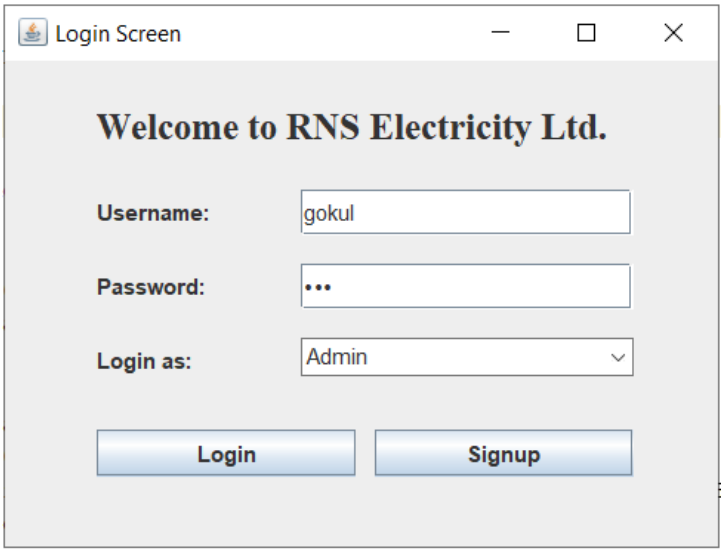


Fig 5.7 : Login Page (refers to the login table) ,Here Customer and Admin can login to their respective accounts. The dropdown menu allows to choose whether to login as admin or as a customer.

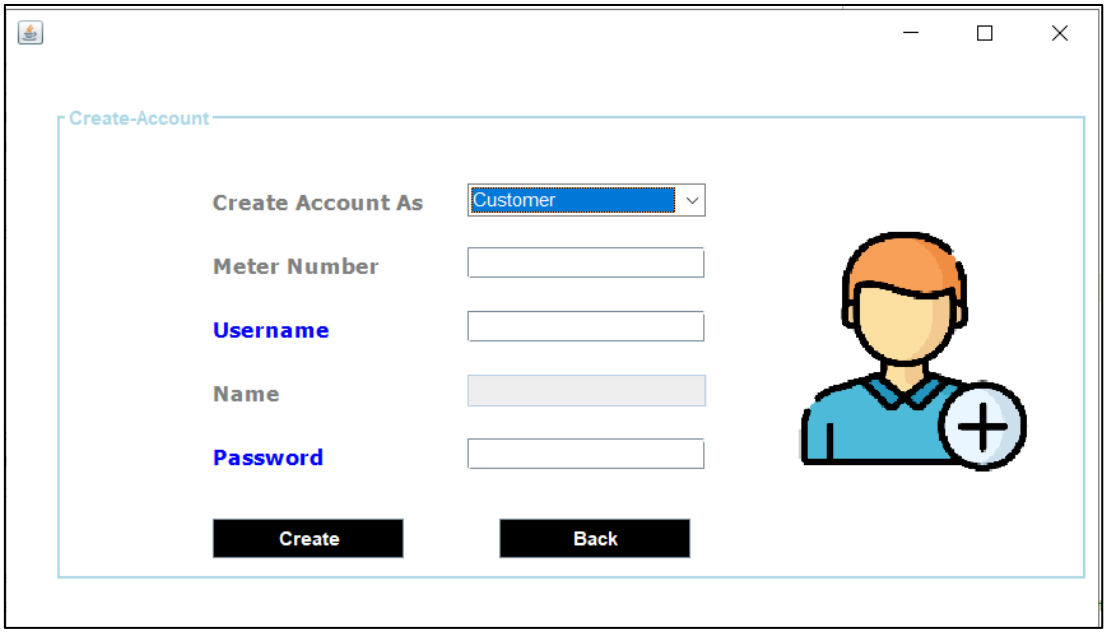


Fig 5.8 Sign Up Page, New customers will signup to access their accounts.
User have to enter username, name, password. Every user must enter their unique Meter Number

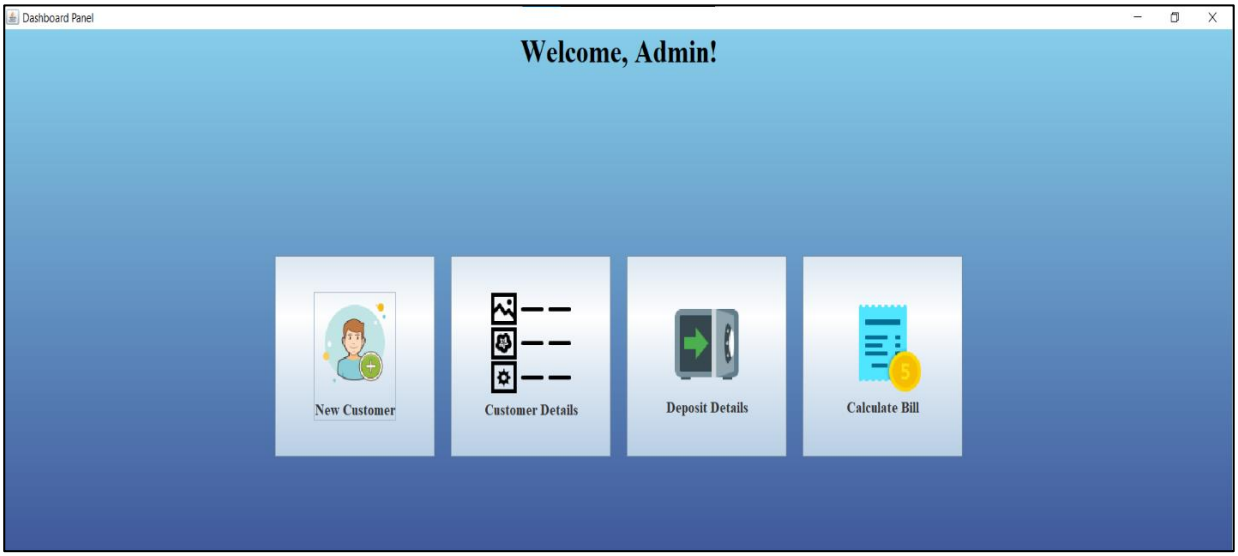


Fig 5.9 : Admin Dashboard, admin user lands on this page after successful login. Options available are add, update and delete customer and meter details

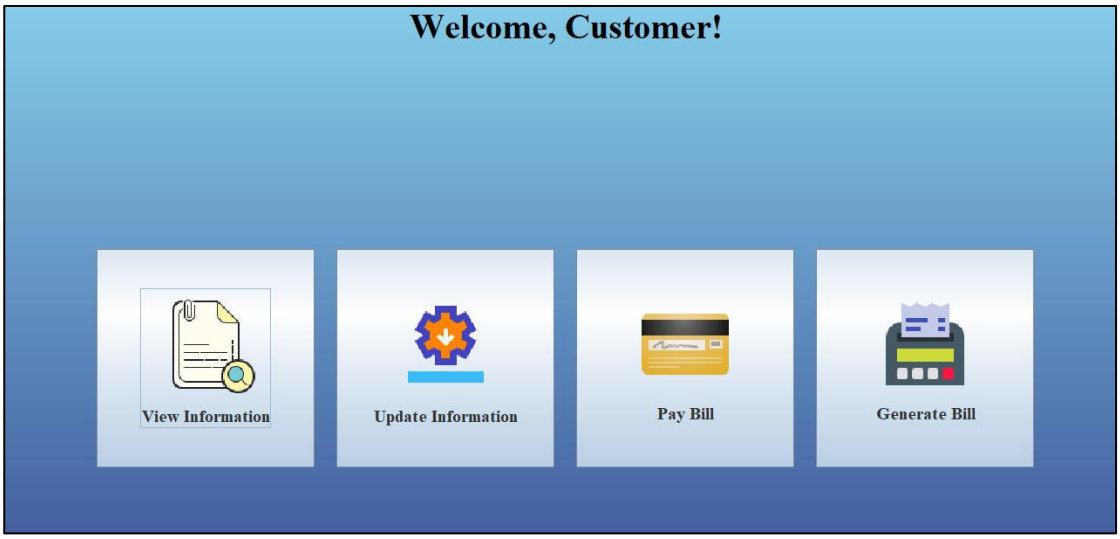
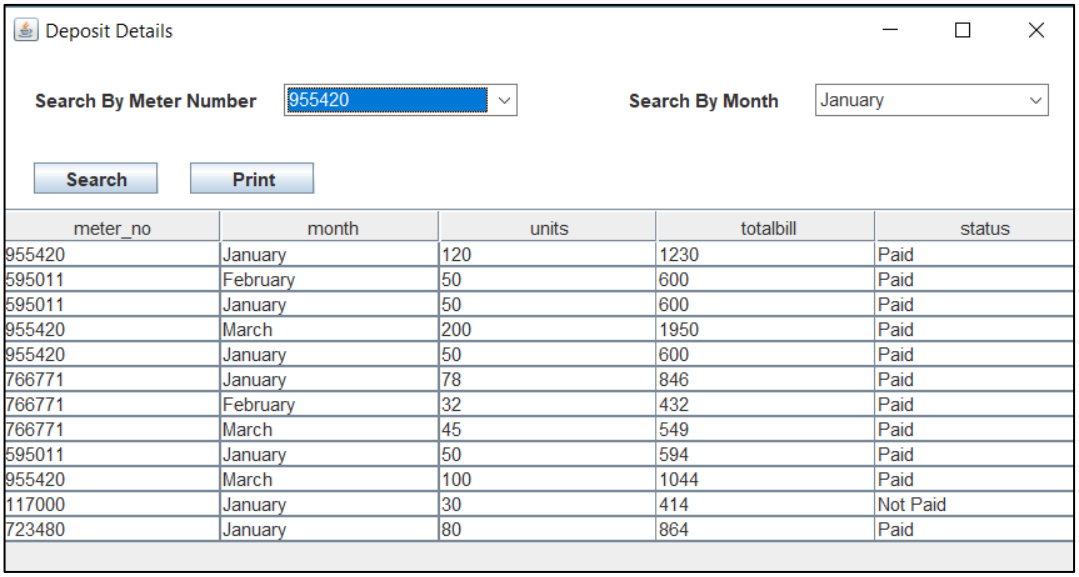


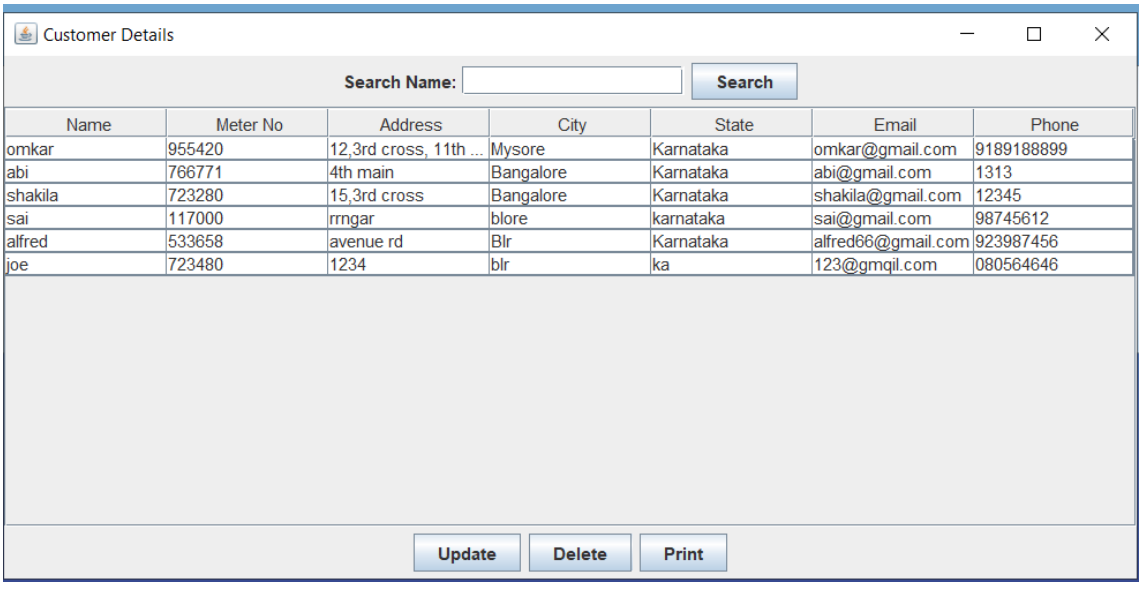
Fig 5.10 : Customer Dashboard, Customer lands on this page after successful login. Here they can view, update and pay their bills

Fig 5.11 Add new Customer page, admin can add new customers into the customer table



meter_no	month	units	totalbill	status
955420	January	120	1230	Paid
595011	February	50	600	Paid
595011	January	50	600	Paid
955420	March	200	1950	Paid
955420	January	50	600	Paid
766771	January	78	846	Paid
766771	February	32	432	Paid
766771	March	45	549	Paid
595011	January	50	594	Paid
955420	March	100	1044	Paid
117000	January	30	414	Not Paid
723480	January	80	864	Paid

Fig 5.12 : Deposit details page, displays the contents of the bill table in the database



Name	Meter No	Address	City	State	Email	Phone
omkar	955420	12,3rd cross, 11th ...	Mysore	Karnataka	omkar@gmail.com	9189188899
abi	766771	4th main	Bangalore	Karnataka	abi@gmail.com	1313
shakila	723280	15,3rd cross	Bangalore	Karnataka	shakila@gmail.com	12345
sai	117000	rrngar	blore	karnataka	sai@gmail.com	98745612
alfred	533658	avenue rd	Blr	Karnataka	alfred66@gmail.com	923987456
joe	723480	1234	blr	ka	123@gmqil.com	080564646

Fig 5.13 Customer details page, displays the contents from the customer table. Also admin can update or delete the details

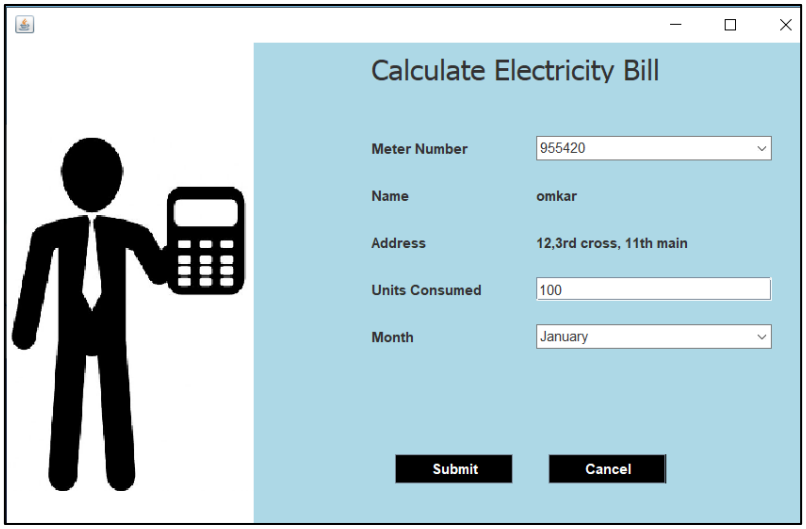


Fig 5.13 Calculate Bill page, to calculate their respective monthly bills

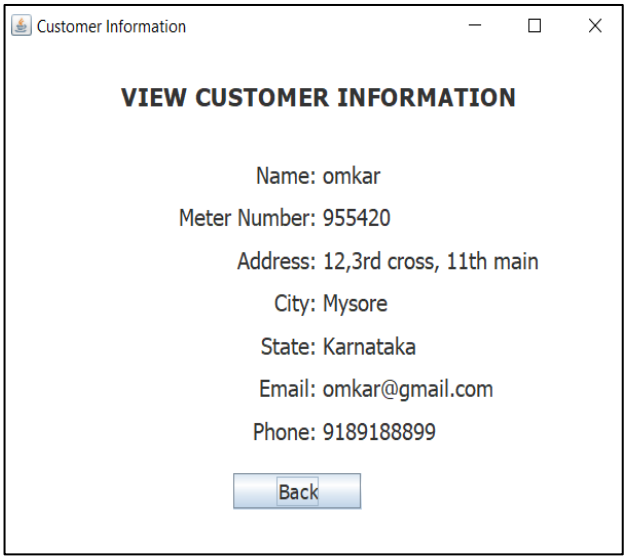


Fig 5.14 : View Details, customer can see their entered information

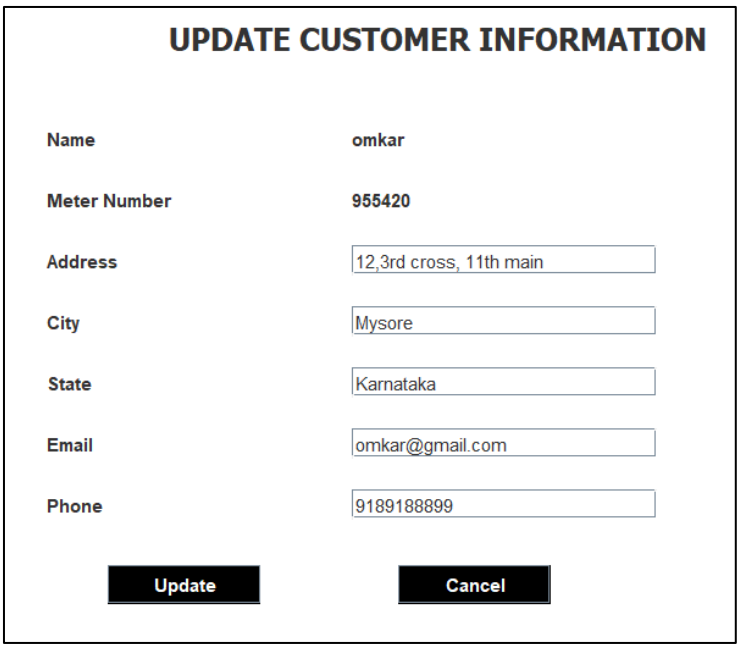


Fig 5.15 Update Details page, to update their existing info

Fig 5.16 Pay bill screen, for processing payment for selected bill

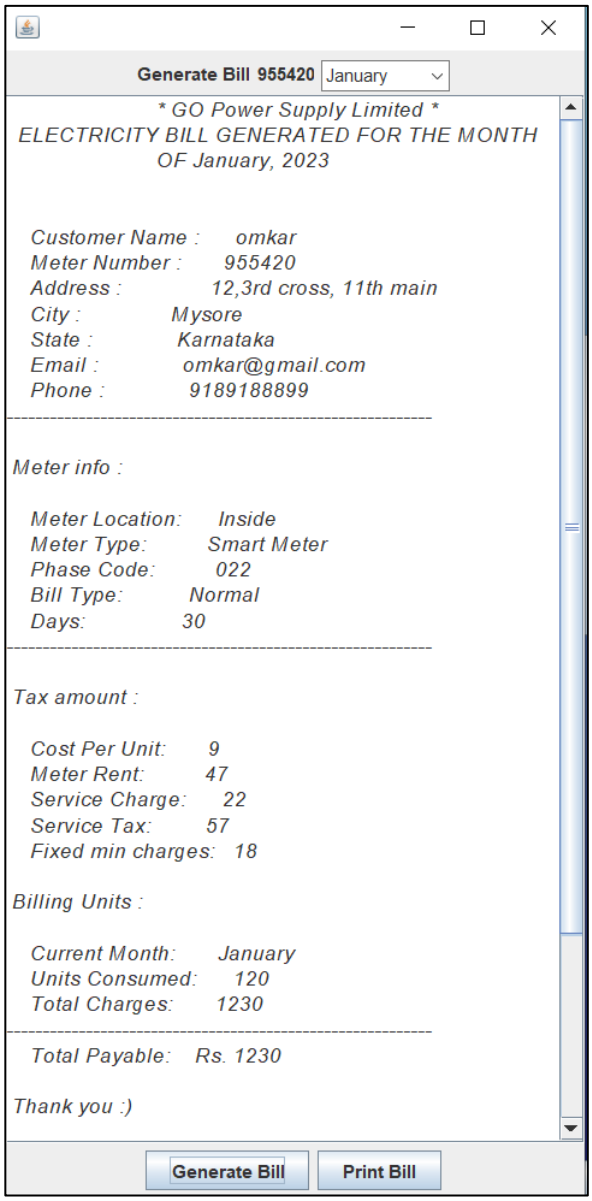


Fig 5.17 Generate bill page, fetch details from the database and generates the monthly bill

CHAPTER - 6

CONCLUSION AND FUTURE ENHANCEMENTS

In the culmination of our efforts to construct the Electricity Bill Management System mini-project, we celebrate the successful integration of Database Management Systems (DBMS) with innovative technology to create a streamlined and efficient platform for managing electricity consumption and billing processes. Through meticulous design and implementation, we have harnessed the power of SQL queries and database functionalities to achieve data accuracy, reliability, and accessibility, thus laying the foundation for effective energy management.

- **Future Enhancements:**

While our mini-project represents a commendable achievement, there are several avenues for future enhancements and optimizations. Some potential areas for improvement include:

1. Performance Optimization: Fine-tuning SQL queries and database indexing to enhance query performance and reduce latency, ensuring swift response times even under high load conditions.

2. User Interface Refinement: Iteratively refining the user interface to improve usability, accessibility, and visual appeal, thus enhancing the overall user experience.

3. Enhanced Reporting Capabilities: Expanding reporting functionalities to include more comprehensive analytics, graphical representations, and customizable reporting options for stakeholders.

4. Integration with External Systems: Integrating with external systems, such as smart meters or online payment gateways, to automate data collection, improve billing accuracy, and facilitate seamless payment transactions.

5. Data Security Measures: Implementing robust data encryption, access control mechanisms, and audit trails to enhance data security and compliance with regulatory standards.

By prioritizing these future enhancements, we can further elevate the functionality, performance, and usability of our Electricity Bill Management System mini-project, ensuring its continued relevance and effectiveness in addressing the evolving needs of users and stakeholders.

In conclusion, our journey to develop the Electricity Bill Management System mini-project has been marked by innovation, collaboration, and a relentless pursuit of excellence.

REFERENCES

[1] Ramez Elmarsi and Shamkant B. Navathe,*"Fundamentals of Database Systems"* , Pearson, 7thedition,2017.

[2] Abraham Silberschatz,Henry F. Korth, S. Sudarshan *Database System Concepts*,MC GrawHill,7th Edition, 2021

[3] Mark L. Gillenson,*Fundamentals of Database Management Systems*, 3rd Edition,Wiley,2023.

Websites

- http://www.github.com
- https://www.youtube.com/watch?v=iWitVuW2D1o&t=4s
- www.stackoverflow.com
- www.google.com
- http://www.javatpoint.com/