

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA
MADHYA PRADESH, BHOPAL - 462033**



Major Project 1

“ELECTRICITY BILLING SYSTEM”

*Submitted in the partial fulfilment of the requirements for the award of the Degree of
Bachelor of Engineering in Computer Science and Engineering*

Submitted by

Manish Kumar(0177CS201069)

Under the guidance of
Asist Prof. Aishwarya Mishra
Project Guide



**Department of Computer Science and Engineering
IES College of Technology, Bhopal
Kalkheda, Ratibad Main Road Bhopal-462044
2023-2024**

Manish Kumar(0177CS201069)
Kaif Ahmad(0177CS201062)
Md Aman(0177CS201077)
Mokhtar Alam(0177CS201096)
Arman Husain Ansari(0177CS201030)

IES College of Technology, Bhopal

Affiliated to

Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal

Department of Computer Science and Engineering



CERTIFICATE

Certified that the project work entitled “**ELECTRICITY BILLING SYSTEM**”

carried out by **Manish Kumar**,

benefited students of IES College of Technology, Bhopal in partial fulfilment for the award of the Degree of **Bachelor of Engineering in Computer Science and Engineering** of the **Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal** during the 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 academic requirements in respect of project work prescribed for the said Degree.

Prof. Aishwarya Mishra

Dr Nikhat Raza

Dr Anuprita Mishra

Project Guide

HOD Dept. of CSE

Principal, IES

External Viva

Name of the Examiners

Signature with Date

1. _____

2. _____

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IES COLLEGE OF TECHNOLOGY, BHOPAL

Kalkheda, Ratibad Main Road Bhopal-462044

(Approved by AICTE accredited by NBA, NAAC & Affiliated to RGPV)



Department Vision

To create technocrats in the field of Computer Science & Engineering, through an effective teaching-learning process to make them competent in software skills and professional ethics.

Department Mission

To provide appropriate facilities and environment for the effective teaching-learning process.

To ensure the availability of intellectual assets in terms of qualified faculty committed to developing competent students.

IES COLLEGE OF TECHNOLOGY, BHOPAL

Kalkheda, Ratibad Main Road Bhopal-462044

Affiliated to RGPV

Department of Computer Science and Engineering



DECLARATION

We, the students of seven semester B.E, in the Department of Computer Science and Engineering, **IES College of Technology, Bhopal** declare that the Project work entitled “**ELECTRICITY BILLING SYSTEM**” has been carried out by us and submitted in partial fulfilment of the course requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** discipline of **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal** during the academic year **2023-2024**. Further, the matter embodied in the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

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 Java swings as the base programming language which can be used to develop websites, web applications and web services. The Microsoft Structured Query Language (SQL) server is also used for creating back-end database. The system would be having two logins: the administrative and user login. The administrator 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.

ACKNOWLEDGEMENT

A project is a job of great enormity and it cannot be accomplished by an individual all by them. Eventually, we are grateful to several individuals whose professional guidance, assistance and encouragement have made it a pleasant endeavor to undertake this project.

It gives us great pleasure in expressing our deep sense of gratitude to our respected Chairman **Er B.S Yadav**, for having provided us with great infrastructure and well-furnished labs.

We take this opportunity to express our profound gratitude to our respected Principal **Dr Anuprita Mishra** for her support. We are grateful to the Head of the Department **Dr. Nikhat Raza**, for her unfailing encouragement and suggestion given to us during our project work.

Guidance and deadlines play a very important role in successful completion of the project on time. We also convey our gratitude to our internal project guide, **Asist Prof. Aishwarya Mishra**, for having constantly guided and the development of the project.

Finally, a note of thanks to the Department of Computer Science Engineering, both teaching and non-teaching staff for their co-operation extended to us. We thank our parents for their constant support and encouragement. Last, but not the least, we would like to thank our peers and friends.

TABLE OF CONTENTS

Abstract	i
Acknowledgment	ii
Table of contents	iii
List of figures	iv
1. Introduction	
1.1 Preamble	1-2
1.2 Problem statement	2
1.3 Proposed solution	2
2. Analysis and System Requirements	
2.1 Existing and Proposed System	3
2.2 Software & Hardware Requirements	3
3. System Design and Modelling	
3.1 Preliminary Design	
3.1.1 Entity-Relationship Diagram	4-6
3.1.2 Schema Diagram	7-8
3.2 Normalization	
3.2.1 First normal form(1NF)	9
3.2.2 Second normal form(2NF)	9
3.2.3 Third normal form(3NF)	9
4. Implementation	
4.1 Implementation of operations	10
4.2 Implementation of SQL statements	10-11
4.3 Algorithm or pseudocode of implementation	12-15
5. Testing	
5.1 Testing process	16
5.2 Testing objectives	16
5.3 Levels of Testing	
5.3.1 Unit Testing	16-18
5.3.2 Integration testing	18-19
5.3.3 System testing	19-20
6. Discussion and Snapshots	
6.1 Tables	21-23
6.2 Snapshots	23-33
Conclusion	34
Bibliography	35

LIST OF FIGURES

FIG 3.1.1: Schema diagram of Electricity Billing System	6
FIG 3.1.2: ER diagram of Electricity Billing System	8
FIG 6.1: Snapshot of List of tables	21
FIG 6.2: Snapshot login table description	21
FIG 6.3: Snapshot of customer table description	22
FIG 6.4 Snapshot of tax table description	22
FIG 6.5: Snapshot of rent table description	22
FIG 6.6: Snapshot of bill table description	22
FIG 6.7: Snapshot of meter_info table description	23
FIG 6.8: Snapshot of Splash page	23
FIG 6.9: Snapshot of Login page	24
FIG 6.10: Snapshot of Signup page	24
FIG 6.11: Snapshot of ForgotPassword page	25
FIG 6.12: Snapshot of Admin home page	25
FIG 6.13: Snapshot of New Customer page	26
FIG 6.14: Snapshot of Meter Info page	26
FIG 6.15: Snapshot of Customer Details page	27
FIG 6.16: Snapshot of Tax Details page	27
FIG 6.17: Snapshot of Calculate Bill page	28
FIG 6.18: Snapshot of Delete Customer page	28
FIG 6.19: Snapshot of Customer Home page	29
FIG 6.20: Snapshot of Update Customer Details page	29
FIG 6.21: Snapshot of View Customer Details page	30
FIG 6.22: Snapshot of View Customer Details page	30
FIG 6.23: Snapshot of PayBill page	31
FIG 6.24: Snapshot of Paytm page	31
FIG 6.25: Snapshot of Bill Details page	32
FIG 6.26: Snapshot of Generate Bill page	32
FIG 6.27: Snapshot of Deposit Bill page	33
FIG 6.28: Snapshot of About page	33

LIST OF TABLES

Table 5.1: Negative test case for phone number insertion	17
Table 5.2: Positive test case for phone number insertion	17
Table 5.3: Negative test case for email insertion	17
Table 5.4: Positive test case for email insertion	17
Table 5.5: Negative test case for customer name insertion	18
Table 5.6: Positive test case for customer name insertion	18
Table 5.7: Test case on basis of generation of bill	18
Table 5.8: Test case on basis of Deposit Details	19
Table 5.9: Test cases for the project	iv
	19-20

Chapter 1

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 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.

1.1 Preamble

We, the owners of our project, respect all customers and make them happy with our service.

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 Customer.

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

1.2 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 laborious as one must enter same information at different places.

1.3 Proposed Solution

- This project system excludes the need of maintaining paper electricity bill as all the electricity bill records are managed electronically.
- Administrator does not have to keep a manual track of the users. The system automatically calculates fine.
- Users do not 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

ANALYSIS AND SYSTEM REQUIREMENT

2.1 Existing and Proposed System

The conventional system of electricity billing is not so effective; one staff must visit each customer's house to note the meter readings and collect the data. Then, another staff must compute the consumed units and calculate the money to be paid. Again, the bills prepared are to be delivered to customers. Finally, individual customer must go to electricity office to pay their dues.

Hence, the conventional electricity billing system is uneconomical, requires many staffs to do simple jobs and is a lengthy process overall. In order to solve this lengthy process of billing, a web based computerized system is essential. This proposed electricity billing system project overcomes all these drawbacks with the features. It is beneficial to both consumers and the company which provides electricity.

With the new system, there is reduction in the number of staffs to be employed by the company. The working speed and performance of the software is faster with high performance which saves time. Furthermore, there is very little chance of miscalculation and being corrupted by the staffs.

2.2 Software & Hardware Requirements

Hardware Requirements:

- Hardware Specification: -Processor Intel Pentium i3 or higher
- Clock Speed: -1.7 GHz or more
- System Bus: -64 bits
- RAM: -8GB
- HDD: -1TB
- Monitor: -LCD Monitor
- Keyboard: -Standard keyboard
- Mouse: -Compatible mouse

Software Requirements:

- Operating System: -Windows 11
- Software: -Microsoft SQL Server
- Front End: -Java core/swings (NetBeans)
- Back End: -MySQL

CHAPTER 3

SYSTEM DESIGN AND MODELLING

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 the system. It is also the plan or blueprint for how to obtain answer to the question being asked. The design specifies various type of approach.

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 in the 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 a new schema or you need to document our existing structure.

If you have an existing database you need to document, you create a database diagram 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 concept about which you want to store information.
- ❖ A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

- ❖ Actions, which are represented by diamond shapes, show how two entities share information in the database.
- ❖ In some cases, entities can be self-linked. For example, employees can supervise 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, an employee entity can have multiple skill values.
- ❖ A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.
- ❖ Connecting lines, solid lines that connect attributes to show the relationships of 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.

Figure 3.1.1 describes the ER diagram of Electricity Billing System. It has 5 entities namely login, customer, tax, bill, and meter info. The entities have attributes which are primary and foreign and attributes. The primary attributes are underlined.

3.1.1 ER Diagram

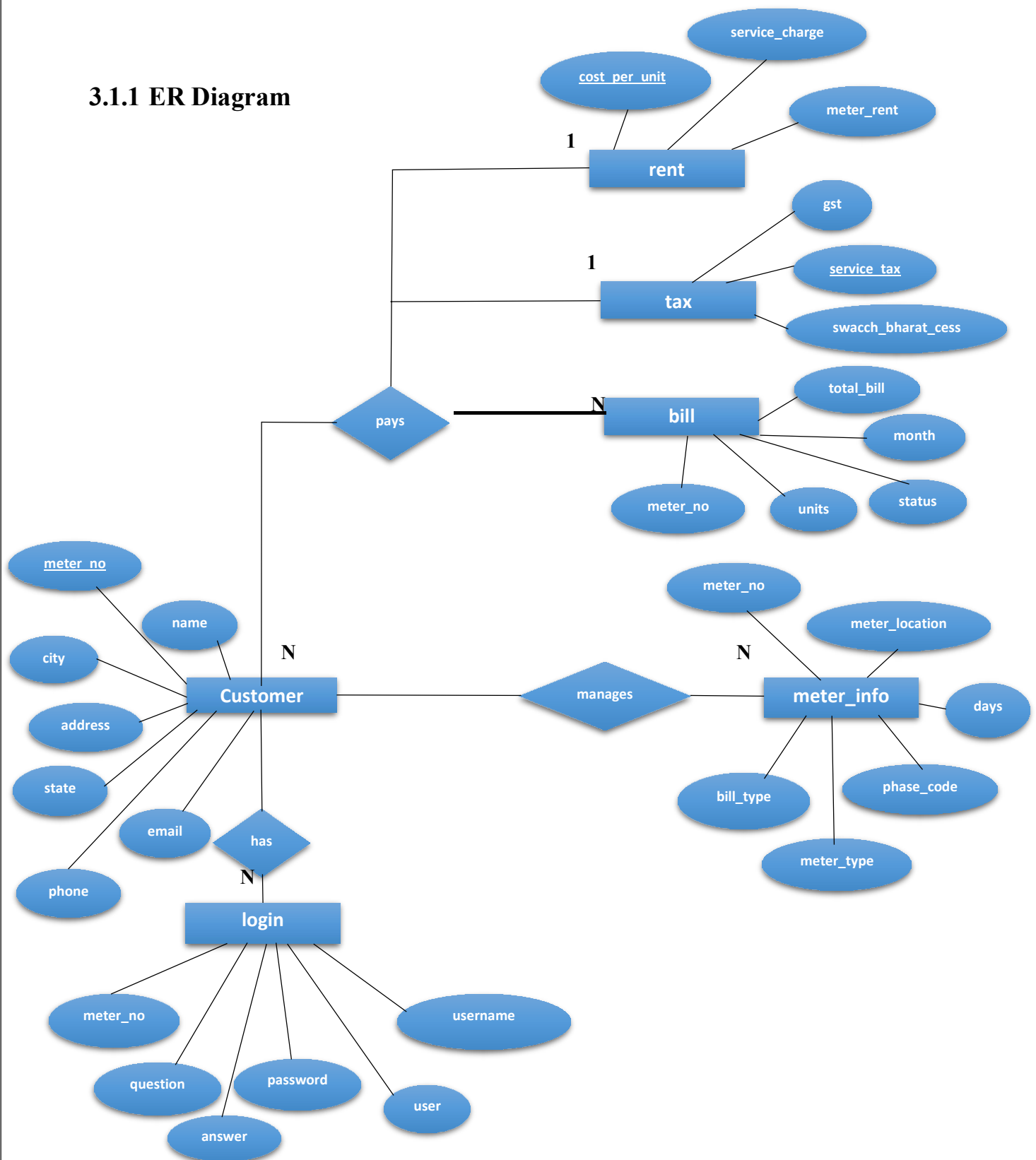


FIG 3.1.1: Schema diagram of Electricity Billing System

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. The fact that many foreign key values repeat simply reflects the fact that its one-to-many relationship. In one-to-many relationship, the primary key has the one value and foreign key has many values.

Figure 3.1.2 is a Schema diagram of Electricity Billing System which has six tables i.e., login, customer, tax, rent, bill, and meter_info where each table contain attributes some with primary key, foreign key. In the login table there are 6 attributes "meter_no", "username", "password", "user", "question", "answer". The customer table has 7 attributes "name", "meter_no"(primary key), "address", "city", "state", "email", "phone". The rent table has 3 attributes "cost_per_unit"(primary key), "meter_rent", "service_charge". The tax table has 3 attributes "service_tax", "swacch_bharat_cess", "gst". The bill table has 5 attributes "meter_no"(foreign key that references the primary key of the customer table meter_no), "month", "units", "total_bill", "status". The meter_info table has 6 attributes "meter_no"(foreign key that references the primary key of the customer table meter_no), "meter_location", "meter_type", "phase_code", "bill_type", "days".

3.1.2 Schema Diagram

Login

meter_no	Username	password	user	question	Answer
----------	----------	----------	------	----------	--------

customer

Name	<u>meter_no</u>	Address	City	state	Email	phone
------	-----------------	---------	------	-------	-------	-------

rent

<u>cost_per_unit</u>	meter_rent	service_rent
----------------------	------------	--------------

tax

<u>service_tax</u>	swacch_bharat_cess	gst
--------------------	--------------------	-----

bill

meter_no	Month	units	total_bill	Status
----------	-------	-------	------------	--------

meter_info

meter_no	meter_location	meter_type	phase_code	bill_type	Days
----------	----------------	------------	------------	-----------	------

FIG 3.1.2: Schema diagram of Electricity Billing System

3.2 Normalization

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

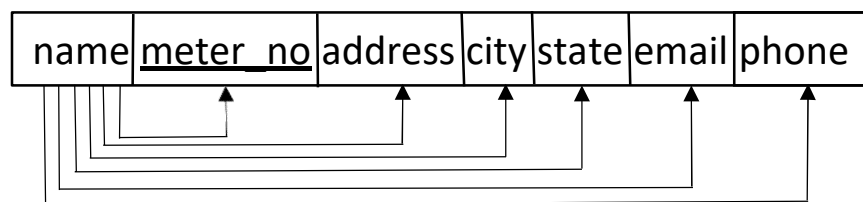
Let us discuss about anomalies first then we will discuss normal forms with examples. Anomalies in DBMS There are three types of anomalies that occur when the database is not normalized. These are –Insertion, update, and deletion anomaly.

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 functionally dependent 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 the primary/composite key, not by other keys.

CHAPTER 4

IMPLIMENTATION

4.1 Implementation of operations

- ❖ **Adding Customer:** Here admin can add new customer to the customer list who started using electricity bill system.
- ❖ **Searching Deposit Details:** Here admin can search according to meter number and month to view deposit details.
- ❖ **Viewing Details:** Here admin and user can view customer details and about details.
- ❖ **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_name VALUES (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: delete from 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 defined for the columns while creating the table.
- The syntax would be as follows: CREATETABLE table_name (column1datatype, column2datatype, column3 datatype, columnN datatype, PRIMARY KEY (one or more columns)).
 - The following SQL statement creates a table “customer” table: create table customer (name varchar (30), meter_no varchar (20) primary key, address varchar (50), city varchar (20), state varchar (30), email varchar (30), phone varchar (30));
 - The following SQL statement creates a table “login” table: create table login (meter_no varchar (30), username varchar (30), password varchar (30), user varchar (30), question varchar (40), answer varchar (30));
 - The following SQL statement creates a table “tax” table: create table tax (cost_per_unit int (20) primary key, meter_rent int (20), service_charge int (20), service_tax int (20), swacch_bharat_cess int (20), gst int (20));
 - The following SQL statement creates a table “bill” table: create table bill (meter_no varchar (20), foreign key(meter_no) references customer(meter_no) on delete cascade, month varchar (20), units int (20), total_bill int (20), status varchar (40));
 - The following SQL statement creates a table “meter_info” table: create table meter_info (meter_no varchar (30), foreign key(meter_no) references customer(meter_no) on delete cascade, meter_location varchar (10), meter_type varchar (15), phase_code int (5), bill_type varchar (10), days int (5));

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 successful otherwise need to be signup.
- If admin forgets password, it can be retrieved by giving username and answer for security question.
- After successful login the admin will be redirected to admin portal page where he/she can do following activities.

NewCustomer:

- This program will allow the admin to enter the customer details and automatically 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.

CustomerDetails:

- 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.

DepositDetails:

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

TaxDetails:

- 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 cancel option.

CalculateBill:

- This program will allow the admin to calculate total_bill when units consumed are inserted where meter_no and month is selected.
insert the values into bill
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 cancel option.

DeleteBill:

- This Program will allow the admin to delete the customer info when meter_no is selected.
- If we need to delete the particulars that has been saved click onto delete option.
- If we need to cancel the particulars that has been entered click onto back option.

About:

- This program will allow the admin to view details of the project in short.
- If we need to exit the particulars that has been viewed click onto exit option.

Algorithm or pseudocode of Customer:

Login:

- This program will allow the customer to enter the username and password. If the entered credentials are correct, then the login will be successful otherwise need to be sign up 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.

UpdateInfo1:

- This program will allow the customer to update the customer details. If customer address, city, state, email and phone number is updated, update the values into customer else print error update the details of customer that has been updated by clicking onto update button.
- If we need to cancel the particulars that has been updated, click onto back option.

ViewInfo:

- 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 onto back option.

PayBill:

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

BillDetails:

- 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.

GenerateBill:

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

About:

- This program will allow the customer to view details of the project in short.
- If we need to exit the particulars that has been viewed click onto exit option.

NOTE: Utility (notepad, browser, calculator), query and logout is given to both customer and admin portals.

CHAPTER 5

TESTING

This chapter gives the outline of all the testing methods that are carried out to get a bug free application.

5.1 Testing process

Testing is an integral part of software development. Testing process, in a way certifies, whether the product, that is developed, compiles with the standards, that it was designed to. Testing process involves building of test cases, against which, the product must be tested. In some cases, test cases are done based on the system requirements specified for the product/software, which is to be developed.

5.2 Testing objectives

The main objectives of testing process are as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding a yet undiscovered error.
- A successful test is one that uncovers a yet undiscovered error.

5.3 Levels of Testing

Different levels of testing are used in the testing process; each level of testing aims to test different aspects of the system. The basic levels are unit testing, integration testing, system testing and acceptance testing.

5.3.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design the module. The software built, is a collection of individual modules. In this kind of testing exact flow of control for each module was verified. With detailed design consideration used as a guide, important control paths are tested to uncover errors within the boundary of the module.

Table 5.1: Negative test case for phone number insertion

Function Name	Input	Expected Output	Error	Resolved
Input phone number	98977	Phone number is invalid	Length of phone number is not equal to 10	Consume ()
Input phone number	98977agv	Phone number is invalid	Alphabets are being taken as input for phone number	—

Table 5.2: Positive test case for phone number insertion

Function Name	Input	Expected Output	Error	Resolved
Input phone number	9897778988	Expected output is seen	—	—

Table 5.3: Negative test case for email insertion

Function Name	Input	Expected Output	Error	Resolved
Input email	Sail.in	Email is invalid	Email is not in a format given	Consume ()

Table 5.4: Positive test case for email insertion

Function Name	Input	Expected Output	Error	Resolved
Input email	aki123@gmail.com	Expected output is seen	—	—

Table 5.5: Negative test case for customer name insertion

Function Name	Input	Expected Output	Error	Resolved
Input customer name	Sana123	Name is invalid	Numbers are being taken as input for name	Consume ()

Table 5.6: Positive test case for customer name insertion

Function Name	Input	Expected Output	Error	Resolved
Input customer name	Gowthu	Expected output is seen	—	—

5.3.2 Integration testing

The second level of testing is called integration testing. In this, many class-tested modules are combined into subsystems, which are then tested. The goal here is to see if all the modules can be integrated properly. We have been identified and debugged.

Table 5.7: Test case on basis of generation of bill

Function Name	Input	Expected Output	Error	Resolved
Negative searching of total bill	12334(meter_no) January(month)	Details seen but not total bill	Output not seen	Consume ()
Positive searching of total_bill	12334(meter_no) January(month)	Must display full generated bill with total_bill	—	—

Table 5.8: Test case on basis of deposit details

Function Name	Input	Expected Output	Error	Resolved
Negative searching of depositedetails	12334(meter_no) January(month)	Details not seen	Output not seen	Consume ()
Positive searching of total_bill	12334(meter_no) January(month)	Must display depositedetails	—	—

5.3.3 System testing

Here the entire application is tested. The reference document for this process is the requirement document, and the goal is to see IF the application meets its requirements. Each module and component of ethereal was thoroughly tested to remove bugs through a system testing strategy. Test cases were generated for all possible input sequences and the output was verified for its correctness.

Table 5.9: Test cases for the project

Steps	Action	Expected output
Step1 choice	The screen appears when the users run the program. 1.If admin login 2. If customer login	A page with different menus appears. 1.Admin panel opens and 2. Customer panel opens
Step 2	The screen appears when the admin logs in and selects any one of the menus from the click of the mouse.	A window for adding new customer, inserting tax, calculate bill, view deposit details etc
Selection 1	<ul style="list-style-type: none"> ❖ New Customer ❖ Customer Details ❖ Deposit Details ❖ Calculate Bill ❖ Tax Details ❖ Delete Customer 	

Step 2.1	The screen appears when the customer login and selects any one of the menus from the click of the mouse	A window for generating bill, update customer details, view details, generating bill
Selection 2	❖ Update Details ❖ View Details	
Selection 2a	❖ Generate Bill	
Selection 2b	❖ Pay Bill ❖ Bill Details	

CHAPTER 6

DISCUSSION AND SNAPSHOTS

6.1 TABLES:

The given below table is a snapshot of backend view of the localhost and the structures 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 with meter_no.
- ✓ The customer is used to store details of customer.
- ✓ The tax is used to store tax values.
- ✓ The rent is used to store rent values.
- ✓ The bill is used to store details of bill of meter.
- ✓ The meter_info is used to store information of meter placed.

```
mysql> show tables;
+-----+
| Tables_in_ebs |
+-----+
| bill           |
| customer       |
| login          |
| meter_info     |
| tax            |
+-----+
```

FIG 6.1: List of tables

Login Table:

```
mysql> desc login;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| meter_no   | varchar(30)   | YES  |     | NULL    |       |
| username   | varchar(30)   | YES  |     | NULL    |       |
| password    | varchar(30)   | YES  |     | NULL    |       |
| user       | varchar(30)   | YES  |     | NULL    |       |
| question   | varchar(40)   | YES  |     | NULL    |       |
| answer     | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

FIG 6.2:login table description

Customer Table:

```
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name | varchar(30) | YES | | NULL | |
| meter_no | varchar(20) | NO | PRI | NULL | |
| address | varchar(50) | YES | | NULL | |
| city | varchar(20) | YES | | NULL | |
| state | varchar(30) | YES | | NULL | |
| email | varchar(30) | YES | | NULL | |
| phone | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

FIG 6.3: customer table description**Tax Table:**

```
mysql> desc tax;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| service_tax | int | NO | PRI | NULL | |
| swacch_bharat_cess | int | YES | | NULL | |
| gst | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

FIG 6.4: tax table description**Rent Table:**

```
mysql> desc rent;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cost_per_unit | int | NO | PRI | NULL | |
| meter_rent | int | YES | | NULL | |
| service_charge | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

FIG 6.5: rent table description**Bill Table:**

```
mysql> desc bill;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| meter_no | varchar(20) | YES | MUL | NULL | |
| month | varchar(20) | YES | | NULL | |
| units | int | YES | | NULL | |
| total_bill | int | YES | | NULL | |
| status | varchar(40) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

FIG 6.6: bill table description

meter_info Table:

```
mysql> desc meter_info;
```

Field	Type	Null	Key	Default	Extra
meter_no	varchar(30)	YES	MUL	NULL	
meter_location	varchar(10)	YES		NULL	
meter_type	varchar(15)	YES		NULL	
phase_code	int	YES		NULL	
bill_type	varchar(10)	YES		NULL	
days	int	YES		NULL	

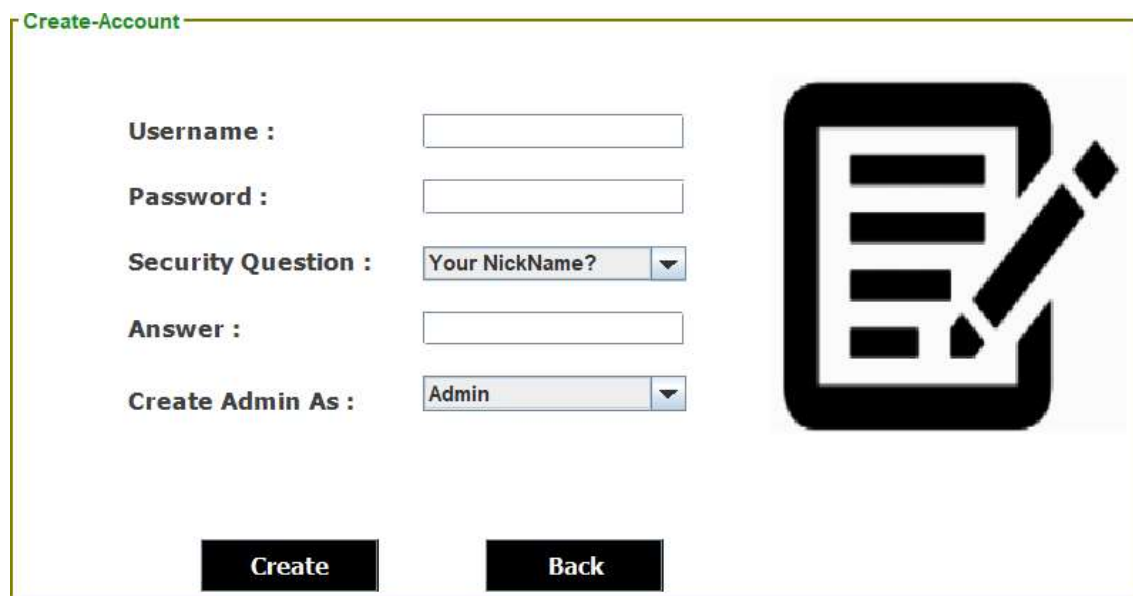
6 rows in set (0.00 sec)

FIG 6.7: meter_info table description**6.2 SNAPSHOTS:****FIG 6.8: Splash page of Electricity Billing System**



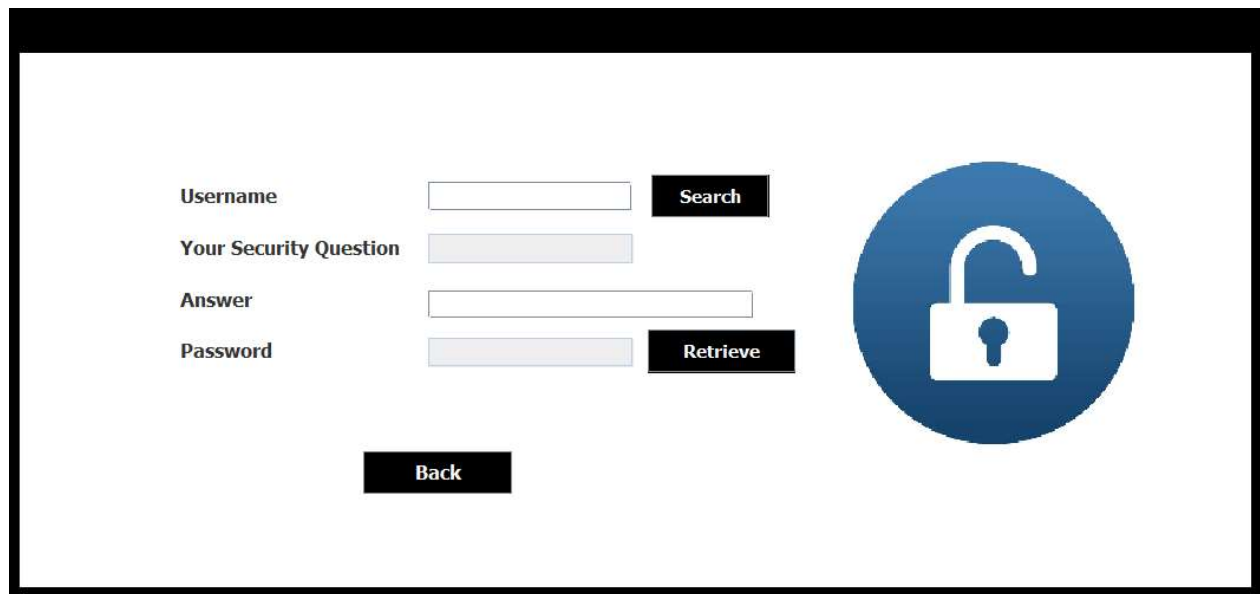
The screenshot shows a web browser window titled "Login Page". On the left is a silhouette of a person in a suit. To the right are three input fields: "Username", "Password", and "Logging in as" (a dropdown menu showing "Admin"). Below these fields are three buttons: "Login" (with a key icon), "Cancel" (with a red X icon), and "Signup" (with a document icon). At the bottom is a "Forgot Password" button (with a lock icon).

FIG 6.9: Login page



The screenshot shows a web browser window titled "Create-Account". On the left are five input fields: "Username :", "Password :", "Security Question :" (a dropdown menu showing "Your NickName?"), "Answer :", and "Create Admin As :" (a dropdown menu showing "Admin"). To the right is a large icon of a document with a pencil. At the bottom are two buttons: "Create" and "Back".

FIG 6.10: Signup page

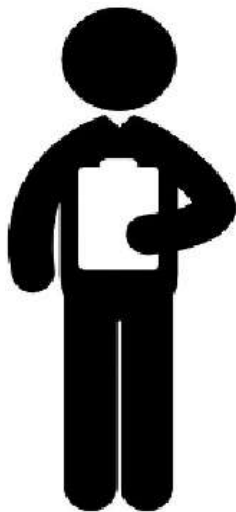


The image shows a web form for a 'Forgot Password' page. It is enclosed in a black rectangular frame. On the left side, there are four labels with corresponding input fields: 'Username' with a white text box, 'Your Security Question' with a light gray text box, 'Answer' with a white text box, and 'Password' with a light gray text box. To the right of the 'Username' field is a black button labeled 'Search'. To the right of the 'Password' field is a black button labeled 'Retrieve'. Below the 'Password' field and to the left of the 'Retrieve' button is a black button labeled 'Back'. On the right side of the form, there is a large blue circular icon containing a white padlock with a keyhole, indicating a security or login-related function.

FIG 6.11: Forgot Password page



FIG 6.12: Admin home page



New Customer

Customer Name

Meter No 673692

Address

City

State

Email

Phone Number

FIG 6.13: New customer page

Meter Information

Meter Number 904764

Meter Location

Meter Type

Phase Code

Bill Type

Days 30

Note: By default Bill is calculated for 30 days only



A small inset image on the left side of the form shows a hand holding a clipboard with a meter reading. The meter is a digital display showing '10000'. The hand is holding a pen and appears to be writing on the clipboard.

FIG 6.14: Meter Info page

Customer Details						
name	meter_no	address	city	state	email	phone
Manish Kumar	407362	Ashoka Garden	Bhopal	Madhya Pradesh	manish@gmail.com	7632078885
Noman	462382	Khawaspur	Siwan	Bihar	noman@yahoo.com	9973829227
md	853313	bhopal	bhopal	mp	mdshamshadali134@gmail.com	9905844190

FIG 6.15: Customer Details page

TAX DETAILS



Cost Per Unit

Meter Rent


Service Charge

Service Tax

Swacch_Bharat_Cess

GST

FIG 6.16: Tax Details page



Calculate Electricity Bill

Meter Number	407362
Name	Manish Kumar
Address	Ashoka Garden
Units Consumed	100
Month	January


Submit **Cancel**

FIG 6.17: Calculate Bill page

DeleteDetails Page

DELETE CUSTOMER DETAILS

Meter_no : 413098



Delete **Back**

FIG 6.18: Delete Customer page



FIG 6.19: Customer Home page

A screenshot of a web application window titled "Update Customer Information". The window has a red header bar with standard window controls. The form contains the following fields and values:

Field	Value
Name	Manish Kumar
Meter Number	407362
Address	Ashoka Garden
City	Bhopal
State	Madhya Pradesh
Email	manish@gmail.com
Phone	7632078885

At the bottom of the form are two buttons: "Update" and "Cancel". To the right of the form is a large black icon representing a customer profile, featuring a white silhouette of a person's head and shoulders next to a white document icon with three horizontal lines.

FIG 6.20: Update customer details page

VIEW CUSTOMER INFORMATION

Name	Manish Kumar	State	Madhya Pradesh
Meter Number	407362	Email	manish@gmail.c...
Address	Ashoka Garden	Phone	7632078885
City	Bhopal		

Cancel





FIG 6.21: View Customer Details page

[illegible]

FIG 6.22: Query page

— □ ×

Electricity Bill

Meter Number	407362
Name	Manish Kumar
Month	<div>December ▾</div>
Units	80
Total Bill	870
Status	Not Paid

Pay

Back

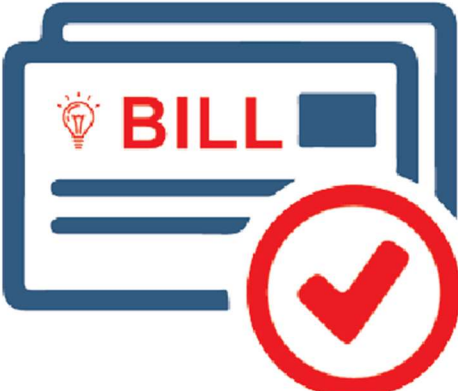


FIG 6.23: Pay Bill page



FIG 6.24: Paytm page

meter_no	month	units	totalbill	status
407362	Octorber	120	1230	paid
407362	December	80	870	Not Paid
407362	March	76	834	Not Paid
407362	August	120	1230	Not Paid

FIG 6.25: Bill Details page

Generate Bill 407362 December

Adani Power Limited
ELECTRICITY BILL GENERATED FOR THE MONTH
OF December, 2023

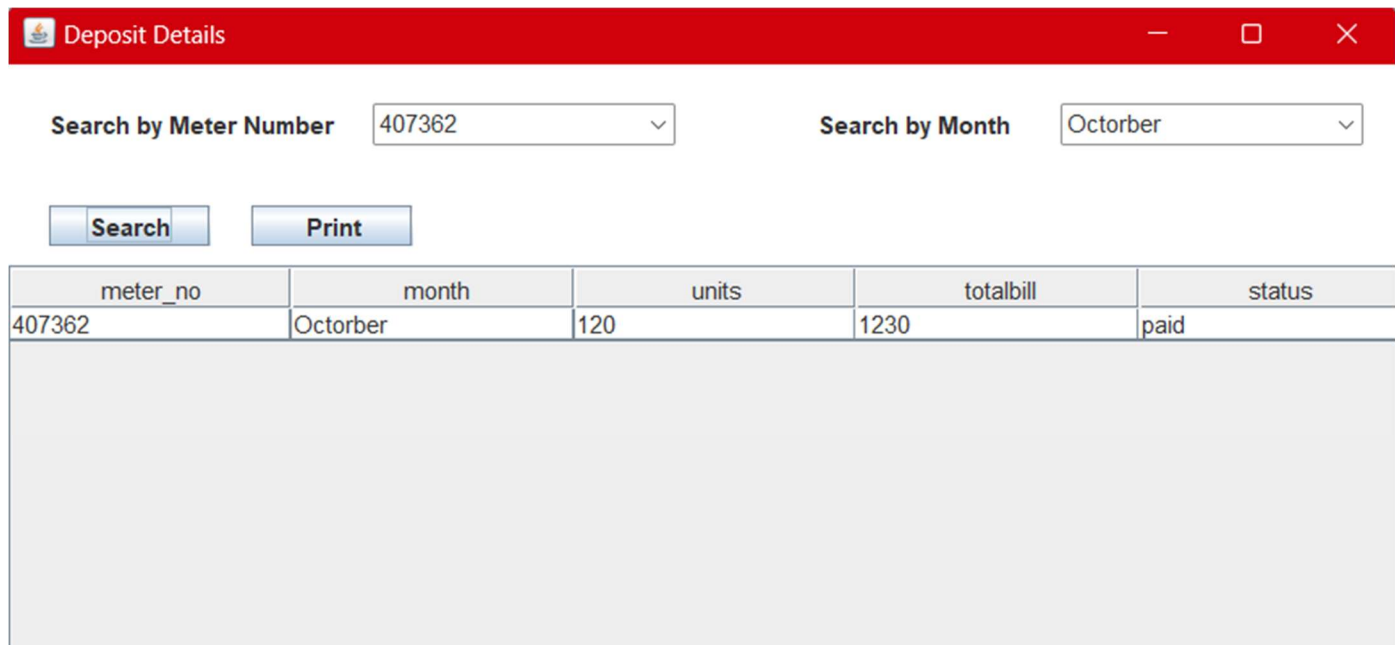
Customer Name: Manish Kumar
Meter Number: 407362
Address: Ashoka Garden
City: Bhopal
State: Madhya Pradesh
Email: manish@gmail.com
Phone: 7632078885

Meter Location: Inside
Meter Type: Smart Meter
Phase code: 055
Bill Type: Normal
Days: 30

Cost Per Unit: 9
Meter Rent: 47
Service Charge: 22
Service Tax: 57
Swacch Bharat Cess: 6
Fixed Tax: 18

Generate Bill

FIG 6.26: Generate Bill page



The image shows a Java Swing window titled "Deposit Details" with a red header bar. It contains two search criteria: "Search by Meter Number" with a dropdown menu showing "407362" and "Search by Month" with a dropdown menu showing "October". Below these are "Search" and "Print" buttons. A table displays the search results with columns: meter_no, month, units, totalbill, and status. The table has one row with the values: 407362, October, 120, 1230, and paid. The rest of the window area is a large, empty light gray rectangle.

meter_no	month	units	totalbill	status
407362	October	120	1230	paid

FIG 6.27: Deposit Details page

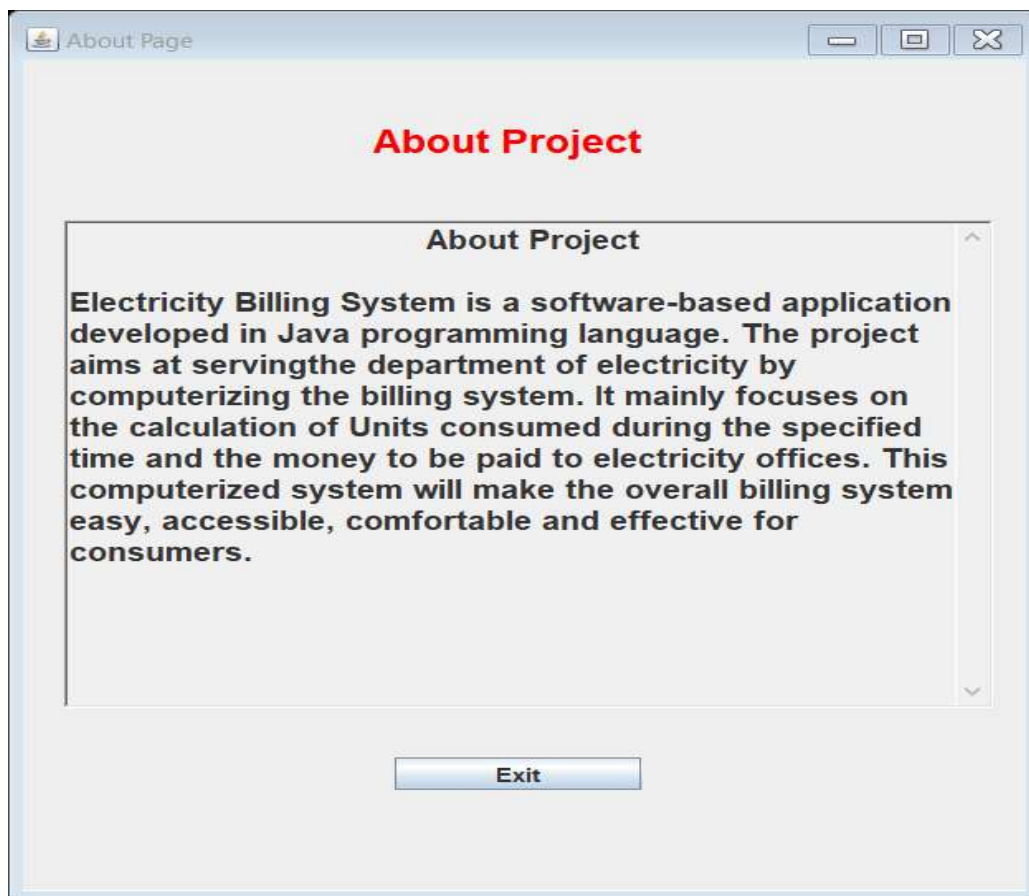


FIG 6.28: About page

CONCLUSION

After all the hard work is done for electricity bill management system is here. It is a software which helps the user to work with the billing cycles, paying bills, managing different DETAILS under which are working etc.

This software reduces the amount of manual data entry and gives greater efficiency. The User Interface of it is very friendly and can be easily used by anyone.

It also decreases the amount of time taken to write details and other modules.

BIBLIOGRAPHY

REFERENCES

Book Reference

Database System Concepts 5th Edition by S. Sudarshan
(TEXTBOOK).

Websites

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