

# TABLE OF CONTENTS

1	Introduction.....	4
1.1	Preamble.....	4
1.2	Problem statement.....	5
1.3	Proposed solution.....	5
2	Analysis and System Requirements.....	6
2.1	Existing and Proposed System .....	6
2.2	Software & Hardware Requirements.....	6
3	System Design and Modelling.....	7
3.1	Preliminary Design.....	7
3.2	Entity-Relationship Diagram.....	7
3.3	Schema Diagram.....	9
3.4	Normalization.....	10
3.5	First normal form(1NF).....	10
3.6	Second normal form(2NF).....	10
3.7	Third normal form(3NF).....	10
4	Implementation.....	11
4.1	Implementation of operations.....	11
4.2	Implementation of SQL statements.....	11
4.3	Algorithm or pseudocode of implementation .....	12
5	Testing.....	16
5.1	Testing process.....	16
5.2	Testing objectives.....	16
5.3	Levels of Testing.....	16
5.4	Unit Testing.....	16
5.5	Integration testing.....	18
5.6	System testing.....	19
6	Discussion and Snapshots.....	21
6.1	Tables.....	21
6.2	Snapshots.....	24
7	Conclusion.....	36

## LIST OF FIGURES

<b>Sr. No.</b>	<b>Name of Figures</b>	<b>Page No.</b>
3.1	Schema diagram of Electricity Billing System	8
3.2	ER diagram of Electricity Billing System	10
6.1	Snapshot of List of tables	21
6.2	Snapshot login table description	22
6.3	Snapshot of customer table description	22
6.4	Snapshot of tax table description	22
6.5	Snapshot of rent table description	23
6.6	Snapshot of bill table description	23
6.7	Snapshot of meter info table description	23
6.8	Snapshot of Splash page	24
6.9	Snapshot of Login page	24
6.10	Snapshot of Signup page	25
6.11	Snapshot of Forgot Password page	25
6.12	Snapshot of Admin home page	26
6.13	Snapshot of New Customer page	26
6.14	Snapshot of Meter Info page	27
6.15	Snapshot of Customer Details page	27
6.16	Snapshot of Tax Details page	28
6.17	Snapshot of Calculate Bill page	28
6.18	Snapshot of Delete Customer page	29
6.19	Snapshot of Customer Home page	29
6.20	Snapshot of Update Customer Details page	30
6.21	Snapshot of View Customer Details page	30
6.22	Snapshot of View Customer Details page	31
6.23	Snapshot of Pay Bill page	31
6.24	Snapshot of Paytm page	32
6.25	Snapshot of Bill Details page	32
6.26	Snapshot of Generate Bill page	33
6.27	Snapshot of Deposit Bill page	34
6.28	Snapshot of about page	35

## LIST OF TABLES

<b>Sr. No.</b>	<b>Name of Table</b>	<b>Page No.</b>
5.1	Negative test case for phone number insertion	17
5.2	Positive test case for phone number insertion	17
5.3	Negative test case for email insertion	17
5.4	Positive test case for email insertion	17
5.5	Negative test case for customer name insertion	18
5.6	Positive test case for customer name insertion	18
5.7	Test case on basis of generation of bill	18
5.8	Test case on basis of Deposit Details	19
5.9	Test cases for the project	20

# CHAPTER 1

## 1.0 INTRODUCTION

Electricity Billing System is a software-based application.

- This 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 charged by the electricity offices.
- 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, and 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 Propose Solution**

- This project system excludes the need of maintaining paper electricity bill as all the 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**

### **ANALYSIS AND SYSTEM REQUIREMENT**

#### **2.1 Existing and Propose 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 V or higher
- Clock Speed: -1.7 GHz or more
- System Bus: -64 bits
- RAM: -16GB
- HDD: -2TB
- Monitor: -LCD Monitor
- Keyboard: -Standard keyboard

##### **Software Requirements:**

- Mouse: -Compatible mouse
- Operating System: -Windows 10
- Software: -Microsoft SQL Server
- Front End: -Java core/swings (NetBeans)
- Back End: -My SQL

# 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 functionality 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.

### 3.2 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 database 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 be 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.

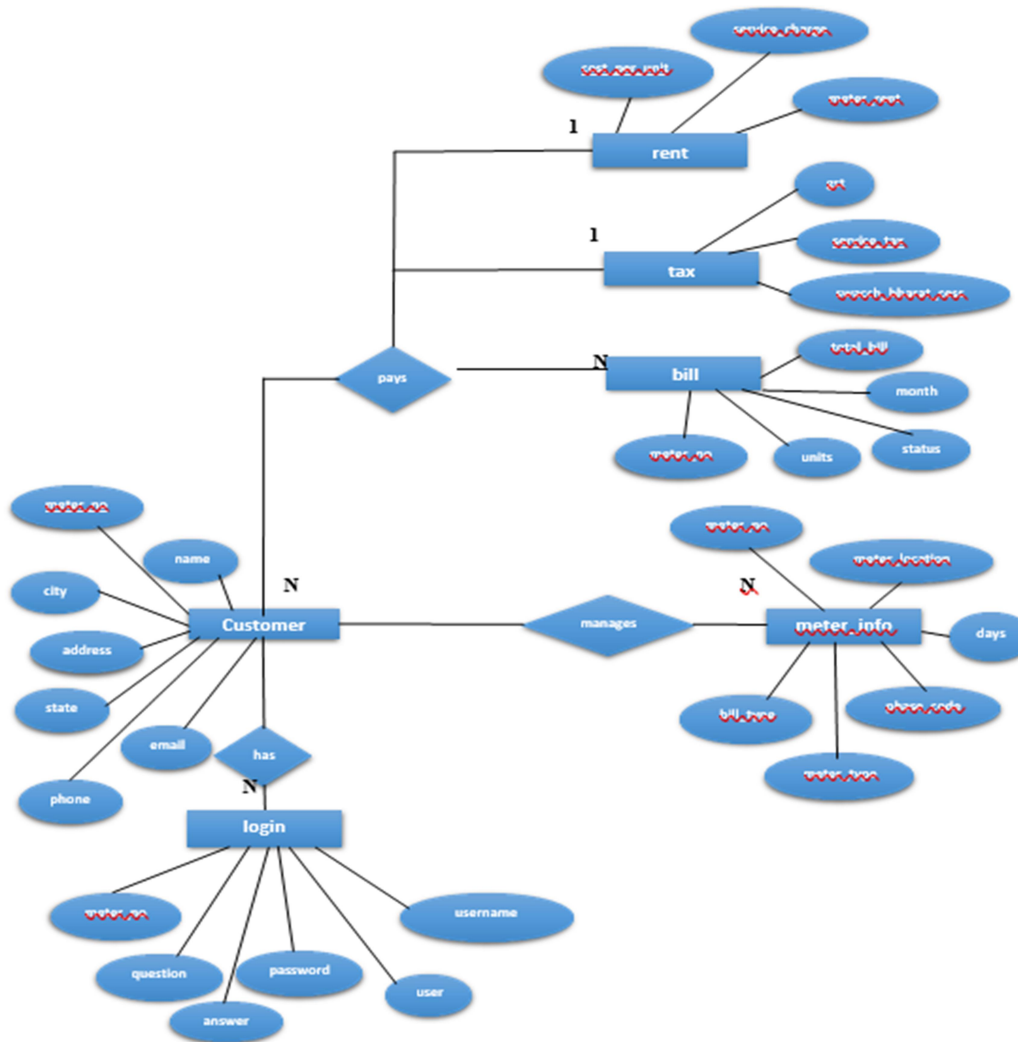


Fig 3.1: Schema diagram of Electricity Billing System



### 3.3 Schema Diagram

Database schema is described as database connections and constraints. It contains attributes. Every database has state instances 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 information where each table contain attributes some with primary key, foreign key. In the login table there are 6 attributes "meter number", "username", "password", "user", "question", "answer". The customer table has 7 attributes "name", "meter number"(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", "Swachh Bharat chess", "gst". The bill table has 5 attributes "meter number"(foreign key that references the primary key of the customer table meter number), "month", "units", "total bill", "status". The meter information table has 6 attributes "meter number"(foreign key that references the primary key of the customer table meter number), "meter location", "meter type", "phase code", "bill type", "days".

#### 3.3 Schema Diagram Login

Meter number	Username	password	User	question	Answer
--------------	----------	----------	------	----------	--------

#### Customer

Name	meter no	Address	City	state	Email	phone
------	----------	---------	------	-------	-------	-------

#### Rent

Cost per unit	meter rent	service rent
---------------	------------	--------------

#### tax

service tax	Swacch Baharat chess	gst
-------------	----------------------	-----

## Bill

Moter number	Month	units	Total bill	status
--------------	-------	-------	------------	--------

## Meter information

Meter number	Meter location	Meter type	Zip code	Bill type	
--------------	----------------	------------	----------	-----------	--

Fig 3.2: Schema diagram of Electricity Billing System

## 3.4 Normalization

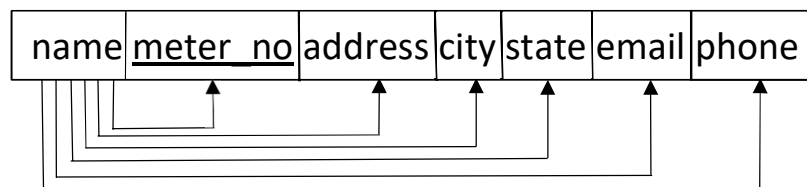
Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

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.

## 3.5 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



## 3.6 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.

## 3.7 Third normal

- 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 number of the customer.
- **Delete Customer:** Here admin can delete details based on meter number.

#### 4.2 Implementation of 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 ("sonu","12345"," bet"," 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 number ="12345".
- **Delete Statement:**
- The DELETE statement is used to delete existing records in a table.
- The DELETE syntax would be as: DELETE FROM table name WHERE condition.

- The following SQL statement delete's a record in the “customer” table: delete from customer where meter number=12345.
- Create Statement
- The CREATE TABLE Statement is used to create tables to store data. Integrity Constraints like primary key, unique key, and 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, column N datatype, PRIMARY KEY (one or more columns)).
- The following SQL statement creates a table “customer” table: createtable customer (name varchar (30), meter number 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 number 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 tabletax (cost per unit int (20) primary key, meter rent int (20), service charge int (20), service tax int (20), Swachh Bharat cess int(20), get int (20));
- The following SQL statement creates a table “bill” table: create table bill (meter number varchar (20), foreign key (meter number) references customer (meter number) on delete cascade, month varchar (20), units int (20), total bill int (20), status varchar (40));
- The following SQL statement creates a table “meter information” table: create table meter information (meter number varchar (30), foreign key (meter number)references customer (meter number) 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.
- New Customer:
- 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 information details else print meter information error
- Submit the details of customer that has been entered by clicking onto next button.
- If we need to cancel the particulars that have 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 have been viewed click onto print option.
- Deposit Details:
- This program will allow the admin to view bill details. If we need to sort the particulars based on meter number and month.
- If we need to search the particulars that have been viewed click onto search option.
- If we need to print the particulars that have 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 have been entered click onto canceloption.
- Calculate bill:
- This program will allow the admin to calculate total bill when units consumed are inserted where meter number 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 have been entered click onto canceloption.
- Delete Bill:
- This Program will allow the admin to delete the customer info when meter number is elected.
- If we need to delete the particulars that has been saved click onto deleteoption.
- If we need to cancel the particulars that have been entered click onto backoption.
- About:
- This program will allow the admin to view details of the project in short.
- If we need to exit the particulars that have 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 number 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 Information:
- 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 have been updated, click onto backoption.
- View Information:
- 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.
- 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 have 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 have been viewed click onto print option.
- Generate Bill:
- This program will allow the customer to generate bill when meter number and month is selected.
- Generate the details by clicking on generate bill button.
- About:
- This program will allow the customer to view details of the project in short.
- If we need to exit the particulars that have been viewed click onto exit option.
- NOTE: Utility (notepad, browser, and calculator), query and logout is given to bothcustomer 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, complies with the standards, that it was designed to. Testing process involves building of test cases, against which, the product has to 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 an as yet undiscovered error.
- A successful test is one that uncovers an as 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.

### 5.4 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.



FunctionName	Input	ExpectedOutput	Error	Resolved
Input phonenumber	98977	Phone number is invalid	Length of phone number is not equal to 10	Consume ()
Input phonenumber	98977	Phone number is invalid	Alphabets are being taken as input for phone number	-----

Table 5.1: Negation test case for phone number insertion

FunctionName	Input	Expected Output	Error	Resolved
Input phonenumber	9897778988	Expected output is seen	-----	-----

Table 5.2: Positive test case for phone number insertion

FunctionName	Input	ExpectedOutput	Error	Resolved
Input email	Sai1.in	Email is invalid	Email is not in a format given	Consume ()

Table 5.3: Negative test case for email insertion

FunctionName	Input	ExpectedOutput	Error	Resolved
Input email	aki123@gmail.com	Expected output is seen	-----	-----

Table 5.4: Positive test case for email insertion

FunctionName	Input	ExpectedOutput	Error	Resolved
Input customername	Sana123	Name isinvalid	Numbers are being taken asinput for name	Consume ()

Table 5.5: Negative test case for customer name insertion

FunctionName	Input	ExpectedOutput	Error	Resolved
Input customer name	Growth	Expected output is seen	-----	-----

Table 5.6: Positive test case for customer name insertion

## 5.5 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 havebeen identified and debugged.

FunctionName	Input	Expected Output	Error	Resolved
Negative searching total bill	12334(meter no) January(month)	Details seenbut not total bill	put notseen	Consume ()
Positive searching of total bill	12334(meter no) January(month)	Must display full generated bill with Total bill	-----	-----

Table 5.7: test case on basis of generation of bill

FunctionName	Input	ExpectedOutput	Error	Resolved
Negative searching of deposit details	2334(meter no) January(month)	ails notseen	put notseen	Consume ()
Positive searching of total bill	2334(meter no) January(month)	Must display deposit details	-----	-----

Table 5.8: Test case on basis of deposit details

## 5.6 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.

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 menu's 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	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	-----

Table 5.9: Test cases for the project

## 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 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_elect |
+-----+
| bill             |
| customer         |
| login            |
| meter_info       |
| rent             |
| tax              |
+-----+
6 rows in set (0.03 sec)
```

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 descriptions

### 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 descriptions

### Meter Information 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 table description

## 6.2 Snapshots



Fig 6.8 Splash page of electricity billing System

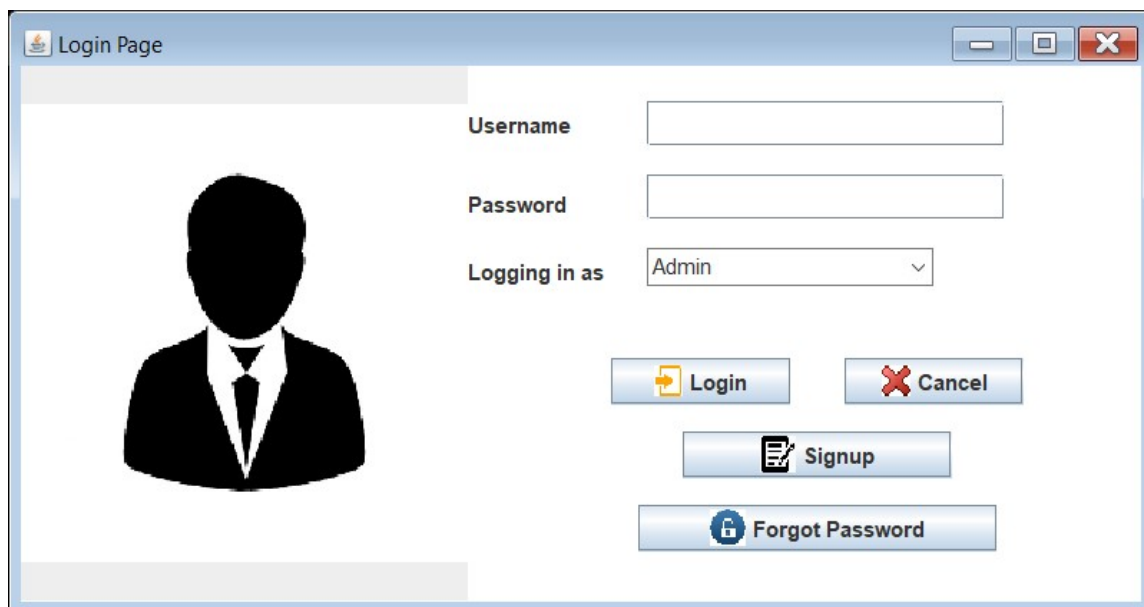
A screenshot of a web application login page titled "Login Page". The page features a silhouette of a person in a suit on the left. To the right, there are input fields for "Username", "Password", and a dropdown menu for "Logging in as" with "Admin" selected. Below these fields are four buttons: "Login" (with a key icon), "Cancel" (with a red X icon), "Signup" (with a document icon), and "Forgot Password" (with a lock icon). The page has a standard window border with minimize, maximize, and close buttons in the top right corner.

Fig 6.9: Login page



Create-Account

Username :

Password :

Security Question :

Answer :

Create Admin As :




Fig 6.10 Signup page

Username

Your Security Question

Answer

Password




Fig 6.11: Forgot Password page



Fig 6.12: Admin home page




### New Customer

Customer Name	<input type="text"/>
Meter No	673692
Address	<input type="text"/>
City	<input type="text"/>
State	<input type="text"/>
Email	<input type="text"/>
Phone Number	<input type="text"/>

Fig 6.13: New customer page




## TAX DETAILS



Cost Per Unit	<input type="text" value="9"/>
Meter Rent	<input type="text" value="47"/>
Service Charge	<input type="text" value="22"/>
Service Tax	<input type="text" value="57"/>
Swacch_Bharat_Cess	<input type="text" value="6"/>
GST	<input type="text" value="18"/>

Fig 6.16: Tax Details Page

CalculateBill Page



### Calculate Electricity Bill

Meter No	<input type="text" value="413098"/>
Name	<input type="text" value="aki"/>
Address	<input type="text" value="btm layout"/>
Units Cosumed	<input type="text"/>
Month	<input type="text" value="January"/>

Fig 6.17: Calculate Bill page

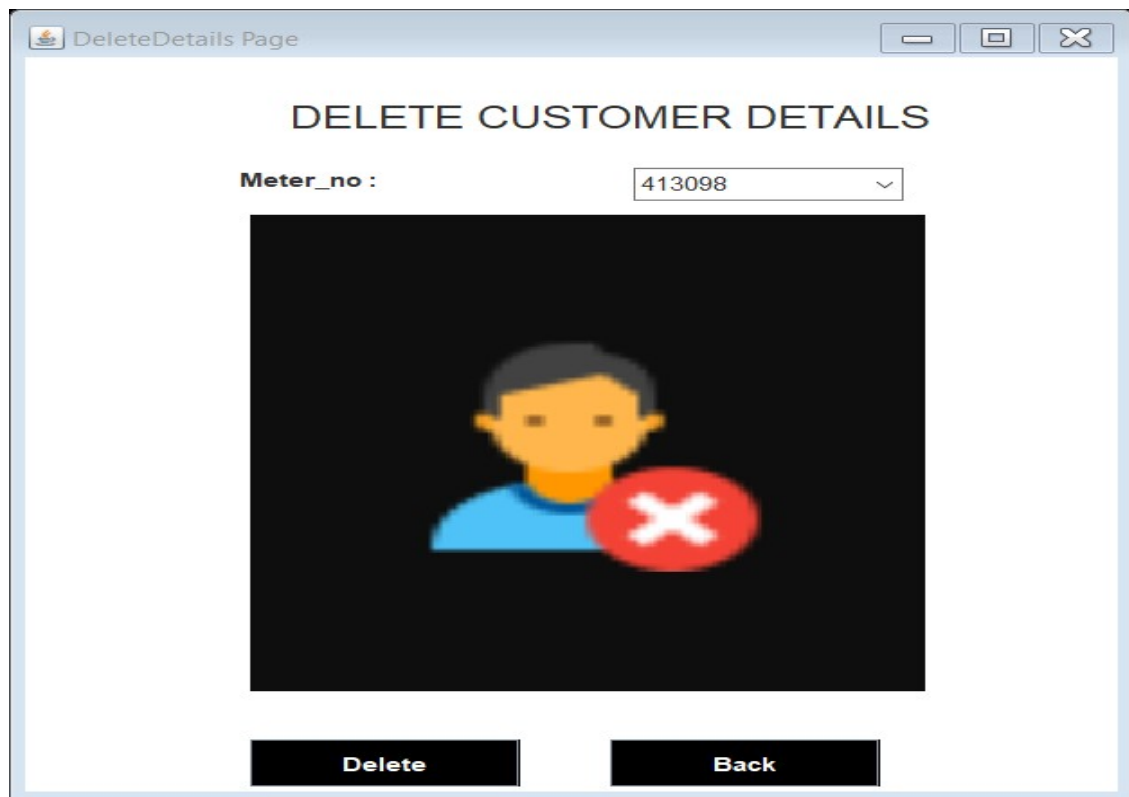


Fig 6.18: Delete Customer page



Fig 6.19: Customer Home page

UPDATE CUSTOMER DETAILS

Name :	aki
Meter Number :	413098
Address :	<input type="text" value="btm layout"/>
City :	<input type="text" value="bangalore"/>
State :	<input type="text" value="karnataka"/>
Email :	<input type="text" value="aki@gmail.com"/>
Phone :	<input type="text" value="8989998888"/>

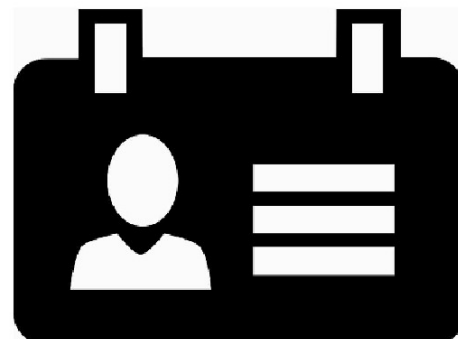


Fig 6.20: Update customer details page

VIEW CUSTOMER DETAILS

Name :	aki	State :	karnataka
Meter Number :	413098	Email :	aki@gmail.com
Address :	btm layout	Phone :	8989998888
City :	bangalore		



Fig 6.21: View Customer Details page

[illegible]

Fig 6.22: query page of customer details

PayBill Page

Electricity Bill

Meter Number

413098

Name

aki

Month

January

Units

25

Total Bill

Rs 375

Status

Not Paid

Pay

Back

BILL

Fig 6.23: Pay Bill page



Fig 6.24: Paytm page

[illegible]

Fig 6.25: Bill Details page



GenerateBill Page

Generate Bill 413098 January

City: bangalore  
Email: aki@gmail.com  
Phone Number: 8989998888

---

Meter Location: Inside  
Meter Type: Electric Meter  
Phase Code: 11  
Bill Type: Normal  
Days: 30

---

Meter Rent:	9
MCB Rent:	47
Service Tax:	22
Service Tax:	57
GST@9%:	18

Current Month :	January
Units Consumed:	25
Total Charges :	375

---

**TOTAL PAYABLE :** 375

Generate Bill

Fig 6.26: Generate Bill page

DepositDetails Page

Sort By Meter Number 413098 ▾

Sort By Month January ▾

Search Print

meter_no	month	units	total_bill	status
413098	January	25	375	Paid

Fig 6.27: Deposit Details page

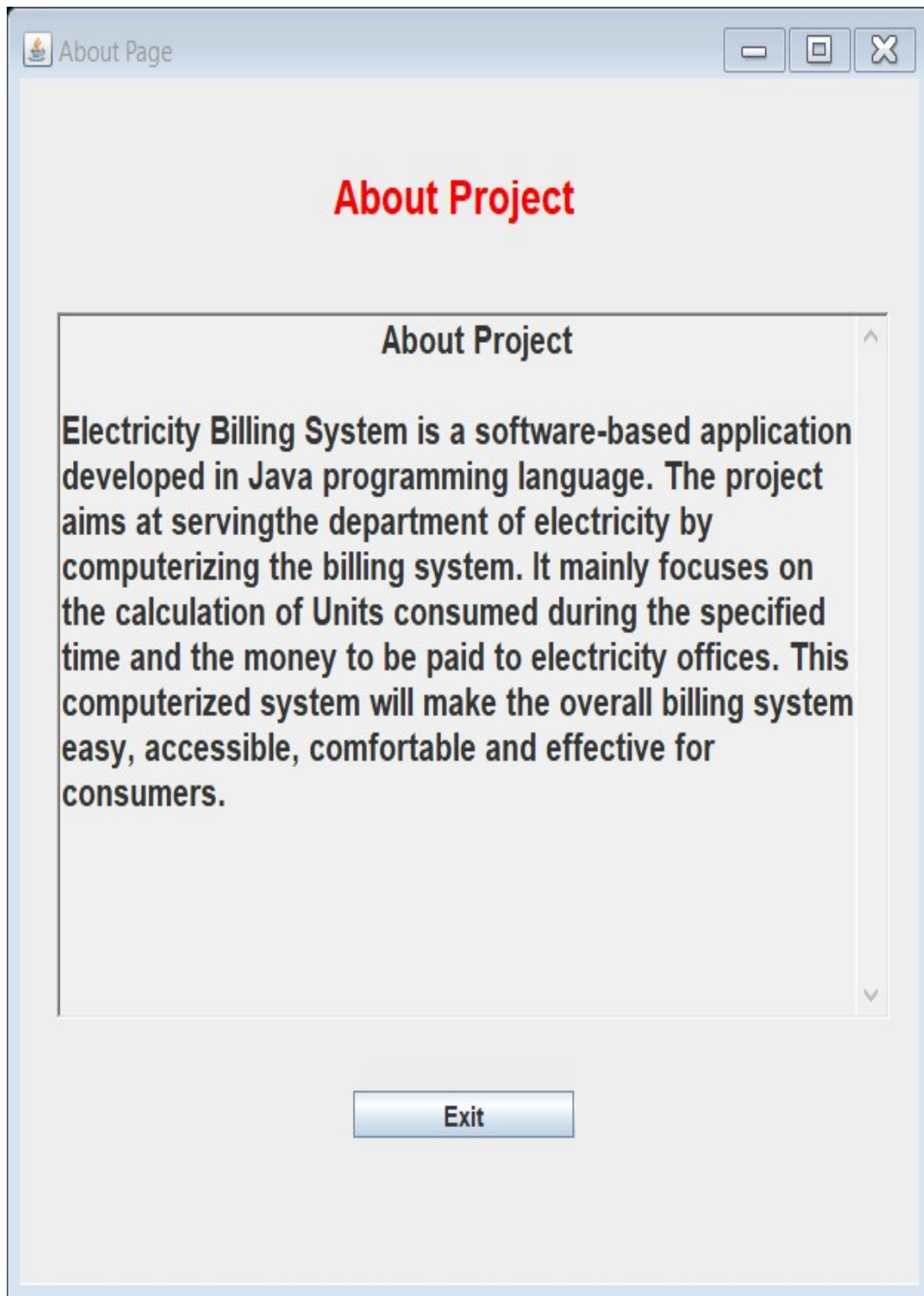


Fig 6.28: About page

## **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.