. RAYAT SHIKSHAN SANSTHA’S

C.D. JAIN COLLEGE OF COMMERCE, SHRIRAMPUR

A project Report On

“Electricity Bill Management System”

Submitted To

University of Pune

In Partial Fulfilment of the Requirement

Of BBA (CA)-III

Bachelor of Business Administration

(Computer Application)

Submitted By

Mr. Vishwambhar Sumit Bharat

Mr. Bodkhe Nilesh Laxman

Under Guidance Of

Mr. Lande .R.D

During Academic Year **2022-2023**

RAYAT SHIKSHAN SANSTHA’S

C. D. JAIN COLLEGE OF COMMERCE, SHRIRAMPUR

CERTIFICATE

(Department Of BBA (CA))

This is certify that project entitled “Online Course Registration “ submitted by **Mr. Vishwambhar Sumit Bharat & Mr. Bodkhe Nilesh Laxman** student of **BBA (CA)-III** (Bachelor of Business Administration And Computer Application) had satisfactorily completed the project during the academic year 2022-23.

Date: _ _ /_ _ /_ _ _ _

Project Guide:	**Head of Department**
Mr Lande.R.D	Mr. Chandratre Y.V

Internal Examiner **External Examiner**

ACKNOWLEGEMENT

We are really happy to present project “Online Course Registration” in PHP & MySQL. We would like to thank Chandratre sir (head of Department) for allowing making full use of lab facility. We thank to Mr.Nabge.A.D spends there valuable time for us and they gives us advice and useful suggestions about system. We are very thankful to our Mr.Nabge.A.D guided us throughout the project without of them our project would not be successful. It reach up to the present stage because of their dedication and guidance.

And also special thanks to those who helped us for completing this project successfully they solve all our problems ,error occurred while making a project and they suggested changes due to the which our project look more attractive and give us idea of project.

Place: Shrirampur **Date: / /**

Signature: **Signature:**

Mr. Vishwambhar Sumit Bharat Mr.Bodkhe Nilesh Laxman

DECLARATION

I Mr. Vishwambhar Sumit Bharat & Mr. Bodkhe Nilesh Laxman of **BBA(CA)-** C. D. Jain College of Commerce, Shrirampur declare that the project entitled **“**Online Course Registration**”** have been successfully & this project is submitted towards the partial fulfilment of the requirement of the degree of **BBA(CA).**This project is not submitted for any other Degree, Diploma or other similar title or prize in any other university**.**

Place: Shrirampur

Date: / /

Signature: **Signature:**

Mr. Vishwambhar Sumit Bharat **Mr. Bodkhe Nilesh Laxman**

Index

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 Motivation

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 Purpose Objective and goals

- 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. o Thus, it saves human efforts and resources.

Objectives :

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.

Goals:

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.

1.4 Literature Survey

"An Efficient Electricity Billing System Based on Java Technology" by Smith et al. (2017): This paper proposes an electricity billing system developed using Java technology. The system employs various Java frameworks and libraries to handle bill calculation, payment processing, and customer management. It also incorporates features such as real-time usage monitoring, bill generation, and report generation. The authors claim that their system improves accuracy, reduces human errors, and enhances overall billing efficiency.

"Design and Implementation of Electricity Billing System Using Java and MySQL" by Kumar et al. (2018): This research work presents a Java-based electricity billing system integrated with a MySQL database. The system allows customers to view and pay their bills online while enabling utility companies to manage customer accounts, generate bills, and analyze consumption patterns. The authors highlight the scalability, security, and robustness of their system, which facilitates smooth billing operations and enhances customer satisfaction.

"Smart Electricity Billing and Management System Using Java and IoT" by Sharma et al. (2019): In this study, the authors propose a smart electricity billing and management system that integrates Java programming and Internet of Things (IoT) technology. The system employs IoT devices to monitor electricity consumption in real-time and communicates with a Java-based billing system. It offers features such as automated meter reading, remote bill payment, and energy consumption analysis. The authors emphasize the system's ability to provide accurate bills and promote energy conservation.

"A Comparative Study of Electricity Billing Systems: A Java Perspective" by Patel et al. (2020): This paper presents a comparative study of existing electricity billing systems developed using Java. The authors analyze and compare different systems in terms of their architecture, functionality, scalability, and performance. The study aims to identify the strengths and weaknesses of each system, providing insights into the design and implementation considerations for developing an efficient electricity bill management system.

Conclusion: The literature survey highlights various research works and projects related to electricity bill management systems developed using Java. These works demonstrate the effectiveness of Java programming language in developing robust, scalable, and user-friendly systems for managing electricity bills. The integration of technologies such as databases, IoT, and real-time monitoring enhances the efficiency, accuracy, and convenience of these systems. The findings from this survey can serve as a valuable reference for the development of the Electricity Bill Management System by Mr. Sumit Vishwambhar and Mr. Nilesh Bodkhe.

1.5 Project Scope and limitations

Extensibility: This software is extendable in ways that its original developers may not expect. The following principles enhances extensibility like hide data structure, avoid traversing multiple

Links or methods avoid case statements on object type and distinguish public and private operations.

Reusability: Reusability is possible as and when require in this application. We can update it next version. Reusable software reduces design, coding and testing cost by amortizing effort Over several designs. Reducing the amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both types of reusability:

Sharing of newly written code within a project and reuse of previously written code on new projects.

Understand ability: A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method, which small and coherent helps to accomplish this.

Cost-effectiveness: Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement.

Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

LIMITATIONS:

This application cannot be accessed remotely.

- **This application requires knowledgeable person to use this application.**
- **This application does not have journals.**

2 .System Analysis

2.1 Existing 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.

2.2scope and limitations of existing systems

- The scope of an Electricity Bill Management System project in Java can vary depending on the specific requirements and objectives of the system. However, some common features and functionalities of such a system may include:
- User registration and login: Users can create an account and log in to the system using their username and password.
- Electricity meter reading: The system should allow users to input their electricity meter readings, which can be used to calculate the amount of electricity consumed.
- Bill calculation: Based on the electricity meter reading and the tariff rates, the system should calculate the electricity bill amount for each user.
- Payment management: The system should provide users with various payment options and manage the payment process.

- Customer support: The system should provide customer support services to users, such as responding to queries and complaints.
- However, there are some limitations to existing systems for Electricity Bill Management:
- Limited functionality: Some existing systems may lack certain features or functionalities, such as real-time monitoring of electricity consumption.
- Security concerns: Electricity Bill Management systems deal with sensitive data such as user information and payment details. Therefore, the system must ensure that data is protected against unauthorized access and breaches.
- Compatibility issues: Some existing systems may not be compatible with certain hardware or software configurations, which can limit their functionality.
- Scalability: As the number of users and electricity consumption grows, the system should be able to handle an increasing load. However, some existing systems may struggle to scale up to accommodate higher traffic and usage levels.
- Overall, the scope of an Electricity Bill Management System project in Java is broad and can provide several benefits to users. However, the limitations of existing systems must be taken into consideration when developing a new system to ensure that it meets user needs and expectations.

2.3 Project Perspective, Features

- From a project perspective, an Electricity Bill Management System project in Java can be designed to provide efficient management of electricity bills and related processes. Here are some key features that can be included in such a system:

- User Registration and Login: Allow users to create accounts and log in securely to access their electricity billing information.

- Dashboard and Overview: Provide users with a comprehensive dashboard displaying their electricity consumption history, current usage, pending bills, and payment status.

- Meter Reading Management: Enable users to input their electricity meter readings manually or integrate with smart meters to automate the reading process.

- Bill Calculation: Calculate electricity bills based on the meter readings, tariff rates, and any additional charges or discounts. Provide a breakdown of the bill components for transparency.

- Billing History and Invoices: Maintain a record of past bills and invoices for users to review and download as necessary.

- Payment Options: Integrate with payment gateways to offer various payment methods such as credit/debit cards, net banking, e-wallets, and automatic payment setups.

- Notifications and Reminders: Send notifications and reminders to users for upcoming bill payments, payment confirmations, and any changes in billing or tariff rates.

- Energy Consumption Analysis: Provide insights and analytics on electricity consumption patterns, allowing users to track their usage trends and make informed decisions to reduce energy consumption.

- Customer Support: Include a customer support module to address user queries, complaints, and provide assistance regarding billing, payments, and general system usage.

- Reports and Analytics: Generate reports and analytics on electricity consumption, revenue generation, outstanding bills, and other relevant metrics to assist in decision-making and planning.
- Integration with Utility Providers: Establish integration with utility providers' systems to fetch accurate meter readings and facilitate seamless data exchange.
- Security and Privacy: Implement robust security measures to protect user data, ensure privacy, and prevent unauthorized access.
- Multi-lingual and Multi-platform Support: Provide support for multiple languages and ensure compatibility with various platforms such as desktop, web, and mobile devices.
- These features can enhance the functionality and usability of the Electricity Bill Management System, providing users with a convenient and streamlined experience for managing their electricity bills efficiently.

2.4 Requirement Analysis

Hardware Requirements:

2.3.1 Hardware Specification: -Processor Intel Pentium V or higher
2.3.2 Clock Speed: -1.7 GHz or more
2.3.3 System Bus: -64 bits
2.3.4 RAM: -16GB
2.3.5 HDD: -2TB
2.3.6 Monitor: -LCD Monitor
2.3.7 Keyboard: -Standard keyboard
2.3.8 Mouse: -Compatible mouse

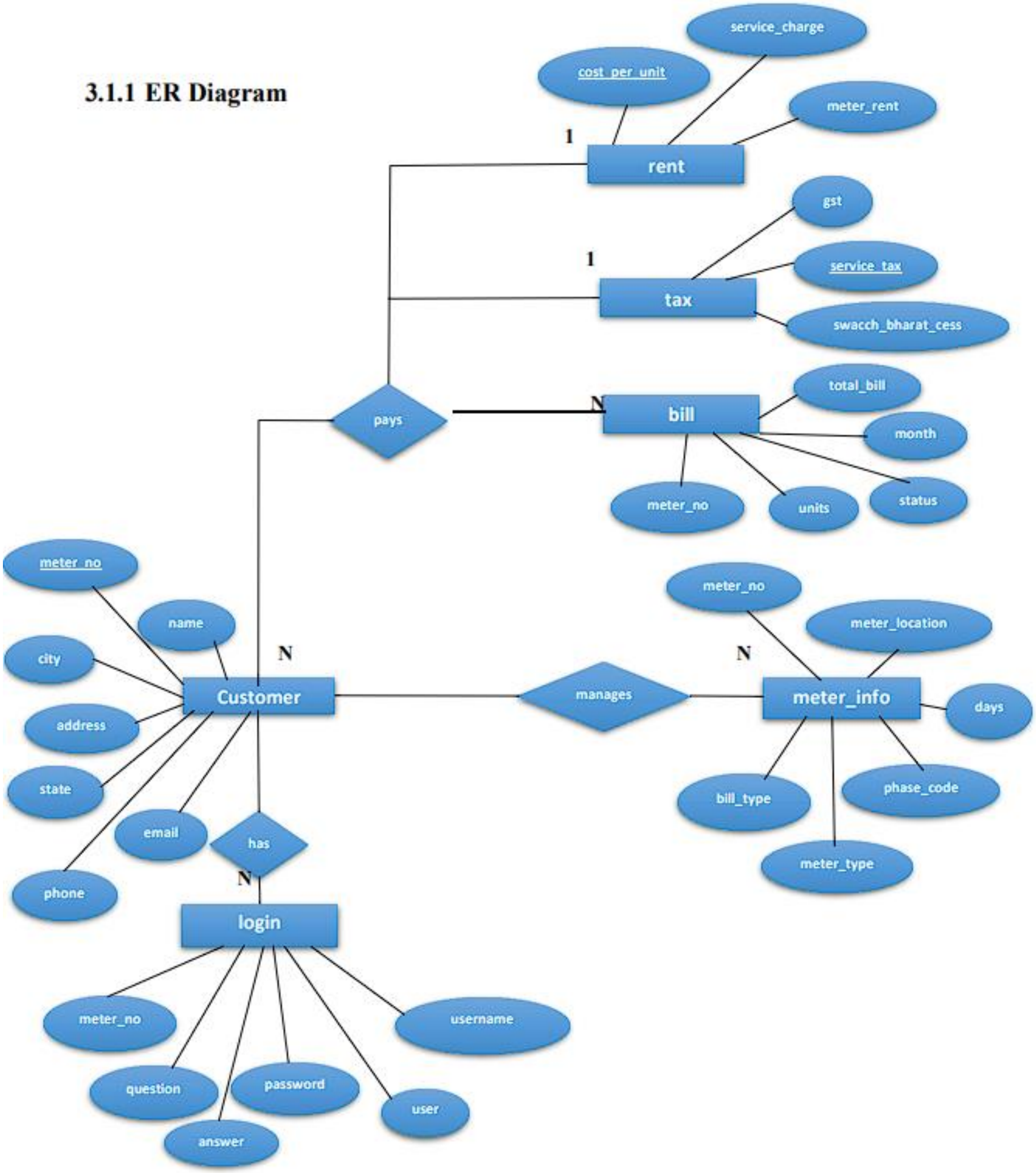
Software Requirements:

2.3.9 Operating System: -Windows 10
2.3.10 Software: -Microsoft SQL Server
2.3.11 Front End: -Java core/swings (NetBeans)
2.3.12 Back End: -My SQL

3. System Design

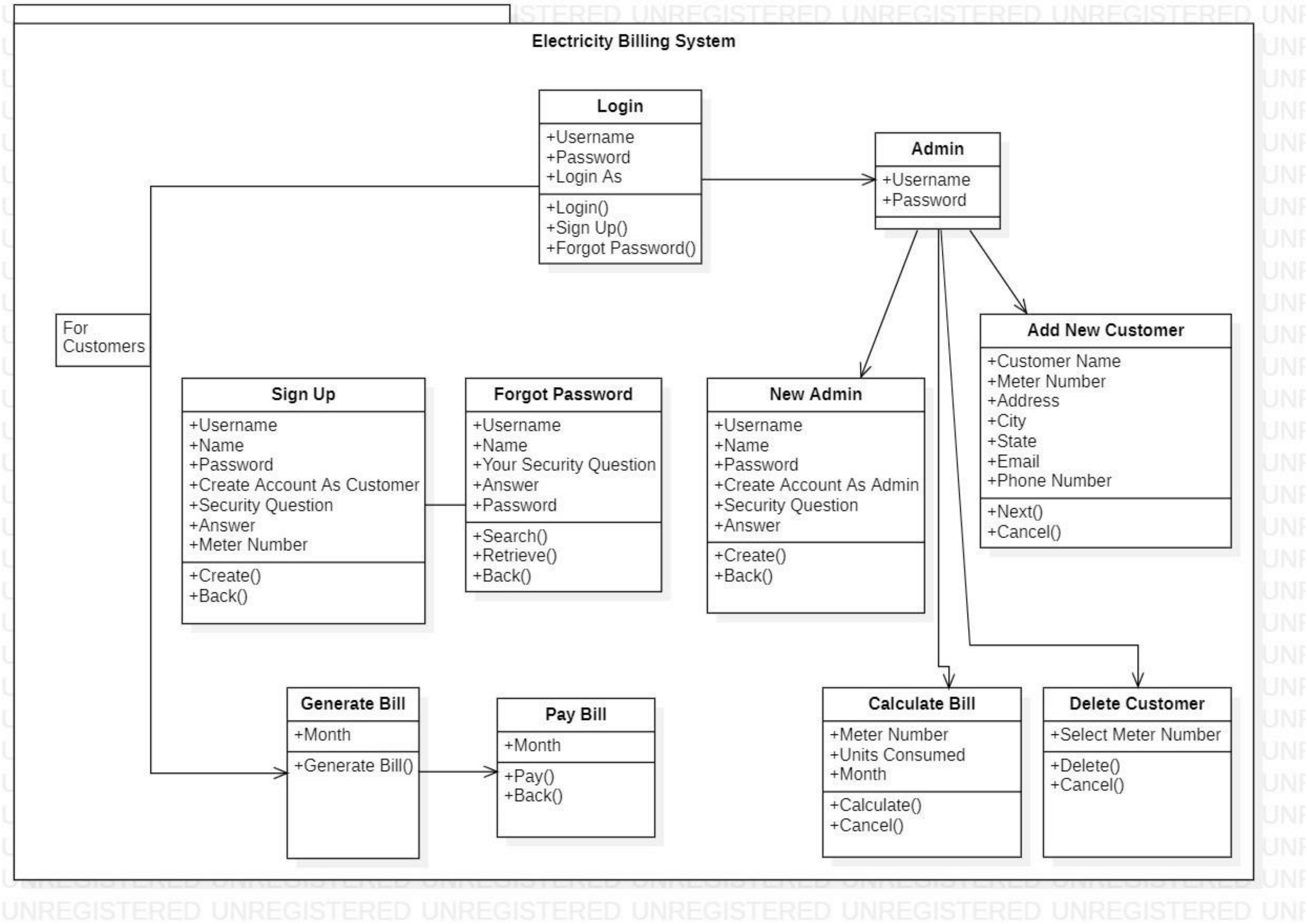
3.1 Design Constraints [ER-Diagram]

3.1.1 ER Diagram

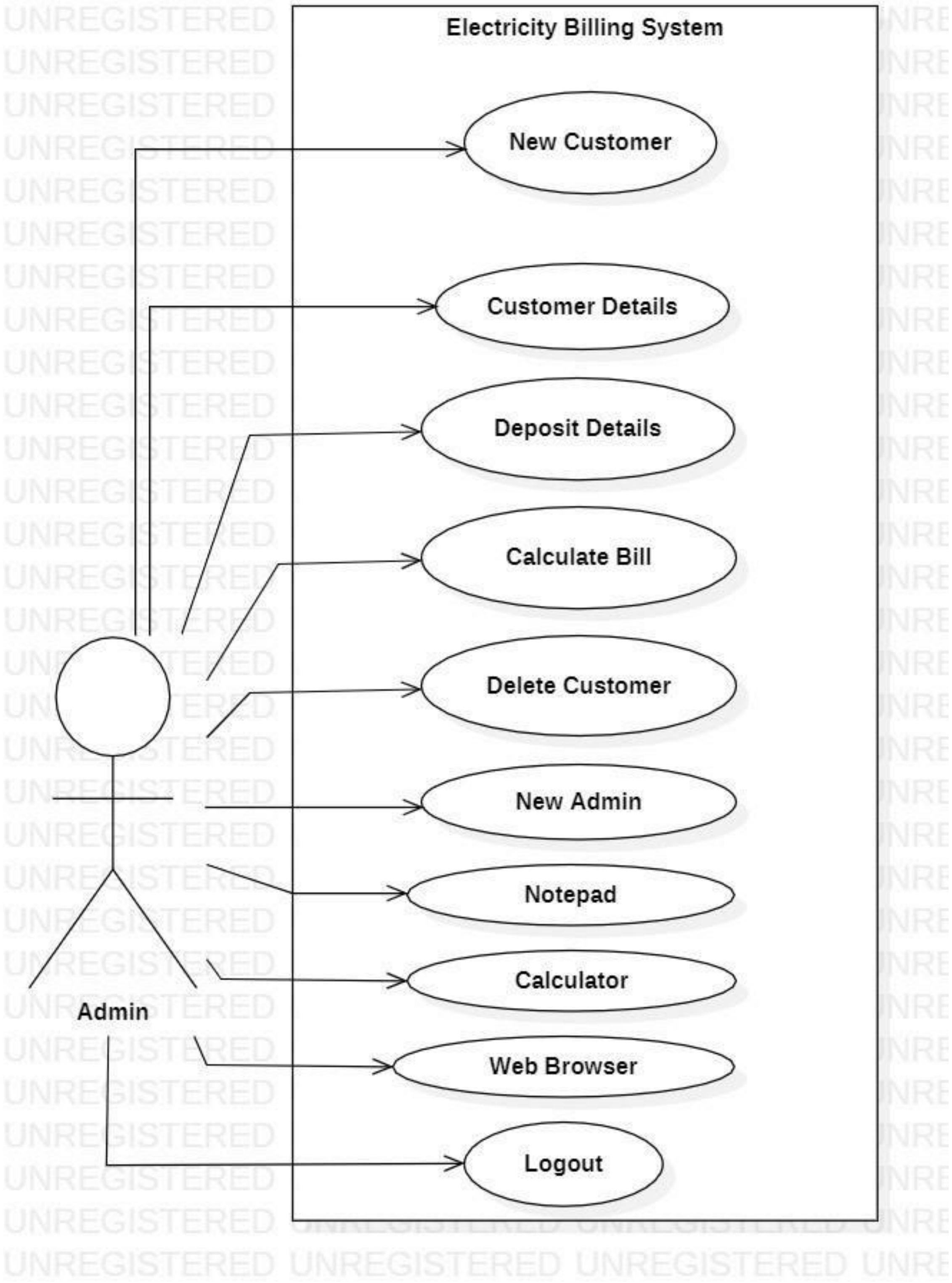


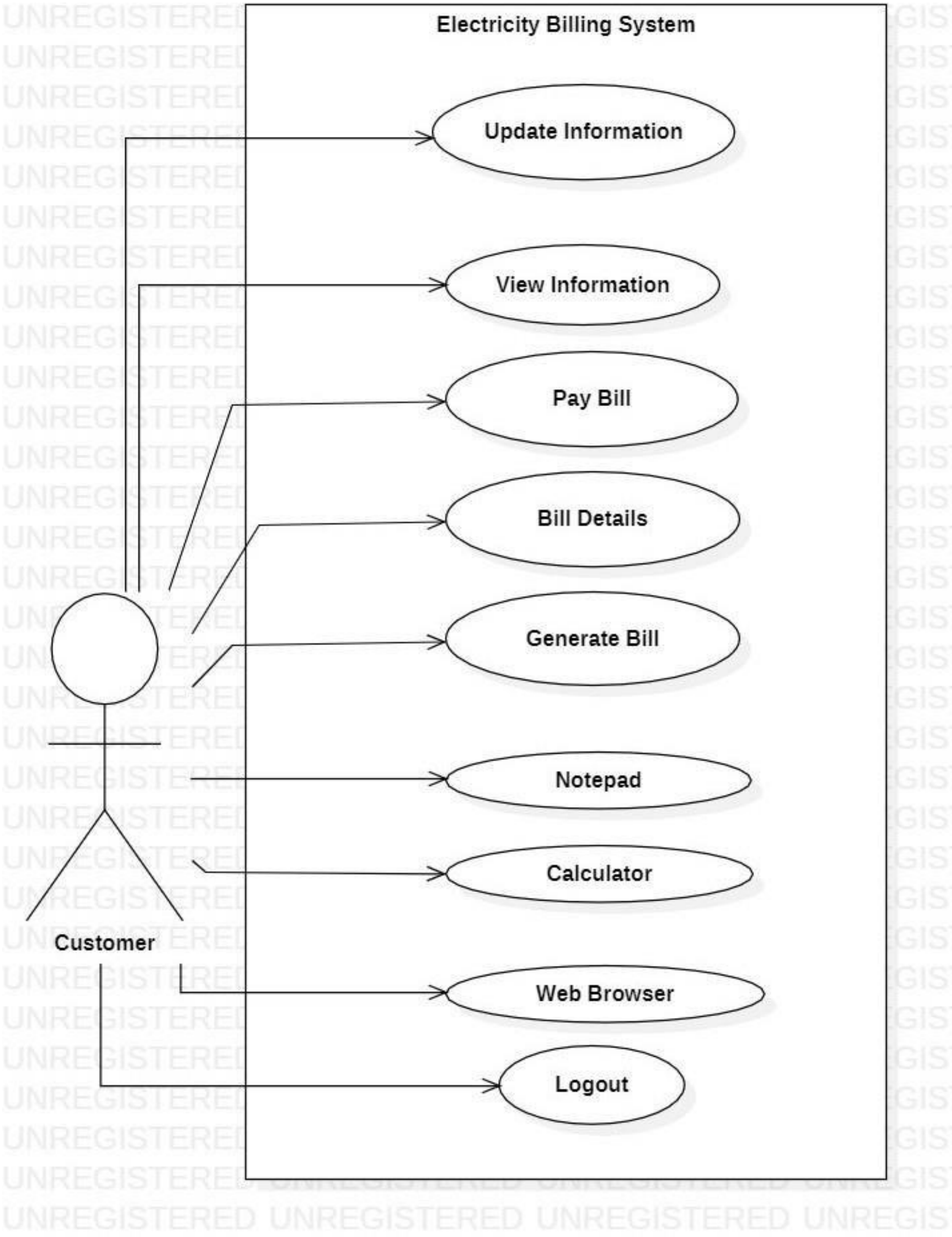
3.2 System Model: UML Diagrams

3.2.1 Class Diagram

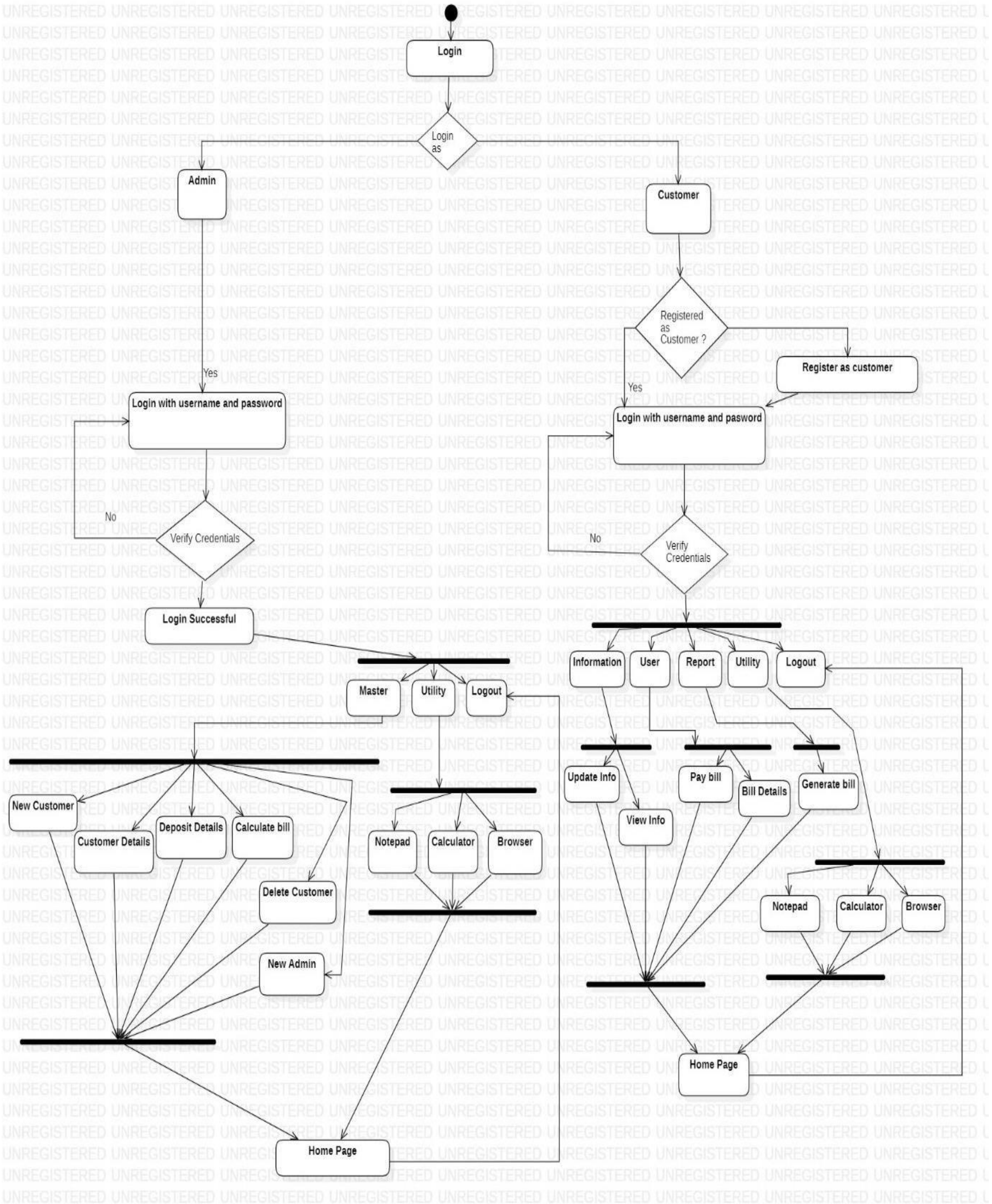


3.2.2 Use Case Diagrams:

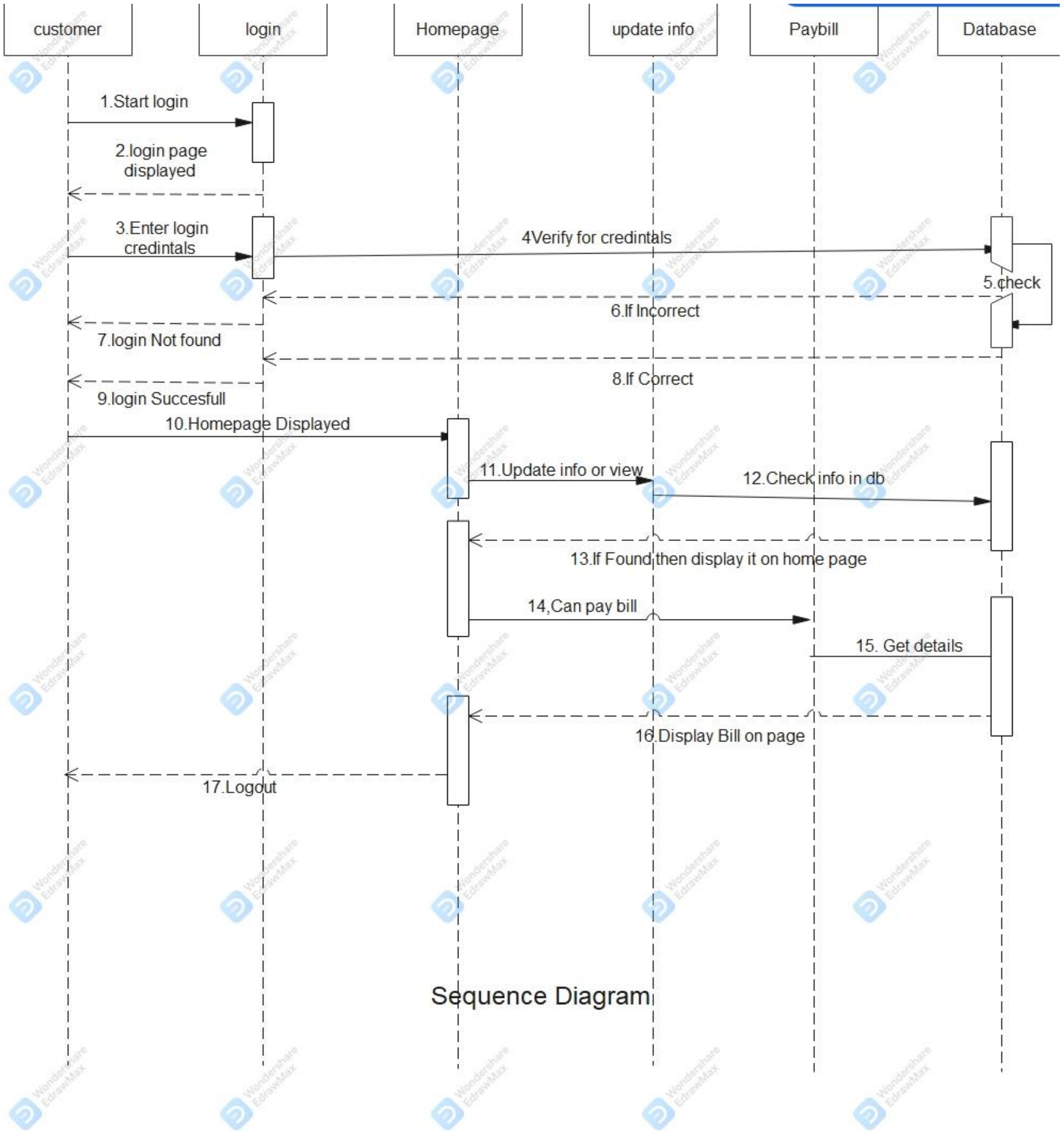



Electricity Billing System
Update Information
View Information
Pay Bill
Bill Details
Generate Bill
Notepad
Calculator
Web Browser
Logout
Customer

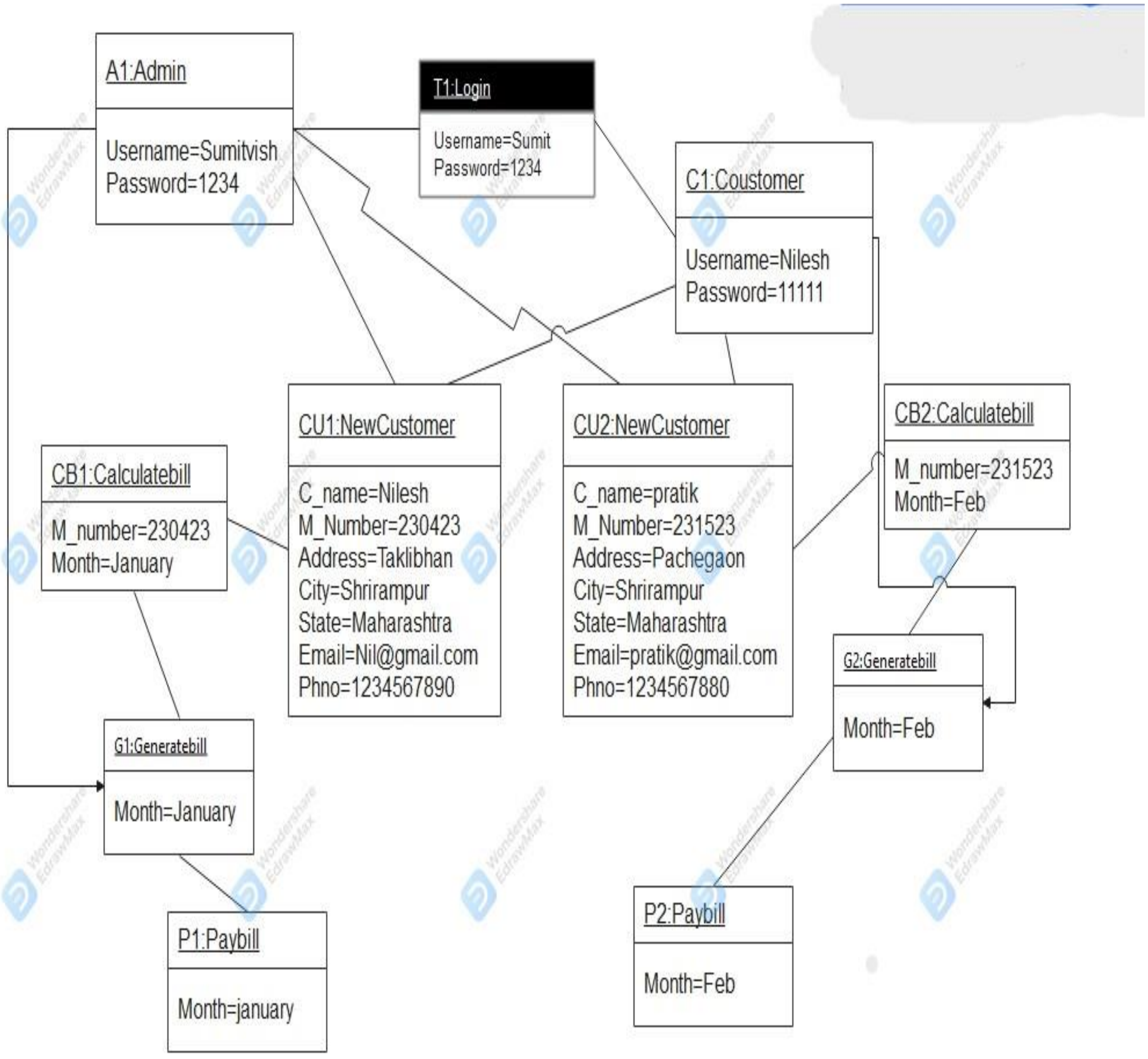
3.2.3 Activity Diagram:



3.2.4 Sequence Diagram

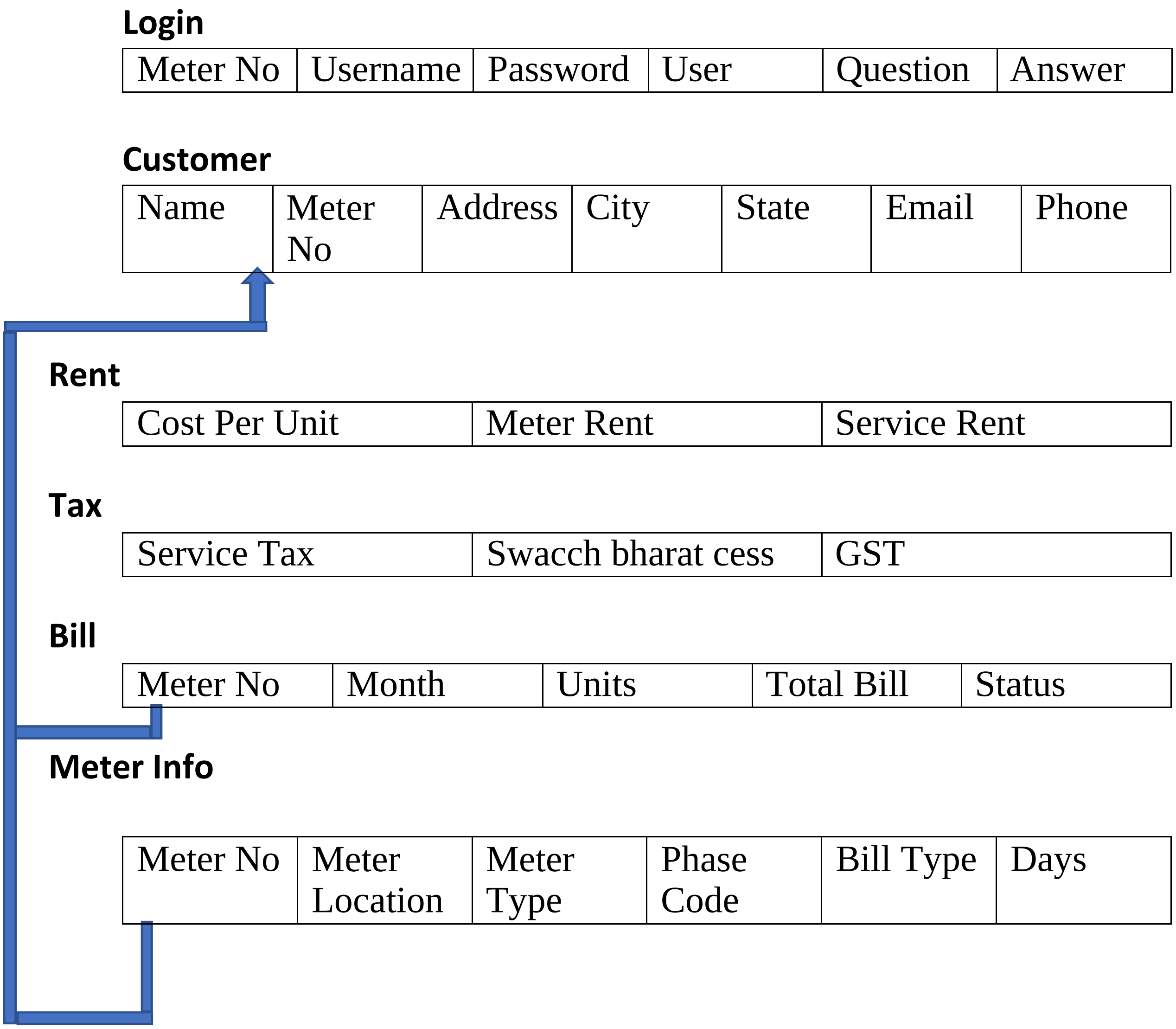


3.2.5. Object Diagram



Object Diagram

3.2.6 Schema Diagram



Login

Meter No	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 bharat cess	GST

Bill

Meter No	Month	Units	Total Bill	Status

Meter Info

Meter No	Meter Location	Meter Type	Phase Code	Bill Type	Days

Schema Diagram

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

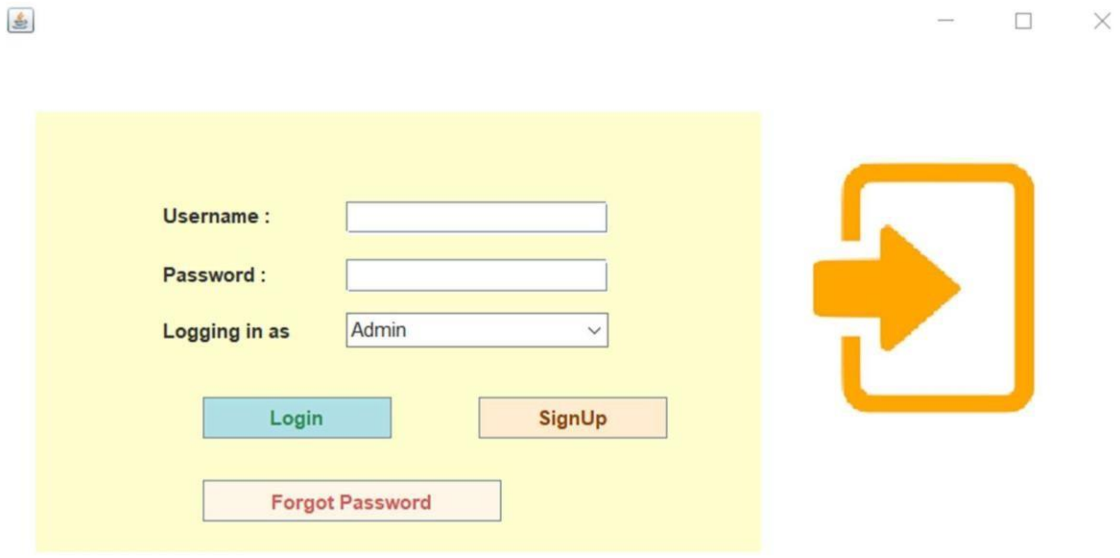
A primary key is a table column that can be used to uniquely identify every row of the table. Any column that has this property, these columns are called candidate key. A composite primary key is a primary key consisting of more than one column. A foreign is a column or combination of columns that contains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

3.2.6 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 ".

3.3 User Interfaces

SNAPSHOTS



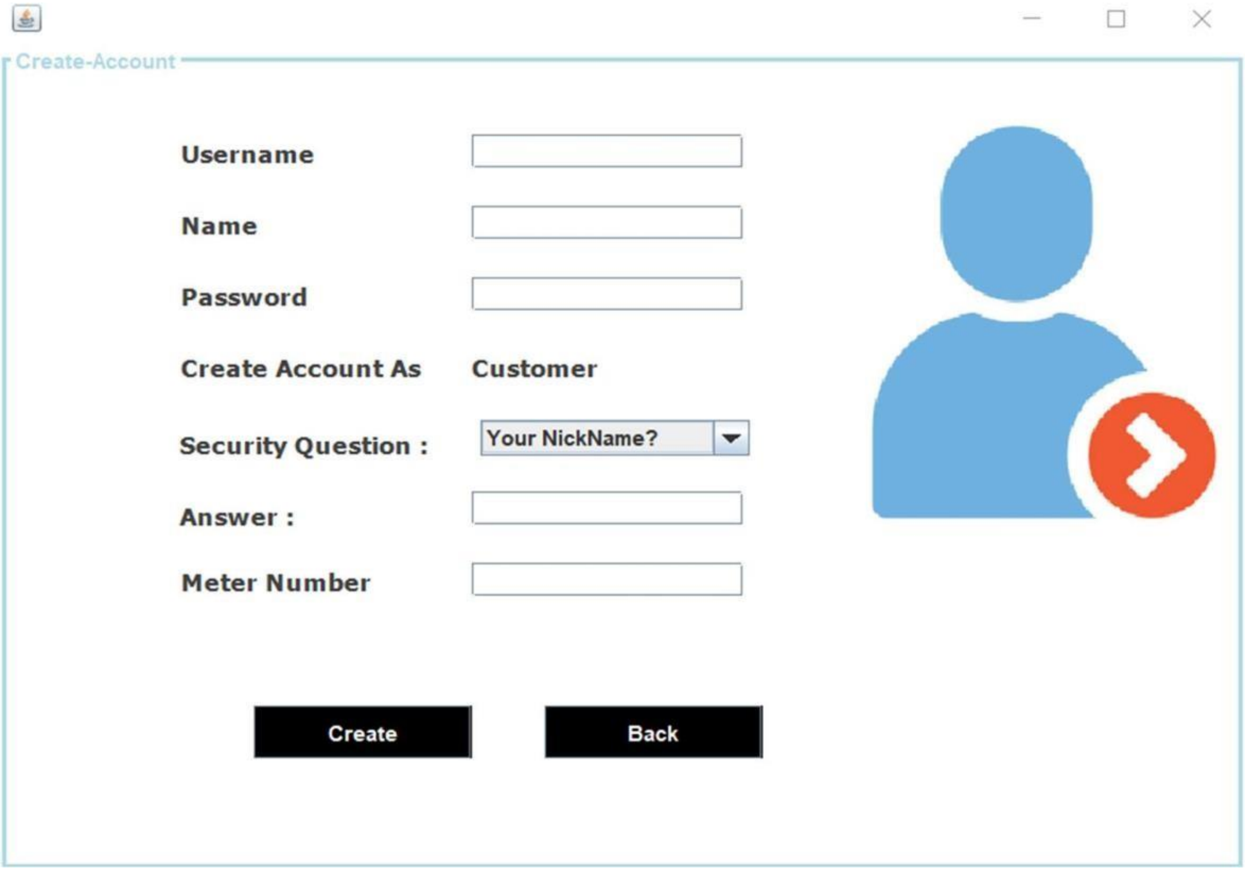
Username :

Password :

Logging in as Admin

Login Screen

Here Customer and Admin can login to their respective accounts. The dropdown menu allows to choose whether to login as an admin or as a customer.



Create Account As Customer

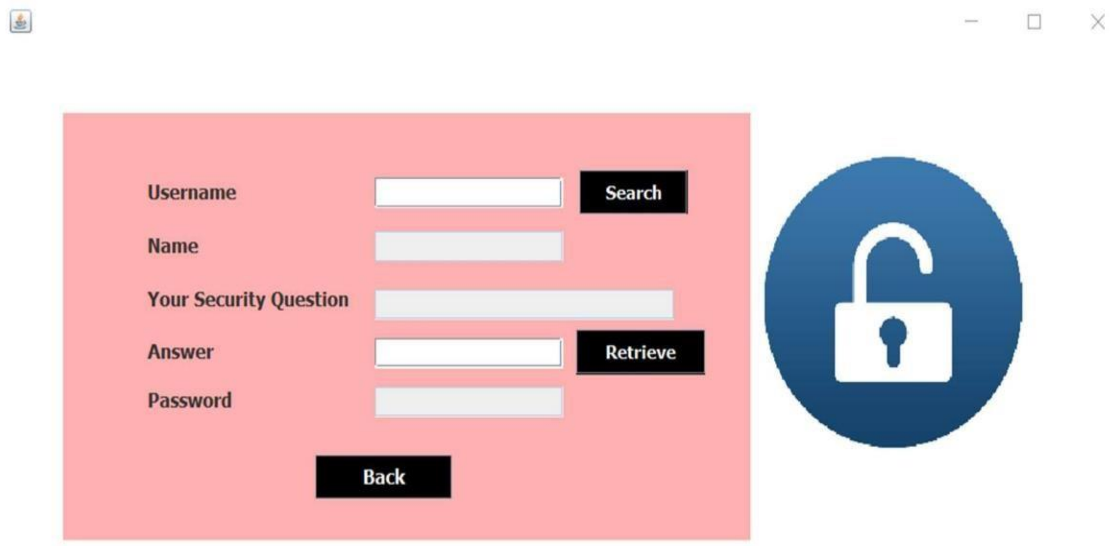
Security Question : Your NickName?

Sign Up Screen

Here New customers will signup to access their accounts.

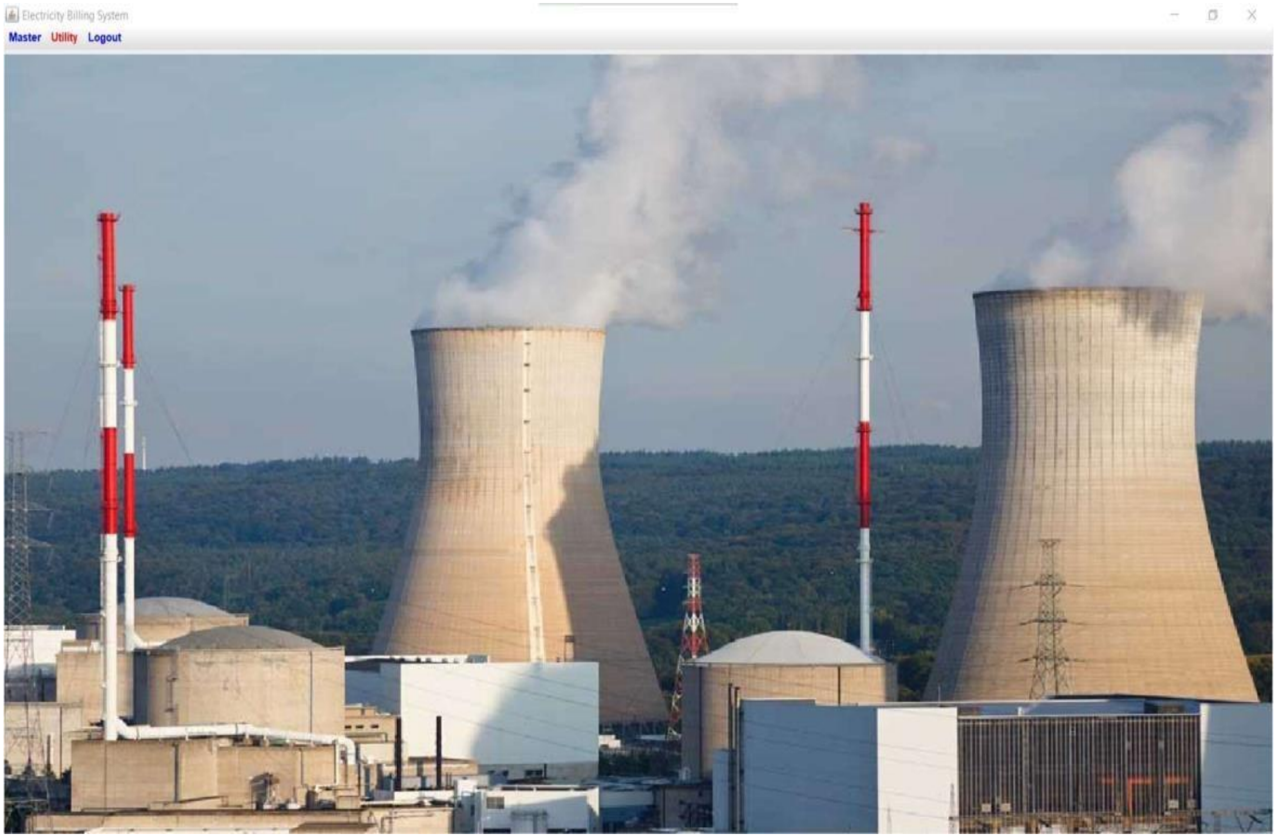
User have to enter username, name, password, choose security question and answer to that question.

Every user must enter their unique Meter Number to complete their signup process.



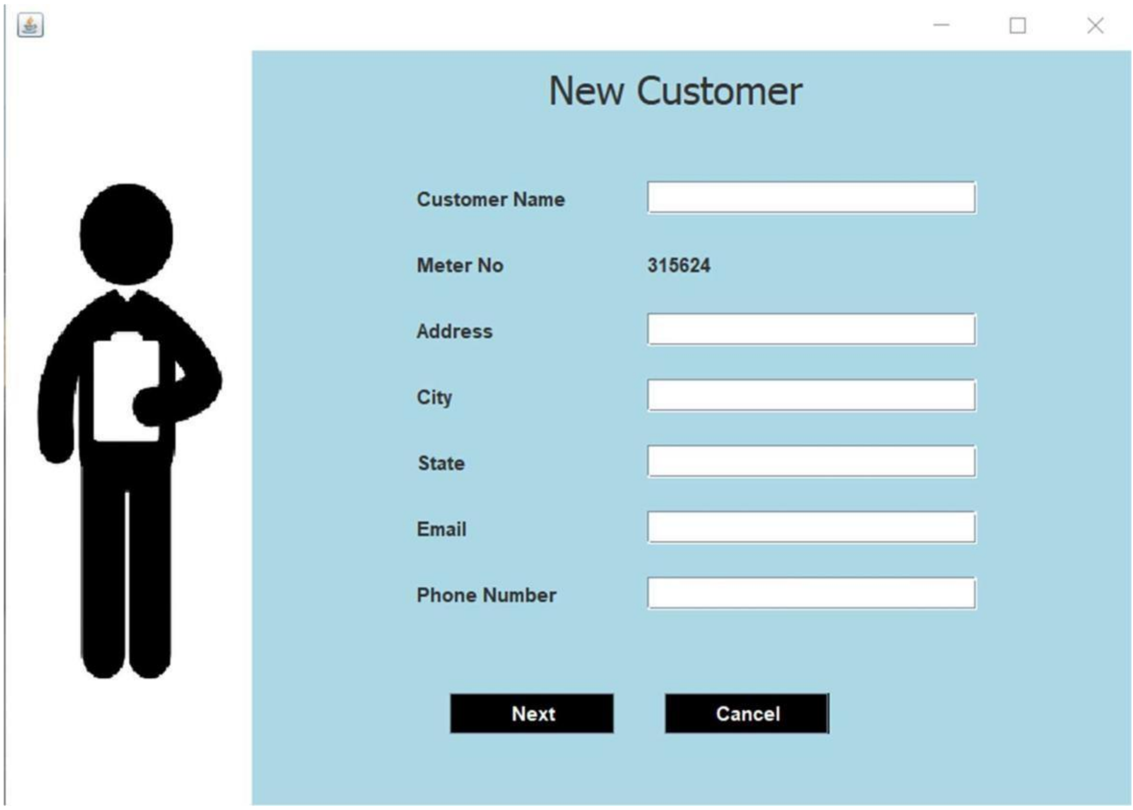
Forgot Password
Screen

Here customers as well as admins can retrieve their passwords in case they fail to remember. Customers and admins have to fill the required details and answer the security question chosen by them to retrieve their password.



Admin's Home Screen

Admin lands on this page after successful login.



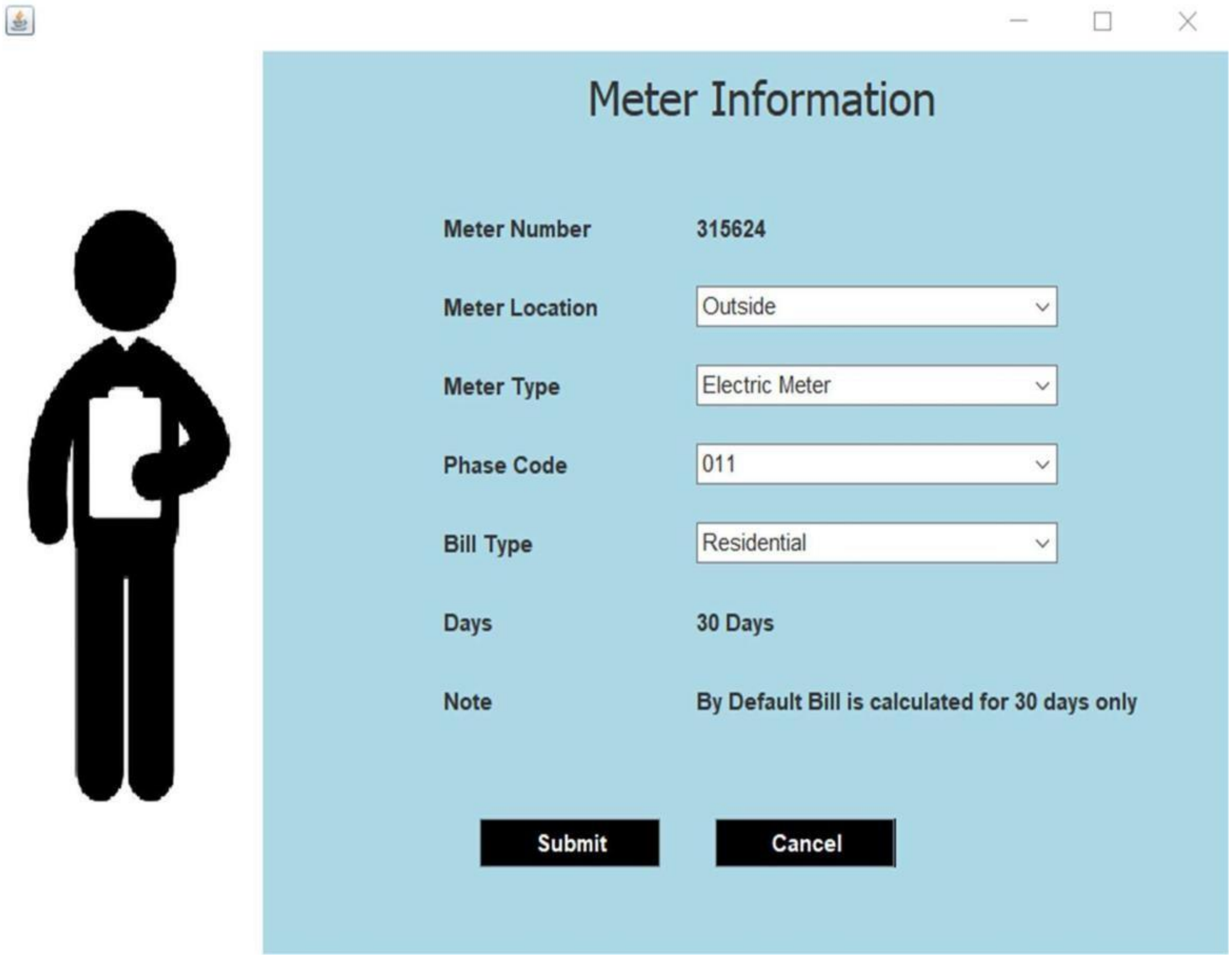
New Customer

Customer Name	
Meter No	315624
Address	
City	
State	
Email	
Phone Number	

New Customer
Screen

Here admin registers new users.

Admin enters Customer’s Name, Address, City, State, Email and Phone Number.



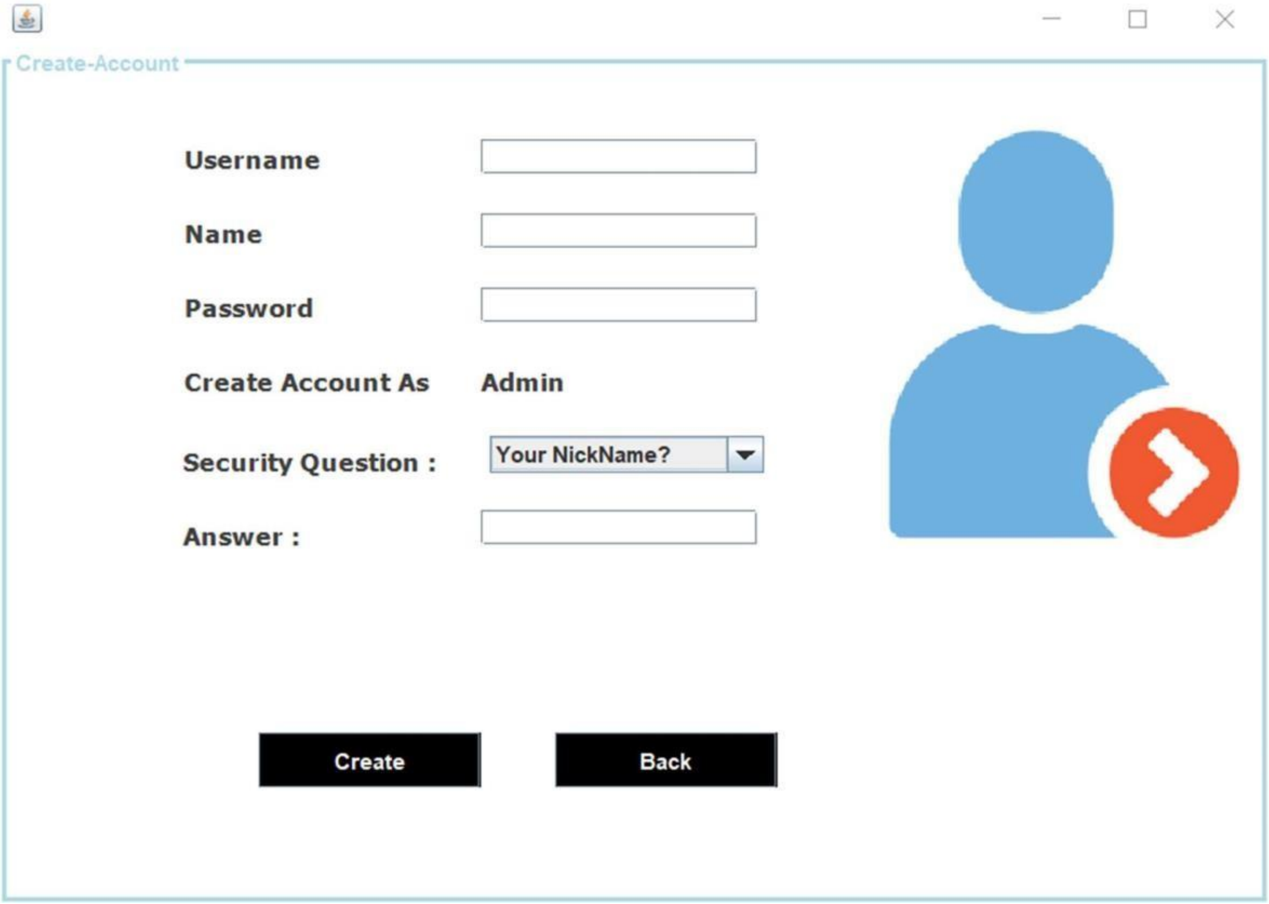
Meter Information

Meter Number	315624
Meter Location	Outside
Meter Type	Electric Meter
Phase Code	011
Bill Type	Residential
Days	30 Days
Note	By Default Bill is calculated for 30 days only

Meter Info Screen

Here Admin selects the location and type of meter installed at the customers end.

Admin also selects the phase code and Bill type i.e. Residential or Commercial/ Industrial.



Create-Account

Username

Name

Password

Create Account As Admin

Security Question : Your NickName?

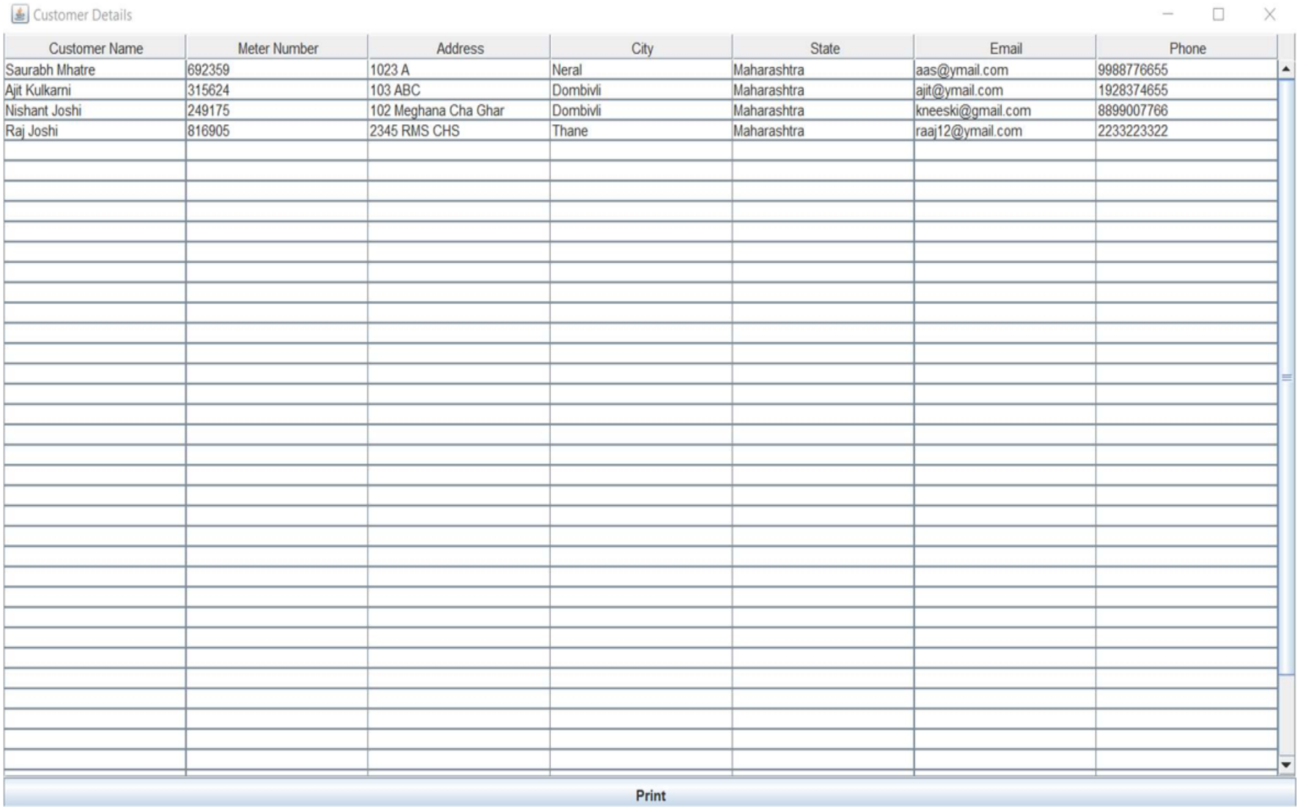
Answer :

Add New Admin
Screen

Here existing admins can add new admins to access the stored data.

New admins have to enter username, name, password, choose security question and answer to that question.

Admin can be added only by existing admins via Admin module only.

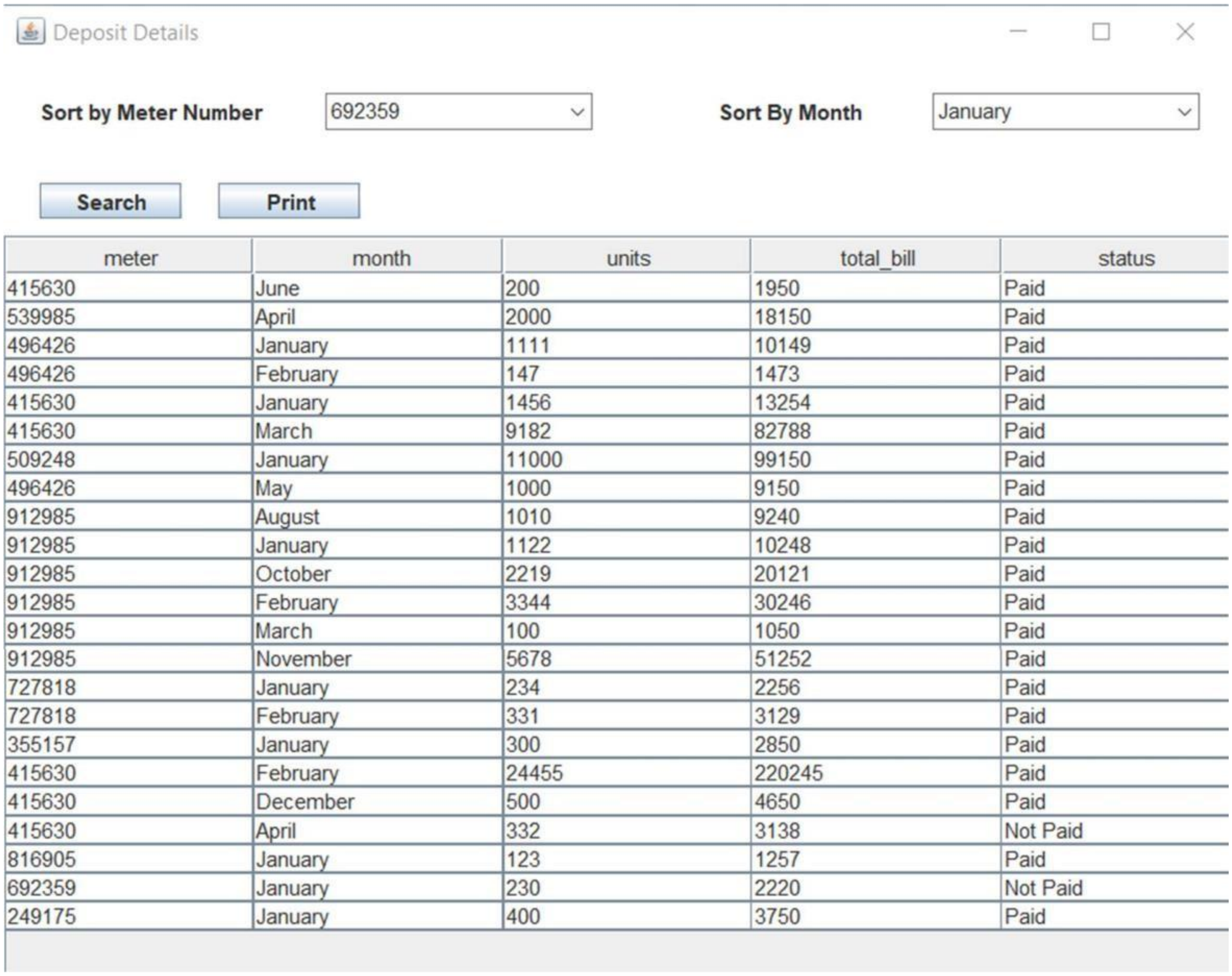


Customer Name	Meter Number	Address	City	State	Email	Phone
Saurabh Mhatre	692359	1023 A	Neral	Maharashtra	aas@ymail.com	9988776655
Ajit Kulkarni	315624	103 ABC	Dombivli	Maharashtra	ajit@ymail.com	1928374655
Nishant Joshi	249175	102 Meghana Cha Ghar	Dombivli	Maharashtra	kneeski@gmail.com	8899007766
Raj Joshi	816905	2345 RMS CHS	Thane	Maharashtra	raaj12@ymail.com	2233223322

Customer Details

Screen

Here Admins can see the details of all registered customers. Admin can print these details in pdf format if the wish.



Sort by Meter Number: 692359

Sort By Month: January

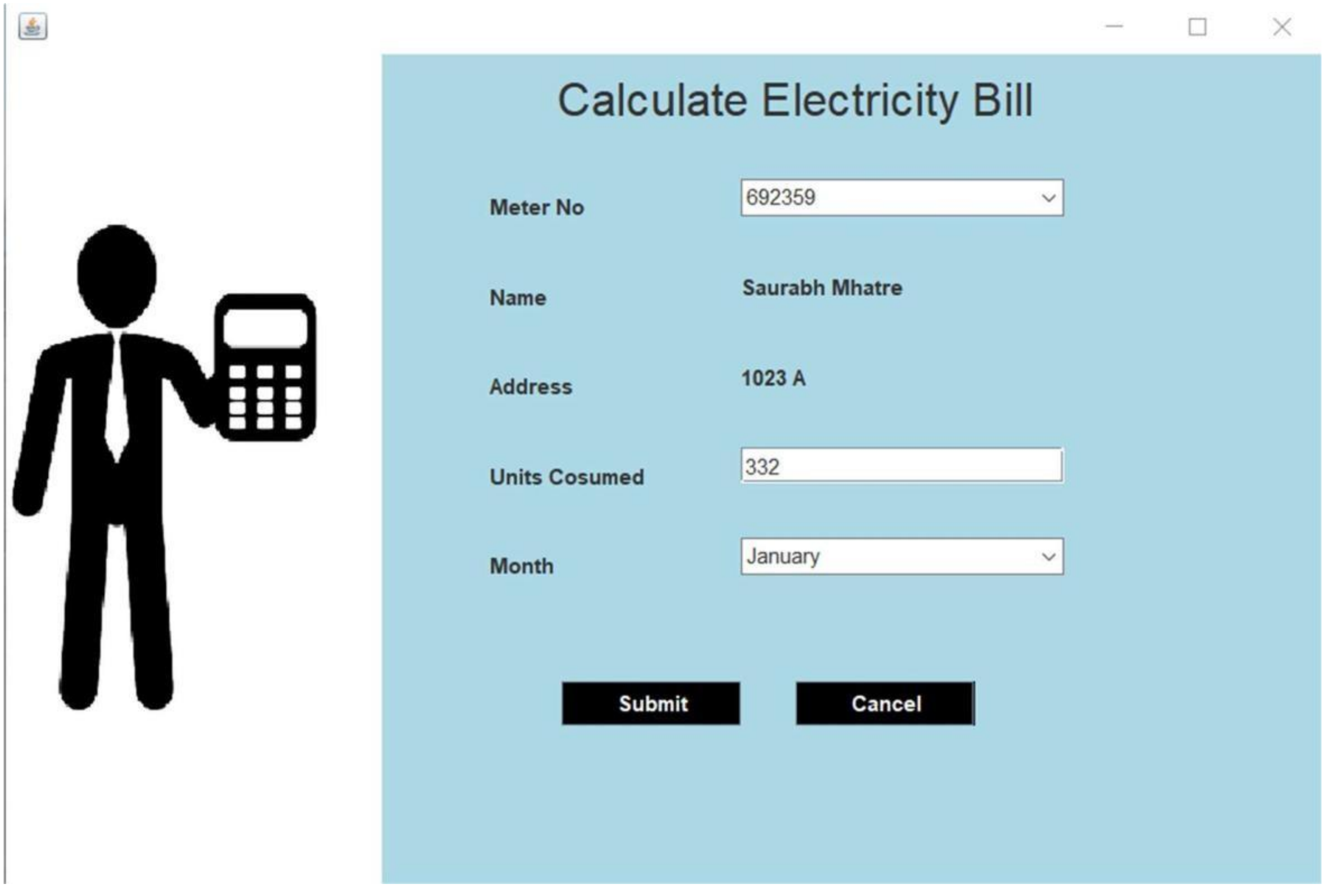
meter	month	units	total_bill	status
415630	June	200	1950	Paid
539985	April	2000	18150	Paid
496426	January	1111	10149	Paid
496426	February	147	1473	Paid
415630	January	1456	13254	Paid
415630	March	9182	82788	Paid
509248	January	11000	99150	Paid
496426	May	1000	9150	Paid
912985	August	1010	9240	Paid
912985	January	1122	10248	Paid
912985	October	2219	20121	Paid
912985	February	3344	30246	Paid
912985	March	100	1050	Paid
912985	November	5678	51252	Paid
727818	January	234	2256	Paid
727818	February	331	3129	Paid
355157	January	300	2850	Paid
415630	February	24455	220245	Paid
415630	December	500	4650	Paid
415630	April	332	3138	Not Paid
816905	January	123	1257	Paid
692359	January	230	2220	Not Paid
249175	January	400	3750	Paid

Deposit Details
Screen

Here Admin can check the status whether customers have paid their bills or not.

His list can be sorted according to individual user's meter number or according to month.

Admin can print these details in pdf format if the wish.

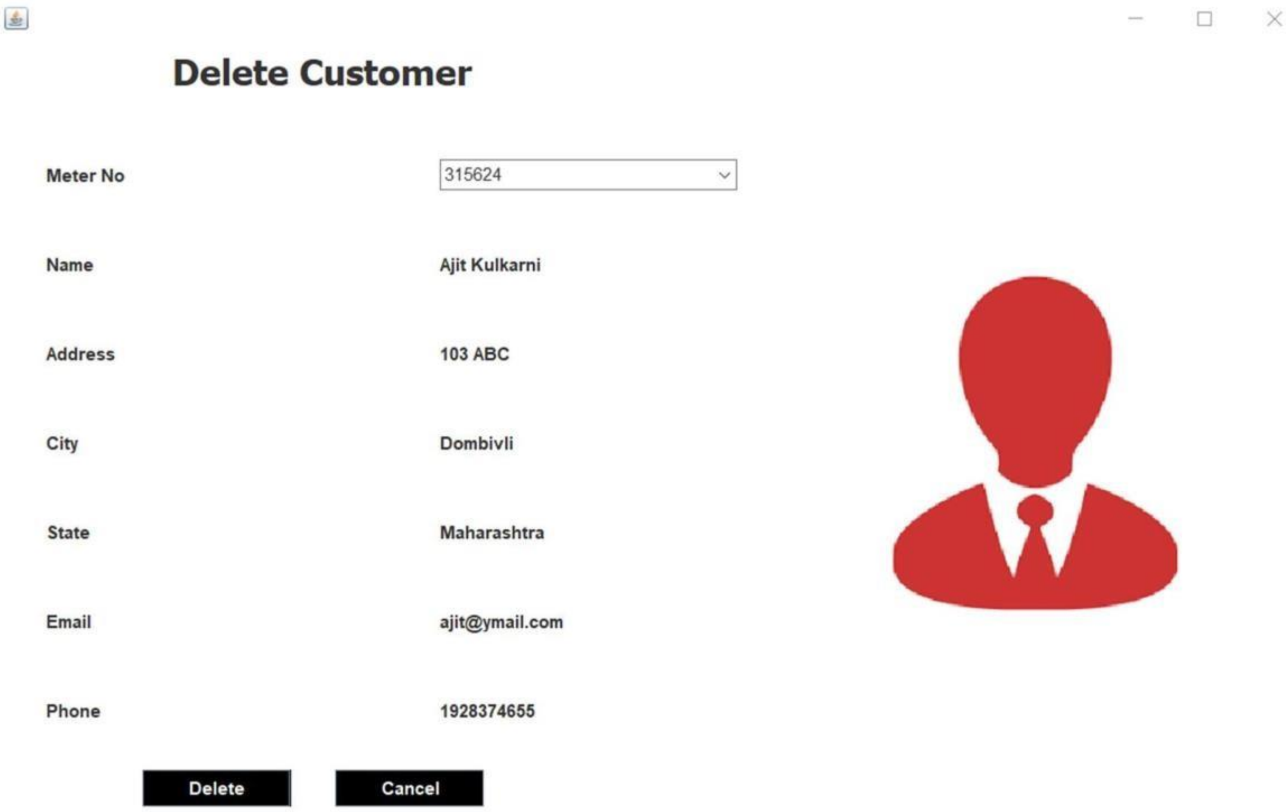


Calculate Electricity Bill

Meter No	692359
Name	Saurabh Mhatre
Address	1023 A
Units Cosumed	332
Month	January

Calculate Bill Screen

Here admin calculate the bill of users by selecting appropriate meter number, units consumed and month.

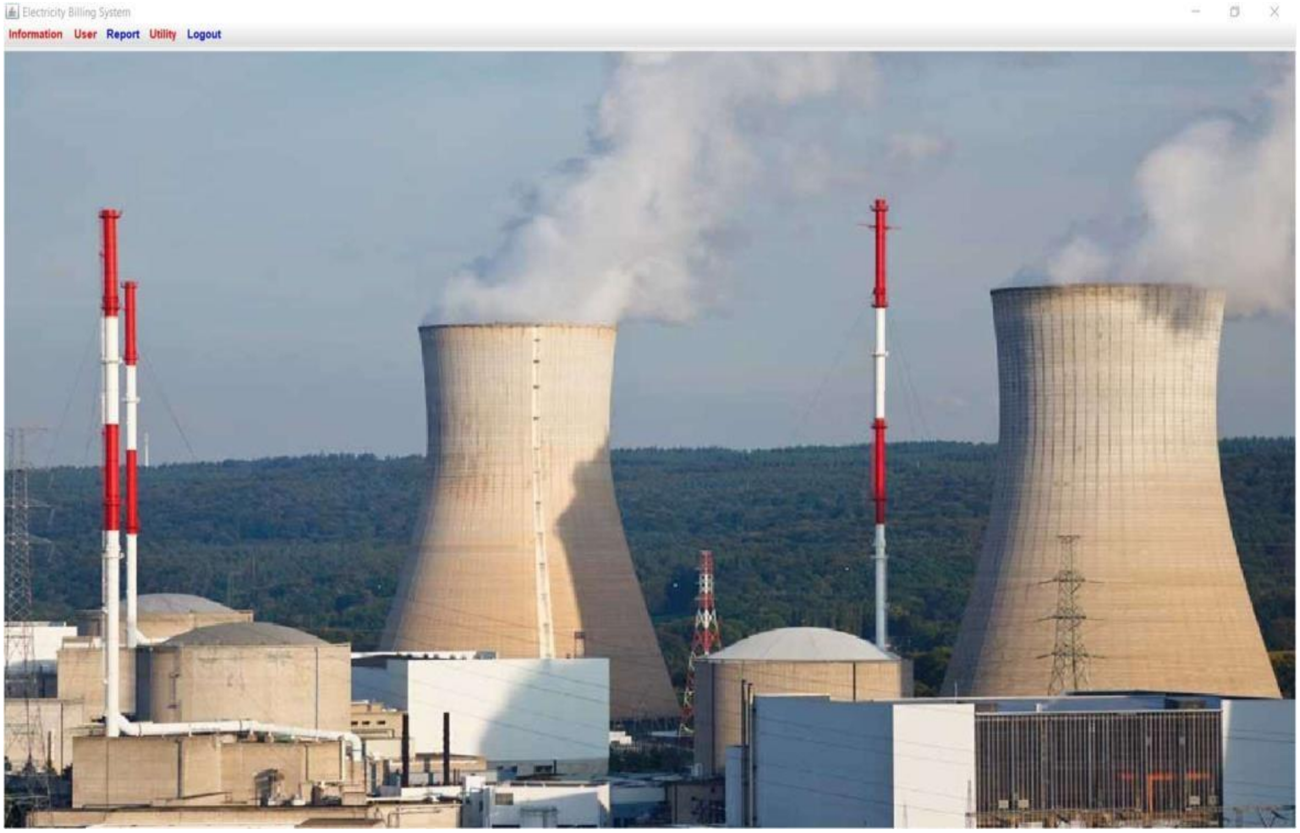


Delete Customer

Meter No	315624
Name	Ajit Kulkarni
Address	103 ABC
City	Dombivli
State	Maharashtra
Email	ajit@ymail.com
Phone	1928374655

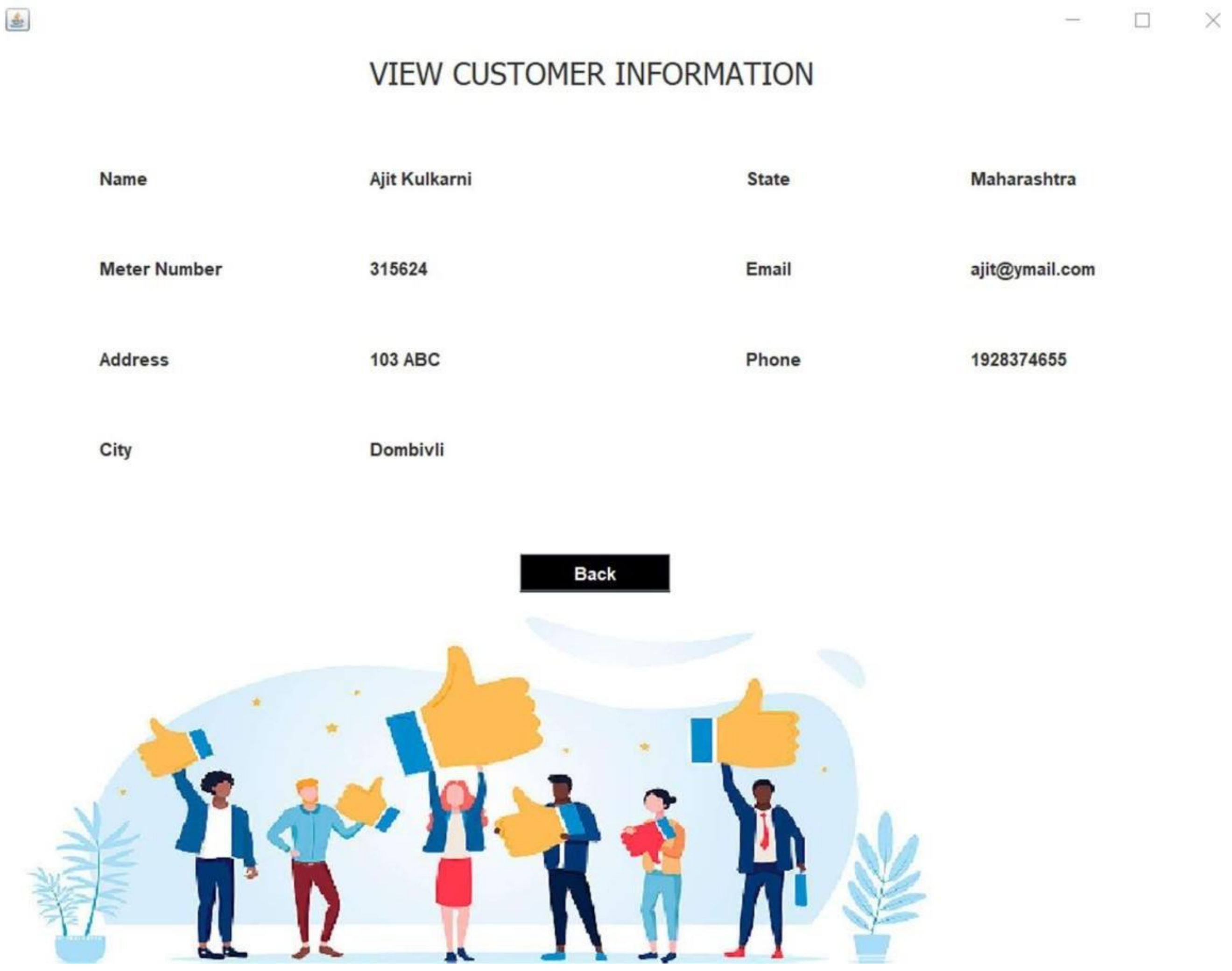
Delete Customer Screen

Here admin can delete any existing customer by choosing appropriate meter number.



Customer’s Home Screen

Customer lands on this page after successful login.

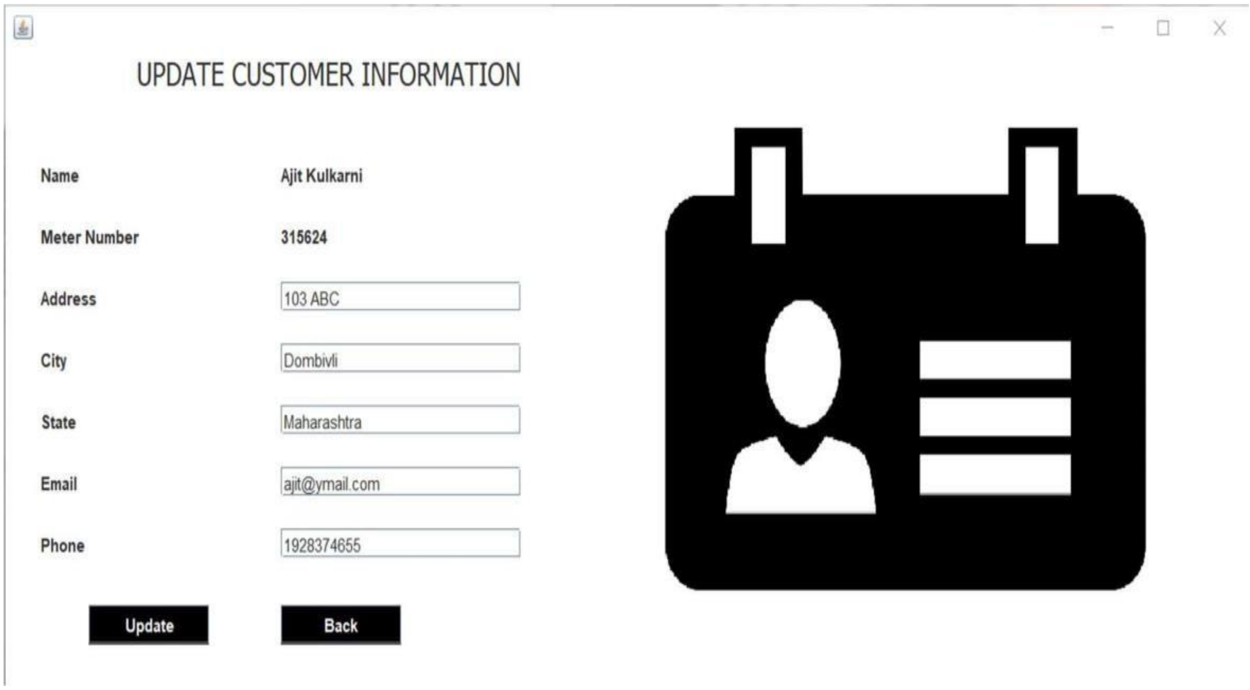


VIEW CUSTOMER INFORMATION

Name	Ajit Kulkarni	State	Maharashtra
Meter Number	315624	Email	ajit@ymail.com
Address	103 ABC	Phone	1928374655
City	Dombivli		

View Customer Info Screen

Here customer can see their entered information such as their name, meter number, address, city , state, email id and phone number.

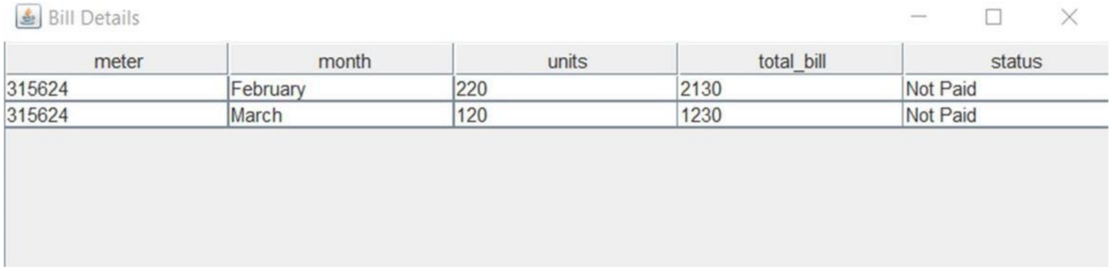


UPDATE CUSTOMER INFORMATION

Name	Ajit Kulkarni
Meter Number	315624
Address	103 ABC
City	Dombivli
State	Maharashtra
Email	ajit@ymail.com
Phone	1928374655

Update Customer Info Screen

Here customer can update their entered information if any correction is needed such as their address, city, state, email id and phone number.

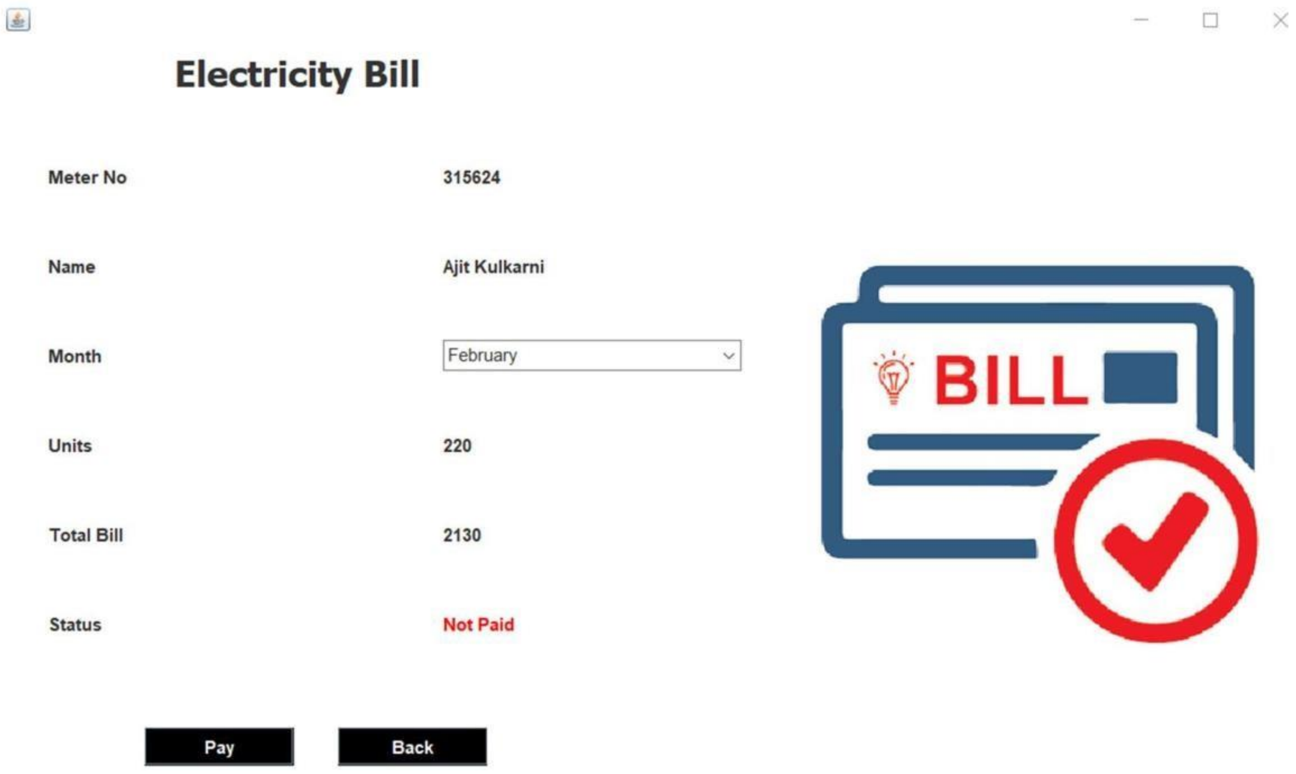


Bill Details

meter	month	units	total_bill	status
315624	February	220	2130	Not Paid
315624	March	120	1230	Not Paid

Bill Details Screen for Customers

Here every customer can check the status of their bills, whether they have paid the bills or not.

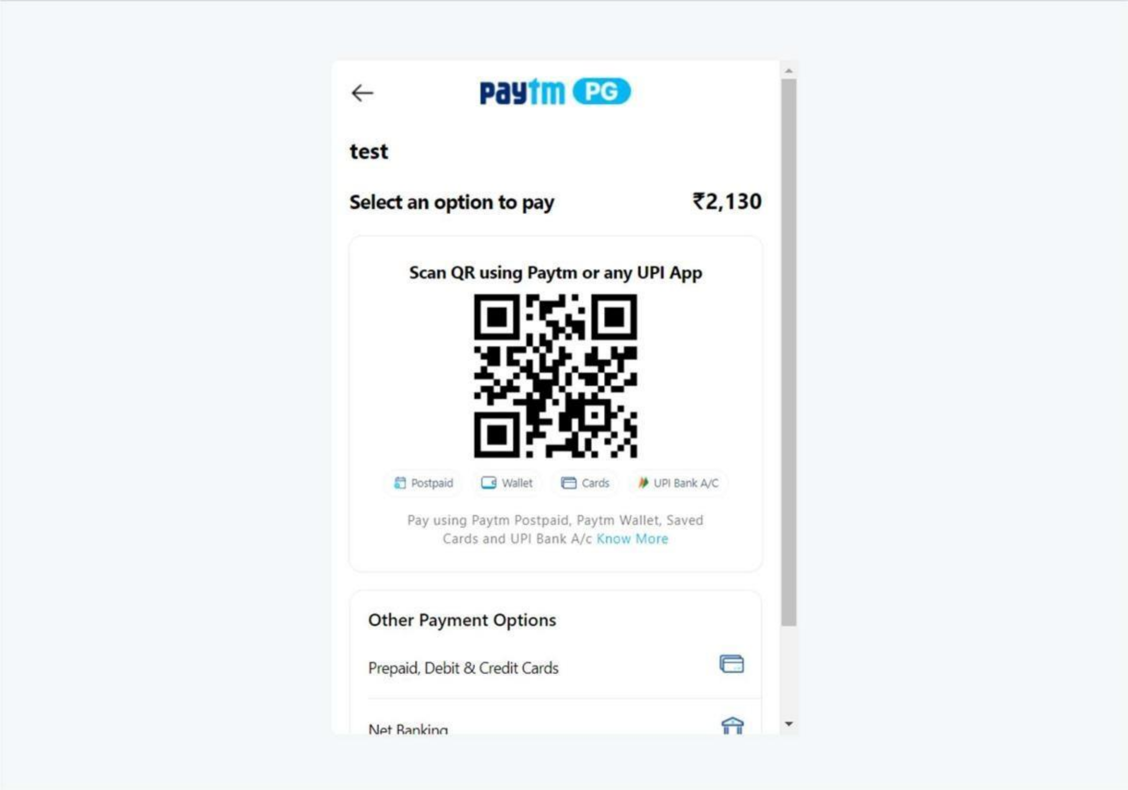


Electricity Bill

Meter No	315624
Name	Ajit Kulkarni
Month	February
Units	220
Total Bill	2130
Status	Not Paid

Pay Bill Screen

Here customers pay their bills by selecting appropriate month.



test

Select an option to pay ₹2,130

Scan QR using Paytm or any UPI App

Pay using Paytm Postpaid, Paytm Wallet, Saved Cards and UPI Bank A/c Know More

Other Payment Options

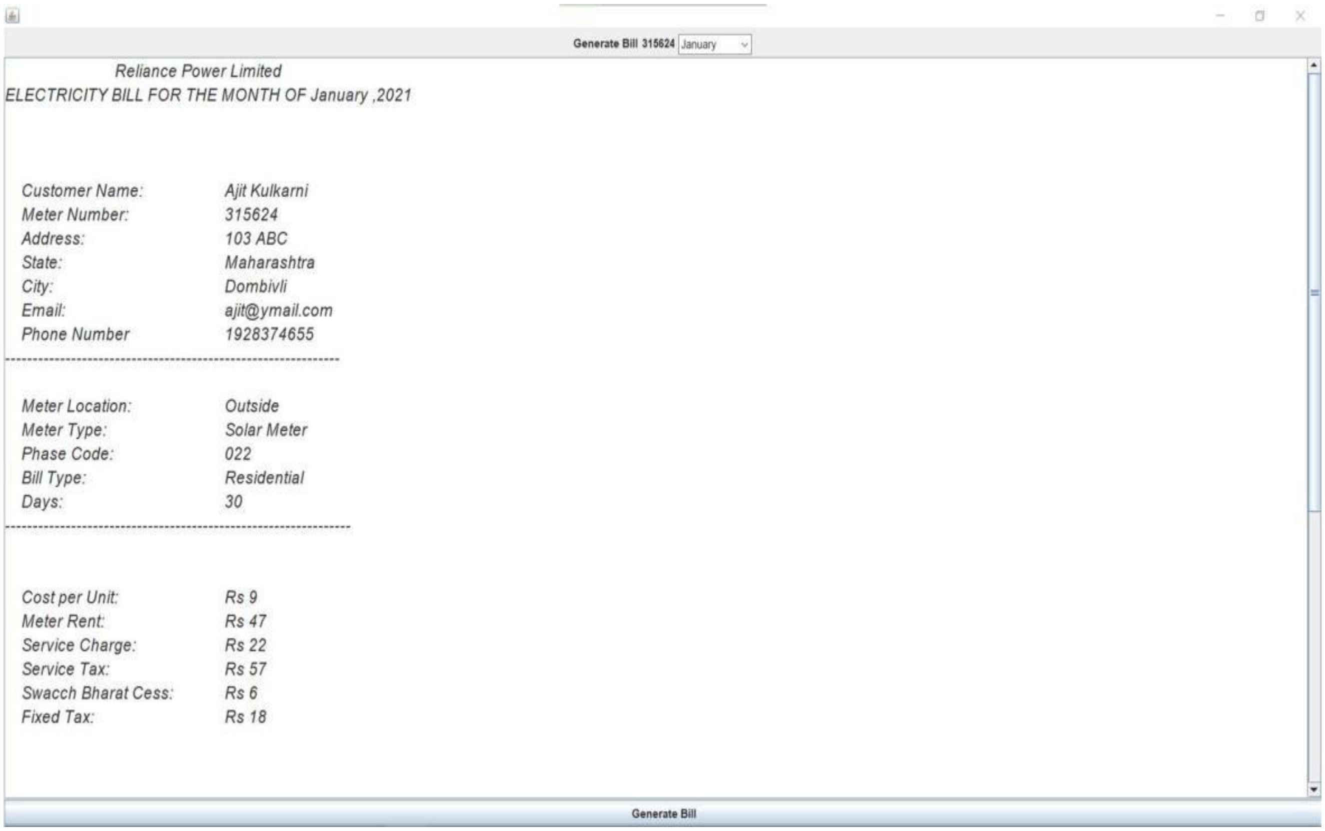
Prepaid, Debit & Credit Cards

Net Banking

Paytm Gateway Screen

Customers can pay via Paytm gateway.

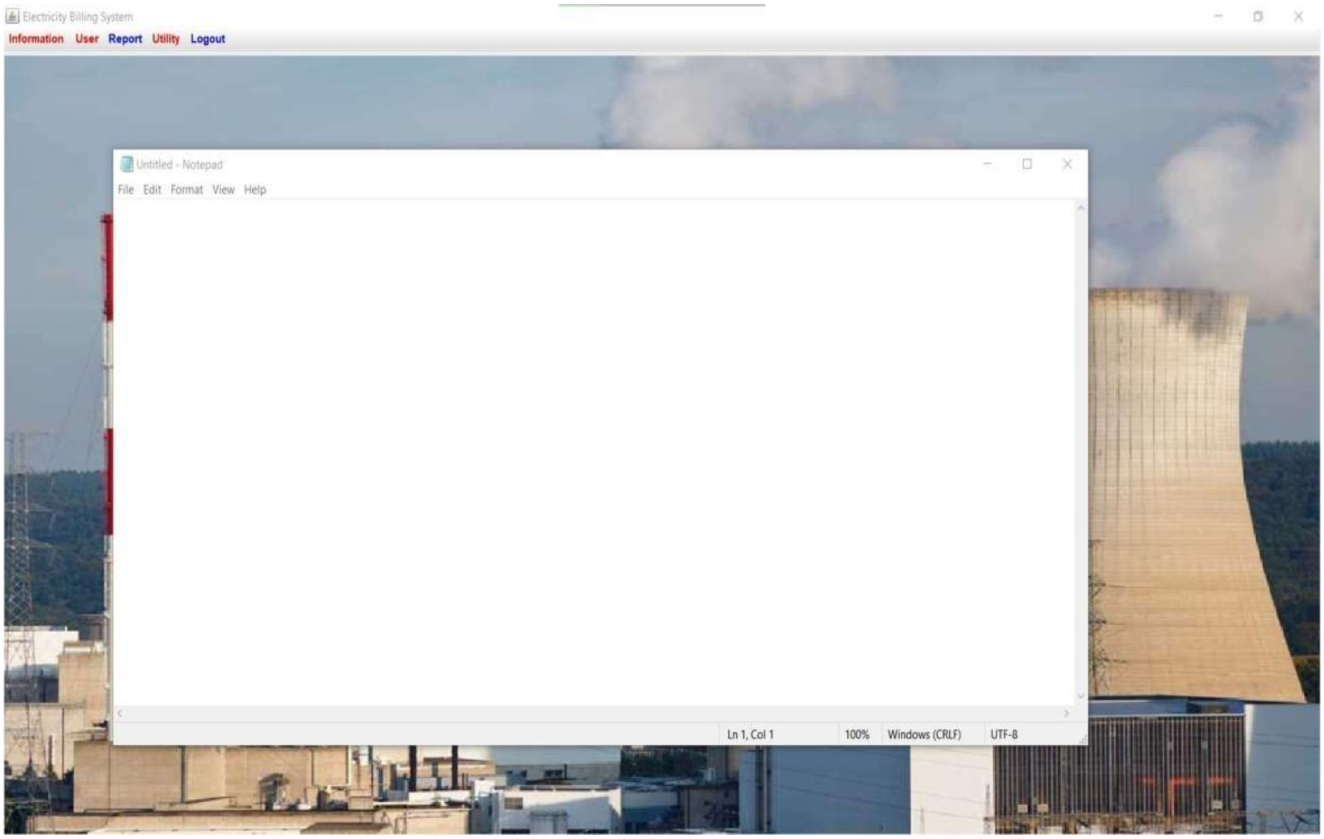
Customers can pay via net banking, wallet, Debit or credit cards.



```
              Reliance Power Limited
ELECTRICITY BILL FOR THE MONTH OF January ,2021



    Customer Name:           Ajit Kulkarni
    Meter Number:            315624
    Address:                 103 ABC
    State:                   Maharashtra
    City:                    Dombivli
    Email:                   ajit@ymail.com
    Phone Number             1928374655
-------------------------------------------------------

    Meter Location:          Outside
    Meter Type:              Solar Meter
    Phase Code:              022
    Bill Type:               Residential
    Days:                    30
---------------------------------------------------------


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

Generate/ Show
Bill Screen

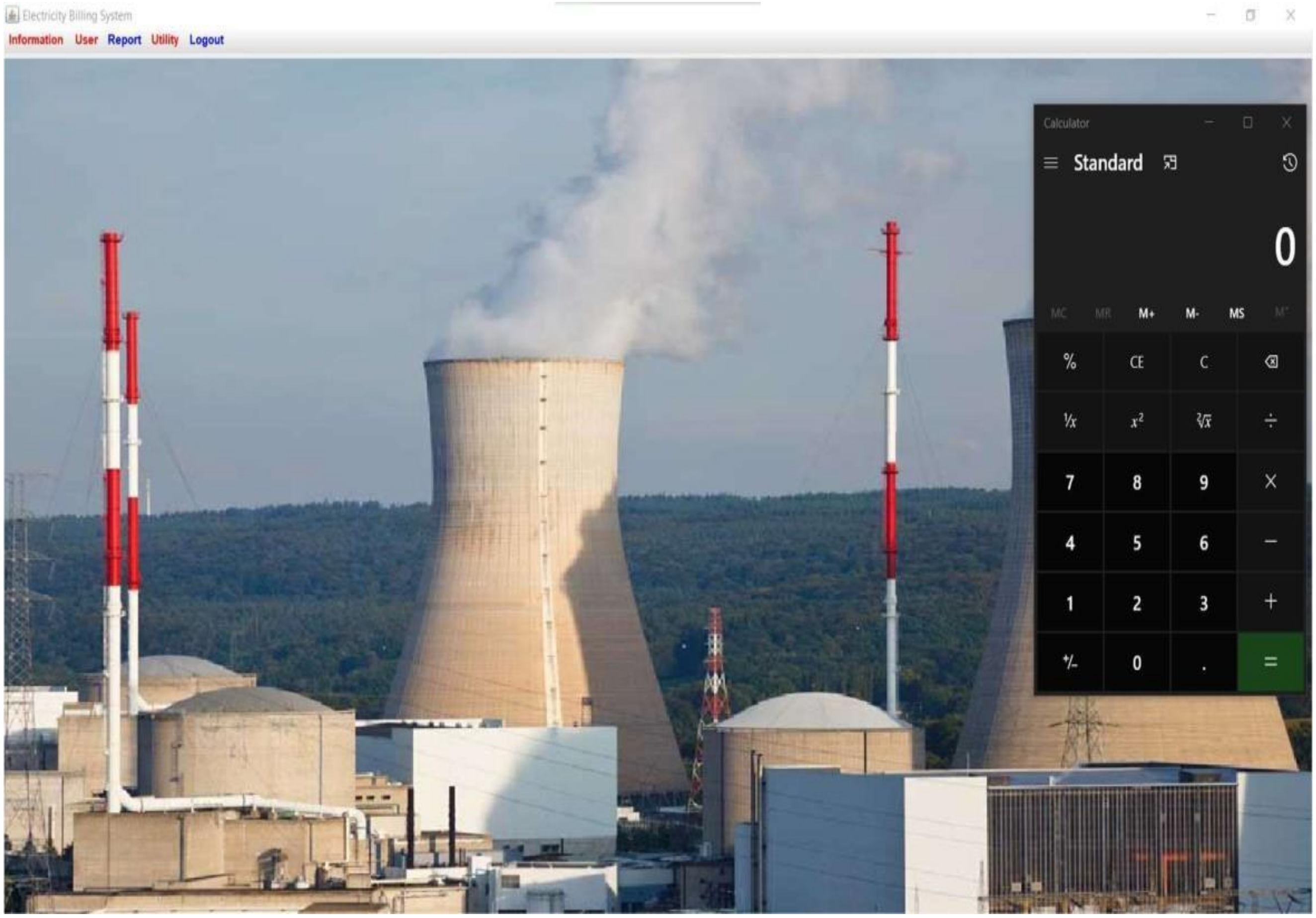
Here customer can generate / see their bill in a proper breakdown of entire amount.



Notepad Screen

When user clicks on notepad option under utilities section, its launches the notepad.

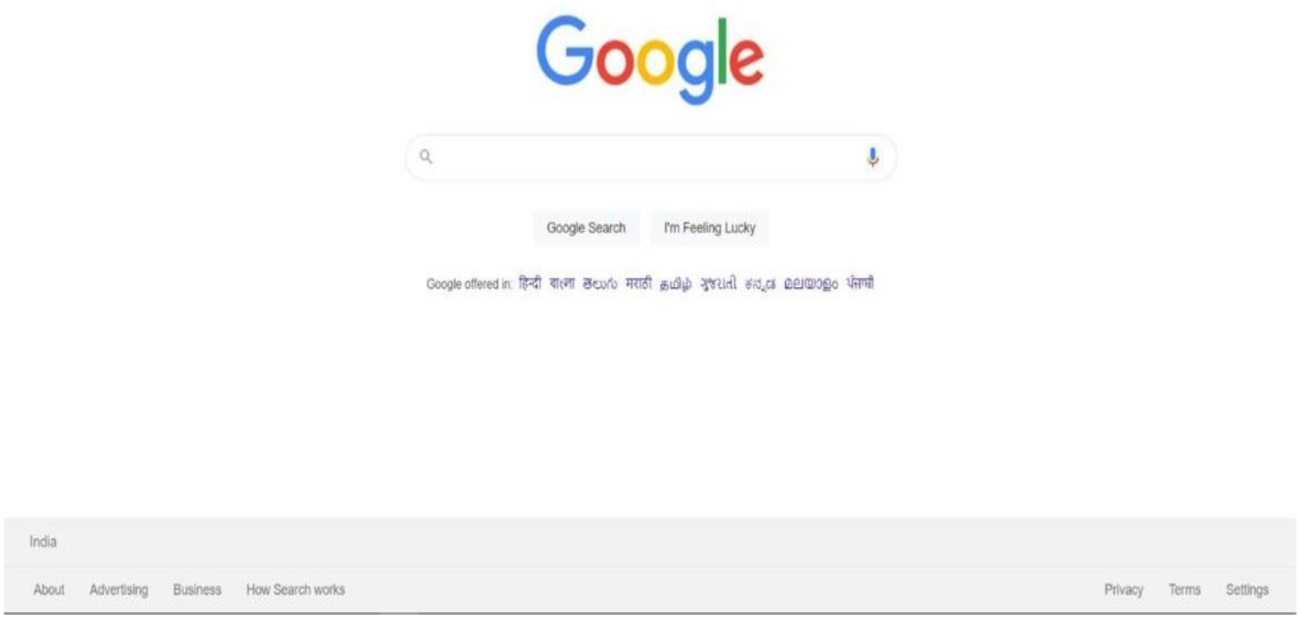
This feature is available to both Admins and Customers.



Calculator Screen

When user clicks on calculator option under utilities section, its launches the calculator.

This feature is available to both Admins and Customers.



Web Browser Screen

When user clicks on Web Browser option under utilities section, its launches the web browser.

This feature is available to both Admins and Customers.

3.4 Data Model

3.4.1 Data Dictionary :

3.4.1.1TABLES:

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.

Login Table:

Field	**Data Type**	**Description**	**constraint**	**size**
Meter_no	**Varchar**	**Customer meter number**	**Primary key**	**(20)**
Username	**Varchar**	**Contains user name**	**Null**	**(30)**
Password	**Varchar**	**Saves password**	**Null**	**(30)**
user	**Varchar**	**Store user info**	**Null**	**(30)**
Question	**Varchar**	**Questions**	**Null**	**(40)**
Answer	**Varchar**	**Answers**	**Null**	**(30)**

Customer Table:

Field	Data Type	Description	constraint	size
Name	varchar	Name of customer	Null	(30)
Meter_no	varchar	Meter number of customer	Primary key	(20)
Address	varchar	Address of customer	Null	(50)
City	varchar	City details	Null	(30)
State	varchar	State details	Null	(30)
Email	varchar	email	Null	(30)
Phone	varchar	Phone number	Null	(30)

Tax Table:

Field	Data Type	Description	constraint	size
Service_tax	int	Tax details	Primary key	(50)
Swatch Bharat css	int	Css	Null	(30)
GST	int	GST details	Null	(40)

Rent Table:

Field	Data Type	Description	constraint	size
Cost_per_Unit	int	Cost details	Primary key	(50)
Meter_rent	int	Meter rent details	Null	(30)
Service_charge	int	Other service charges	Null	(40)

Bill Table :

Field	Data Type	Description	constraint	size
Meter_no	varchar	Meter number details	Primary key	(20)
Month	varchar	Month info	Null	(20)
Unit	int	Units per months	Null	-
Total_bill	int	Total bill	Null	-
Status	varchar	Paid or unpaid status	Null	(40)

Meter Info Table :

Field	Data Type	Description	constraint	size
Meter_no	varchar	Meter number details	Primary key	(20)
Meter_location	varchar	Meter Address	Null	(30)
Meter_Type	varchar	Meter type	Null	(15)
Phase_code	int	Phase code of meter	Null	(4)
Bill_type	varchar	Bill type	Null	(20)
Days	int	Used days	Null	(20)

4. Implementation Details

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

4.2.1 **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").

4.2.2 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, column 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_chargeint (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

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 homepage

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_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 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.

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 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.

Delete Customer:

- 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.

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 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.

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.

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 back option.
- 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 and month is selected.
- Generate the details by clicking on generatebill button.

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

5. Output & Reports 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 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.

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.

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.

5.2 Black Box testing

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.

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 pho ne number	_

Positive test case for phone number insertion

Function Name	Input	Expected Output	Error	Resolved
Input Phone Number	9897778988	Expected Output is Seen	_	_

Negative test case for email insertion

Function Name	Input	Expected Output	Error	Resolved
Input email	Sai1.i n	Email is invalid	Email is not in a format given	Consume ()

Positive test case for email insertion

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

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 ()

Positive test case for customer name insertion

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

5.3 White Box Testing

5.3.1 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.

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	_	_

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	_	_

5.4 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.

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 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 ❖New Admin	

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	

6.Conclusion and Recommendations

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.

7. Future Scope

SOFTWARE SCOPE:

- **Extensibility**: This software is extendable in ways that its original developers may not expect. The following principles enhances extensibility like hide data structure, avoid traversing multiple
Links or methods avoid case statements on object type and distinguish public and private operations.

- **Reusability**: Reusability is possible as and when require in this application. We can update it next version. Reusable software reduces design, coding and testing cost by amortizing effort Over several designs. Reducing the amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both types of reusability:
Sharing of newly written code within a project and reuse of previously written code on new projects.
- **Understand ability:** A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method, which small and coherent helps to accomplish this.
- **Cost-effectiveness:** Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement.

Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

BIBLIOGRAPHY

REFERENCES

Book Reference

Database Management Systems 3rd Edition by Raghu Ramakrishnan.

Websites

https://www.youtube.com/watch?v=iWitVuW2D1o&t=4s

http://www.github.com

www.stackoverflow.com

www.google.com

http://www.javatpoint.com/