Project Report

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.

Proposed 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 the 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.

ANALYSIS AND SYSTEM REQUIREMENT

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.

Software & Hardware Requirements

Hardware Requirements:

- ➢ Hardware Specification: - Intel Core i3 or higher (2.10 GHz)
- ➢ System Bus: -64 bits
- ➢ RAM: -16GB
- ➢ HDD: -2TB
- ➢ Monitor: -LCD Monitor
- ➢ Keyboard: -Standard keyboard
- ➢ Mouse: -Compatible mouse

Software Requirements:

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

SYSTEM DESIGN AND MODELING

Preliminary Design

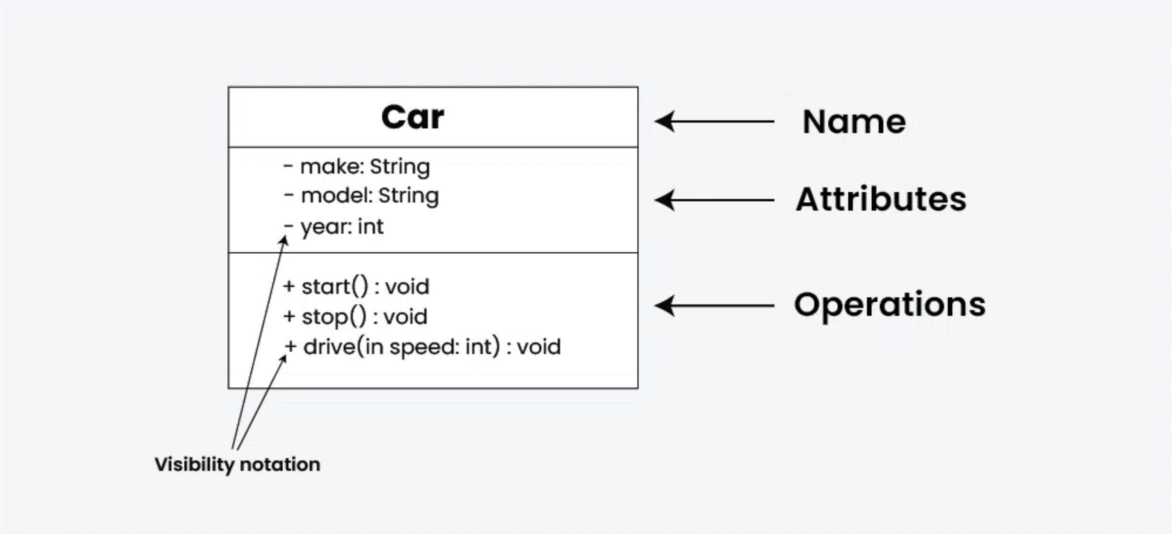
System design is an abstract representation of a system component and their relationship and which describe the aggregated functionally and performance of the system. It is also the plan or blueprint for how to obtain answer to the question being asked. The design specifies various type of approach.

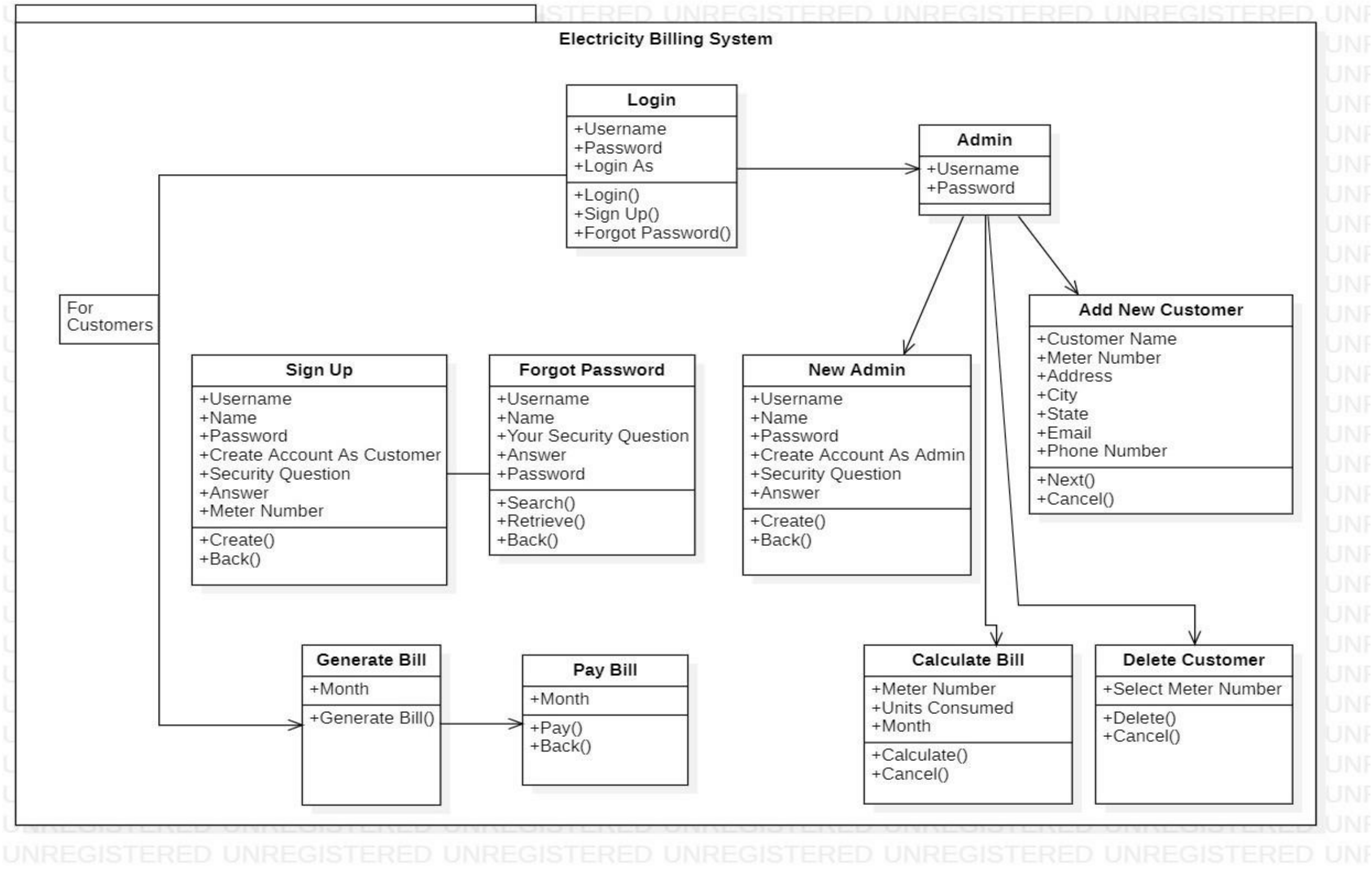
Database design is one of the most important factors to keep in mind if you are concerned with application performance management. By designing your database to be efficient in each call it makes and to effectively create rows of data in the database, you can reduce the amount of CPU needed by the server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

UML Diagrams :-

Class Diagram: -

A class diagram is a visual representation of classes and their relationships, attributes, specifications, and behaviours. Class diagrams are a type of structure diagram in the Unified Modelling Language (UML) and are commonly used by software engineers to document software architecture.



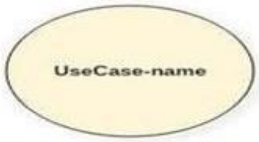


Use Case Diagrams :-

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases.

Use-case diagram notations :

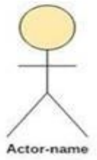
Use cases are used to represent high- level functionalities and how the user will handle the system. A use case represents a distinct functionality of a system, a component, a package, or a class. It is denoted by an oval shape with the name of a use case written inside the oval shape.



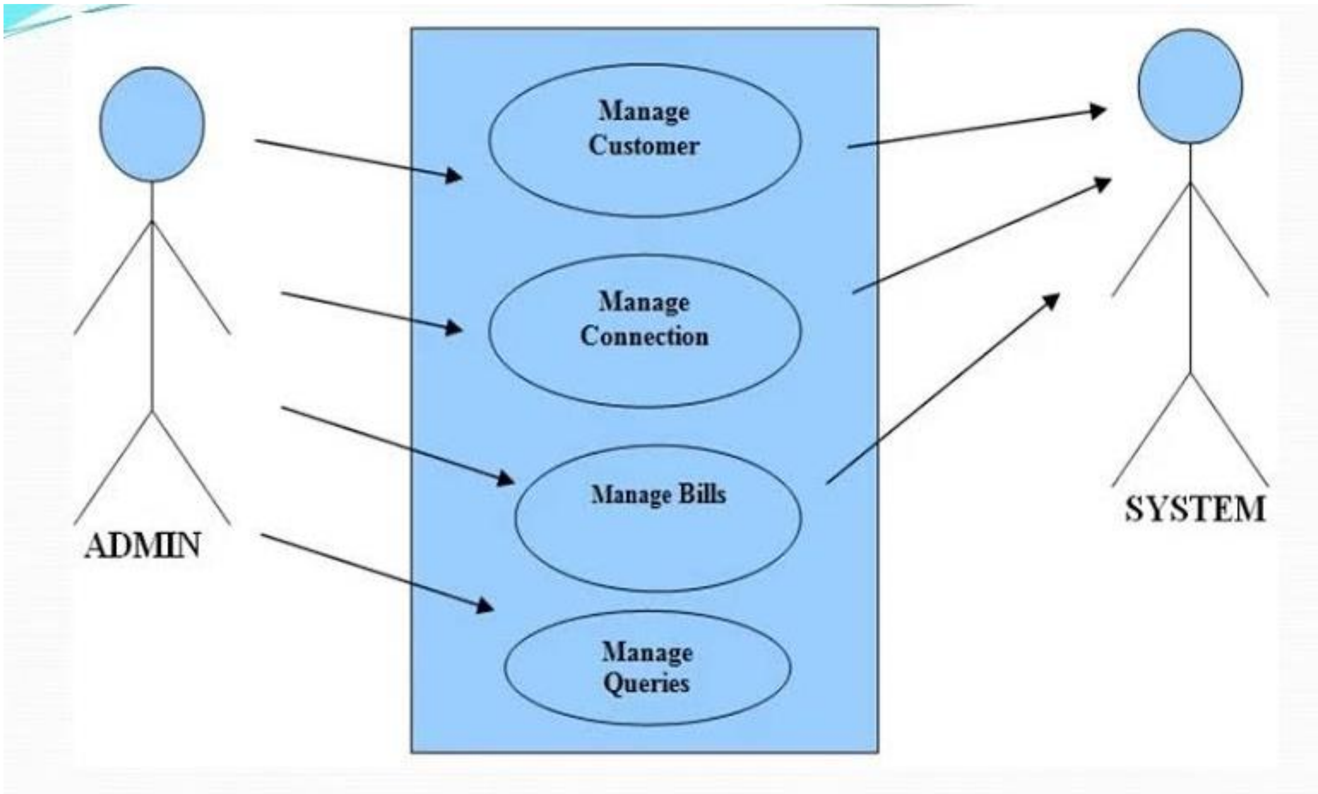
UML Use-Case Notation

Actor:

It is used inside use case diagrams. The actor is an entity that interacts with the system. A user is best example of an actor. An actor is an entity that initiates the use case from outside the scope of a use case.



UML Actor Notation



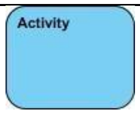
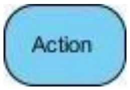
Use Case diagram for Admin and System

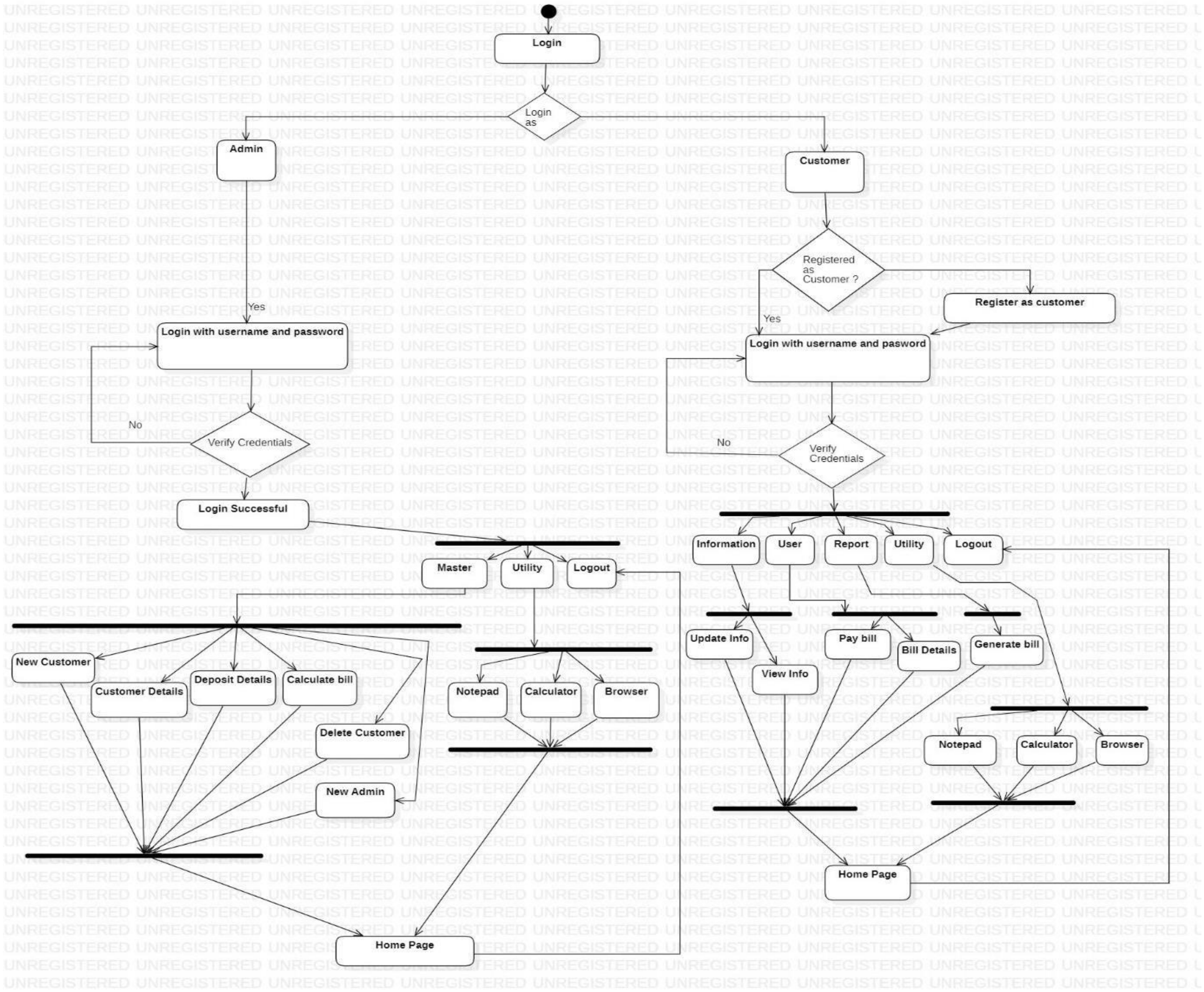
Activity Diagram: -

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modelling how a collection of use cases coordinate to represent business workflows

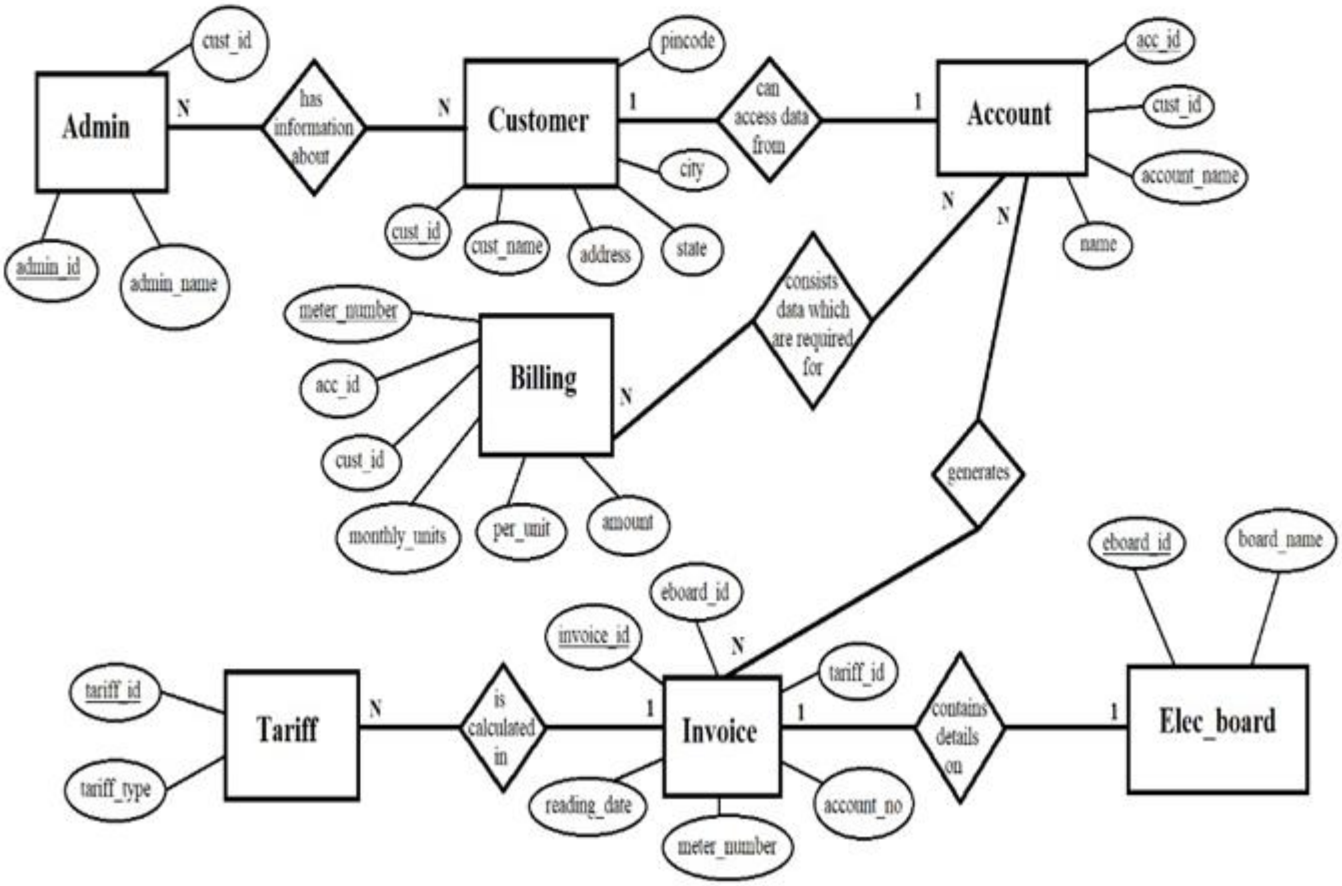
1. Identify candidate use cases, through the examination of business workflows
2. Identify pre- and post-conditions (the context) for use cases
3. Model workflows between/within use cases
4. Model complex workflows in operations on objects
5. Model in detail complex activities in a high level activity Diagram

Activity Diagram Notation Summary

Notation Description	UML Notation
Activity Is used to represent a set of actions	
Action A task to be performed	
Control Flow Shows the sequence of execution	→
Initial Node Portrays the beginning of a set of actions or activities	●
Activity Final Node Stop all control flows and object flows in an activity (or action)	◉


Login
Login as
Admin
Customer
Yes
Login with username and password
No
Verify Credentials
Login Successful
Master
Utility
Logout
New Customer
Customer Details
Deposit Details
Calculate bill
Delete Customer
New Admin
Notepad
Calculator
Browser
Home Page
Registered as Customer ?
Register as customer
Yes
Login with username and pasword
No
Verify Credentials
Information
User
Report
Utility
Logout
Update Info
View Info
Pay bill
Bill Details
Generate bill
Notepad
Calculator
Browser
Home Page

E – R Diagram :-
E – R Diagram stands for Entity Relationship .
It shows the relationship between the entities (table).



Schema Diagram

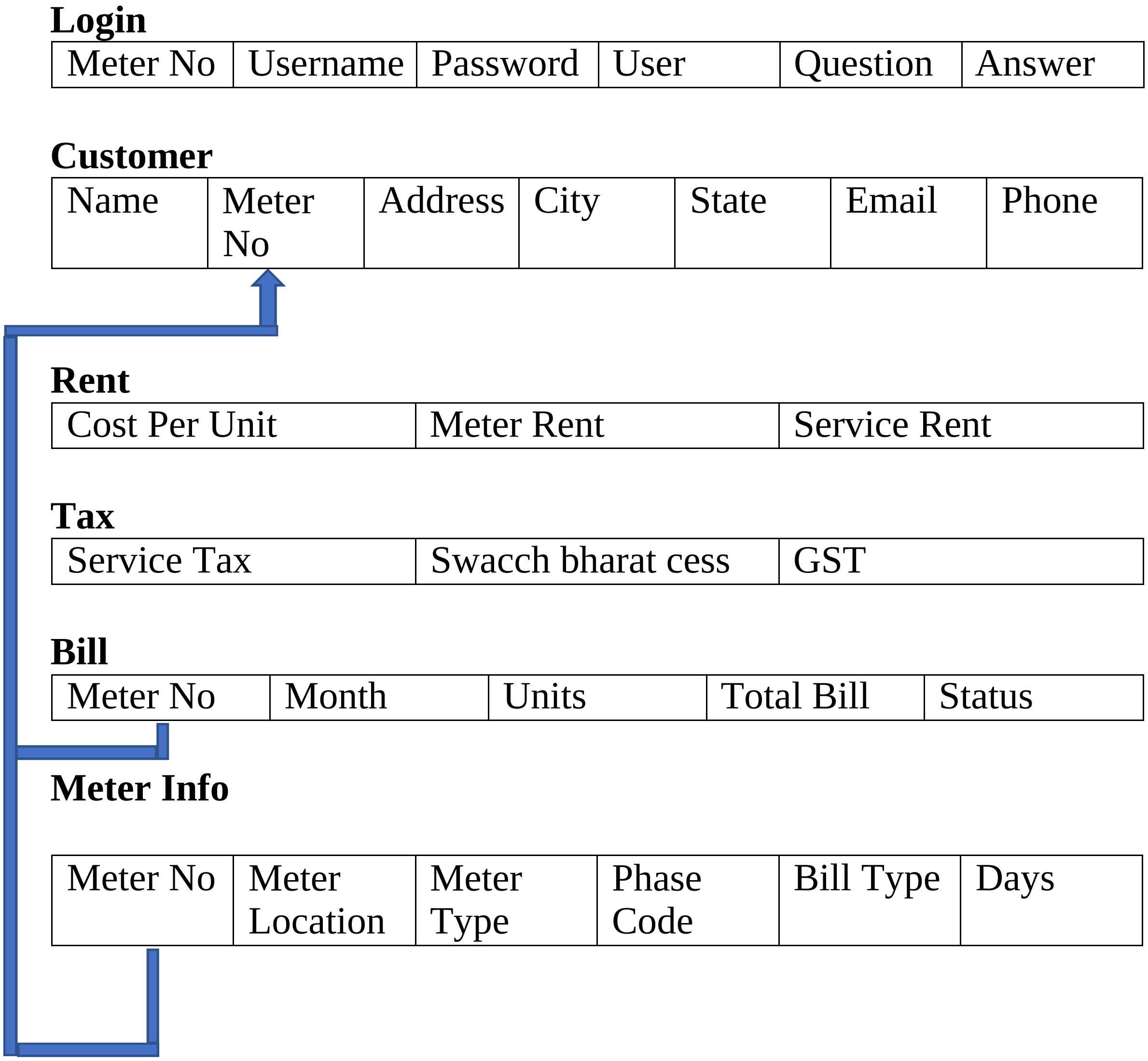
Database schema is described as database connections and constraints. It contains attributes. Every database has a state instances represent current set of databases with values. There are different types of keys in a database schema.

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.

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	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

3.1 Normalization

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

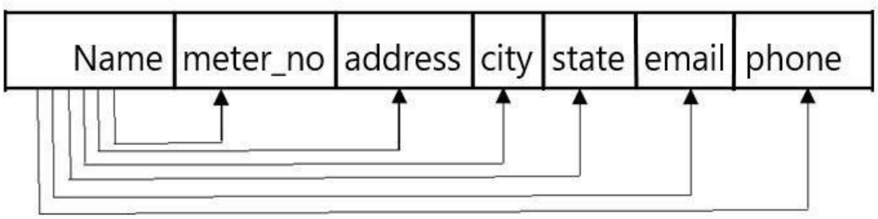
Let’s discuss about anomalies first then we will discuss normal forms with examples. Anomalies in DBMS There are three types of anomalies that occur when the database is not normalized. These are –Insertion, update and deletion anomaly.

3.1.1 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



3.1.2 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.

3.1.3 Third normal form(3NF)

As per the rule of third normal form,

✓ Database must be in First and Second Normal Form.

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

IMPLIMENTATION

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.

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, 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_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

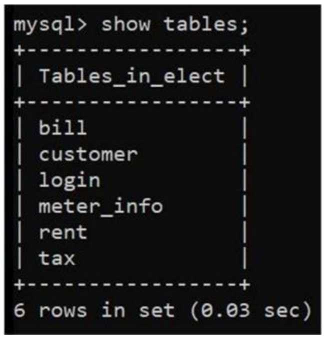
    varchar (10), meter_type varchar (15), phase_code int (5), bill_type varchar (10), days int (5));

DATABASE SNAPSHORT :-

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.

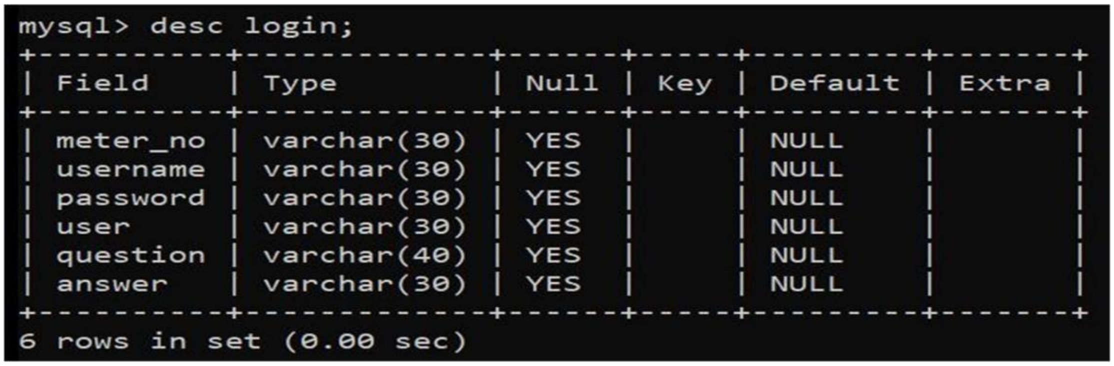
- ✓ The login is used to store the details of login’s admin and customer with meter_no.
- ✓ The customer is used to store details of customer.
- ✓ The tax is used to store tax values.
- ✓ The rent is used to store rent values.
- ✓ The bill is used to store details of bill of meter.
- ✓ The meter_info is used to store information of meter placed.



```
mysql> show tables;
+----------------+
| Tables_in_elect |
+----------------+
| bill           |
| customer       |
| login          |
| meter_info     |
| rent           |
| tax            |
+----------------+
6 rows in set (0.03 sec)
```

List of tables

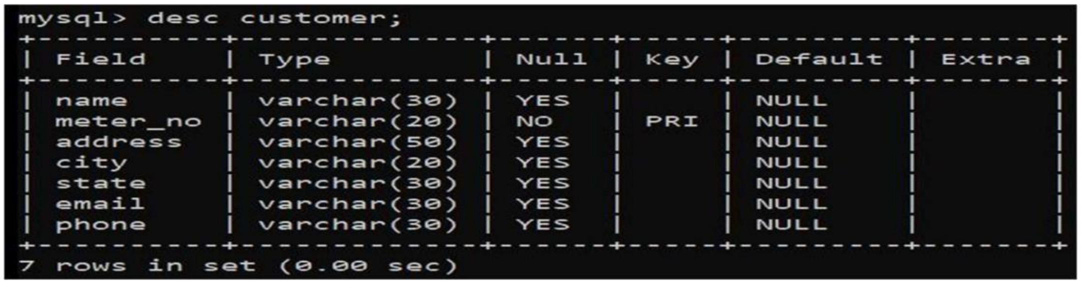
Login Table:



```
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)
```

login table description

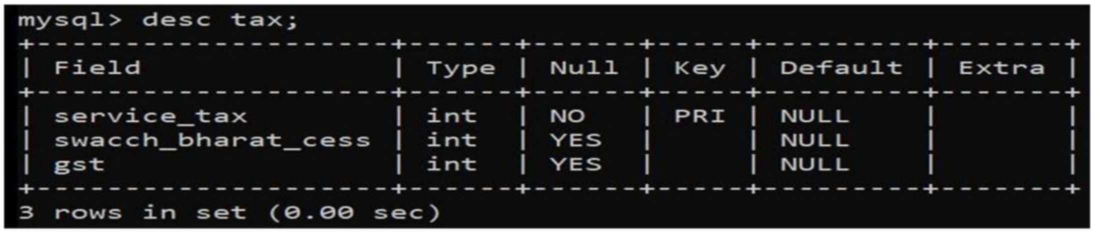
Customer Table:



```
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)
```

customer table description

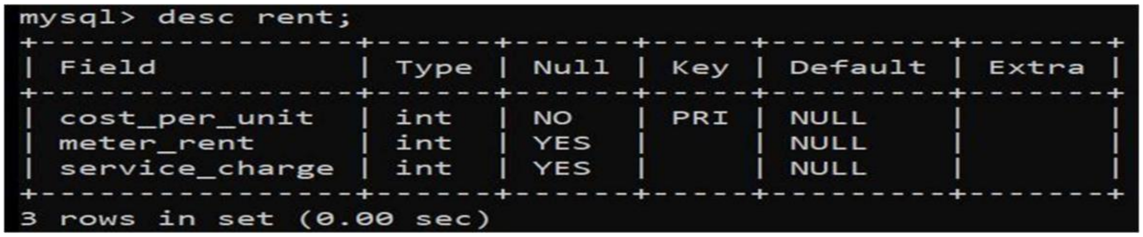
Tax Table:



```
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)
```

tax table description

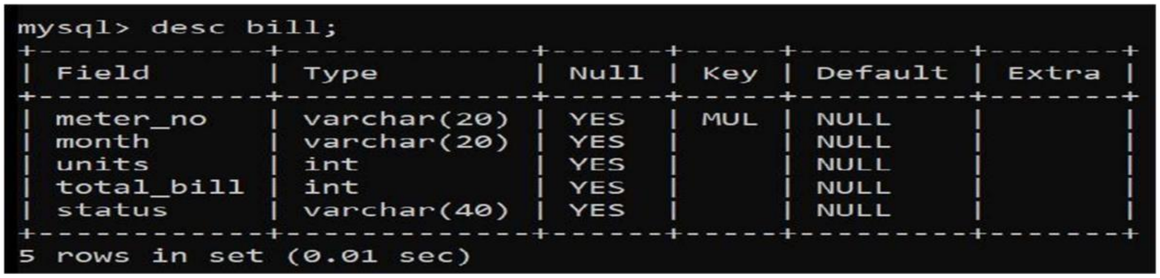
Rent Table:



```
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)
```

rent table description

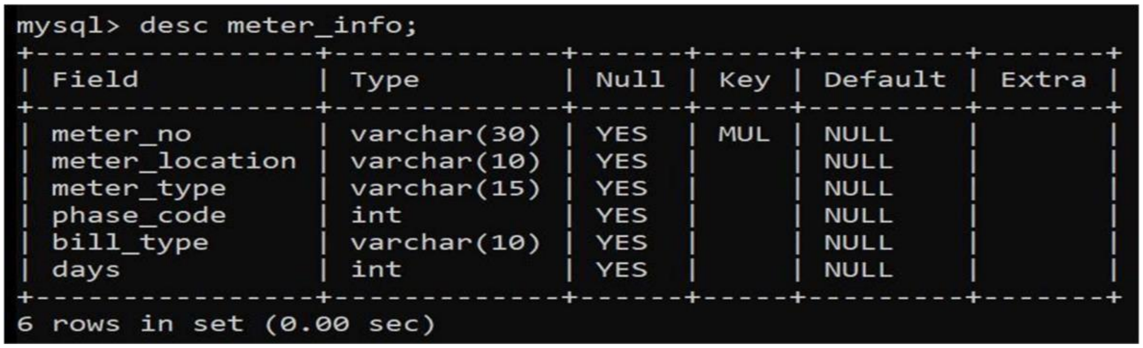
Bill Table:



```
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)
```

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)
```

meter_info table description

GUI SNAPSHORT :-

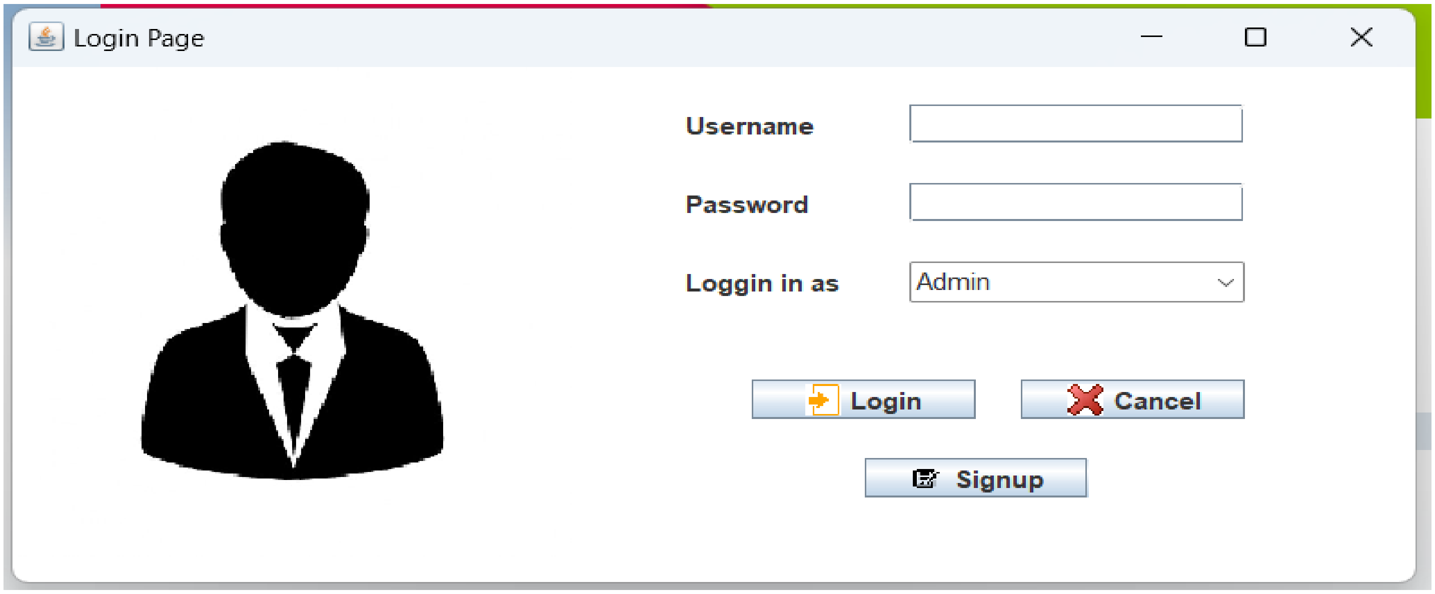
LOGIN FILE Code : -

```
  package electricity.billing.system;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Login extends JFrame implements ActionListener{

    JButton login, cancel, signup;
    JTextField username, password;
    Choice logginin;
    Login() {
        super("Login Page");
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel lblusername = new JLabel("Username");
        lblusername.setBounds(300, 20, 100, 20);
        add(lblusername);

        username = new JTextField();
        username.setBounds(400, 20, 150, 20);
        add(username);

        JLabel lblpassword = new JLabel("Password");
        lblpassword.setBounds(300, 60, 100, 20);
        add(lblpassword);
    password = new JTextField();
    password.setBounds(400, 60, 150, 20);
    add(password);

    JLabel loggininas = new JLabel("Loggin in as");
    loggininas.setBounds(300, 100, 100, 20);
    add(loggininas);

    logginin = new Choice();
    logginin.add("Admin");
    logginin.add("Customer");
    logginin.setBounds(400, 100, 150, 20);
    add(logginin);

    ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icon/login.png"));
    Image i2 = i1.getImage().getScaledInstance(16, 16,
Image.SCALE_DEFAULT);
    login = new JButton("Login", new ImageIcon(i2));
    login.setBounds(330, 160, 100, 20);
    login.addActionListener(this);
    add(login);

    ImageIcon i3 = new
ImageIcon(ClassLoader.getSystemResource("icon/cancel.jpg"));
    Image i4 = i3.getImage().getScaledInstance(16, 16,
Image.SCALE_DEFAULT);
    cancel = new JButton("Cancel", new ImageIcon(i4));
    cancel.setBounds(450, 160, 100, 20);
    cancel.addActionListener(this);
    add(cancel);

    ImageIcon i5 = new
ImageIcon(ClassLoader.getSystemResource("icon/signup.png"));
    Image i6 = i5.getImage().getScaledInstance(16, 16,
Image.SCALE_DEFAULT);
    signup = new JButton("Signup", new ImageIcon(i6));
    signup.setBounds(380, 200, 100, 20);
    signup.addActionListener(this);
    add(signup);

    ImageIcon i7 = new
ImageIcon(ClassLoader.getSystemResource("icon/second.jpg"));
    Image i8 = i7.getImage().getScaledInstance(250, 250,
Image.SCALE_DEFAULT);
    ImageIcon i9 = new ImageIcon(i8);
```

```
        JLabel image = new JLabel(i9);
        image.setBounds(0, 0, 250, 250);
        add(image);

        setSize(640, 300);
        setLocation(400, 200);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == login) {
            String susername = username.getText();
            String spassword = password.getText();
            String user = logginin.getSelectedItem();

            try {
                Conn c = new Conn();
                String query = "select * from login where username = '"+susername+"'
and password = '"+spassword+"' and user = '"+user+"'";

                ResultSet rs = c.s.executeQuery(query);

                if (rs.next()) {
                    String meter = rs.getString("meter_no");
                    setVisible(false);
                    new Project(user, meter);
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Login");
                    username.setText("");
                    password.setText("");
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
        } else if (ae.getSource() == cancel) {
            setVisible(false);
        } else if (ae.getSource() == signup) {
            setVisible(false);

            new signup();
        }
    }

    public static void main(String[] args) {
        new Login();
        }
    }
```

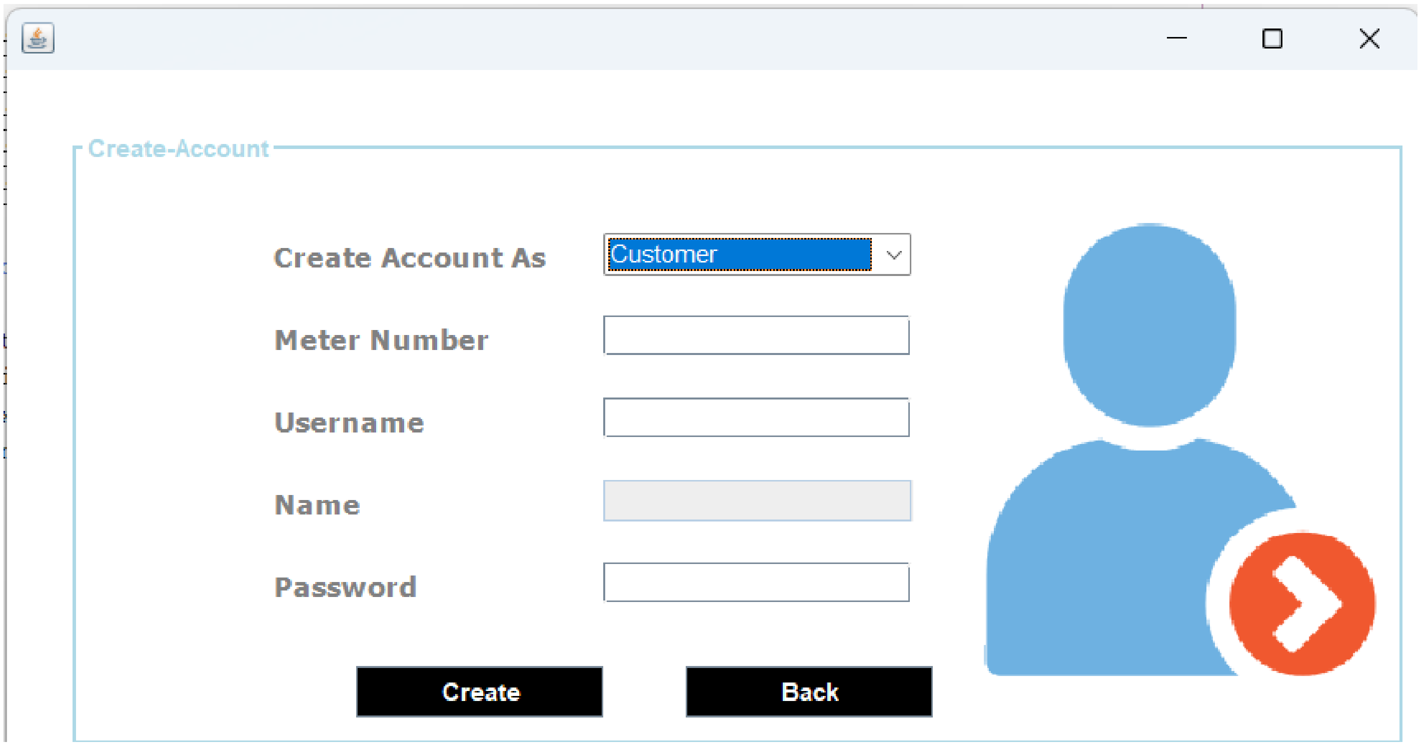
INTERFACE :-

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.



Loggin in as Admin

Login page for Admin and Customer



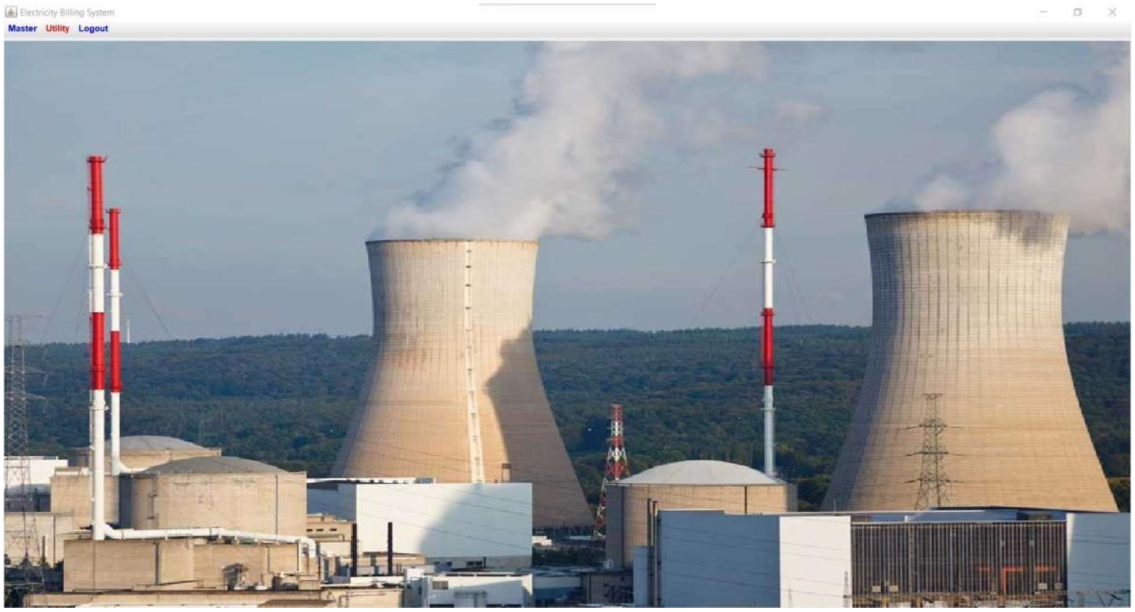
Create Account As Customer

SIGN UP SCREEN

Here New customers will signup to access their accounts.

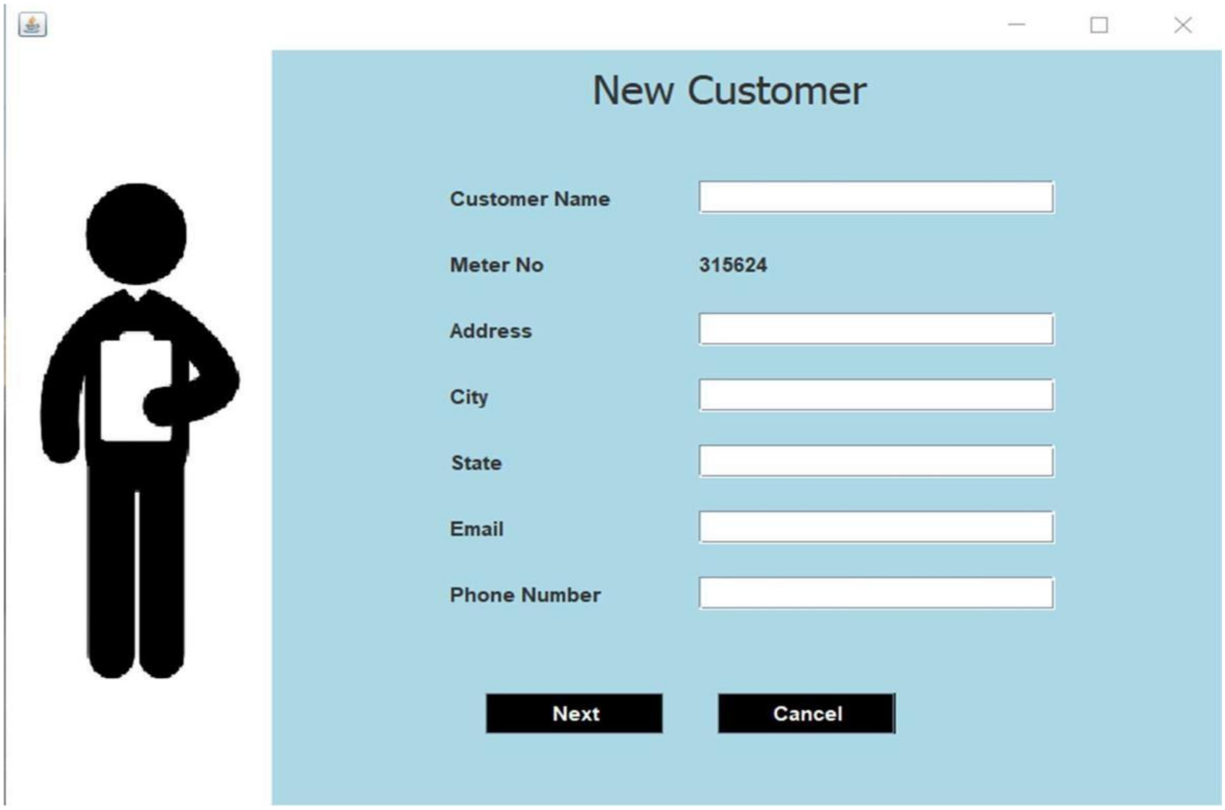
User have to enter username, name, password, choose security question and answer to that question.

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



ADMIN HOME SCREEN

Admin lands on this page after successful login.



Customer Name

Meter No 315624

Address

City

State

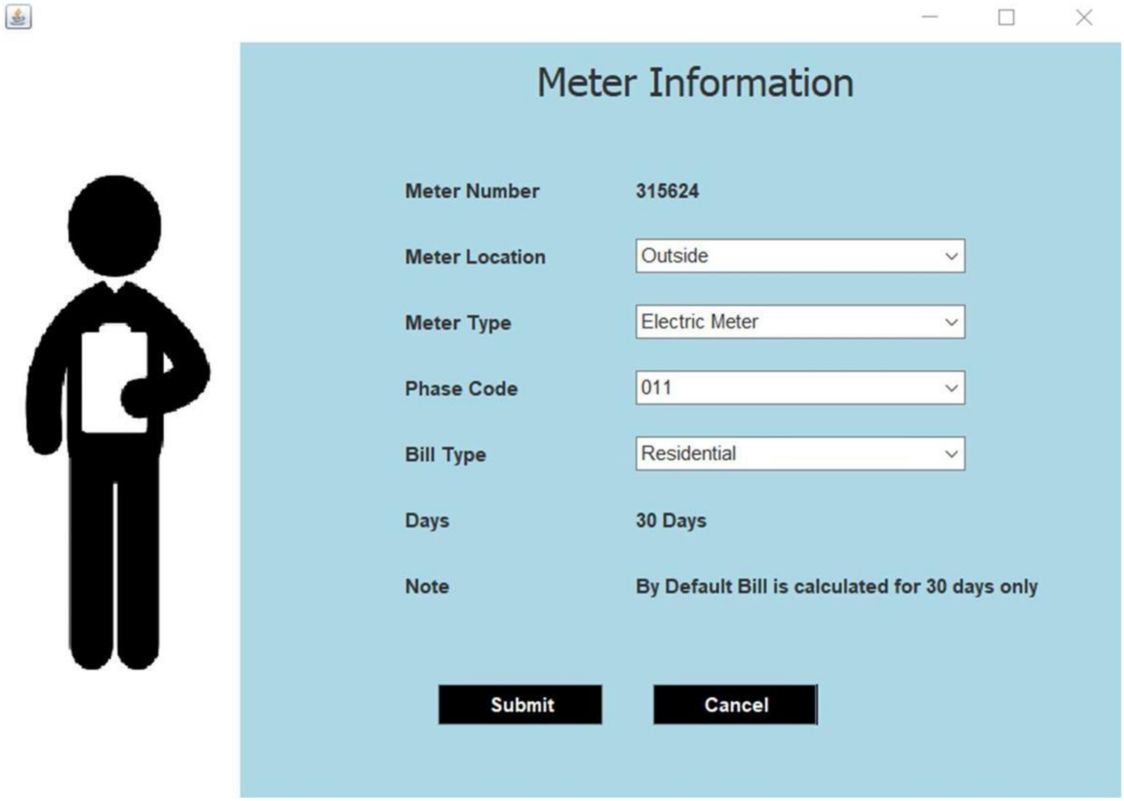
Email

Phone Number

New Customer Screen

Here admin registers new users.

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



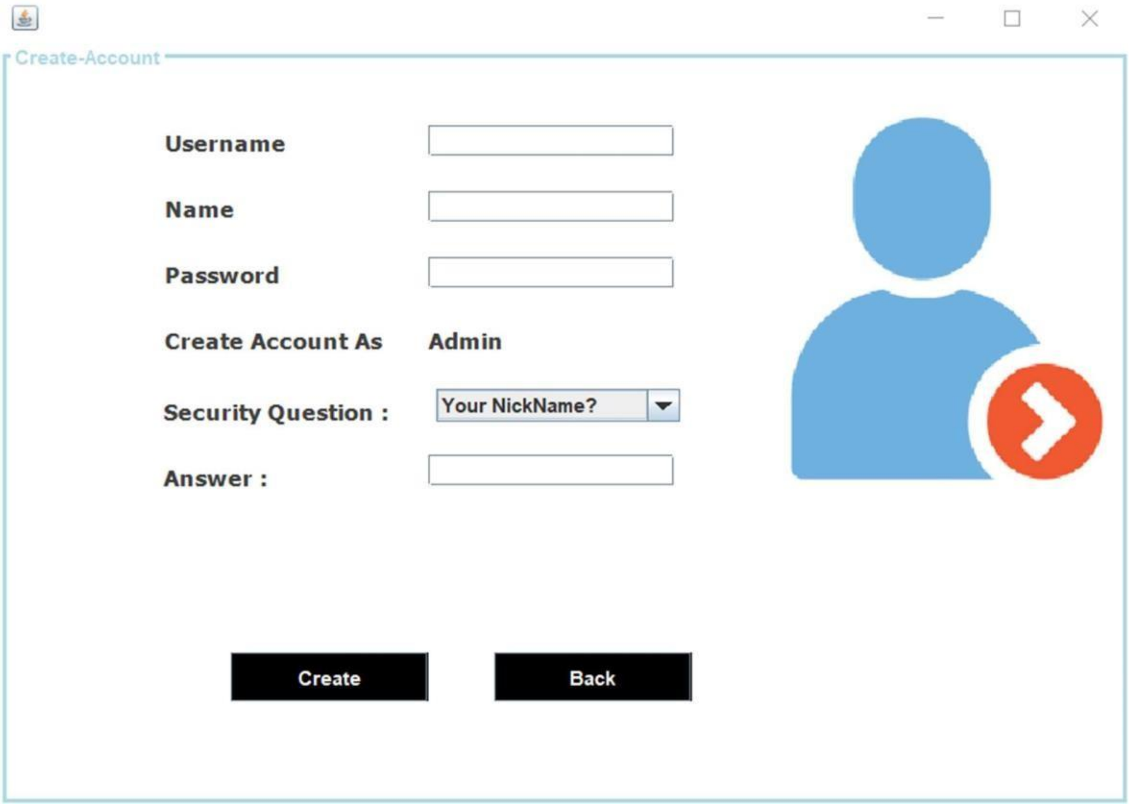
Meter Information

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 As Admin

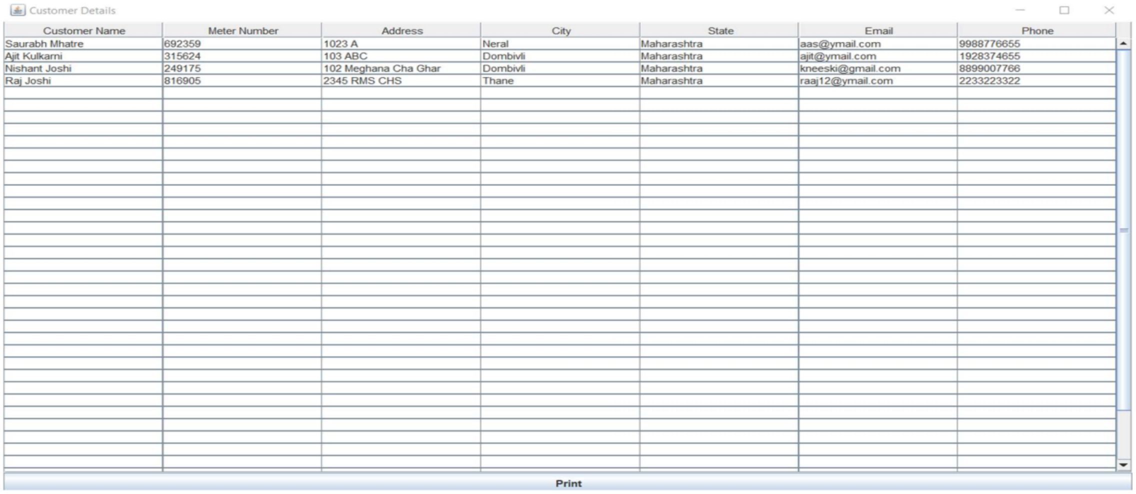
Security Question : Your NickName?

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.

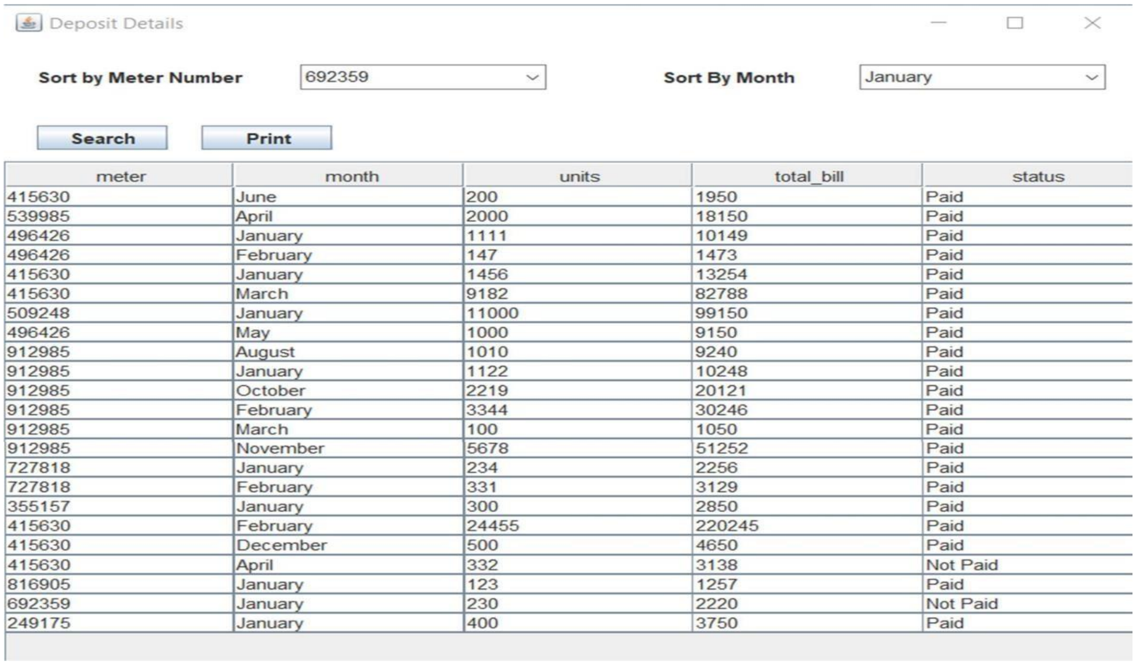
Admin can be added only by existing admins via Admin module only.



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.



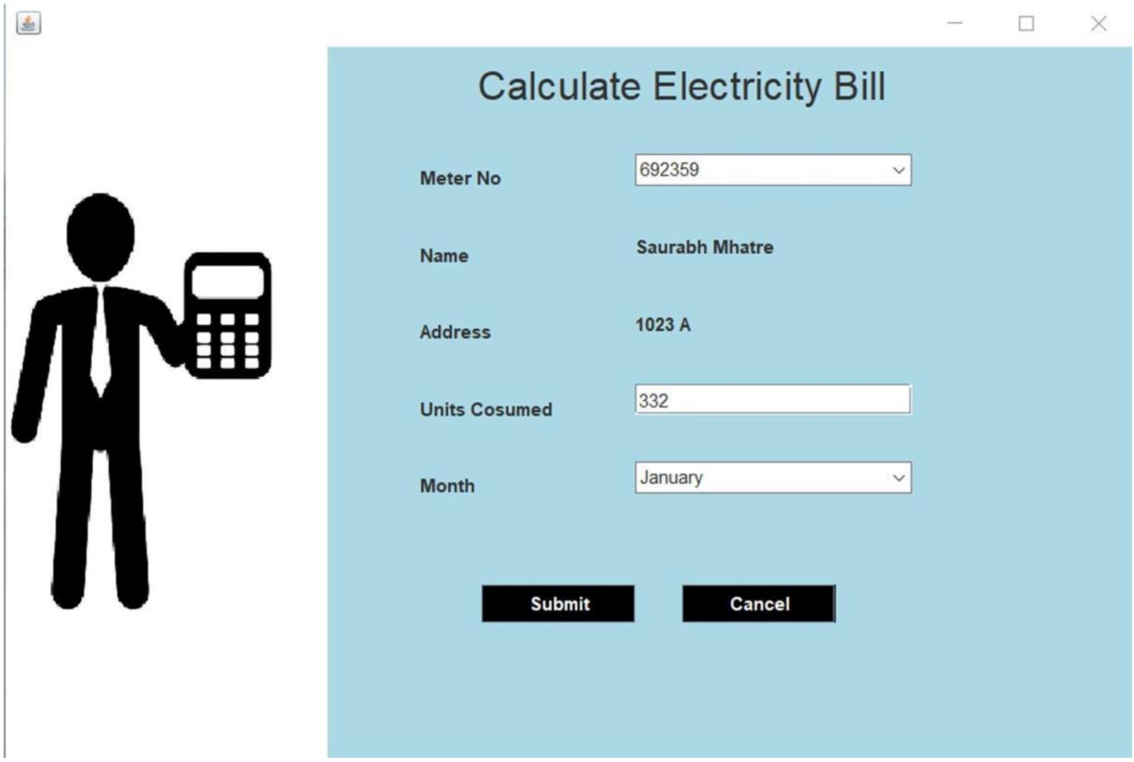
meter	month	units	total_bill	status
415630	June	200	1950	Paid
539985	April	2000	18150	Paid
496426	January	1111	10149	Paid
496426	February	147	1473	Paid
415630	January	1456	13254	Paid
415630	March	9182	82788	Paid
509248	January	11000	99150	Paid
496426	May	1000	9150	Paid
912985	August	1010	9240	Paid
912985	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.

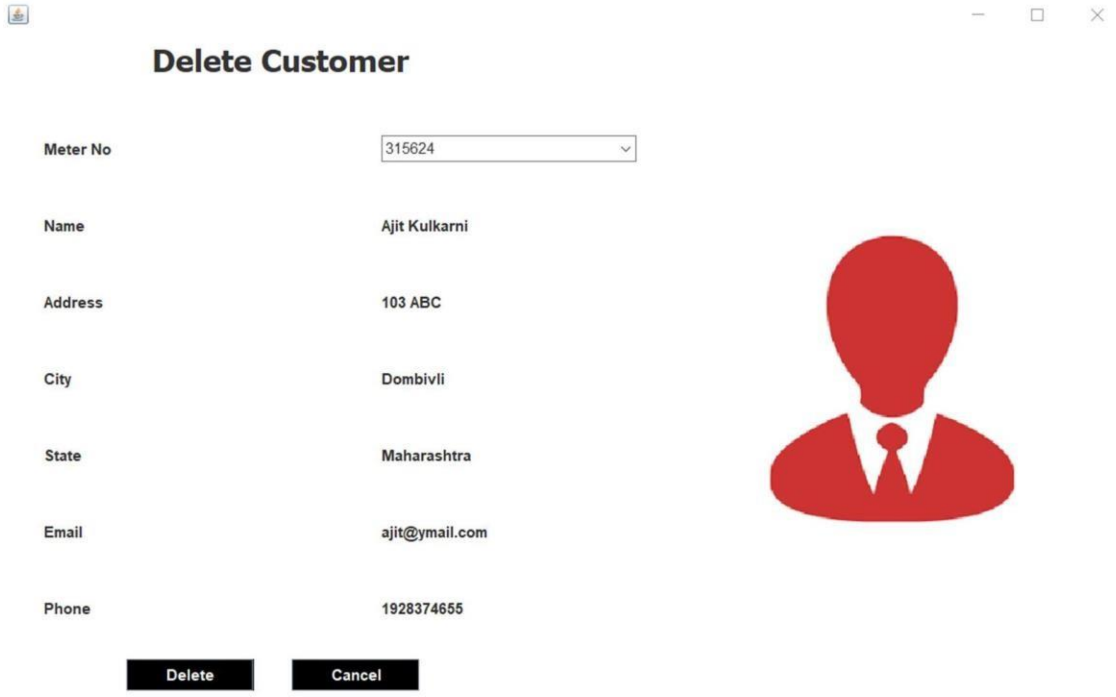


Calculate Electricity Bill

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.

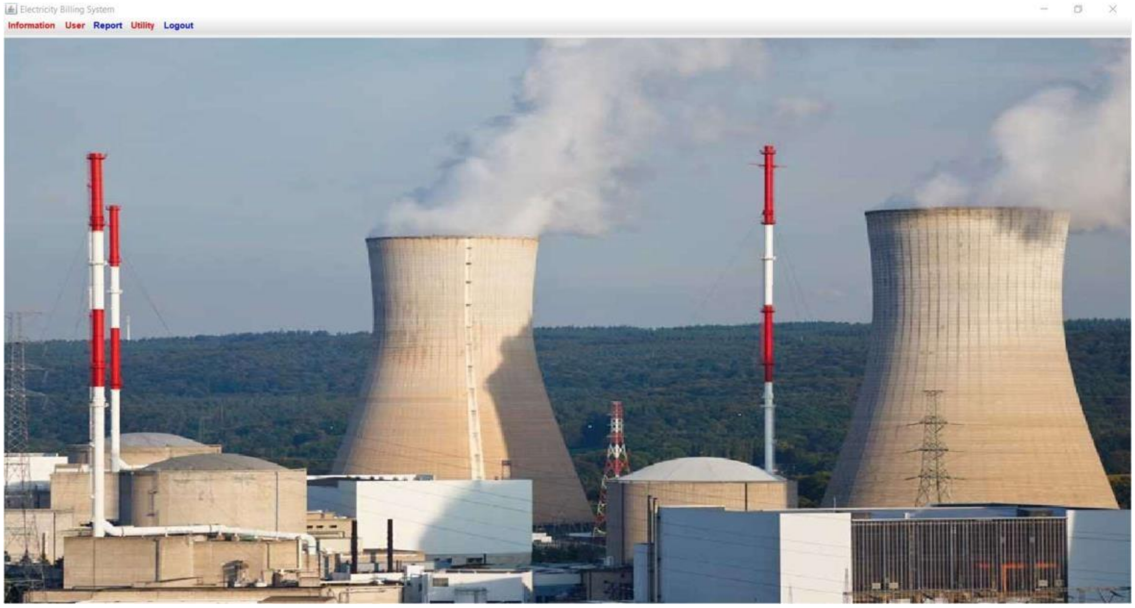


Delete Customer

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

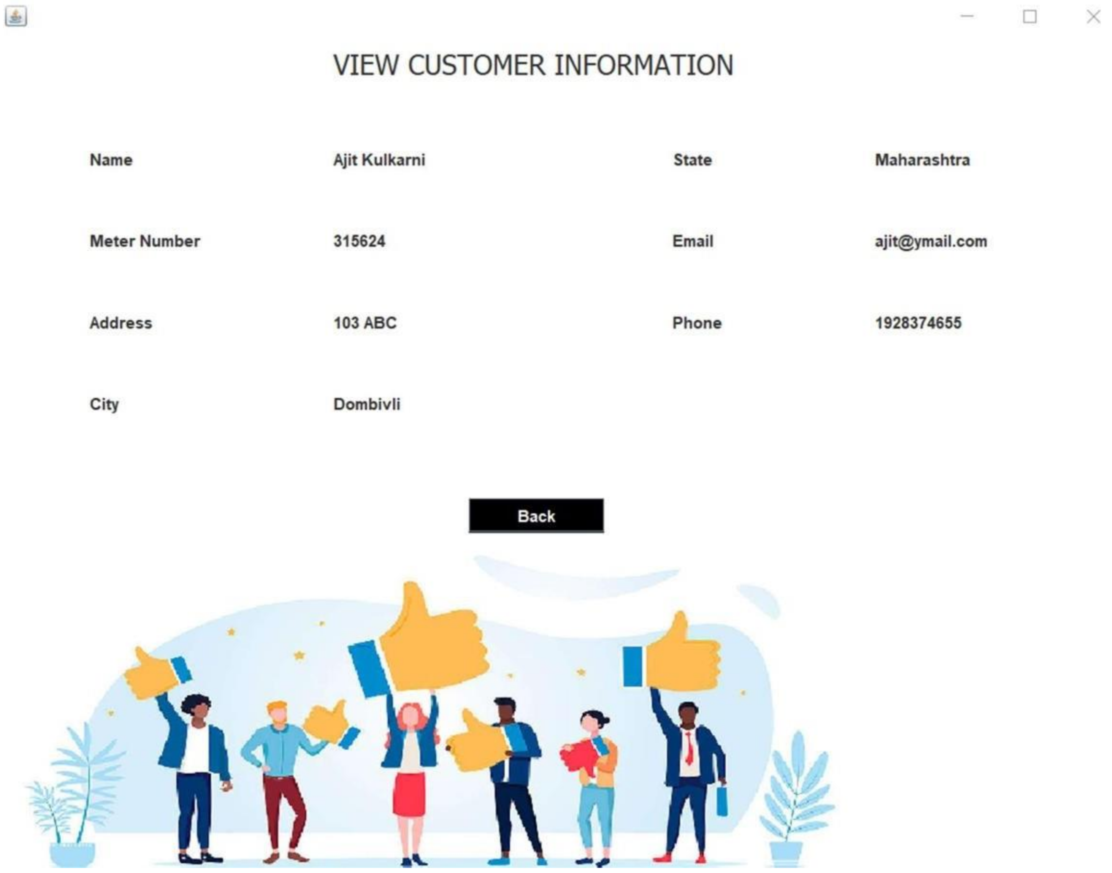
Delete Customer Screen

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



Customer's Home Screen

Customer lands on this page after successful login.

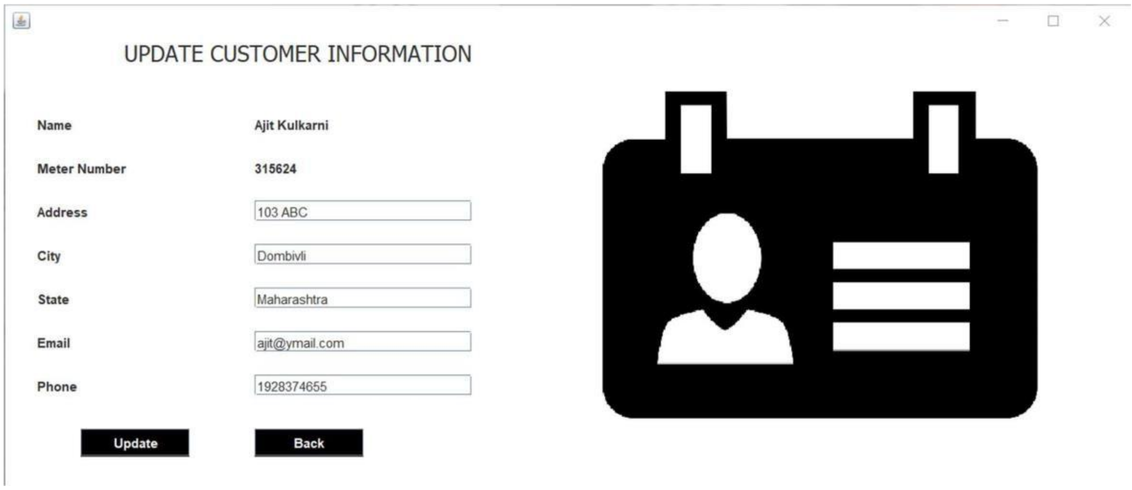


VIEW CUSTOMER INFORMATION

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.

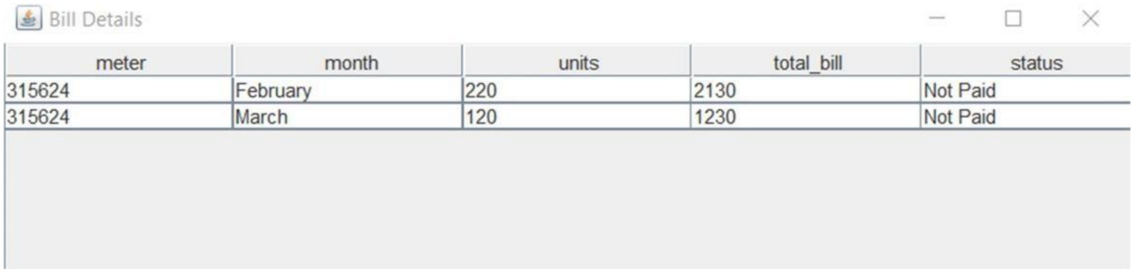


UPDATE CUSTOMER INFORMATION

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

Update Customer Info Screen

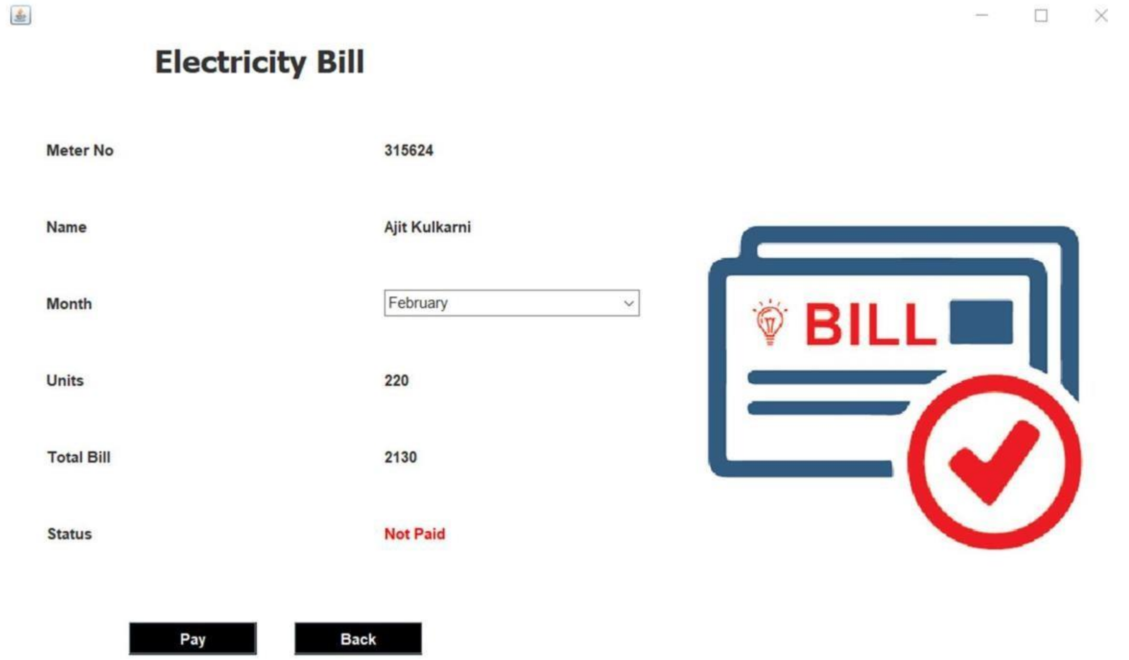
Here customer can update their entered information if any correction is needed such as their address, city, state, email id and phone number.



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.

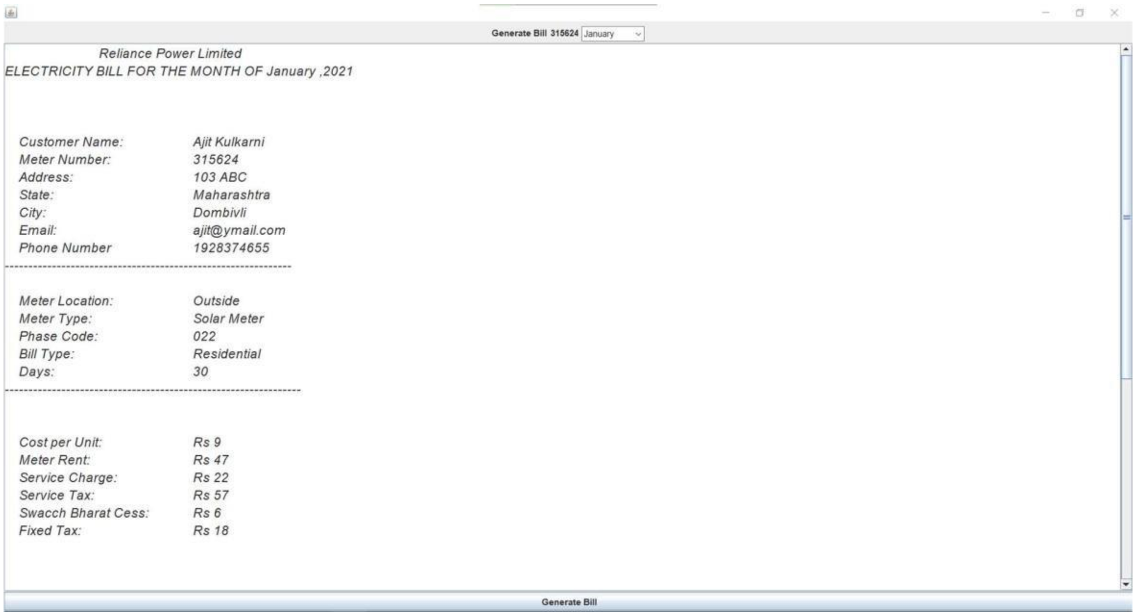


Electricity Bill

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

Pay Bill Screen

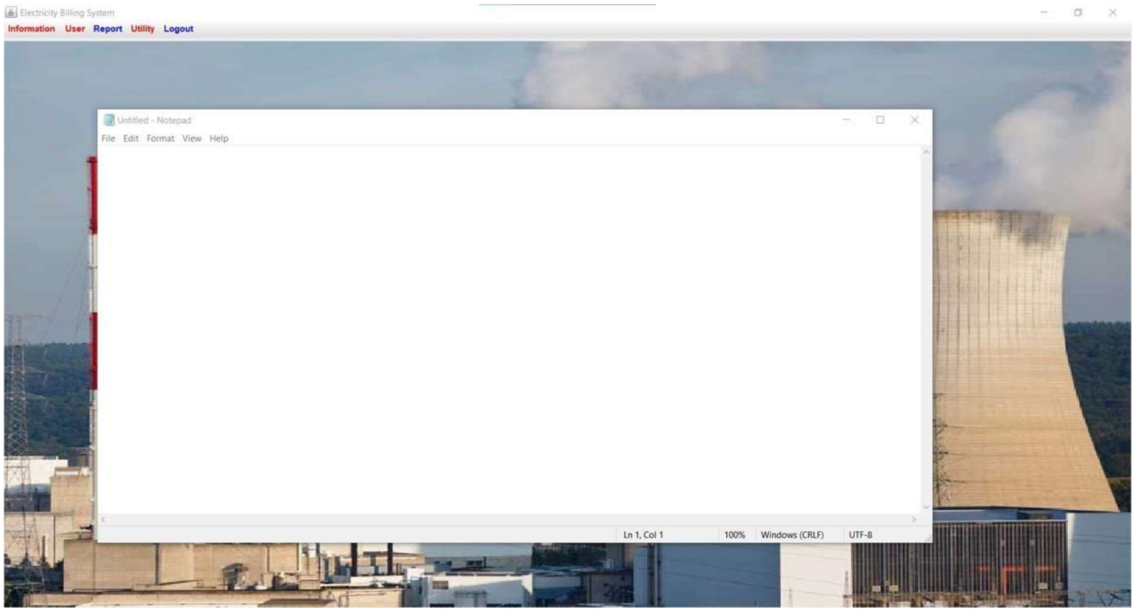
Here customers pay their bills by selecting appropriate month.



```
               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

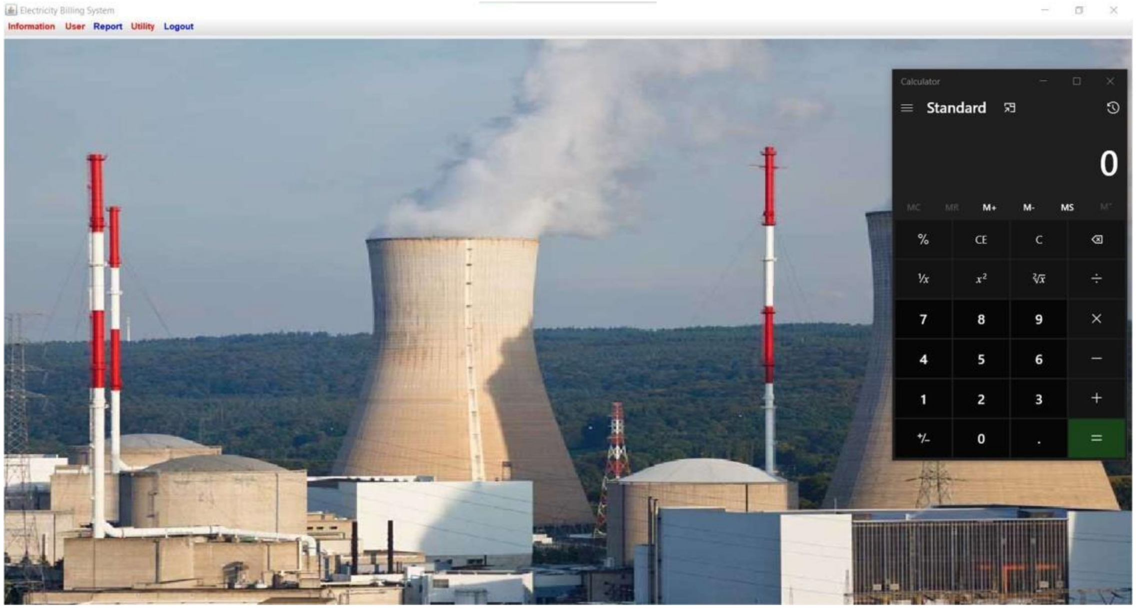
Here customer can generate / see their bill in a proper breakdown of entire amount.



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.