Electricity Billing System

Project

Submitted in partial fulfillment of the requirements
for the degree of

BACHELOR OF TECHNOLOGY

By

Team Members

Name	ID	Email
Anup Kumar Mahato	CL2025010601901674	anupez16@gmail.com
Roumik Raj	CL2025010601915288	2201020306@cgu-odisha.ac.in
Shaswat Kumar Sundaray	CL2025010601900765	2201020303@cgu-odisha.ac.in
Priyanka Priyadarshini	CL2025010601901573	2201020826@cgu-odisha.ac.in
Debasish Tripathy	CL2025010601876269	debasishtripathy7103@gmail.com

Certificate

This is to certify that the **Project** report entitled **Electricity Billing System** submitted by **Anup Kumar Mahato, Roumik Raj, Shaswat Kumar Sundaray, Priyanka Priyadarshini, Debasish Tripathy** has been carried out under my guidance & supervision. The project report is approved for submission towards partial fulfillment of the requirement for the award of degree of **Bachelor of Technology in Computer Engineering(Software Engineering) from C.V. Raman Global University, Bhubaneswar (Odisha).**

Electricity Billing System

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Electricity Billing System is a software-based application.

i. This project aims at serving the department of electricity by computerizing the billing system.
ii. It mainly focuses on the calculation of units consumed during the specified time and the money to be charged by the electricity offices.
iii. This computerized system will make the overall billing system easy, accessible, comfortable, and effective for consumers.

To design the billing system more service oriented and simple, the following features have been implemented in the project. The application has high speed of performance with accuracy and efficiency.

The software provides facility of data sharing, it does not require any staff as in the conventional system. Once it is installed on the system only the meter readings are to be given by the admin where customer can view all details, it has the provision of security restriction.

The electricity billing software calculates the units consumed by the customer and makes bills, it requires small storage for installation and functioning. There is provision for debugging if any problem is encountered in the system.

The system excludes the need of maintaining paper electricity bill, administrator does not have to keep a manual track of the users, users can pay the amount without visiting the office. Thus, it saves human efforts and resources.

Purpose :-

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.

Scope :-

The scope of our project are as follows:

- **To keep the information of consuming unit energy of current month.**
- **To keep the information of Customer.**
- **To keep the information of consuming unit energy of previous month.**
- **To calculate the units consumed every month regularly.**
- **To generate the bills adding penalty and rent. To save the time by implementing payment process online.**

Applicability :-

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.

CHAPTER 2

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:

- ➢ Processor: Intel i5 or above
- ➢ RAM: 8 GB minimum
- ➢ Storage: 256 GB SSD
- ➢ Graphics Card: NVIDIA GTX 1050 (if required for ML or graphics processing)
- ➢ Keyboard: Standard keyboard
- ➢ Mouse: myCompatible mouse

Software Requirements:-

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

CHAPTER 3

SYSTEM DESIGN AND MODELING

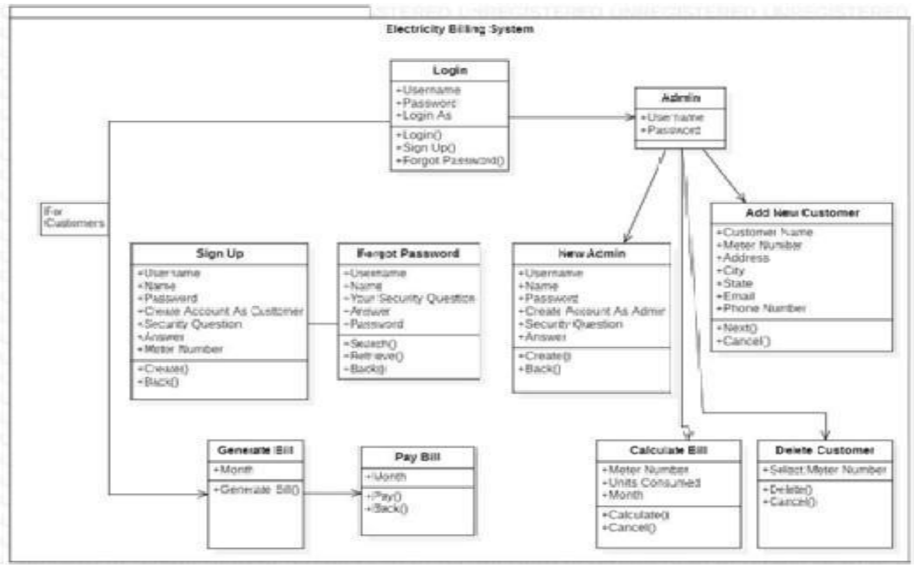
Preliminary Design:-

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 Diagram :-

Class Diagram:



Use Case Diagrams:

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.

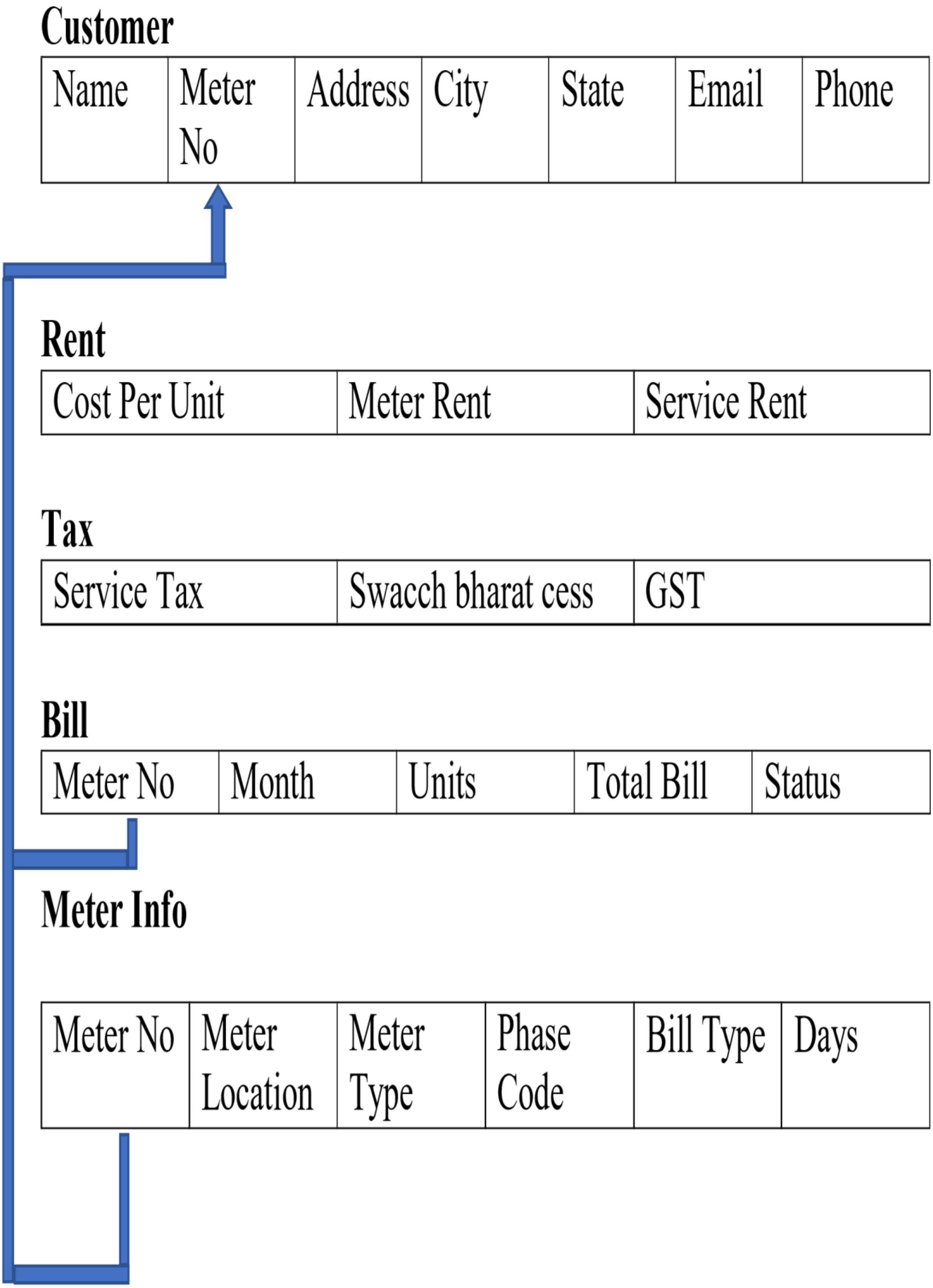
Figure is a Schema diagram of Electricity Billing System which has six tables i.e., login, customer, tax, rent, bill, and meter_info where each table contain attributes some with primary key, foreign key. In the login table there are
6 attributes "meter_no”, "username”, “password”, "user”, "question”, "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 ".

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

Normalization:-

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

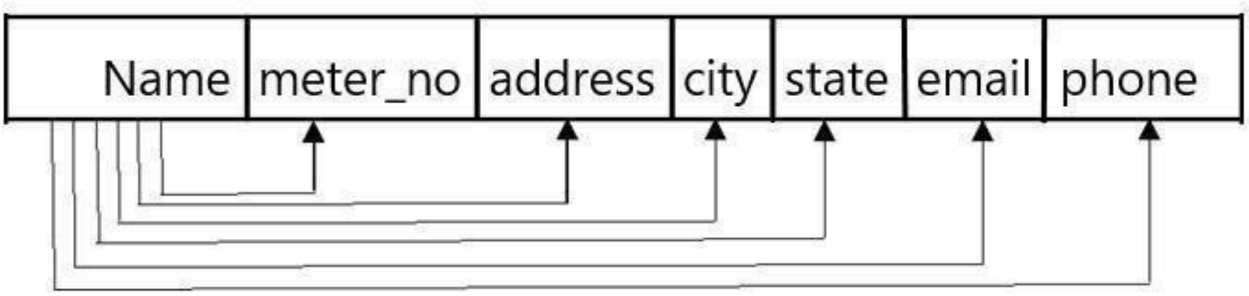
Let's discuss about anomalies first then we will discuss normal forms with examples. Anomalies in DBMS There are three types of anomalies that occur when the database is not normalized. These are –Insertion, update and deletion anomaly.

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:-



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.

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.

CHAPTER 4

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));

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

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 cancel option.

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.

UpdateInfo:-

□ 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.

View Info:

□ This program will allow the customer to view his/her own details.

□ If we need to go back from the particulars that has been viewed click 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.

CHAPTER 5

TESTING

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

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.

Unit Testing:-

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

Table 5.1: Negative test case for phone number insertion

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

Positive test case for phone number insertion

Function Name	**Input**	**Expected Output**	**Error**	**Resolved**
Input Phone Number	9897778900	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	_	_

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 ()

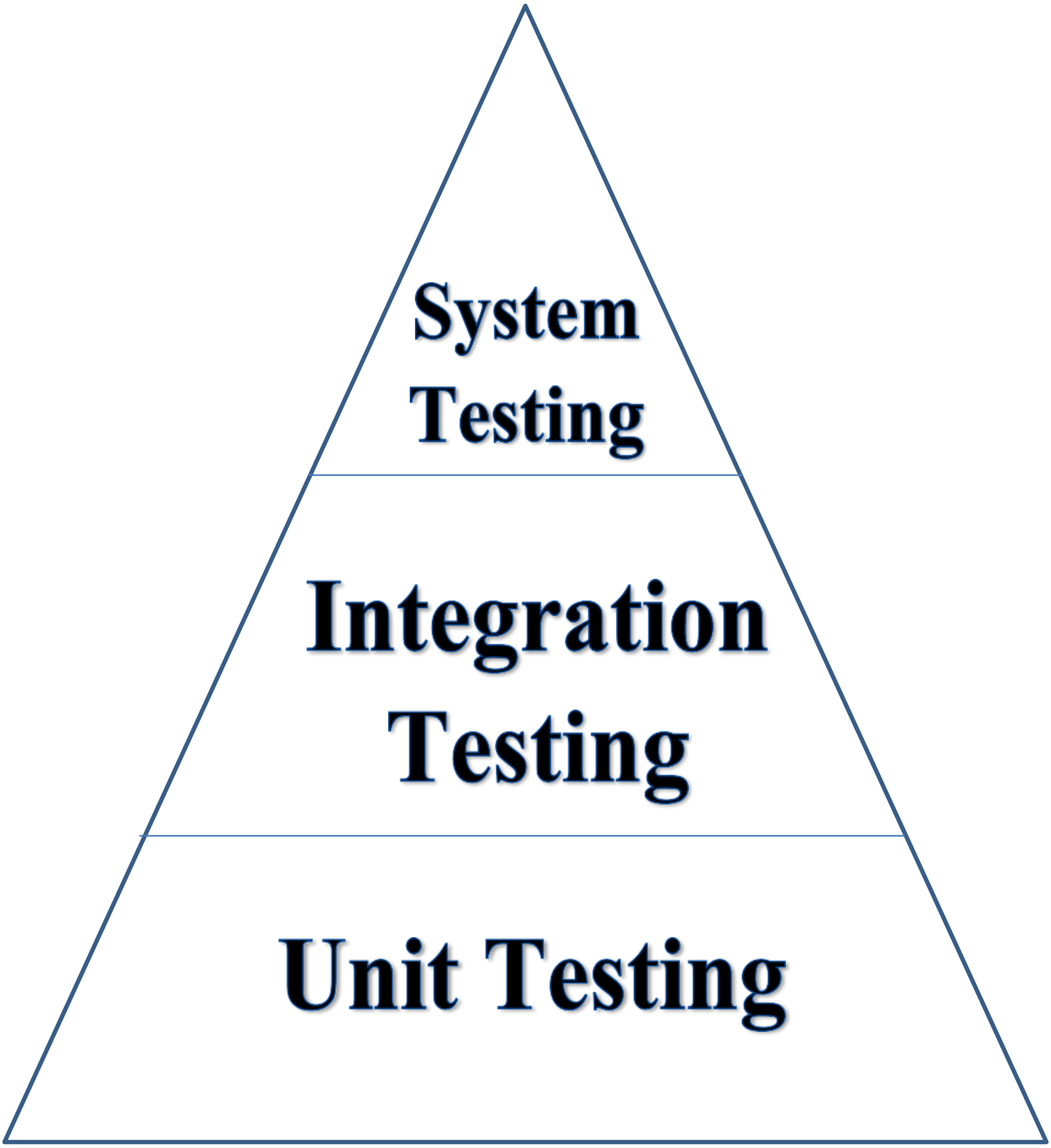
Positive searching of total_bill	12334(meter_no)
	January(month)
		Must display full generated bill with total_bill	_	_

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	_	_



Testing Diagram

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	

CHAPTER 6

DISCUSSION AND SNAPSHOTS

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.

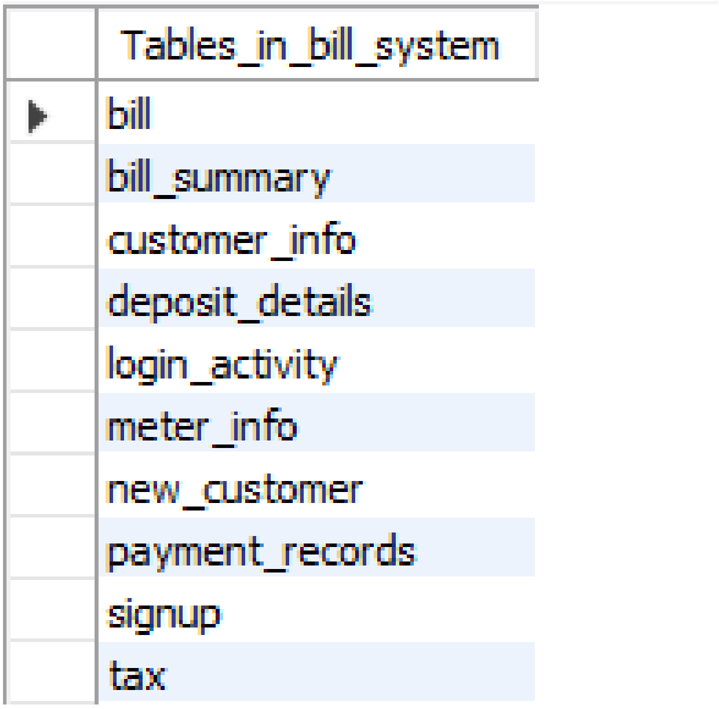
✓ The customer is used to store details of customer.

✓ 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.

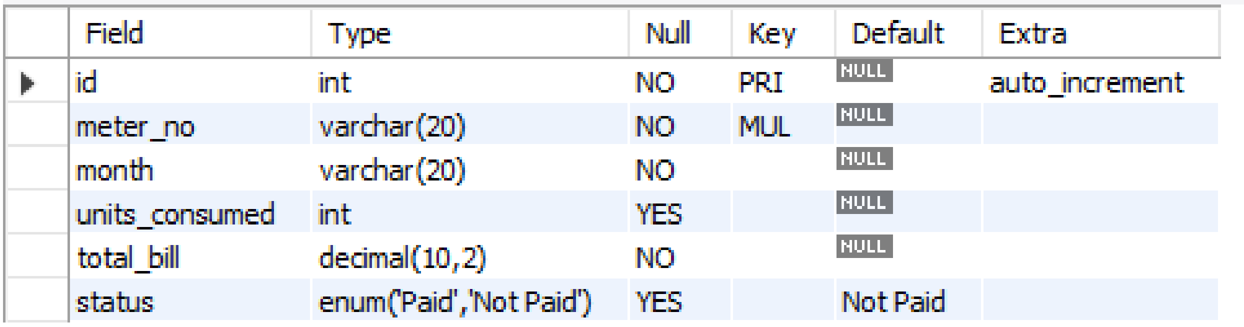
✓ The meter_info is used to store information of meter placed.



	Tables_in_bill_system
▶	bill
	bill_summary
	customer_info
	deposit_details
	login_activity
	meter_info
	new_customer
	payment_records
	signup
	tax

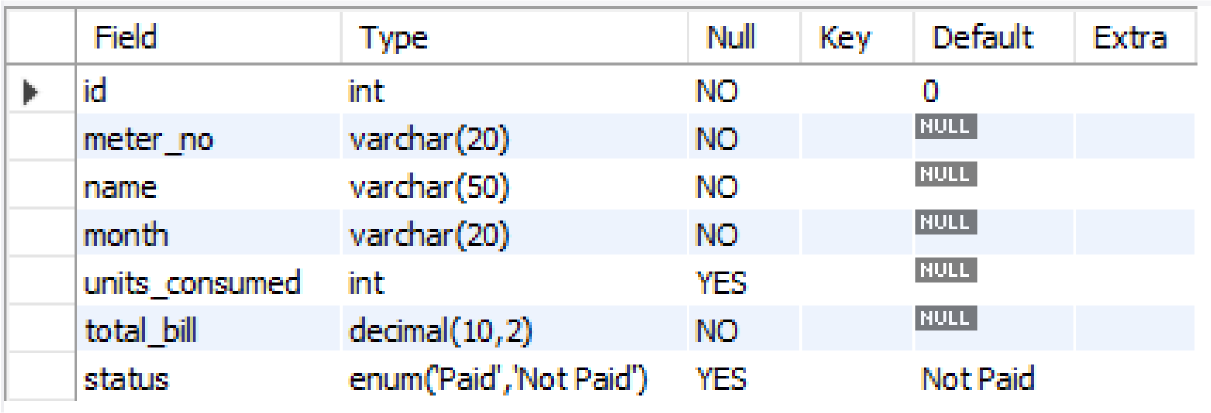
List of tables

1. Bill



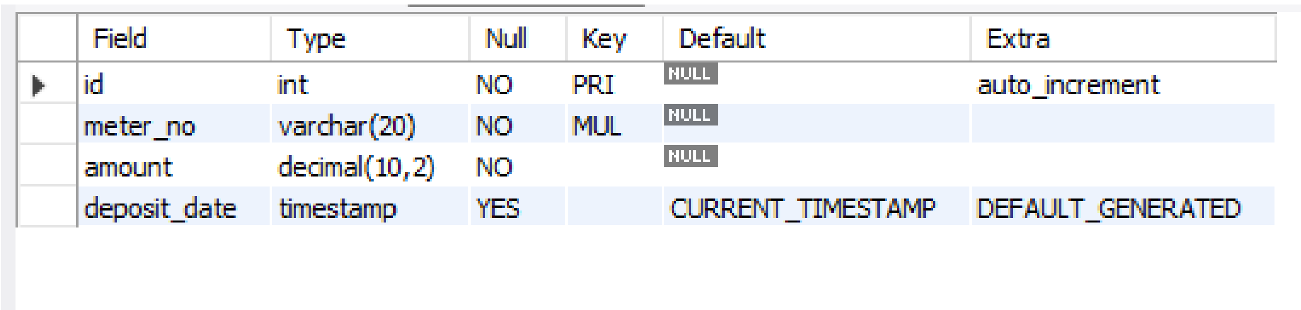
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	meter_no	varchar(20)	NO	MUL	NULL	
	month	varchar(20)	NO		NULL	
	units_consumed	int	YES		NULL	
	total_bill	decimal(10,2)	NO		NULL	
	status	enum('Paid','Not Paid')	YES		Not Paid	

2. bill_summary



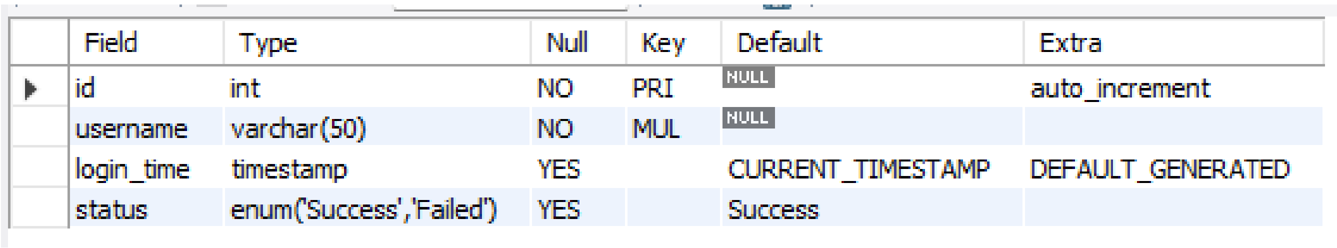
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO		0	
	meter_no	varchar(20)	NO		NULL	
	name	varchar(50)	NO		NULL	
	month	varchar(20)	NO		NULL	
	units_consumed	int	YES		NULL	
	total_bill	decimal(10,2)	NO		NULL	
	status	enum('Paid','Not Paid')	YES		Not Paid	

3. deposit_details



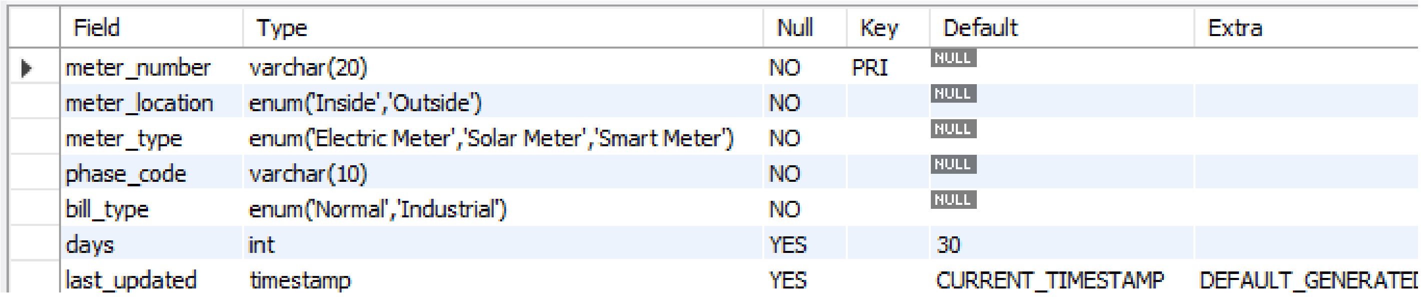
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	meter_no	varchar(20)	NO	MUL	NULL	
	amount	decimal(10,2)	NO		NULL	
	deposit_date	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

4. login_activity



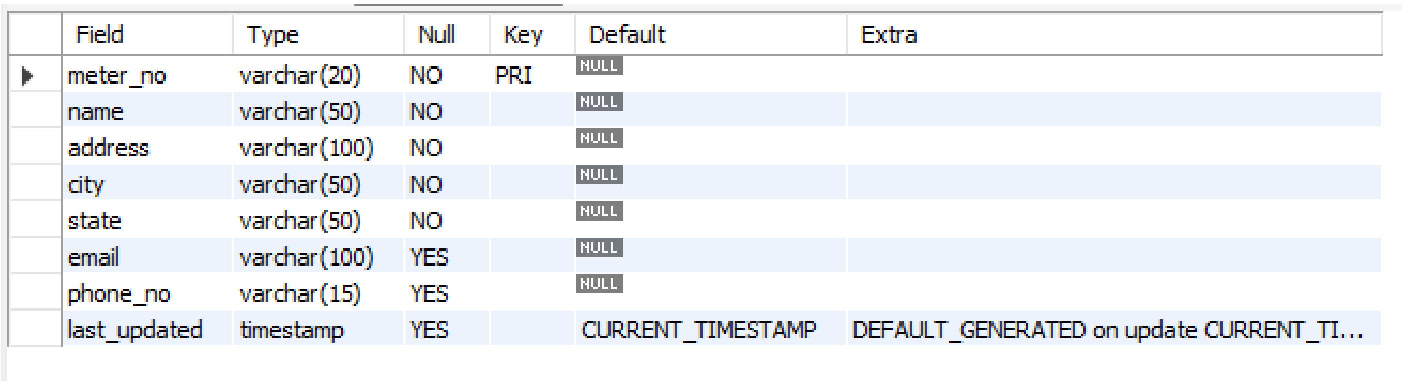
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	username	varchar(50)	NO	MUL	NULL	
	login_time	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
	status	enum('Success','Failed')	YES		Success	

5. meter_info



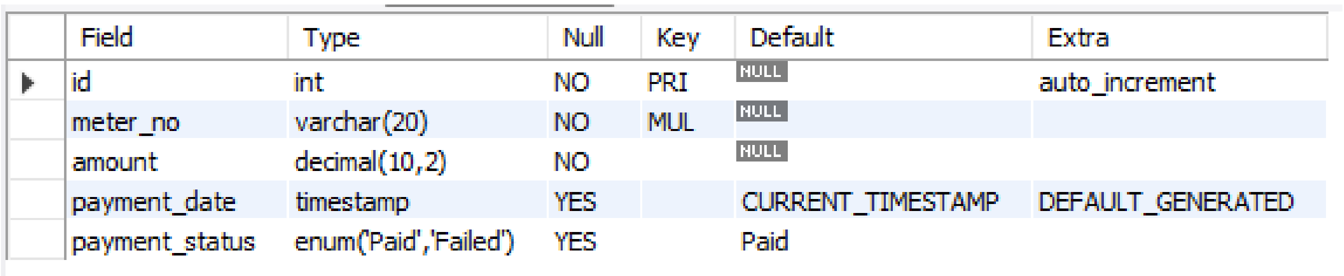
Field	Type	Null	Key	Default	Extra
meter_number	varchar(20)	NO	PRI	NULL	
meter_location	enum('Inside','Outside')	NO		NULL	
meter_type	enum('Electric Meter','Solar Meter','Smart Meter')	NO		NULL	
phase_code	varchar(10)	NO		NULL	
bill_type	enum('Normal','Industrial')	NO		NULL	
days	int	YES		30	
last_updated	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATEI

6. new_customer



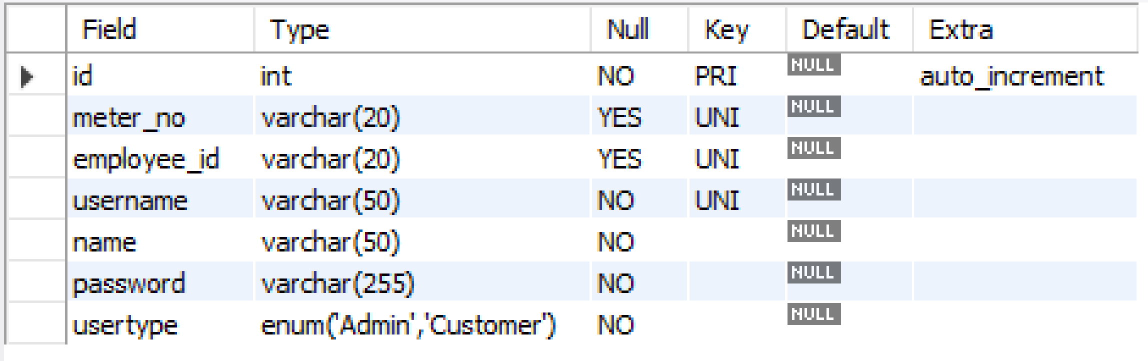
Field	Type	Null	Key	Default	Extra
meter_no	varchar(20)	NO	PRI	NULL	
name	varchar(50)	NO		NULL	
address	varchar(100)	NO		NULL	
city	varchar(50)	NO		NULL	
state	varchar(50)	NO		NULL	
email	varchar(100)	YES		NULL	
phone_no	varchar(15)	YES		NULL	
last_updated	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

7. payment_records



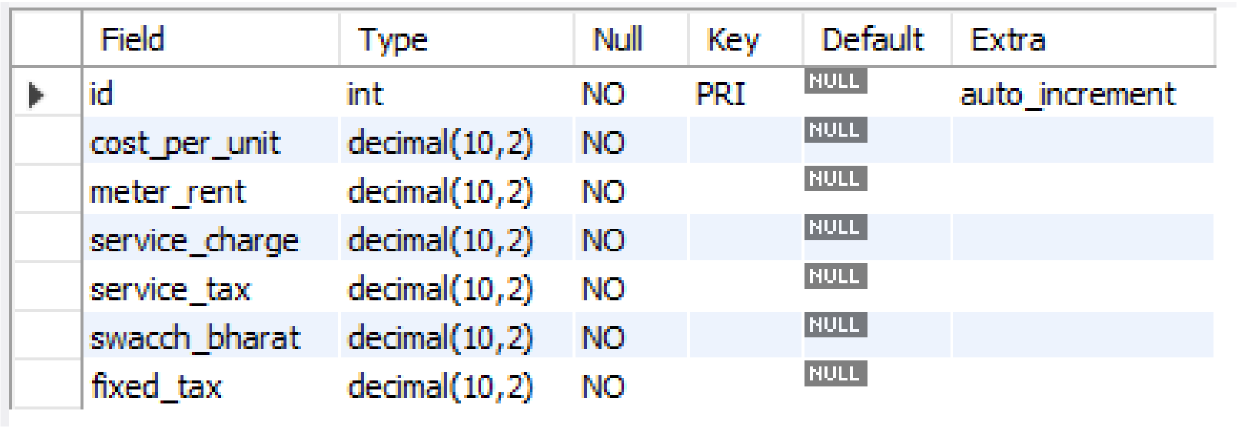
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
meter_no	varchar(20)	NO	MUL	NULL	
amount	decimal(10,2)	NO		NULL	
payment_date	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
payment_status	enum('Paid','Failed')	YES		Paid	

8. Signup



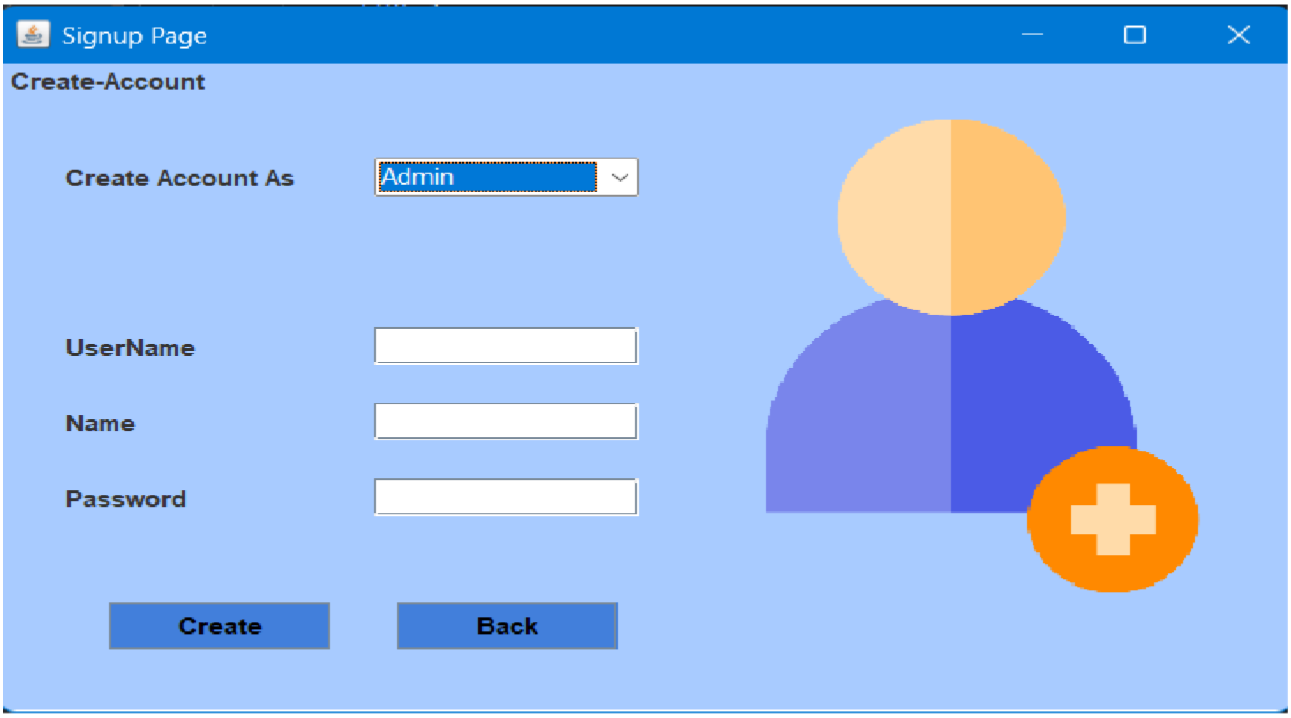
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	meter_no	varchar(20)	YES	UNI	NULL	
	employee_id	varchar(20)	YES	UNI	NULL	
	username	varchar(50)	NO	UNI	NULL	
	name	varchar(50)	NO		NULL	
	password	varchar(255)	NO		NULL	
	usertype	enum('Admin','Customer')	NO		NULL	

9. tax

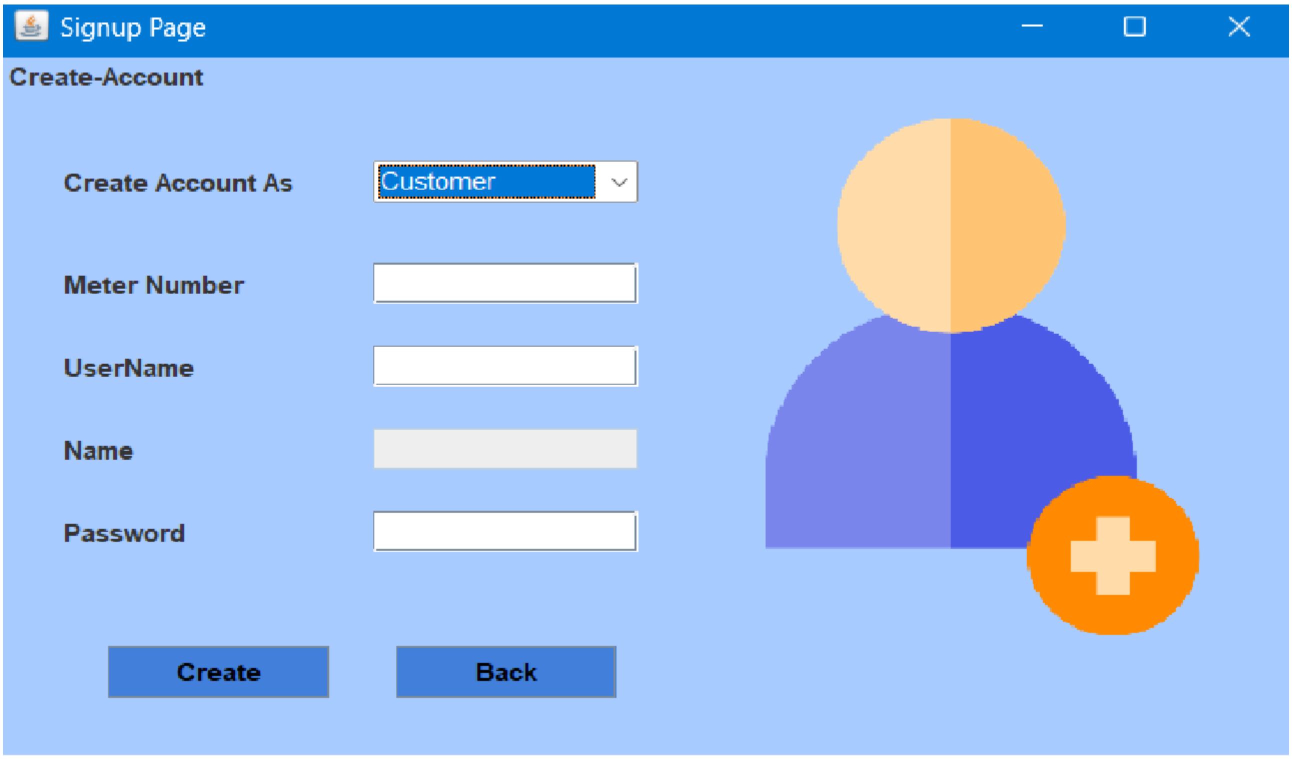


	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	cost_per_unit	decimal(10,2)	NO		NULL	
	meter_rent	decimal(10,2)	NO		NULL	
	service_charge	decimal(10,2)	NO		NULL	
	service_tax	decimal(10,2)	NO		NULL	
	swacch_bharat	decimal(10,2)	NO		NULL	
	fixed_tax	decimal(10,2)	NO		NULL	

SNAPSHOTS:-



SignUp Screen For Admin



SignUp Screen For Customer

Sign Up Screen:-

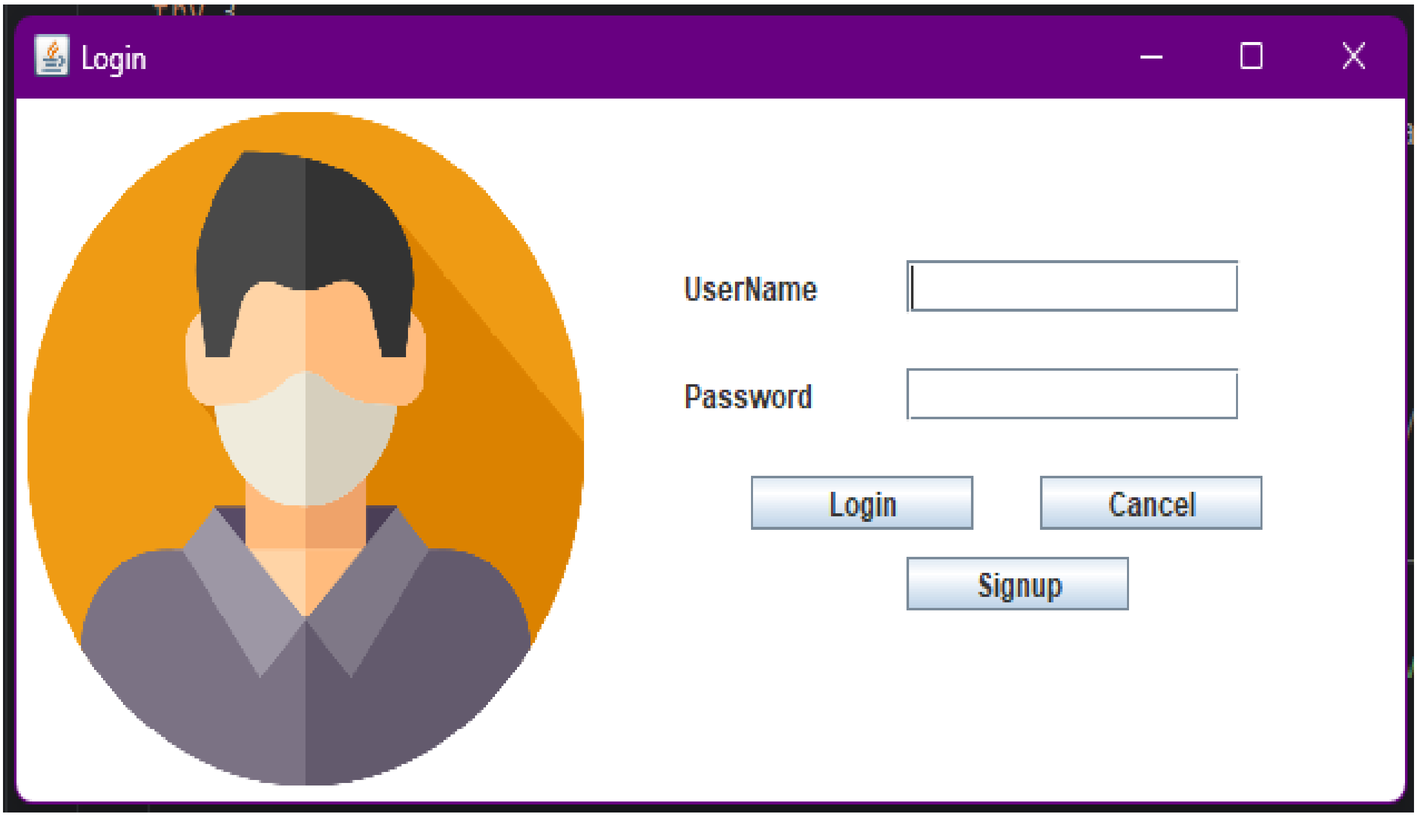
Here New customers will signup to access their accounts.

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.

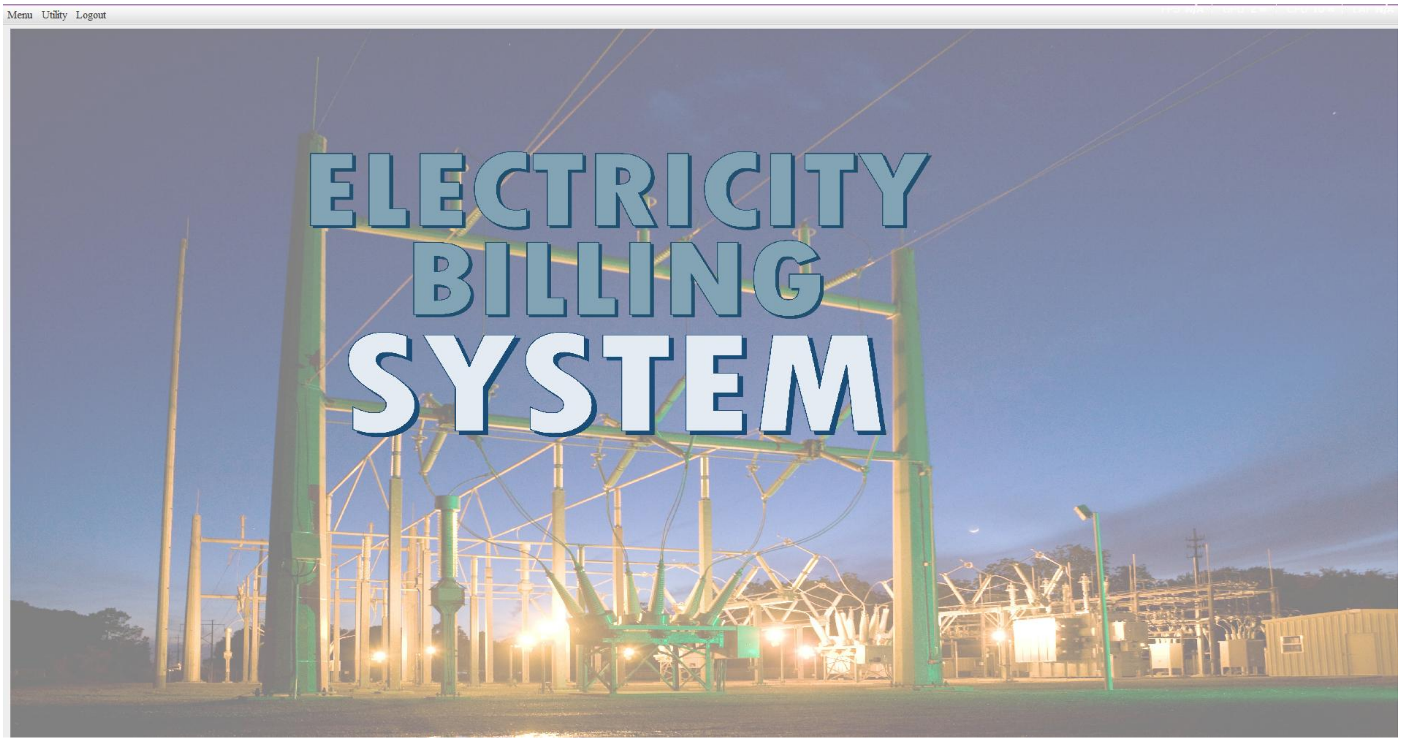
Login Screen:-

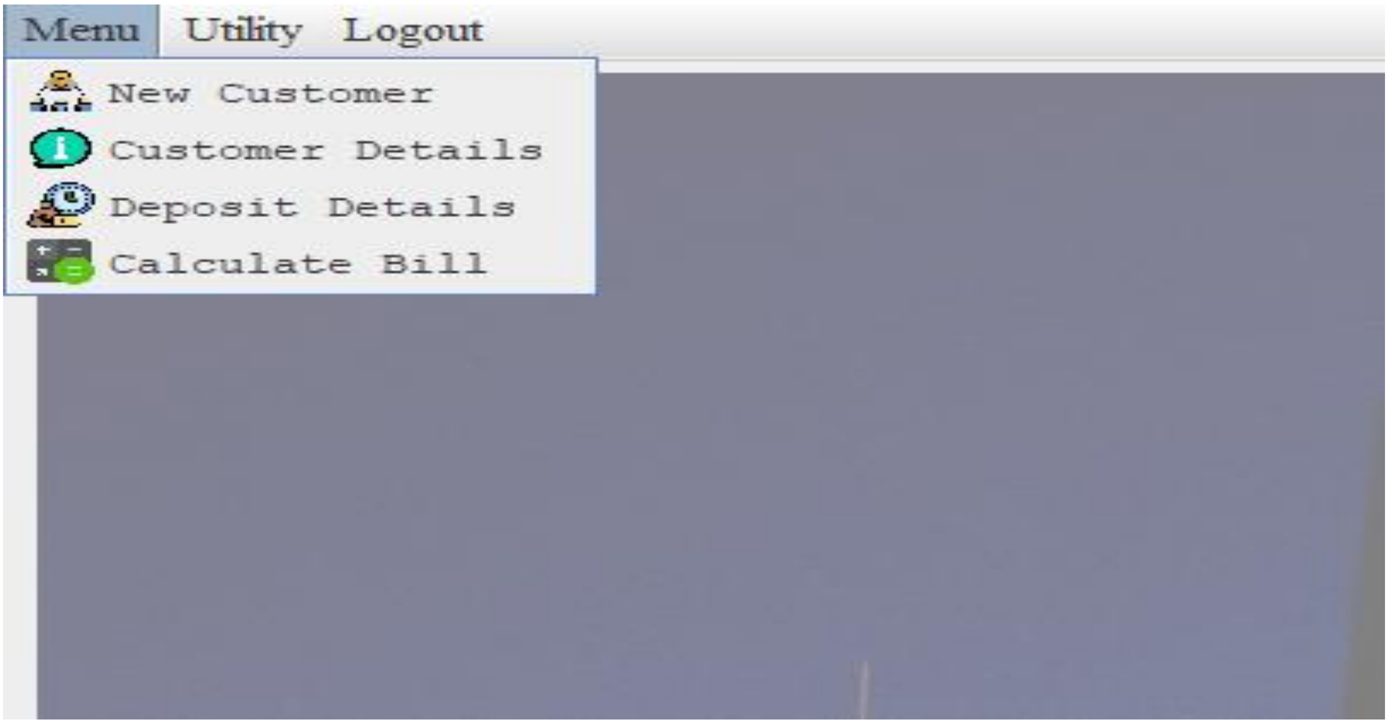
Here Customer and Admin can login to their respective accounts. The dropdown menu allows to choose whether to login as an admin or as a customer.



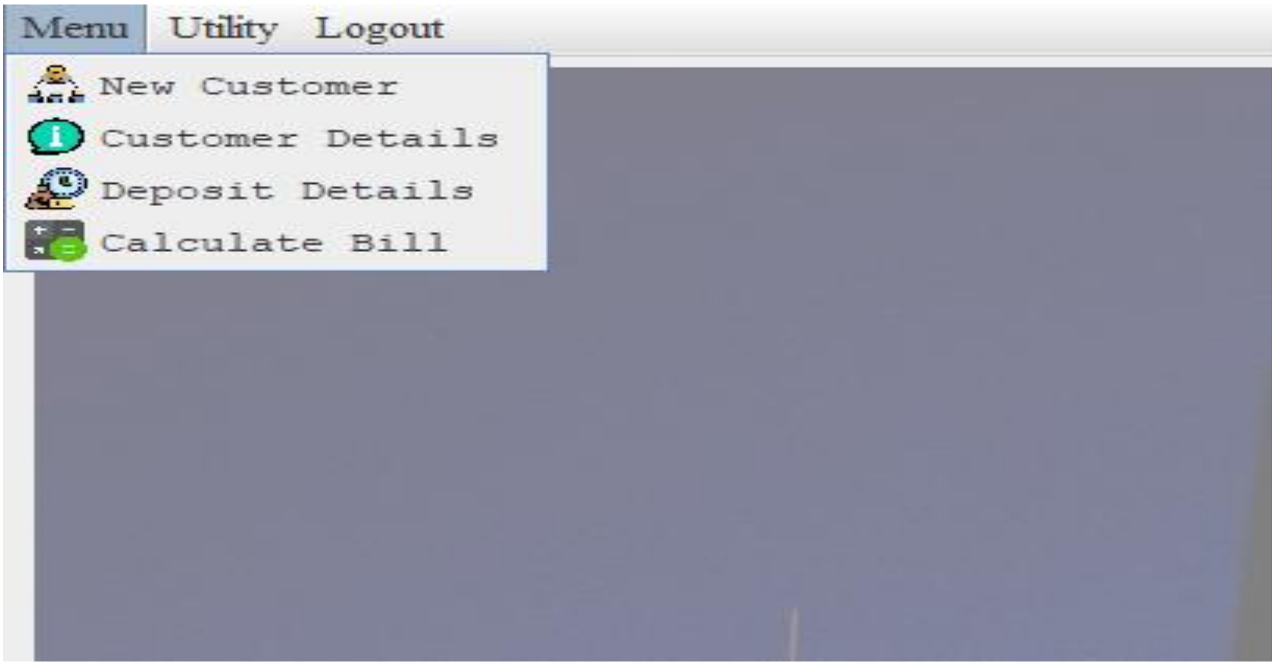
Admin lands on this page after successful login:-

Admin's Home Screen:

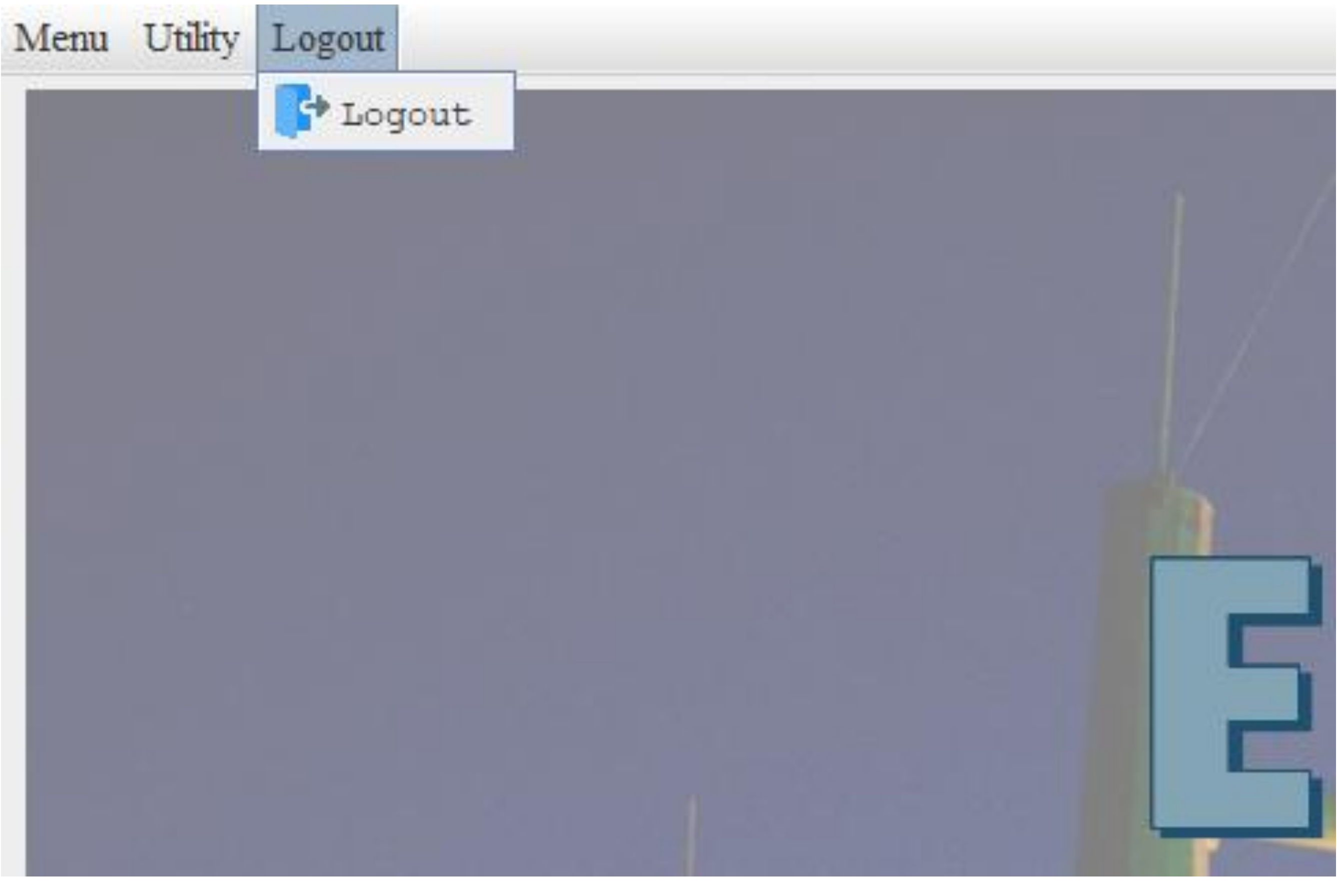




Menu

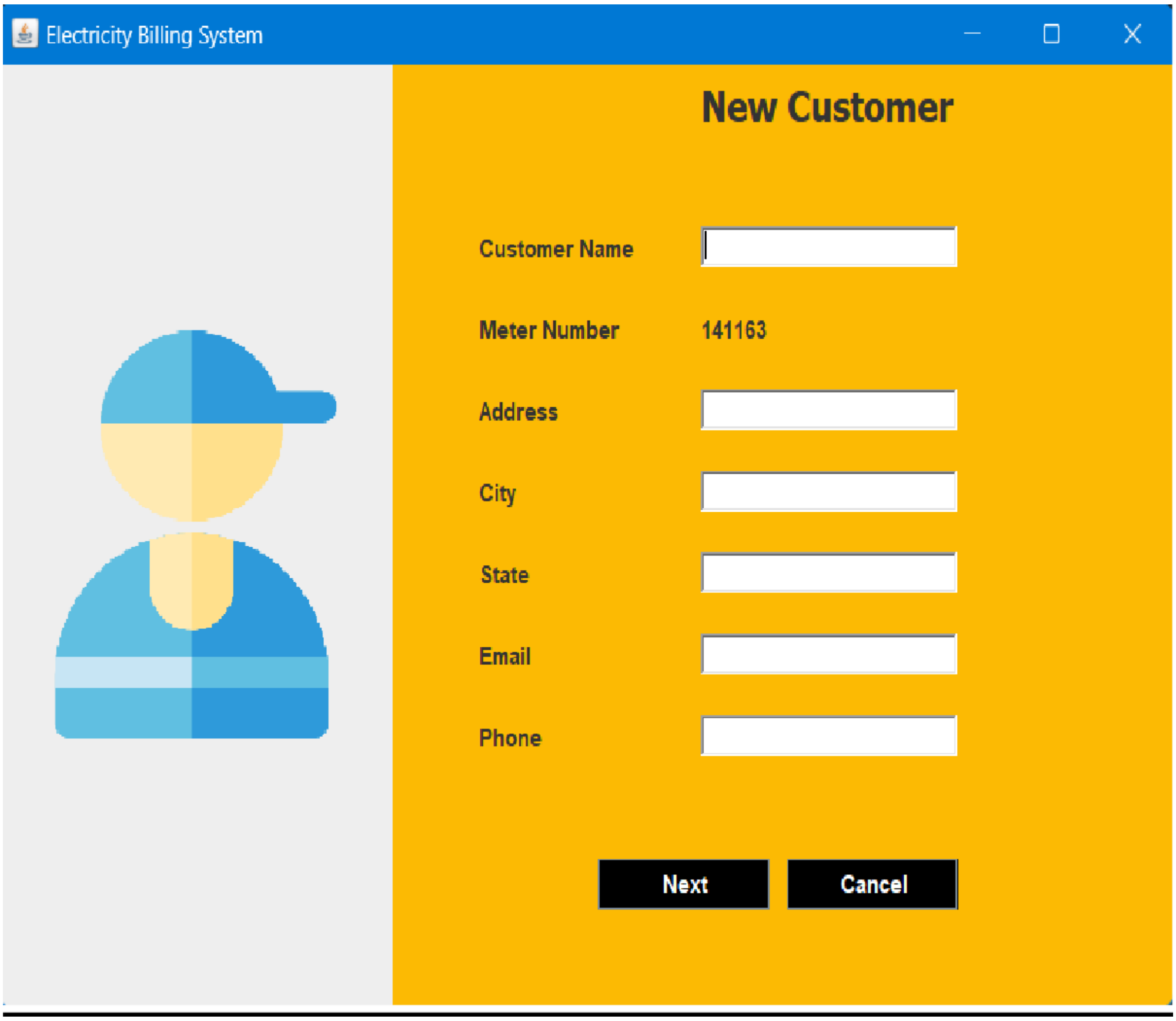


Utility



Exit

New Customer Screen:-



New Customer

Customer Name

Meter Number 141163

Address

City

State

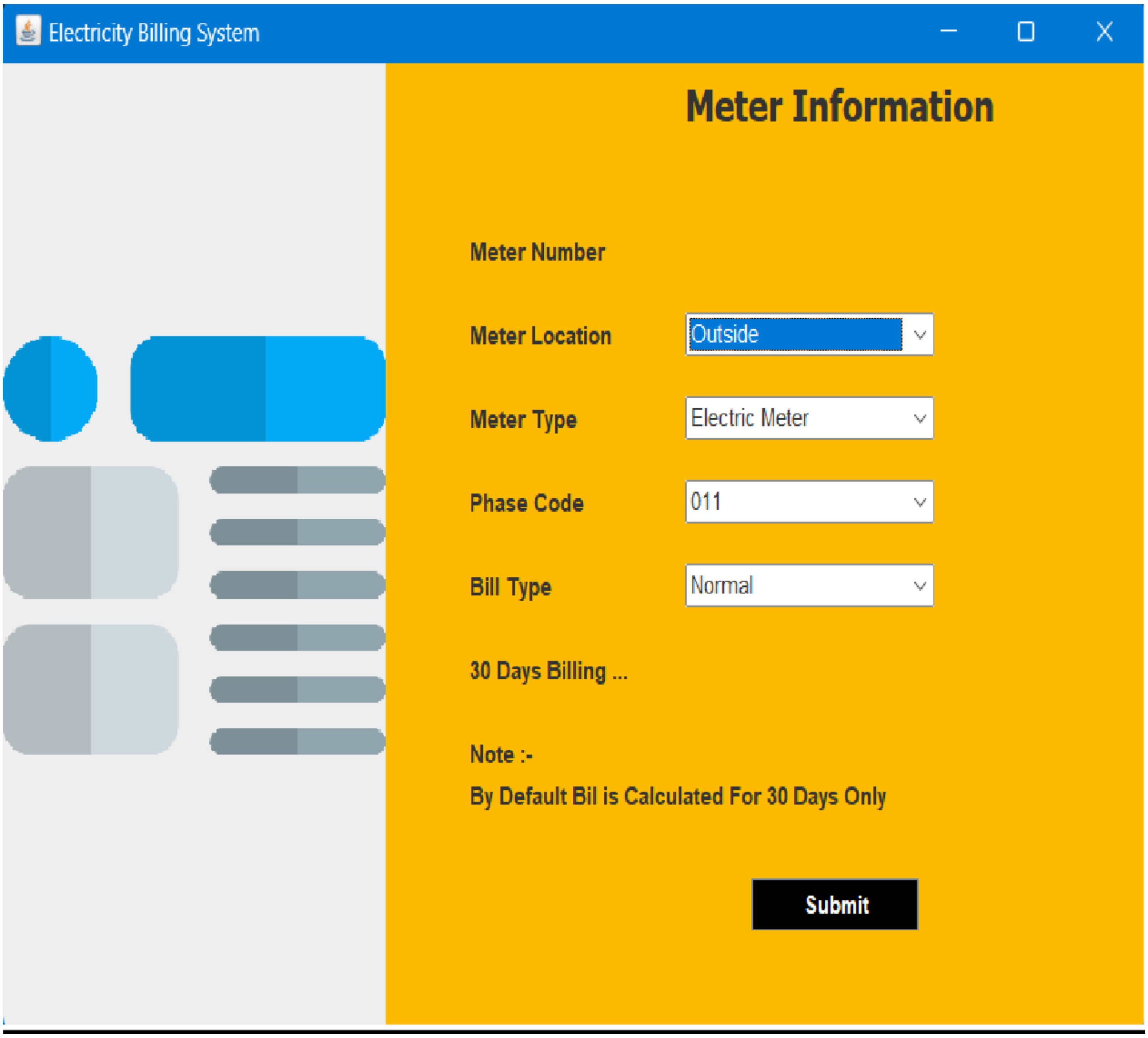
Email

Phone

Here admin registers new users.

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

Meter Info Screen:-



Meter Information

Meter Number

Meter Location: Outside

Meter Type: Electric Meter

Phase Code: 011

Bill Type: Normal

30 Days Billing ...

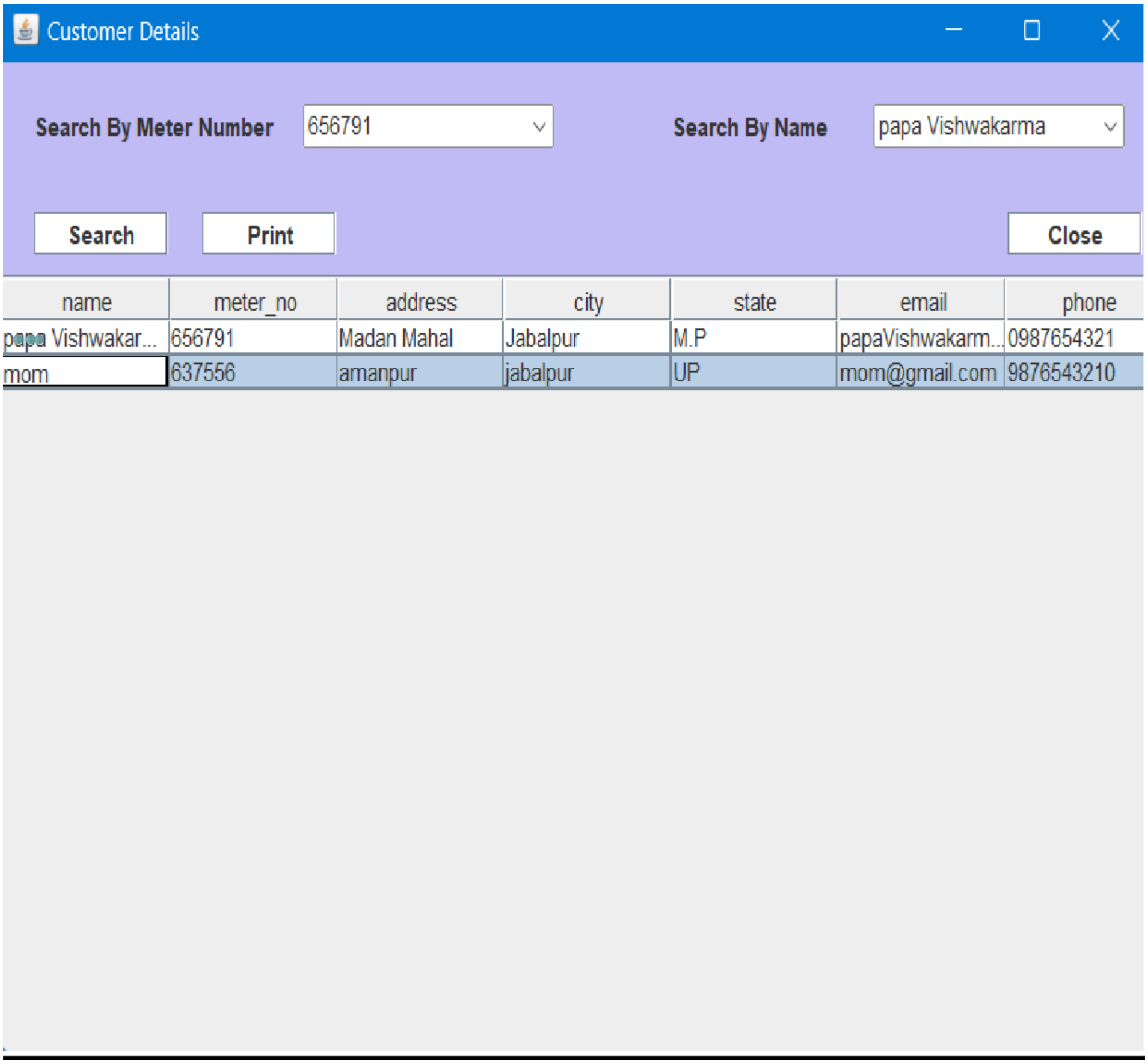
Note :-

By Default Bil is Calculated For 30 Days Only

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.

Customer Details Screen:-



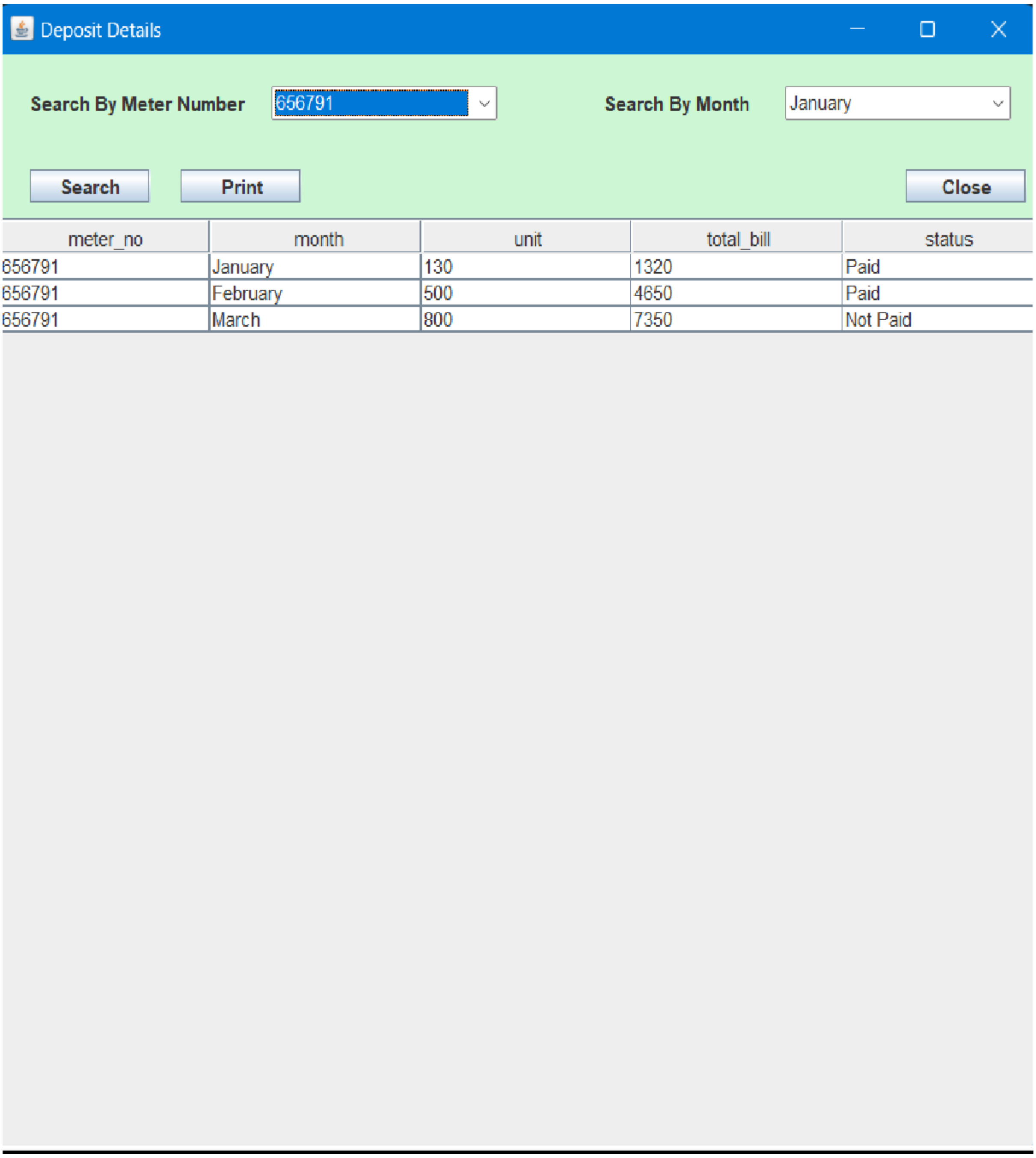
Search By Meter Number: 656791

Search By Name: papa Vishwakarma

name	meter_no	address	city	state	email	phone
papa Vishwakar...	656791	Madan Mahal	Jabalpur	M.P	papaVishwakarm...	0987654321
mom	637556	amanpur	jabalpur	UP	mom@gmail.com	9876543210

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

Deposit Details Screen:-



Search By Meter Number: 656791

Search By Month: January

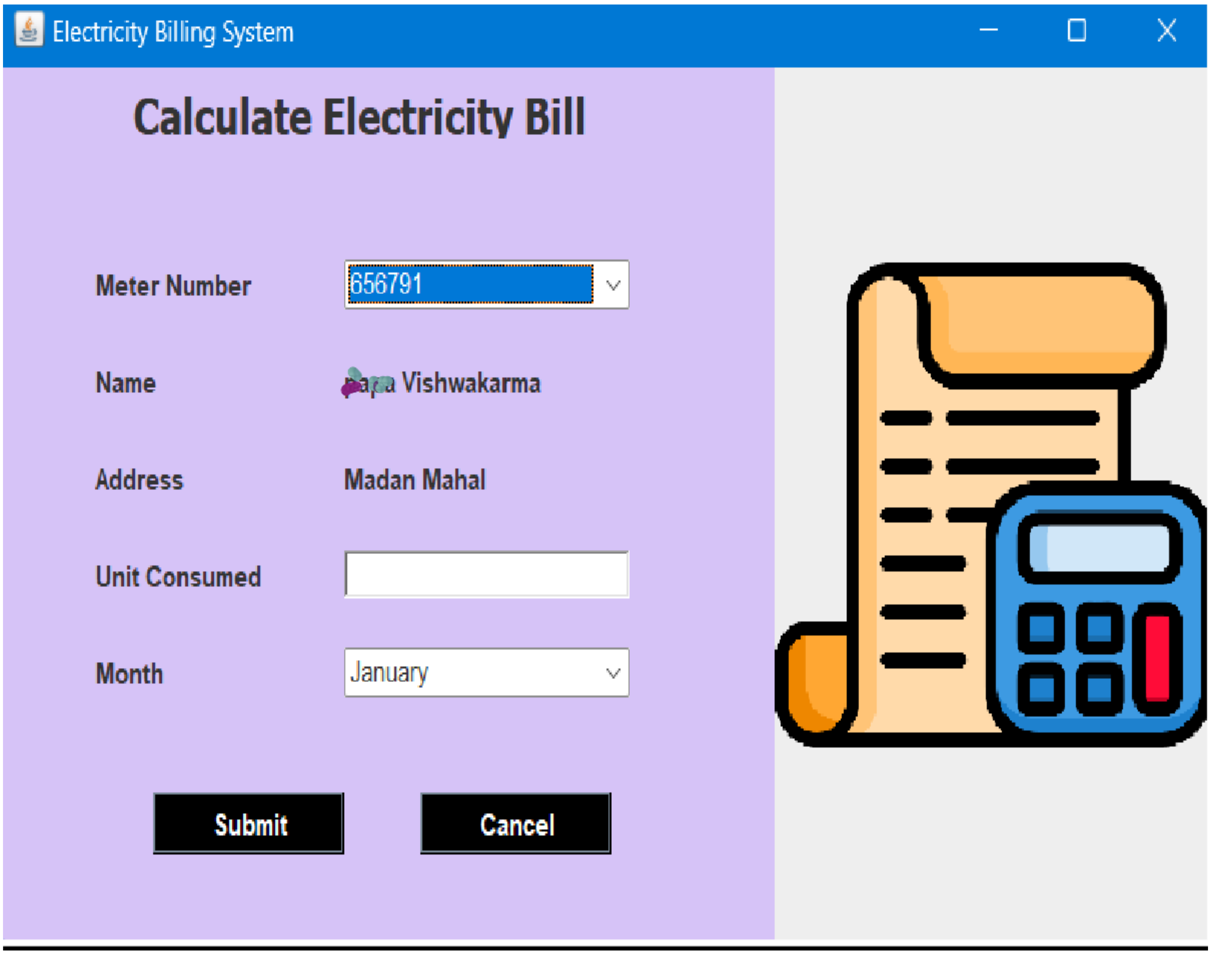
meter_no	month	unit	total_bill	status
656791	January	130	1320	Paid
656791	February	500	4650	Paid
656791	March	800	7350	Not Paid

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 Bill Screen:-



Calculate Electricity Bill

Meter Number 656791

Name papa Vishwakarma

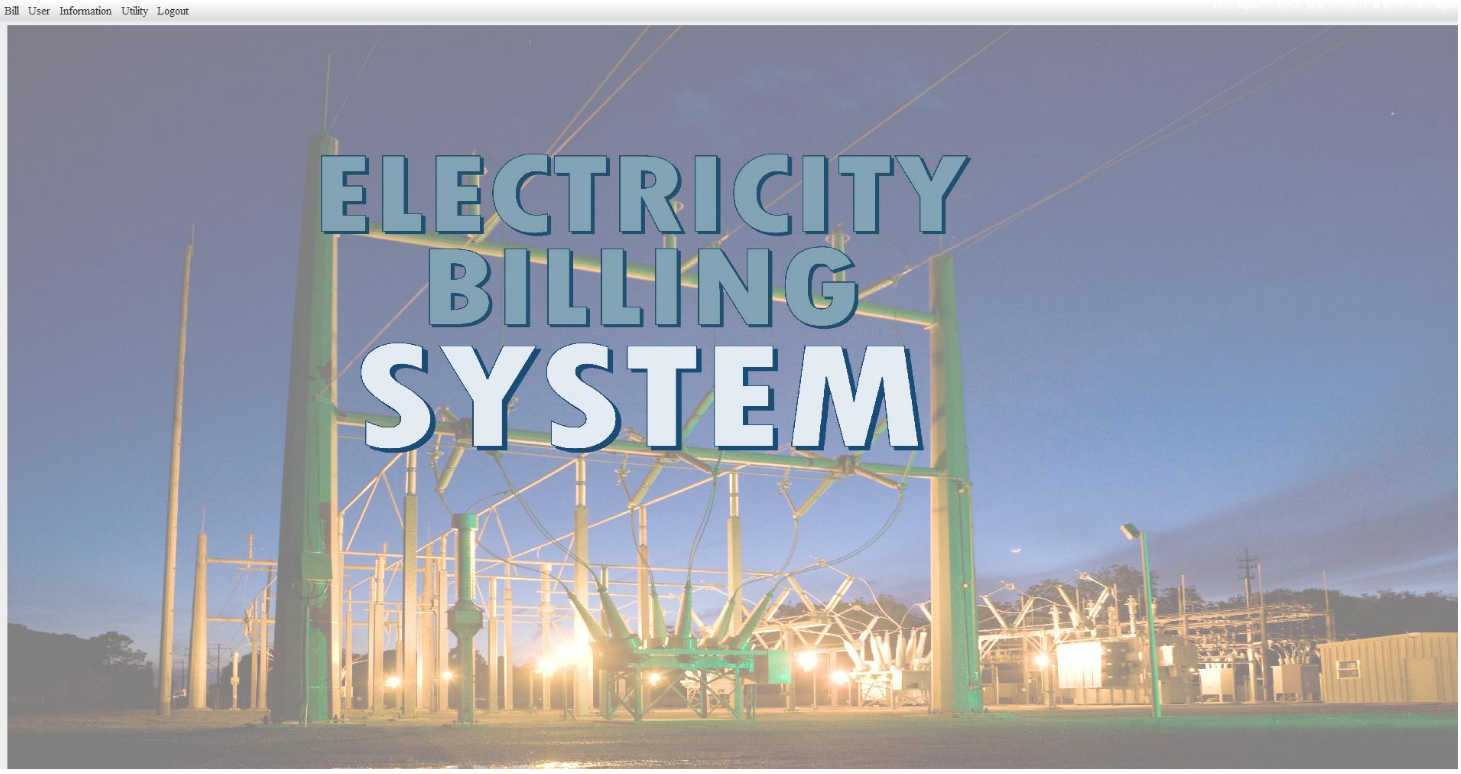
Address Madan Mahal

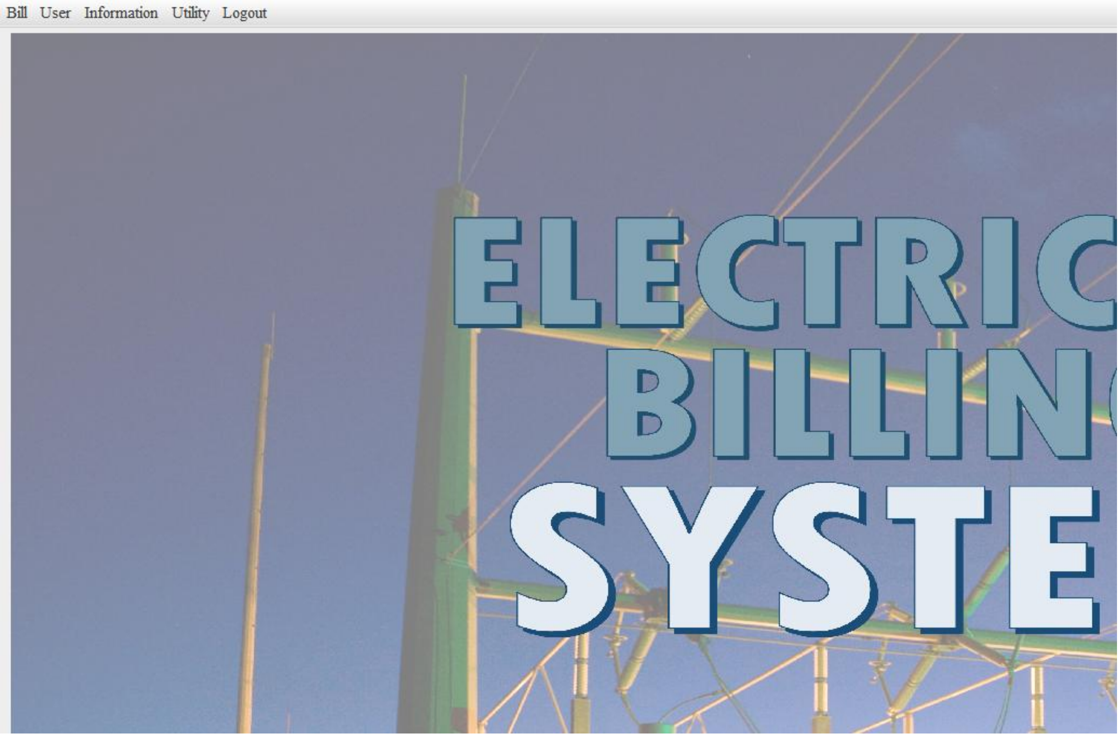
Unit Consumed

Month January

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

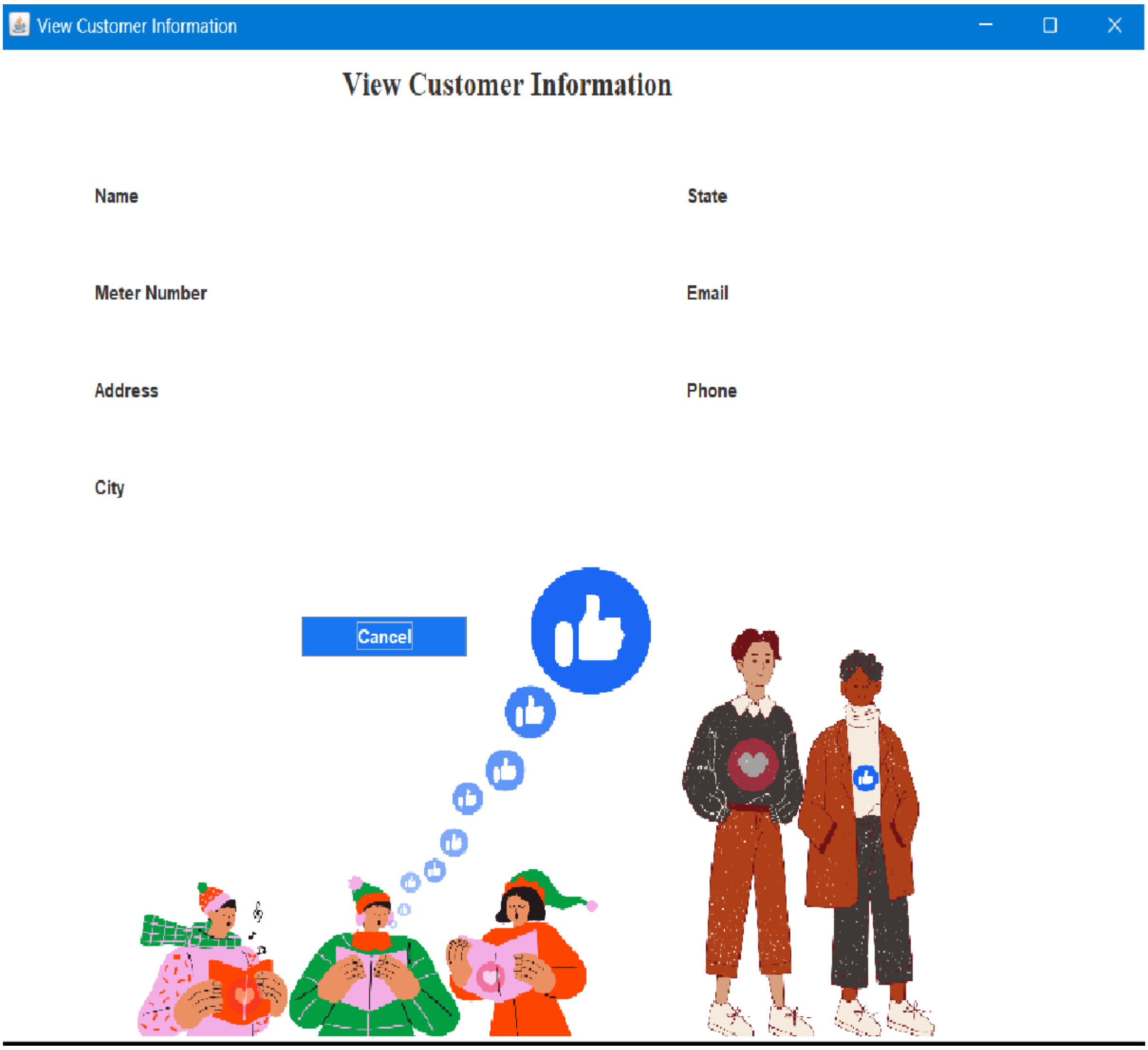
<u>Customer’s Home Screen:-</u>





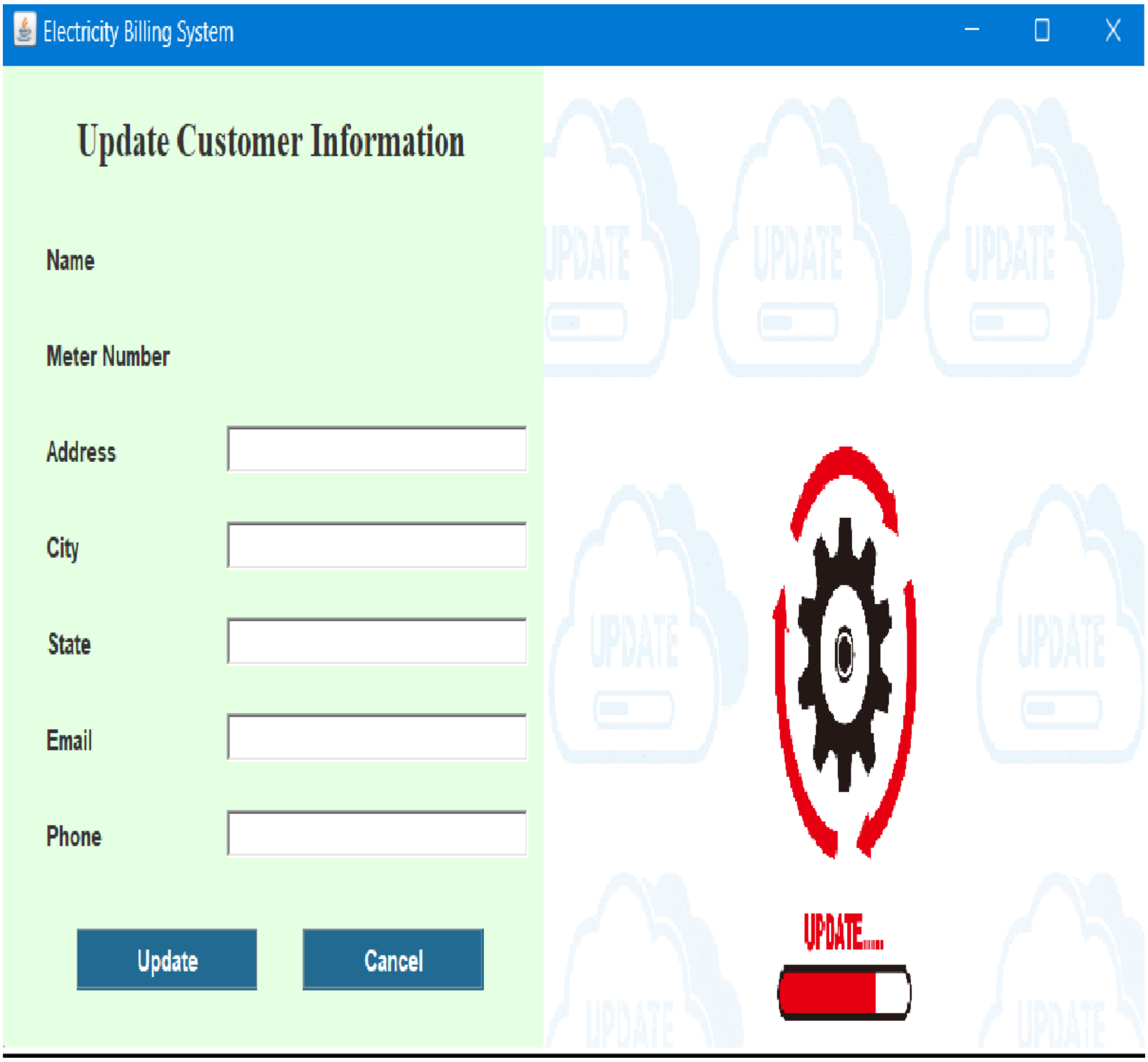
Customer lands on this page after successful login.

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.

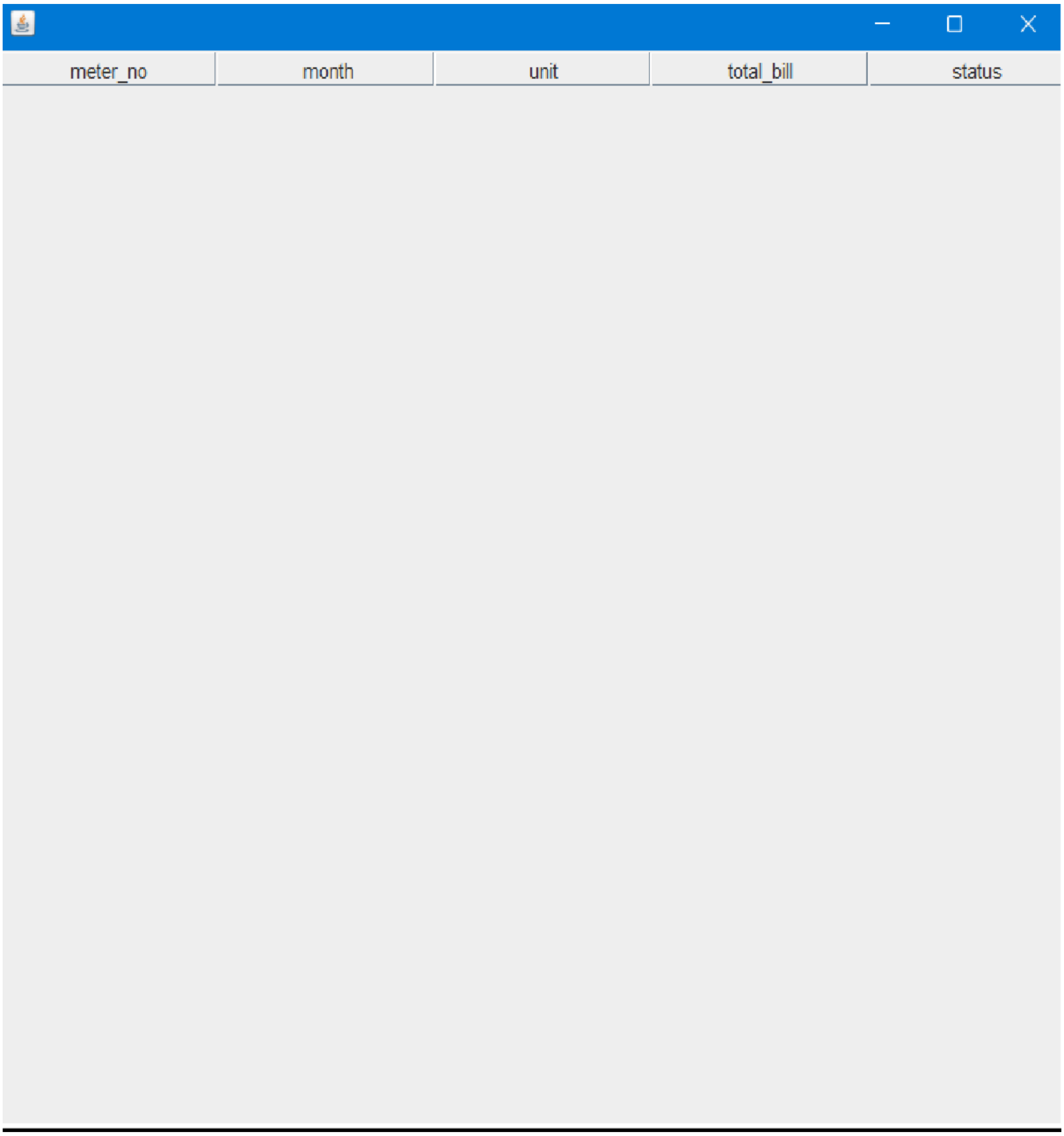
Update Customer Info Screen:-



Name	
Meter Number	
Address	
City	
State	
Email	
Phone	

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

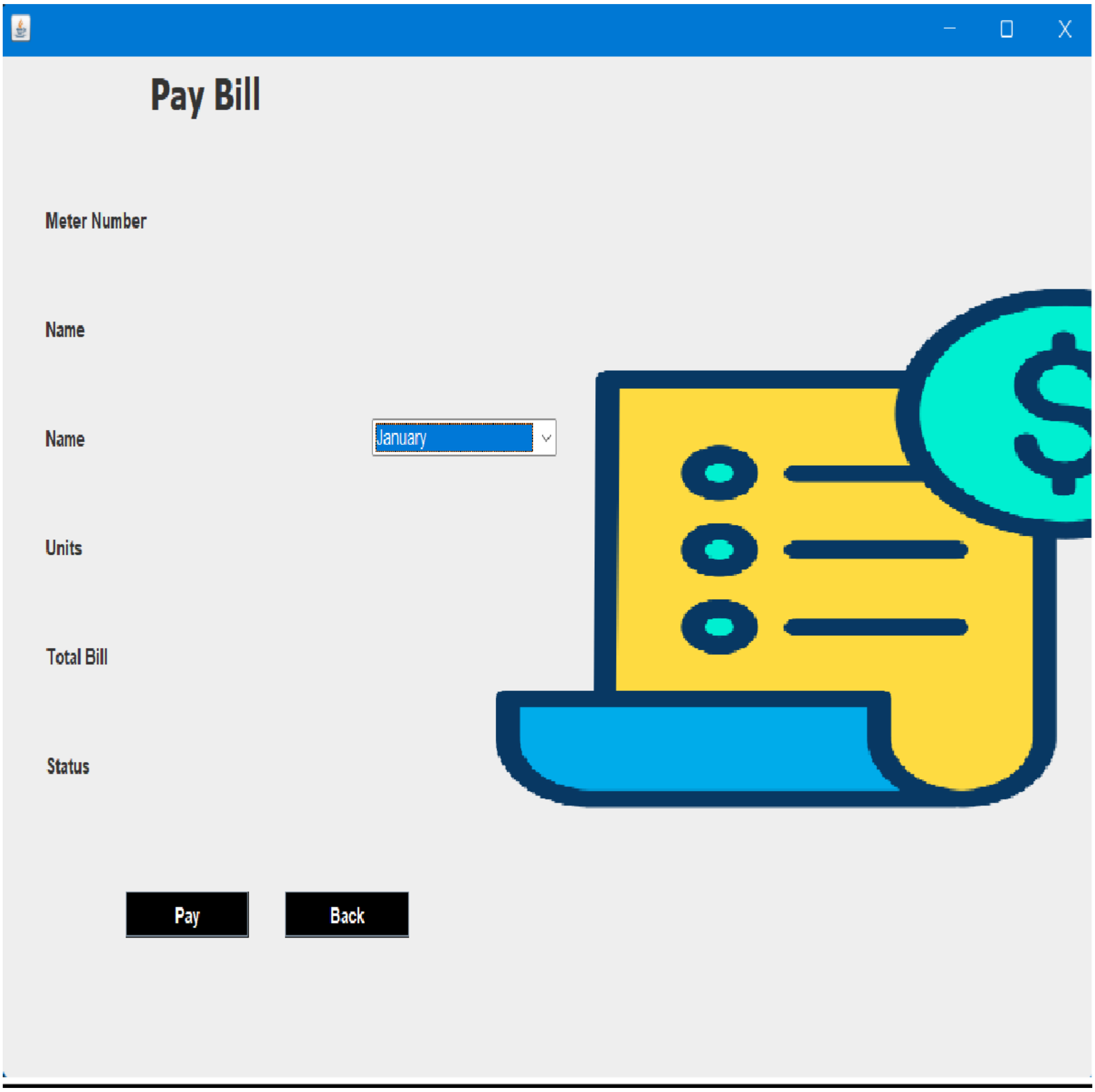
Bill Details Screen for Customers:-



meter_no	month	unit	total_bill	status

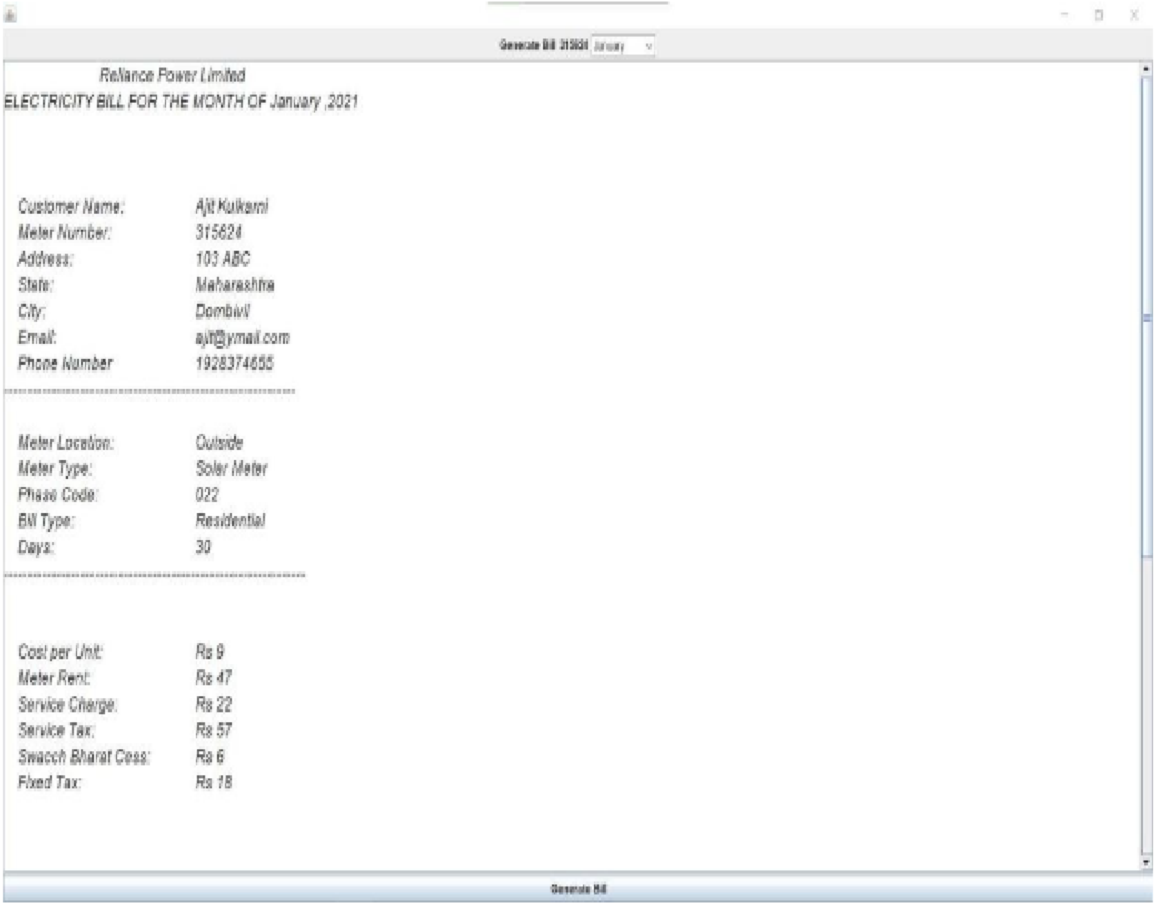
Here every customer can check the status of their bills, whether they have paid the bills or not.

Pay Bill Screen:-



Here customers pay their bills by selecting appropriate month.

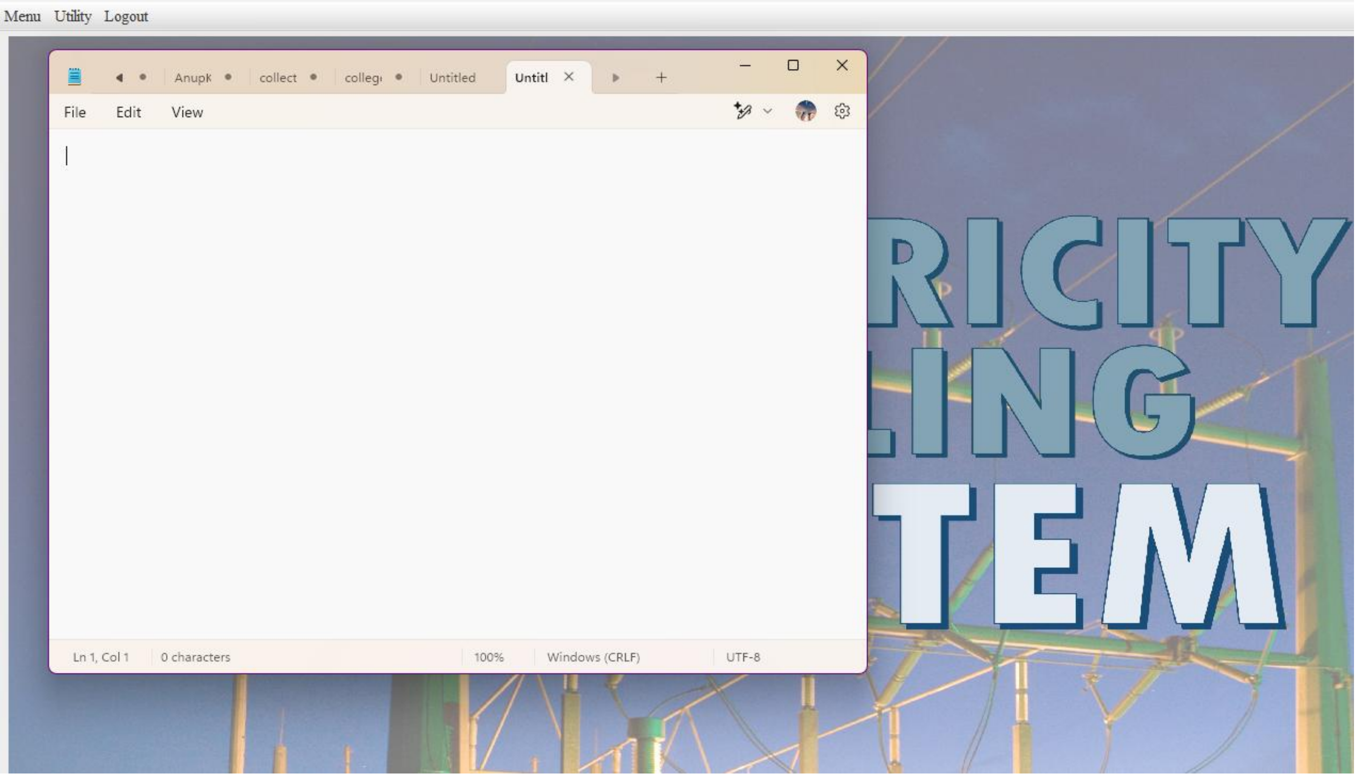
Generate/ Show Bill Screen:-



```
               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
```

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

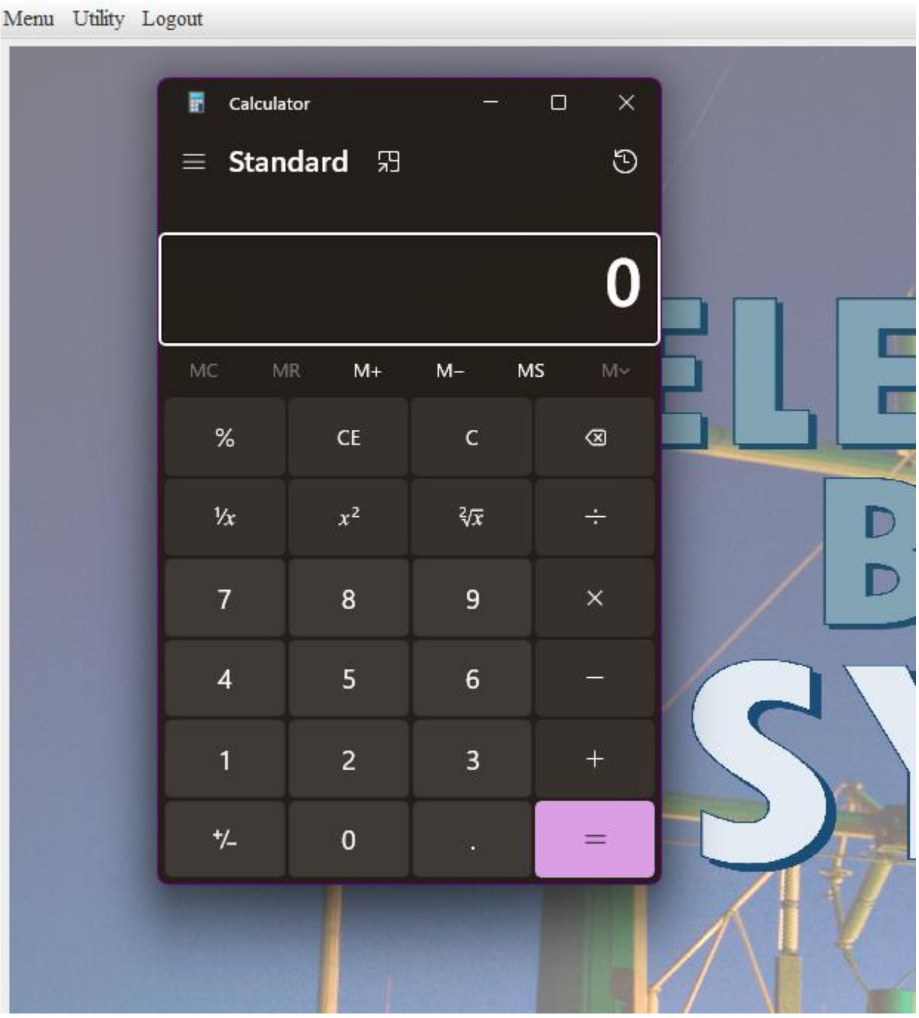
Notepad Screen:-



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.

CHAPTER 7

FUTURE SCOPE AND LIMITATIONS

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.

LIMITATIONS:-

This application cannot be accessed remotely.
** This application requires knowledgeable person to use this application.**
** This application does not have journals.**

CHAPTER 8
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.