HMR INSITUTE OF TECHNOLOGY AND MANAGEMENT

HAMIDPUR, DELHI -110036

Affiliated To

(GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY)

INDUSTRIAL TRAINING REPORT

ON

PROJECT " ELECTRICITY BILLING SYSTEM"

Submitted in partial fulfillment of the
requirements For the award of degree of

Bachelor of

technology In

Computer Science and Engineering 2021-25

Submitted To:
Mr. KARMBIR KHATRI
Assistant Professor

Submitted By:
SHISHIR
02913302721

DECLARATION

I, **SHISHIR GUPTA ,** Student of B Tech (CSE) declare that the project titled "**ELECTRICITY BILLING SYSTEM**" which is submitted by me to Department of computer science and engineering, HMR INSTITUTE OF TECHNOLOGY AND MANAGEMENT, HAMIDPUR DELHI affiliated to Guru Gobind Singh Indraprastha University.

Date: 28TH NOV 2023

SHISHIR
02913302721

ACKNOWLEDGEMENT

The successful completion of this project marks the beginning of an ever going learning experience of converting ideas and concepts into real life, practical system. This project was a quite a learning experience for me at each and every step. At the same time it has given me confidence to work in professional setup. I feel the experience gained during the project will lead me to gain the bright prospect in the future. First of all I would like to give thanks to Head, Education and Training, Engineering Org In for giving me the opportunity to work in this esteemed organization, which has increased my awareness about latest fields. With the deep sense of gratitude, I express my sincere thanks, for their esteemed learning methods and guidance without which it would have been difficult for me to complete this project. I will also like to thank the other working teachers at Engineering Org In. for taking keen interest in my project and giving valuable suggestions and helping me directly or indirectly to complete this project.

SHISHIR GUPTA
02913302721
B Tech CSE

CERTIFICATE

udemy

Certificate no: UC-caf2f1c0-df3f-4db6-bff5-698f3d60e109
Certificate url: ude.my/UC-caf2f1c0-df3f-4db6-bff5-698f3d60e109
Reference Number: 0004

CERTIFICATE OF COMPLETION

45 Real World Java Development Projects Bootcamp Course 2023

Instructors **Engineering Org In**

Shishir Gupta

Date **Nov. 18, 2023**
Length **75.5 total hours**

ABSTRACT

Electricity consumers are often faced with the problem of inaccuracy and delay in monthly billing due to some drawbacks. Thus, it is essential to have an efficient system for such purposes via electronic platform with consideration to proximity. The proposed system automates the conventional process of paying electricity bill by visiting the Electricity Board which is tiresome and time consuming. It is also designed to automate the electricity bill calculation and payment for user convenience. The system is developed with JAVA as the base programming language which can be used to develop websites, web applications and web services. The MYSQL is a relational database management system based on Structured Query Language (SQL) which is used for the purpose of web database. The system would be having two logins: the administrative and user login. The administrator can view the user's account details and can add the customer's information of consuming units of energy of the current month in their account. The Admin must feed the system with the electricity usage data into respective user's account. The system then calculates the electricity bill for every user and updates the information into their account every month. Users can then view their electricity bill and pay before the month end.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1 General

Electricity Billing System is a software-based application.

1) This project aims at serving the department of electricity by computerizing the billing system.
2) It mainly focuses on the calculation of units consumed during the specified time and the money to be charged by the electricity offices.
3) 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 reading 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.

1.2 Preamble

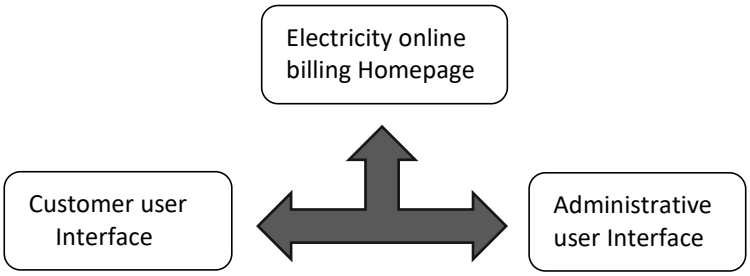
The system excludes the need of maintaining paper electricity bill, administrator does not have to keep a manual track of the users, users can pay the amount without visiting the office. Thus, it saves human efforts and resources. Me, the owner of project, respect all customers and make them happy with my service.

The main aim of my 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:

a) To keep the information of consuming unit energy of current month.
b) To keep the information of Customer.
c) To keep the information of consuming unit energy of previous month.
d) To calculate the units consumed every month regularly.
e) To generate the bills adding penalty and rent.
f) To save the time by implementing payment process online.



1.3 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.

1.4 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.

CHAPTER 2
ANALYSIS AND SYSTEM REQUIREMENT

2.1 Existing and Proposed System

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

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

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

2.2 Feasibility Study

Feasibility study is the phase in which the analyst checks that the candidate system is feasible for the organization or not. This entails identification, description & evaluation of the system. Feasibility study is done to select the best system that meets the performance requirement.

If the feasibility study is to serve as a decision document, it must answer key questions.

1) Is there a new and better way to do the job that will benefit the user?
2) What are the costs and savings of the alternatives?
3) What is recommended?

The most successful system projects are not necessarily the biggest or most visible in the business but rather those truly meet user's expectations.

Feasibility considerations

Three key considerations are involved in the feasibility study. They are as follows:-

Economic Feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system.

We analyse the candidate system (computerized system) is feasible as than the manual system because it saves the money, time and manpower. It also feasible according to cost benefits analysis

Technical Feasibility:

Technical feasibility centers around the technology used. It means the candidate system is technically feasible i.e. it don't have any technical fault and work properly in the given environment. Our system is technically feasible; it is providing us required output.

Behavioral Feasibility:

Behavioral feasibility is the analysis of behavior of the candidate system. In this we analyse that the candidate system is working properly or not. If working than it communicating proper with the environment or not. All this matters are analyzed and a good candidate system is prepared. Due to the change of system what is the change in behaviour of the users, this factors are also analyzed.

2.3 System Development Environment

System development environment shows the hardware and software requirement, which are necessary for developing the software. Necessary software, hardware requirement, which are necessary for making this software are as follows:

2.3.1: Hardware Requirements

- Hardware specification: Intel Pentium Processor
- 32 MB RAM or Higher
- 1.2 GB Hard Disk or Greater
- Video Display Unit
- Keyboard
- Mouse

2.3.1: Software Requirements

- Operating System: Windows 11
- Software: Microsoft SQL Server
- Front End: Jsp Servlet
- Back End: Database – MySQL
  Business Logic- Java

CHAPTER 3
FUNCTIONS, SCOPE, REPORTS AND MODULES

3.1 Functionalities Provided

- Provides the searching facilities based on various factors. Such as Electricity, Meter, Bill, Payment.
- Electricity Billing System also manage the Customer details online for Bill details, Payment details, Electricity.
- It tracks all the information of Connection, Customer, Bill etc.
- Shows the information and description of the Electrical Meter.
    - It deals with monitoring the information and transactions of Bill.
    - Manage the information about Electricity.
    - Editing, adding and updating of Records is improved which results in proper resource management of Electricity data.
    - Manage the information of Bill.
    - Integration of all records of Payment.

3.2 Scope of Project

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Electricity Billing System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

This project aims at Business process automation, i.e. I have tried to computerize various processes of Electricity Billing System.

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.

- The system generates types of information that can be used for various purposes.
- It satisfy the user requirement.
- Be easy to understand by the user and operator.
- Be easy to operate.
- Have a good user interface.
- Be expandable.
- Delivered on schedule within the budget.

3.3 Reports of Project

- It generates the report on Electricity, Connection, Customer
- Provide filter reports on Meter, Bill, Payment
- You can easily export PDF for the Electricity, Customer, Bill
- Application also provides excel export for Connection, Meter, Payment
- You can also export the report into csv format for Electricity, Connection, Payment

3.4 Modules of Project

- Electricity Management Module: Used for managing the Electricity details.
- Payment Module: Used for managing the details of Payment.
- Customer Module: Used for managing the details of Customer.
- Meter Module: Used for managing the meter details.
- Login Module: Used for managing the login details.
- Users Module: Used for managing the users of the system.
- Bill Module: Used for managing the Bill information.
- Signup Module: Used for new customers or admins.

3.5 Input Data and Validation of Project

- All the fields such as Electricity, Meter, Payment are validated and does not take invalid values
- Each form for Electricity, Connection, Customer can not accept blank value fields
- Avoiding errors in data
- Controlling amount of input

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.
- Functionality of the entire module/forms.

CHAPTER 4
SYSTEM DESIGN AND MODELLING

4.1 System Architecture

The system architecture gives the overview of the organizational system that shows the system boundaries, external entities that interact with the system, and the major information that flows between the entities and the system.

4.2 Activity Diagram

Activity diagrams are an indispensable tool for projects due to their ability to visually map out the sequence of activities, actions, and decisions within a system or process. They serve as a graphical representation that simplifies complex workflows, allowing stakeholders and team members to easily comprehend the project's operational structure. These diagrams are crucial for identifying dependencies between various tasks, enabling effective scheduling and resource allocation.

Activity diagrams play a pivotal role in risk management by highlighting potential points of failure or risks within the project, enabling proactive measures to mitigate these issues.

key features and benefits of using activity diagrams:

- They show the sequence of actions or steps required to complete a process or achieve a particular goal.
- These diagrams provide a clear and intuitive visualization of how activities are sequenced, their dependencies, decision points, and parallel activities.
- Each activity or action can be described in detail using text annotations or descriptions, making it easier for stakeholders to understand the purpose of each step.
- These diagrams can be used as documentation for project planning, outlining the steps and activities required for a specific process.
- They assist in organizing and structuring project tasks and activities.
- Activity diagrams can aid in identifying bottlenecks, inefficiencies, or areas for optimization within a process flow.

Fig 4.1 Activity Diagram of system

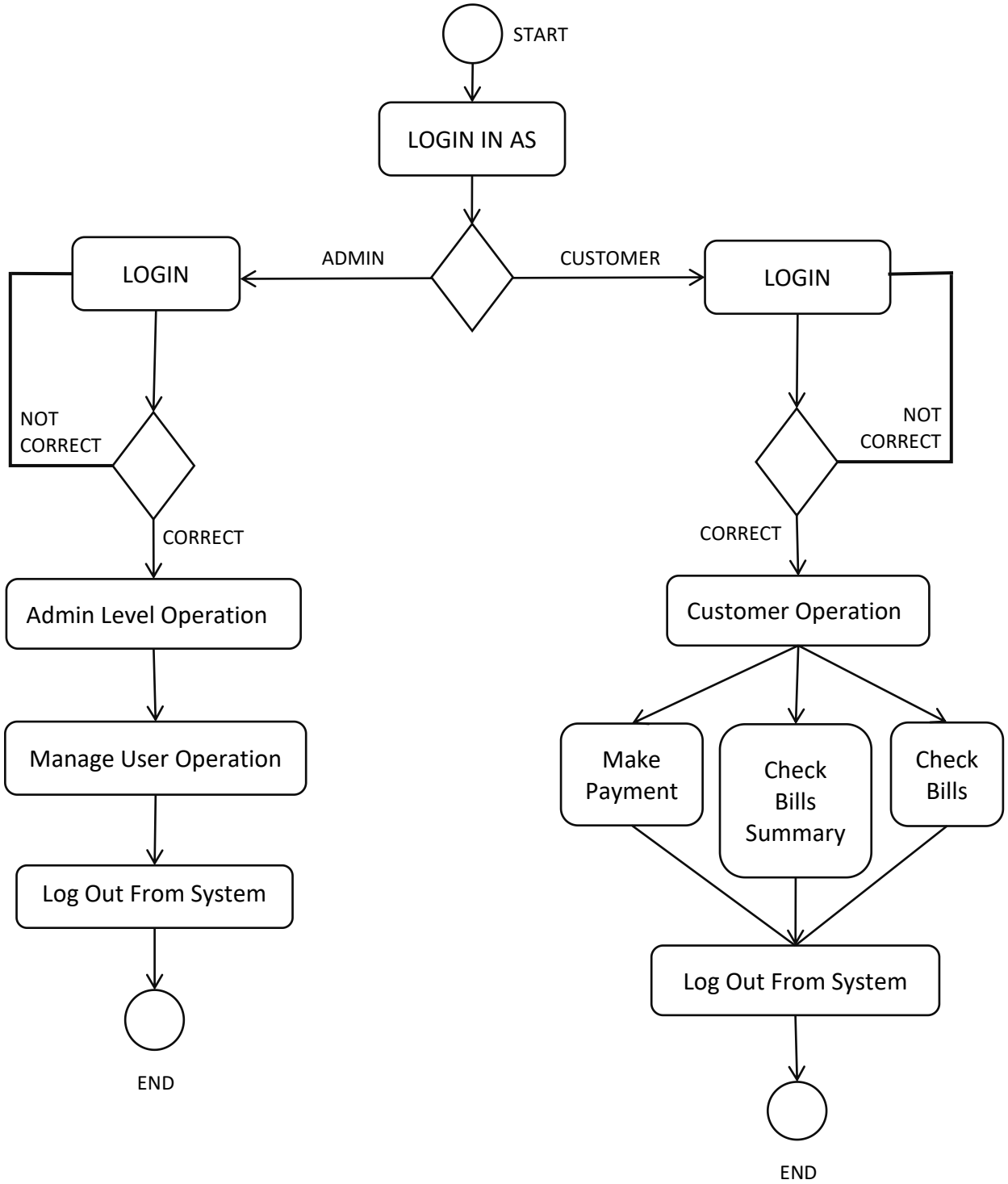


Fig 4.2 Admin Class Diagram

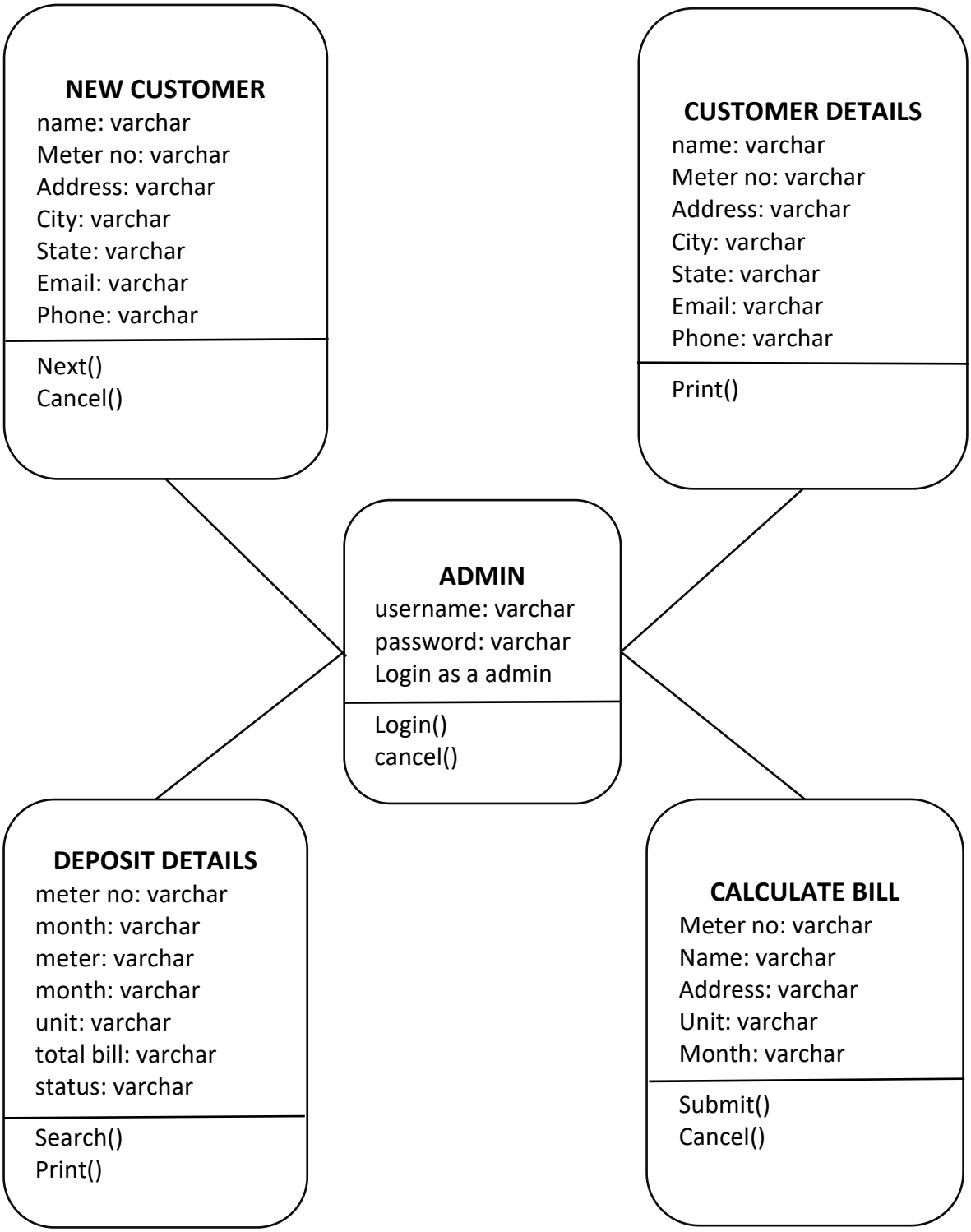


Fig 4.3 Customer Class Diagram

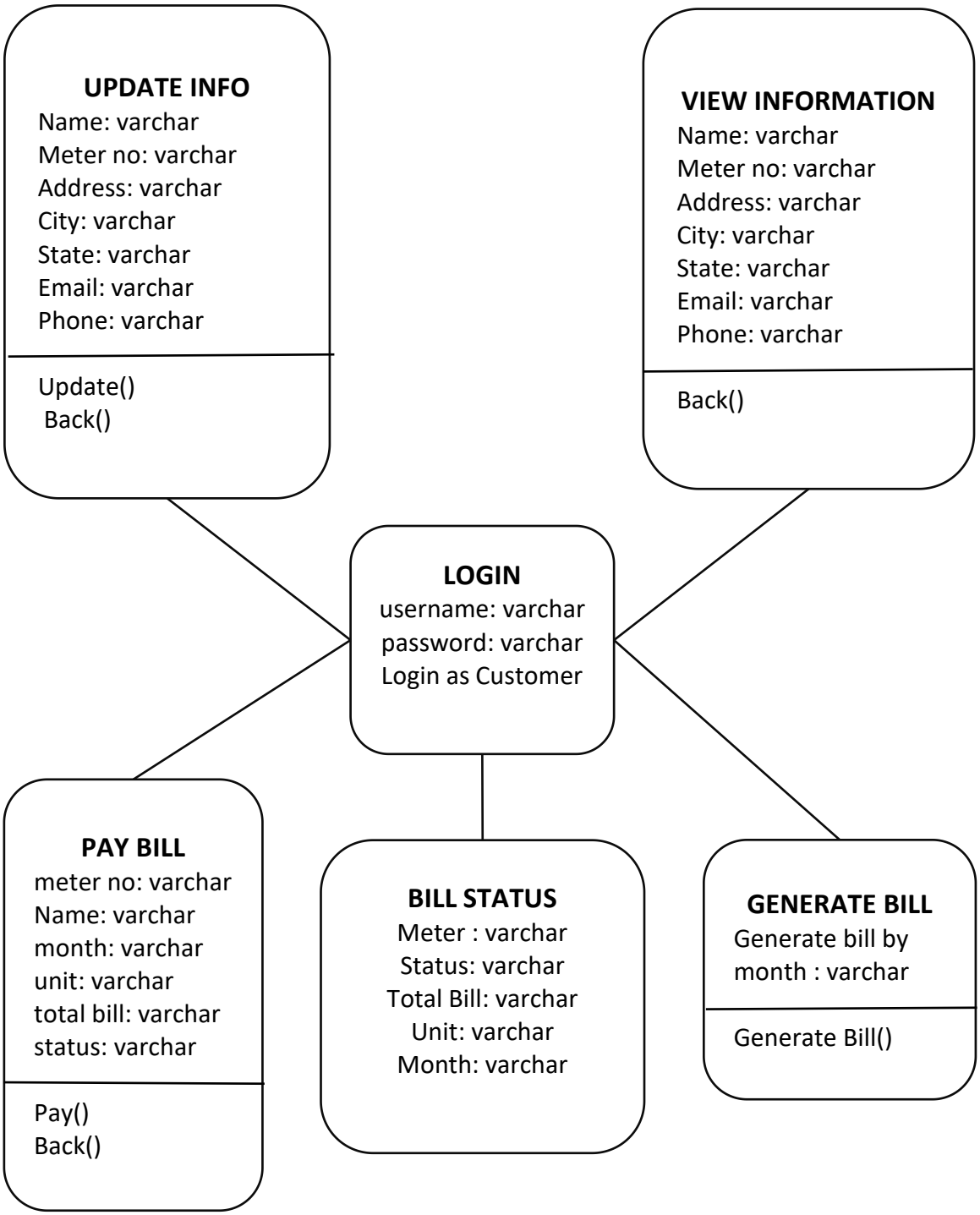


Fig 4.4 Customer Details Activity Diagram

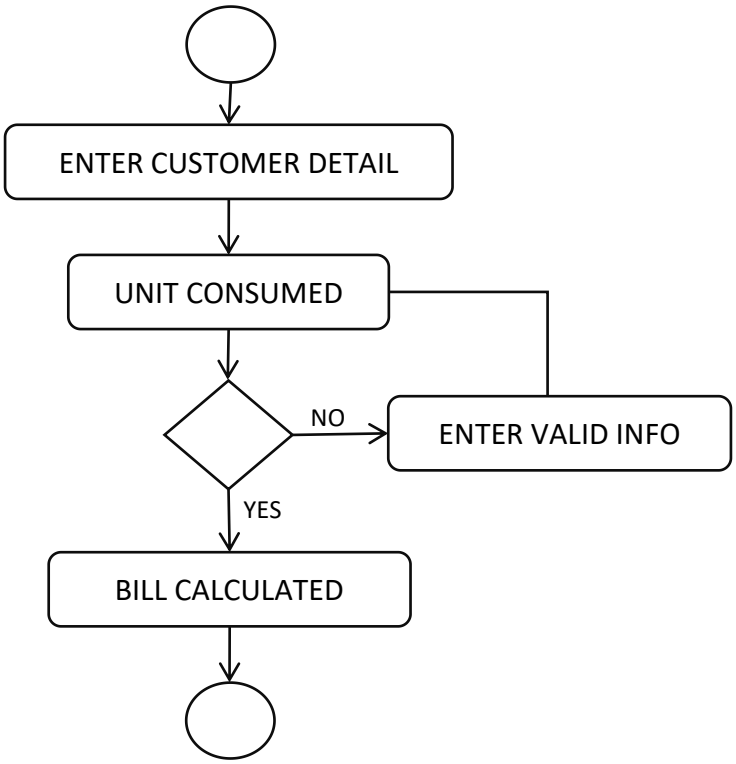


Fig 4.5 Deposit Details Activity Diagram

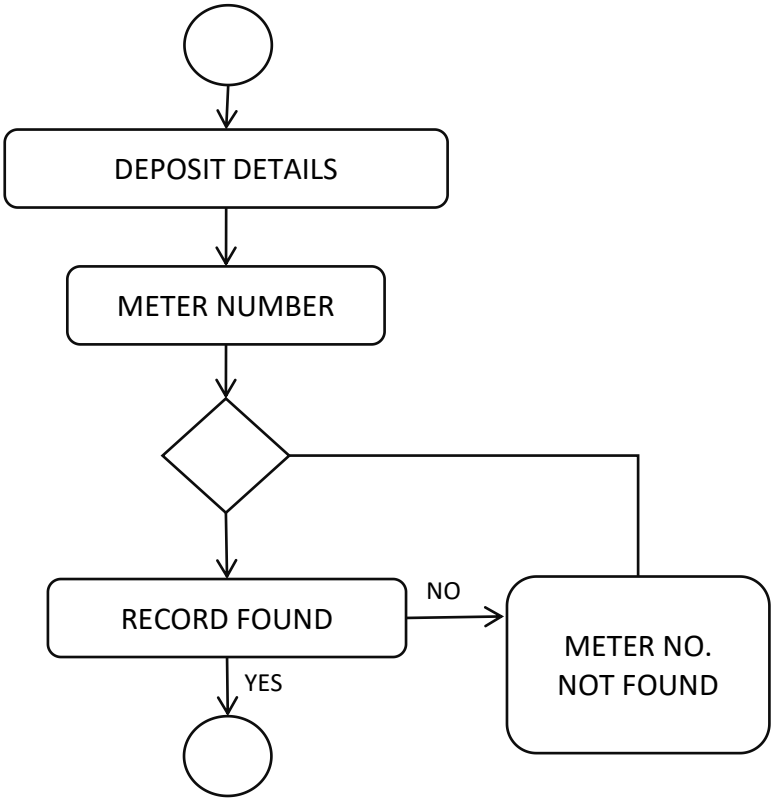
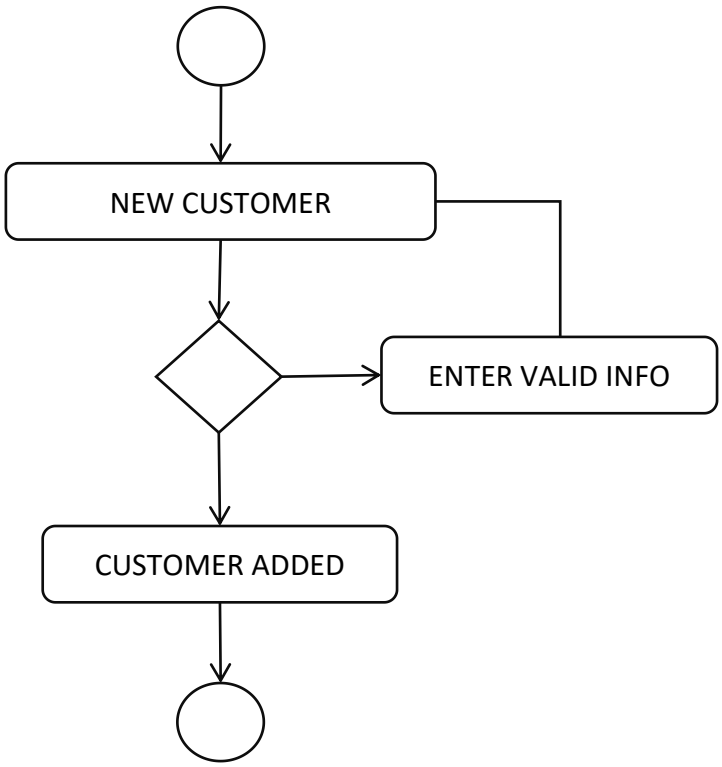


Fig 4.6 Add New Customer Activity Diagram



4.3 Sequence Diagram

Sequence diagrams in project building, particularly in the context of software development or system design, are visual representations that illustrate the interactions and flow of messages between various components or objects within a system. These diagrams showcase the chronological order of these interactions, emphasizing the sequence of events as they occur during the execution of a particular use case, scenario, or function within the project.

Creating a sequence diagram for a project involves depicting interactions between different components, modules, or objects in a system over a specific period. It showcases the flow of messages or actions between these entities.

Sequence diagrams are dynamic and can evolve with the project. They are useful for understanding the flow of a specific scenario or use case within a system and serve as a valuable communication tool among stakeholders, developers, and designers.

Fig 4.7 Customer Sequence Diagram

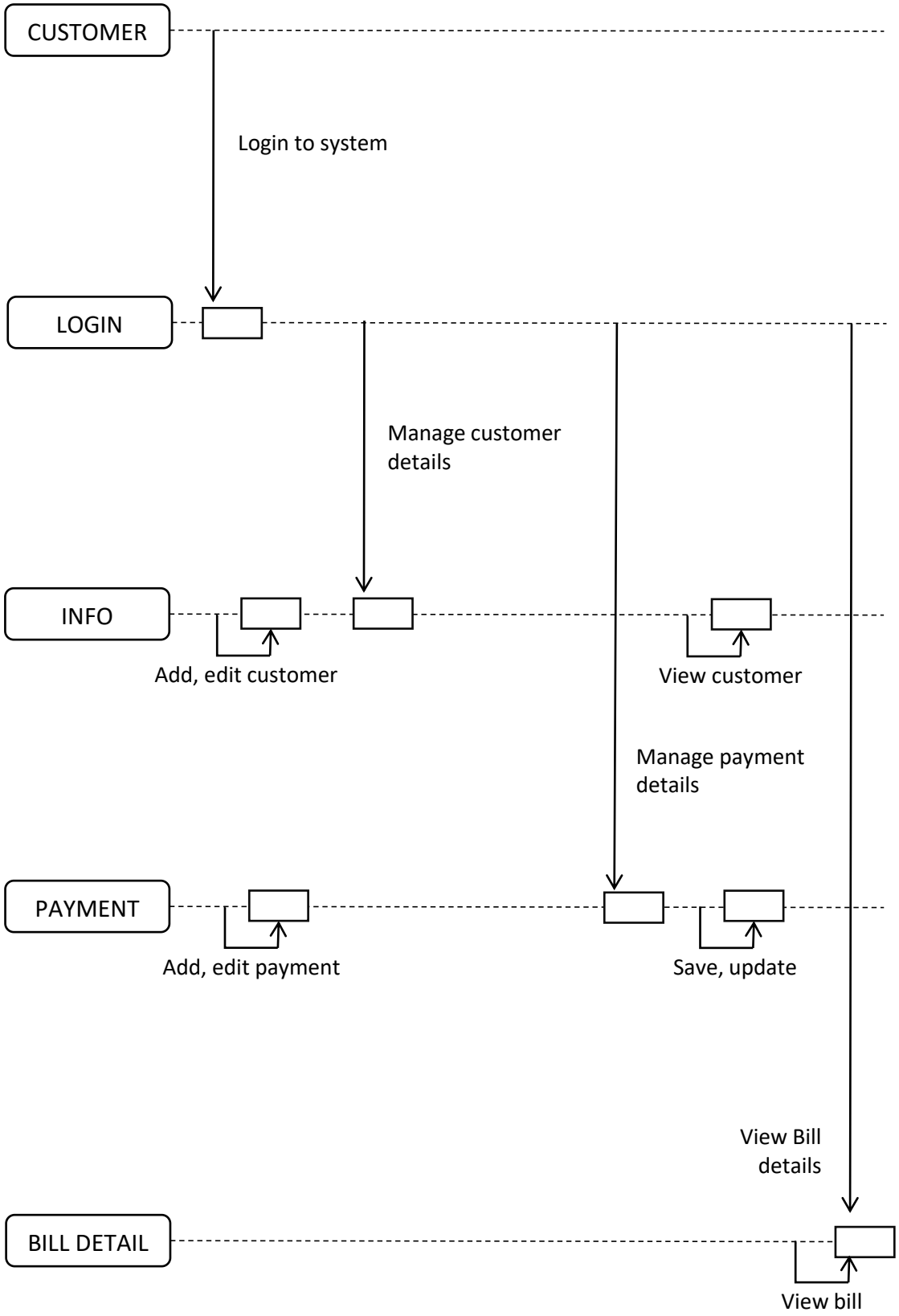
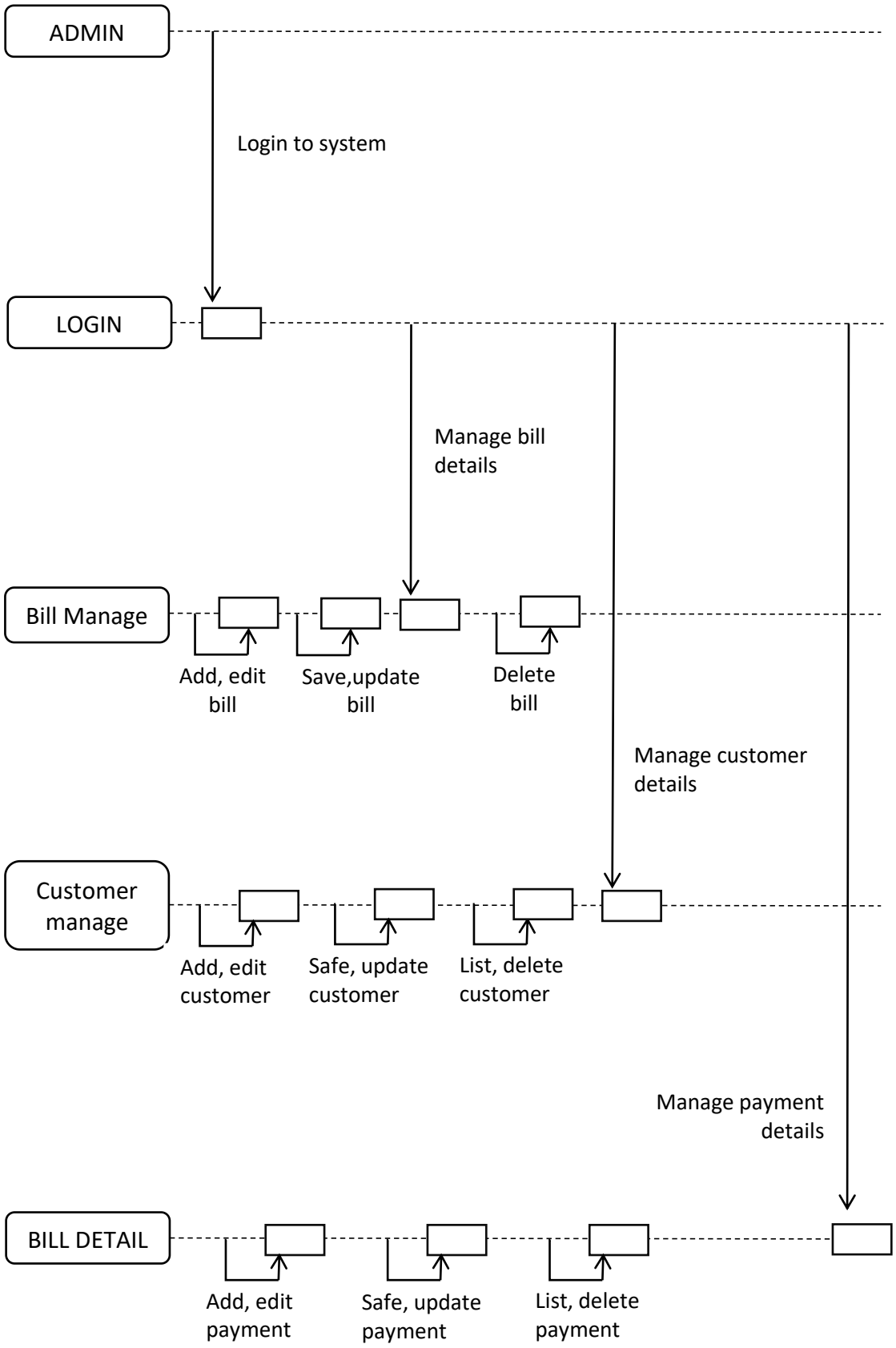


Fig 4.8 Admin Sequence Diagram



4.4 Use Case Diagram

Use case diagrams are instrumental in projects as they offer a high-level visual representation of the functionalities and interactions between users (actors) and the system. These diagrams are crucial for several reasons. Firstly, they provide a comprehensive overview of the system's behavior from a user's perspective, outlining various possible interactions and scenarios. This aids in understanding the system's functionalities and how users will interact with it, serving as a blueprint for system development.

Use case diagrams are valuable in guiding testing efforts, as they help in creating test cases based on the defined user interactions, ensuring comprehensive coverage of system functionalities. Overall, use case diagrams are essential in project development as they facilitate understanding, communication, requirement analysis, scope definition, and testing efforts, contributing significantly to the success of the project.

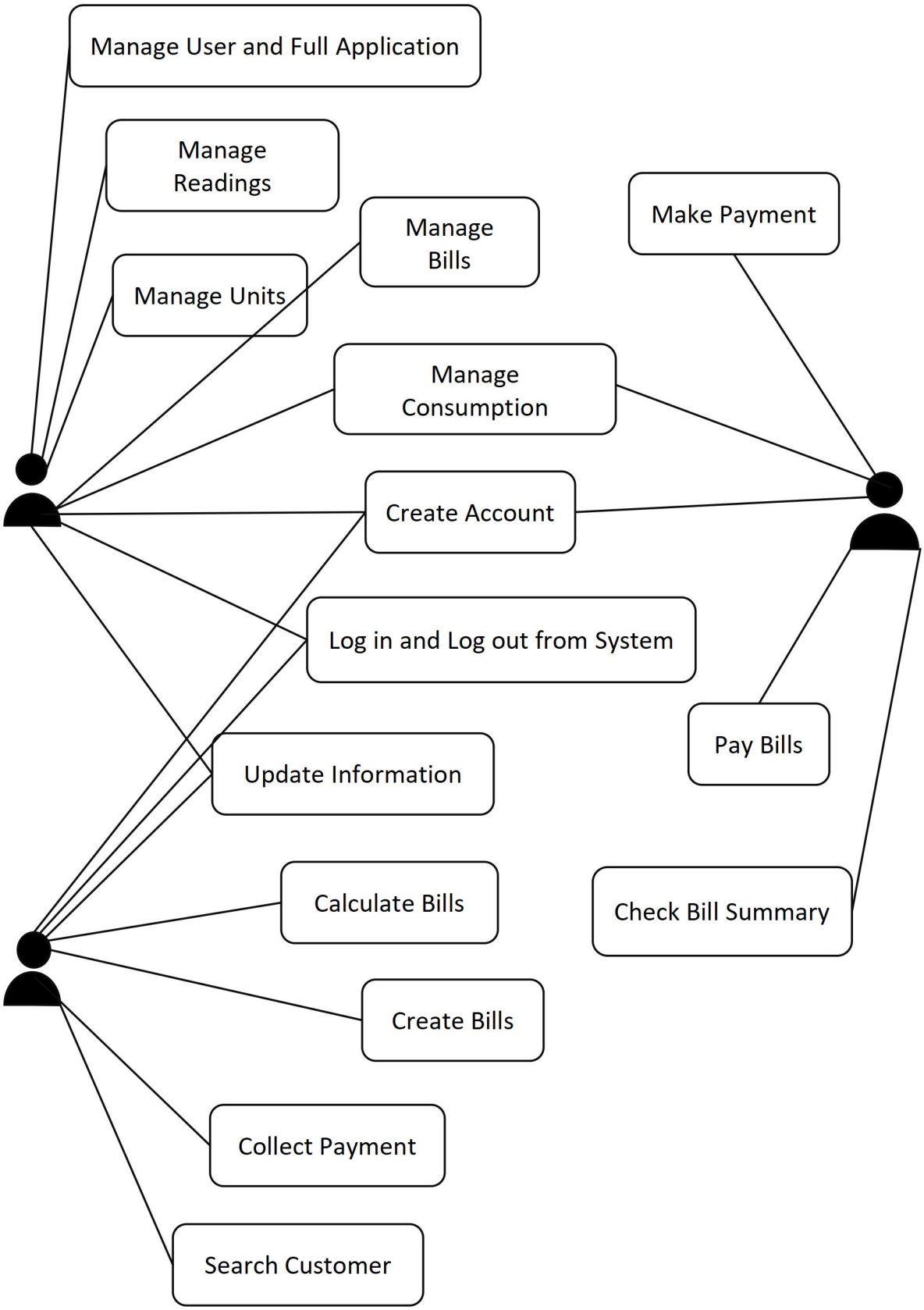
Use case diagrams are beneficial in various ways:

- They help in understanding and documenting user requirements by capturing user-system interactions and functionalities.
- Use case diagrams serve as a communication tool between stakeholders, business analysts, designers, and developers, enabling a shared understanding of system functionalities.
- They aid in designing the system's structure and functionalities by providing an overview of how users will interact with the system.
- Use case diagrams can guide testing efforts by providing a basis for validating whether the system fulfills the specified user requirements.

Key components of a use case diagram include:

- Actors represent the different types of users or external systems interacting with the system
- Use cases represent the specific functionalities or services provided by the system.
- Relationships between actors and use cases are depicted using lines or arrows. An actor interacts with a use case by initiating or participating in it.
- The boundary of the use case diagram represents the scope of the system being modeled. It encloses all the use cases and actors involved within the system.

Fig 4.9 Use Case Diagram of Project



4.5 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.

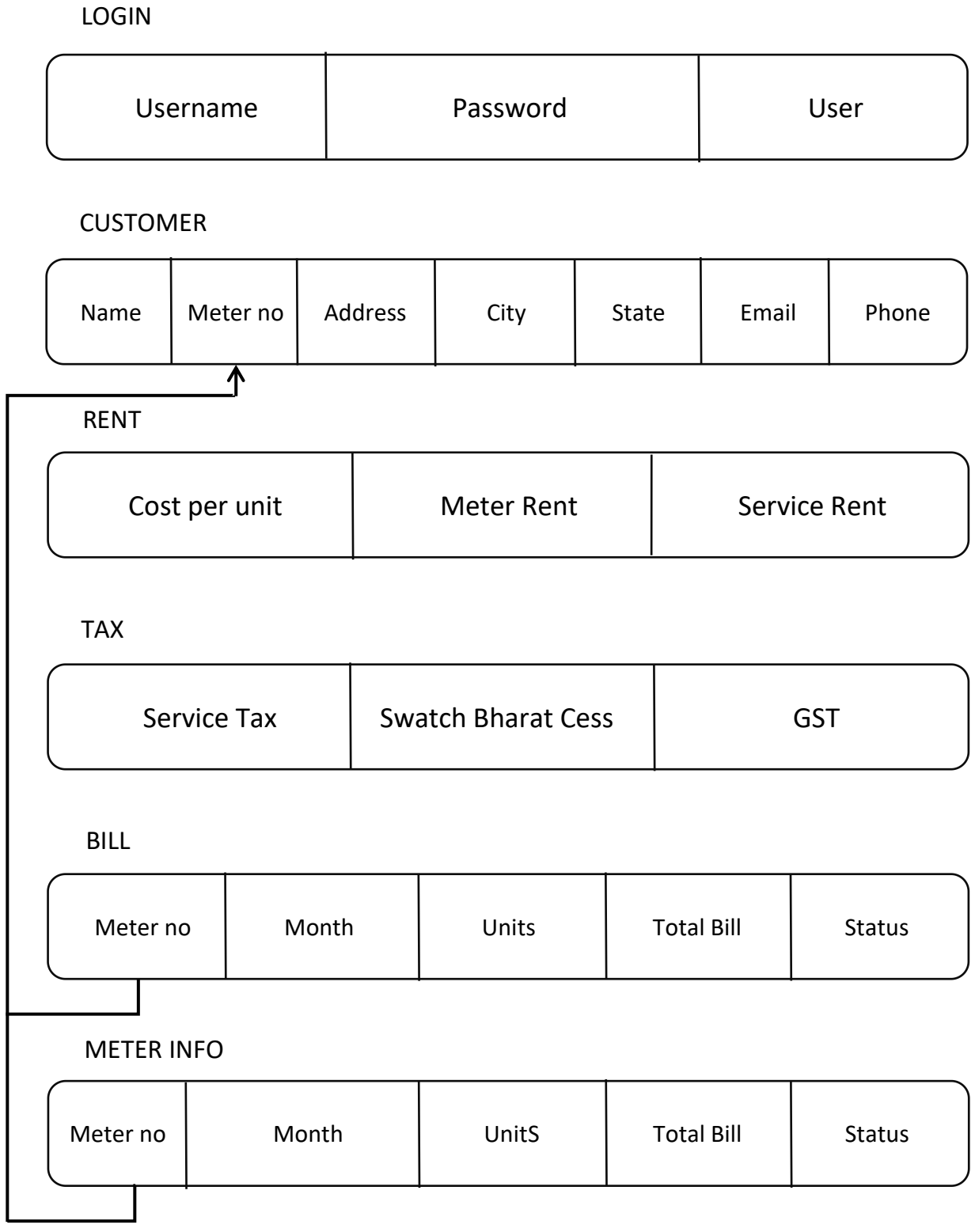
A primary key is a table column that can be used to uniquely identify every row of the table. Any column that has this property, these columns are called candidate key. A composite primary key is a primary key consisting of more than one column. A foreign is a column or combination of columns that contains 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 3.7 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.

- 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”, "swatch_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 ".

Fig 4.10 Schema Diagram of Project



CHAPTER 5 - DEVELOPMENT ENVIORMENT

5.1 Introduction to SQL

Structured Query Language (SQL) is a powerful programming language used for managing and manipulating relational databases. It is designed for storing, manipulating, and retrieving data stored in relational database management systems (RDBMS). SQL provides a standardized way to interact with databases and perform various operations.

SQL is a versatile language used across various industries and applications for managing and querying databases. Understanding SQL fundamentals is essential for working with databases, data analysis, software development, and various data-related roles.

SQL operates through a set of commands categorized into Data Manipulation Language (DML), Data Definition Language (DDL), Data Control Language (DCL), and Data Query Language (DQL). With DML, users can perform operations like selecting, inserting, updating, and deleting data from database tables. DDL enables the creation, alteration, and deletion of database structures such as tables, views, and indexes. DCL manages access permissions and user privileges within the database, granting or revoking access as needed. SQL's primary strength lies in its ability to execute queries effectively, allowing users to retrieve specific data using SELECT statements, apply filtering conditions through WHERE clauses, perform joins across multiple tables, aggregate data using functions like GROUP BY, and sort results using ORDER BY.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

Here's an introduction to the fundamental aspects of SQL:

- **Data Manipulation Language (DML)**:
  - **SELECT**: Retrieves data from one or more tables.
  - **INSERT**: Adds new rows of data into a table.
  - **UPDATE**: Modifies existing data in a table.
  - **DELETE**: Removes rows of data from a table.

✧ **Data Definition Language (DDL)**:

- **CREATE**: Creates new database objects like tables, views, indexes, etc.
- **ALTER**: Modifies existing database objects like tables, adding or by removing columns, etc.
- **DROP**: Deletes database objects like tables, views, etc.

✧ **Data Control Language (DCL)**:

- **GRANT**: Provides users with specific permissions to database objects.
- **REVOKE**: Removes or revokes specific permissions from users.

✧ **Data Query Language (DQL)**:

- **SELECT**: The primary command used for fetching data from a database table.

✧ **Database Concepts**:

- **Tables**: Represents the structure for storing data in rows and columns.
- **Columns/Fields**: Represents specific data attributes within a table.
- **Rows/Records**: Individual entries or instances of data in a table.
- **Constraints**: Rules applied to columns to maintain data integrity (e.g., NOT NULL, UNIQUE, FOREIGN KEY, etc).
- **Indexes**: Improves the speed of data retrieval by creating pointers to data in a table.

✧ **Basic Queries**:

- **SELECT statement**: Retrieves data from one or more tables based on specified criteria.
- **WHERE clause**: Filters data based on specific conditions.
- **JOIN**: Combines data from multiple tables based on a related column between them.
- **GROUP BY**: Groups rows sharing a common value in one or more columns.
- **ORDER BY**: Sorts the result set based on specified columns.
- **Aggregate Functions**: Functions like COUNT, SUM, AVG, MIN, MAX to perform operations on sets of values.

✧ **Sub-queries**:

Queries nested within other queries to retrieve or manipulate data.SQL is a versatile language used across various industries and applications for managing and querying databases. Understanding SQL fundamentals is essential for working with databases, data analysis, software development, and various data-related roles.

5.2 Introduction to JAVA

- Java is an object oriented programming language originally developed by Sun Microsystems and released in 1995.
- Java was originally developed by James Gosling at Sun Microsystems (which has since merge into Oracle Corporation).
- Java programs are platform independent which means they can be run on any operating system with any type of processor as long as the Java interpreter is available on that system.
- Java code that runs on one platform does not need to be recompiled to run on another platform, it's called "write once, run anywhere" (WORA).
- Java virtual machine (JVM) executes Java code, but is written in platform specific languages such as C/C++/ASM etc. JVM is not written in Java and hence cannot be platform independent and Java interpreter is actually a part of JVM.

5.2.1 Uses of JAVA

Earlier, java was only used to design and program small computing devices but later adopted as one of the platform independent programming language and now according to Sun, 3 billion devices run java. Java is one of the most important programming language in today's IT industries.

- SP – Java is used to create web applications like PHP and ASP, JSP(Java Server Pages) used with normal HTML tags, which helps to create dynamic web pages.
- Applets – This is another type of Java program that used within a web page to add many new features to a web browser.
- J2EE – The software Java 2 Enterprise Edition are used by various companies to transfer data based on XML structured documents between one another.

- JavaBeans – This is something like Visual Basic, a reusable software component that can be easily assemble to create some new and advanced application.
- Mobile – Besides the above technology, Java is also used in mobile devices, many kind of games and services built in Java. Today, all leading mobile service provider like Nokia, Siemens, Vodafone are using Java technology.

5.2.2 Types of Java Applications

- Web Application Java is used to create server side web applications. Currently, servlet, jsp, struts, jsfetc technologies are used.
- Standalone Application It is also known as desktop application or window-based application. An application that we need to install on every machine or server such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.
- Enterprise Application An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.
- Mobile Application Java is used to create application software for mobile devices. Currently Java ME is used for creating applications for small devices, and also Java is programming language for Google Android application development.

5.2.3 Facts about Java

- Object Oriented – In java everything is an Object. Java can be easily expanded since it is based on the Object model.
- Platform independent – C and C++ are platform dependency languages hence the application programs written in one Operating system cannot run in any other Operating system, but in platform independence language like Java application programs written in one Operating system can able to run on any Operating system.
- Simple – Java is designed to be easy to learn. If you understand the basic concept of OOP java would be easy to master.
- Secure – With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public key encryption.

- Architectural neutral – Java compiler generates an architecture neutral object file format which makes the compiled code to be executable on many processors, with the presence Java runtime system. Portable – being architectural neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler and Java is written in ANSI C with a clean portability boundary which is a POSIX subset.
- Robust – Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- Multithreaded – With Java's multithreaded feature it is possible to write programs that can do many tasks simultaneously. This design feature allows developers to construct smoothly running interactive applications.
- Interpreted – Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.
- High Performance – With the use of Just-In-Time compilers Java enables high performance.
- Distributed – Java is designed for the distributed environment of the internet.
- Dynamic – Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of runtime information that can be used to verify and resolve accesses to objects on runtime.

5.3 Introduction to NETBEANS 8.1

NetBeans is as software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules.

Applications based on the NetBeans Platform, including the NetBeans integrated development environment can be extended by third party developers. The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris other platforms supporting a compatible JVM.

These modules are part of the NetBeans IDE -

- **NetBeans Profiler**

  The NetBeans Profiler is a tool for the monitoring of Java applications: It helps developers find memory leak sand optimize speed. Formerly downloaded separately, it is integrated into the core IDE since version 6.0

  The Profiler is based on a Sun Laboratories research project that was named JFluid. That research uncovered specific techniques that can be used to lower the overhead of profiling a Java application. One of those techniques is dynamic bytecode instrumentation, which is particularly useful for profiling large Java applications.

  Using dynamic bytecode instrumentation and additional algorithms, the NetBeans Profiler is able to obtain runtime information on applications that are too large or complex for other profilers. NetBeans also support Profiling Points that let you profile precise points of execution and measure execution time.

- **GUI design tool**

  Formerly known as project Matisse, the GUI design-tool enables developers to prototype and design Swing GUIs by dragging and positioning GUI components. The GUI builder has built-in support for JSR 295 (Beans Binding technology), but the support for JSR 296 (Swing Application Framework) was removed in 7.1.

5.3.1 Netbeans features

- **Best Support for Latest Java Technologies**

  NetBeans IDE is the official IDE for Java 8. With its editors, code analyzers and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyzers and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs.

- **Fast & Smart Code Editing**

  An IDE is much more than a text editor. The NetBeans Editor indents lines, matches words and brackets, and highlights source code syntactically and semantically.

✧ **Easy & Efficient Project Management**

Keeping a clear overview of large applications, with thousands of folders and files, and millions of lines of code, is a daunting task. NetBeans IDE provides different views of your data, from multiple project windows to helpful tools for setting up your applications and managing them efficiently, letting you drill down into your data quickly and easily, while giving you versioning tools via Subversion, Mercurial, and Git integration out of the box.

✧ **Write Bug Free Code**

The cost of buggy code increases the longer it remains unfixed. NetBeans provides static analysis tools, especially integration with the widely used Find Bugs tool, for identifying and fixing common problems in Java code. In addition, the NetBeans Debugger lets you place break points in your source code, add field watches, step through your code, run into methods, take snapshots and monitor execution as it occurs.

✧ **Support for Multiple Languages**

NetBeans IDE offers superior support for C/C++ and PHP developers, providing comprehensive editors and tools for their related frameworks and technologies. In addition, the IDE has editors and tools for XML, HTML, PHP, Groovy, Javadoc, JavaScript, and JSP.

✧ **Cross Platform Support**

NetBeans IDE can be installed on all operating systems that support Java, from Windows to Linux to Mac OS X systems. Write Once, Run Anywhere, is as true for NetBeans IDE as it is for your own applications... because NetBeans IDE itself is written in Java, too.

CHAPTER 6 - IMPLIMENTATION

6.1 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.
- ✧ **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.

6.2 Implementation of SQL statements

Insert statement:

- The INSERT INTO statement is used to insert new records in a table.
- The INSERT INTO syntax would be as follows: INSERT INTO table_name VALUES (value1, value2, value3, ...)
- The following SQL statement insert's a new record in the “customer” table: Insert into customer VALUES (“gupta”, “12345”, “btm” ,“Bangalore”, “Delhi”, “gupta@gmail.com”, “9876543833”).

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= gupta@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));

6.3 Algorithm or pseudo code of implementation

Explanation of Algorithm or pseudocode of system:

- Start system
- Enter login name and password
- On clicking the login button
- Connect to database
- Query database to know whether user credentials are correct
- If not, deny access and return login page with an error message
- If correct, check if credentials for administrator
- If yes, allow login
- Set admin session, re-direct administrator to admin login page
- If no, allow login set user session
- Re-direct user to user home page

Algorithm or pseudocode of admin

Login:

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

New Customer:

- This program will allow the admin to enter the customer details and automatically generates unique meter number.
- If customer name, address, city, state, email and phone number is entered, insert the values into customer.
- 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.
- It will allow the admin to delete the customer info when meter_no is selected.

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.

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 generate bill button.

CHAPTER 7 - TESTING

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

7.1 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.

7.2 Testing objectives

The main objectives of testing process are as follows:

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

7.3 Levels of Testing

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

7.3.1 Unit Testing

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

Table 7.1: 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	98977a	Phone number is invalid	Alphabets are being take n as input for phone number	-

Table 7.2: Positive test case for phone number insertion

Function Name	Input	Expected output	Error	Resolved
Input phone number	9897788888	Expected output is seen	-	-

Table 7.3: Negative test case for customer name insertion

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

Table 7.4: Positive test case for customer name insertion

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

7.3.2 Integration Testing

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

Integration testing is a software testing technique used to validate the interactions and interfaces between different modules, components, or subsystems of an application. The primary goal of integration testing is to ensure that individual modules, when combined, work together as expected and produce the desired outcome.

Table 7.5: Test case on basis of generation of bill

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

Table 7.6: Test case on basis of deposit details

Function Name	Input	Expected output	Error	Resolved
Negative searching of Deposit details	12334(meter_no) January(month)	Details not seen	Output not seen	Consume()
Positive searching of total_bill	12334(meter_no) January(month)	Must display deposited details	-	-

7.3.3 System testing

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

System testing is a phase in the software testing process that focuses on evaluating the complete, integrated system to ensure that it meets specified requirements. It involves testing the entire system as a whole, including its functionalities, performance, reliability, security, and other attributes, to validate whether it meets the intended objectives and functions correctly in its intended environment.

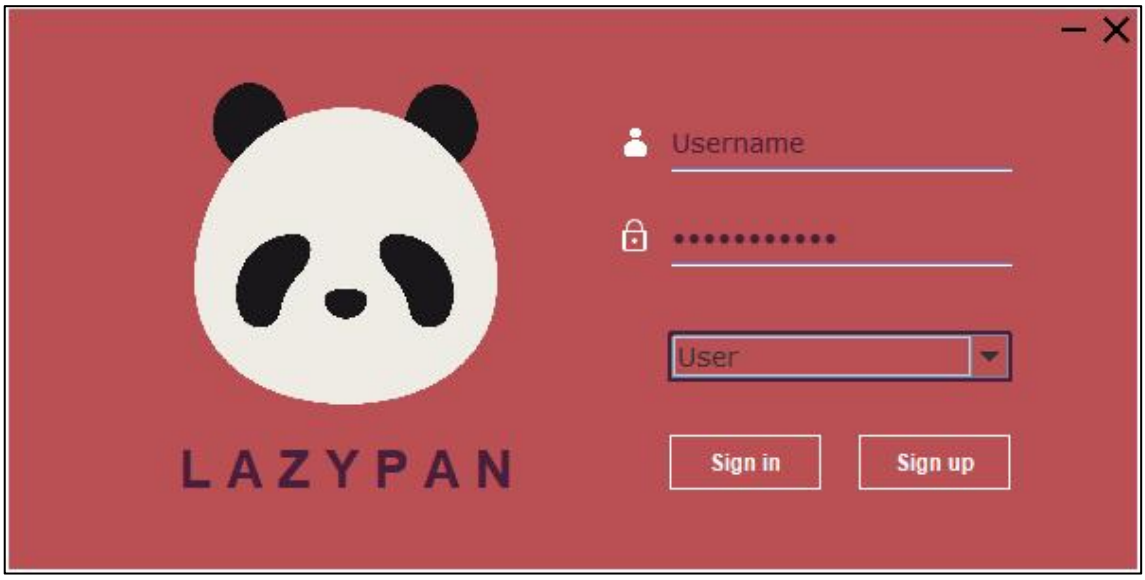
Key aspects and features of system testing:

- It tests all functional and non-functional aspects of the system.
- Evaluates system performance, such as responsiveness, scalability, and resource usage under varying conditions.
- Assesses the system's resilience against unauthorized access, vulnerabilities, and threats.
- Ensures that the system is user-friendly and meets user expectations.
- Verifies the system's stability and reliability under different conditions, including stress and load.

Table 7.7: Test cases for the project

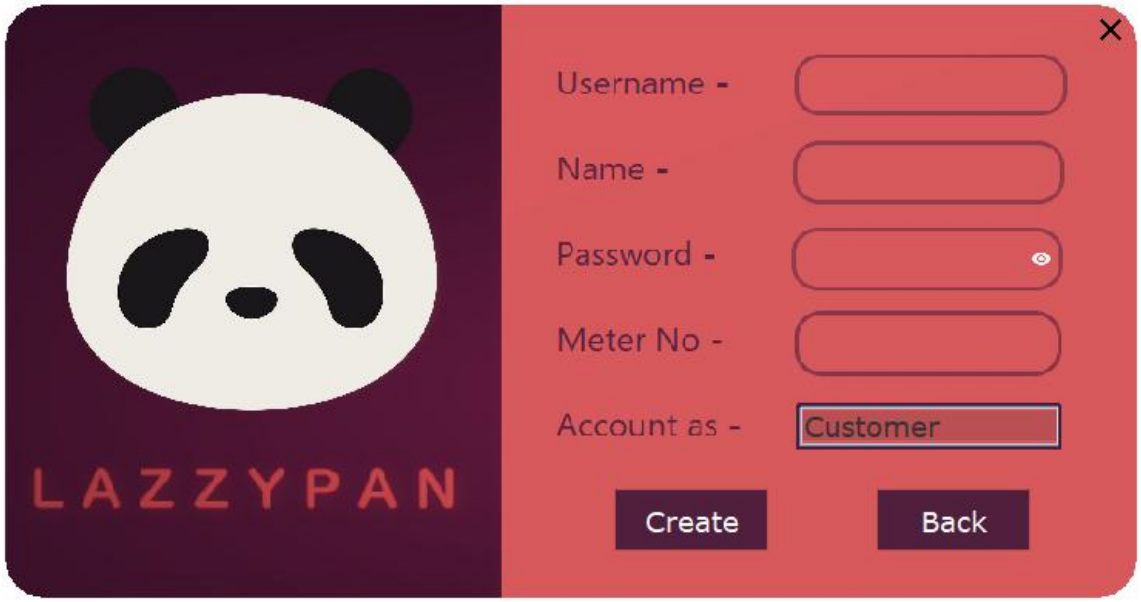
Steps	Action	Expected output
Step 1 Choice	The screen appears when the users run the program. 1. If admin login 2. If customer login	A page with different menu's appears. 1.Admin panel opens and 2.Customer panel opens
Step 2	The screen appears when the admin logs in and selects any one of the menus from 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 8 - SNAPSHOTS



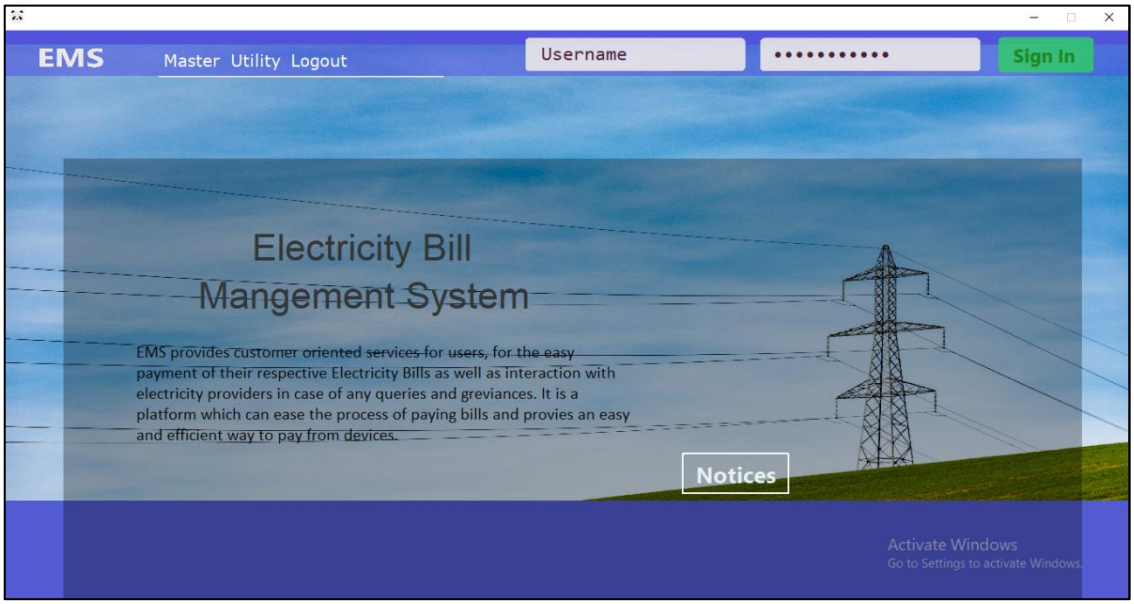
Login Screen

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



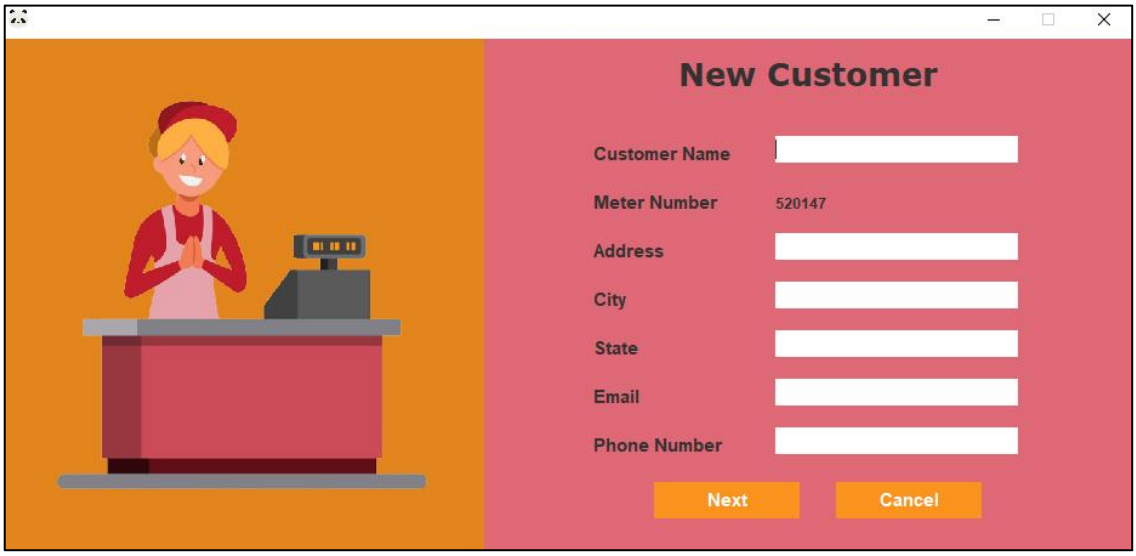
Sign Up Screen

Here New customers will signup to access their accounts. User have to enter username, name, password. Every user must enter their unique Meter Number to complete their signup process.



Admins Home Screen

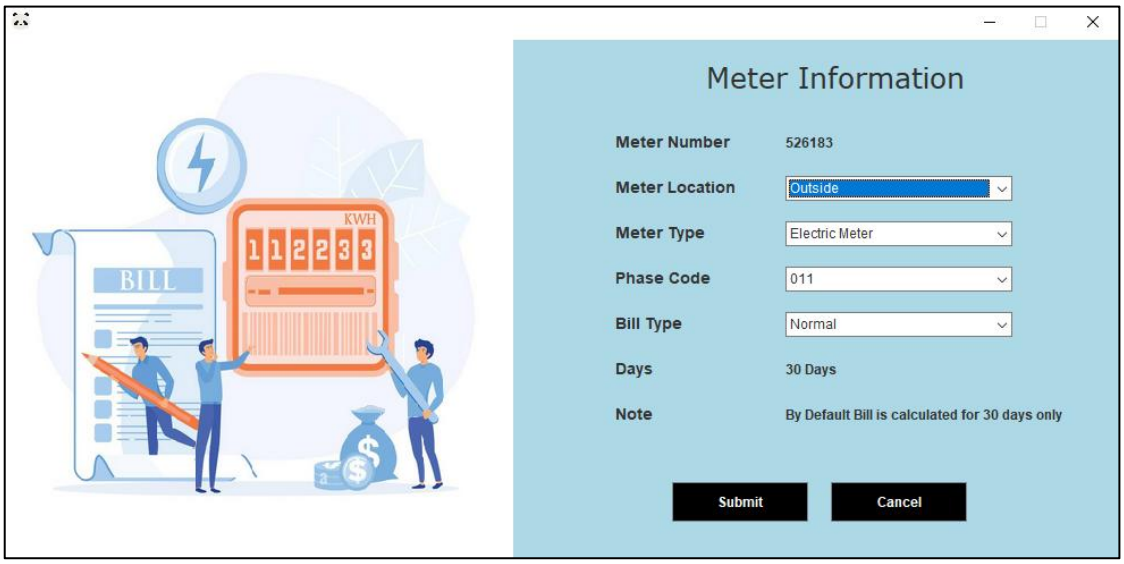
Admin lands on this page after successful login.



New Customer Screen

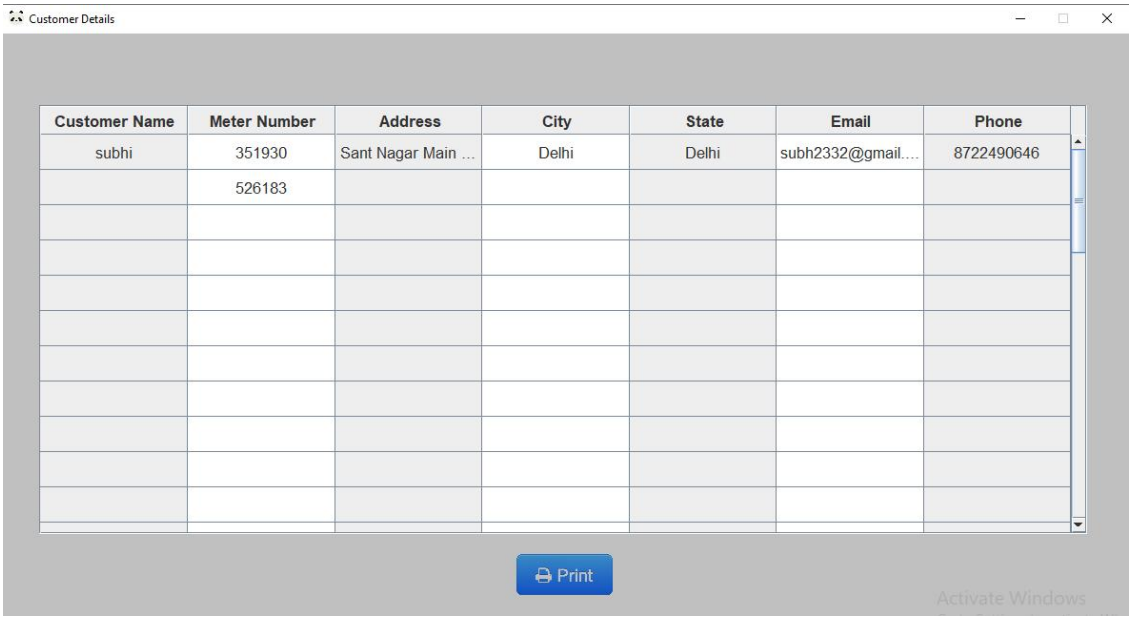
Here admin registers new users.

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



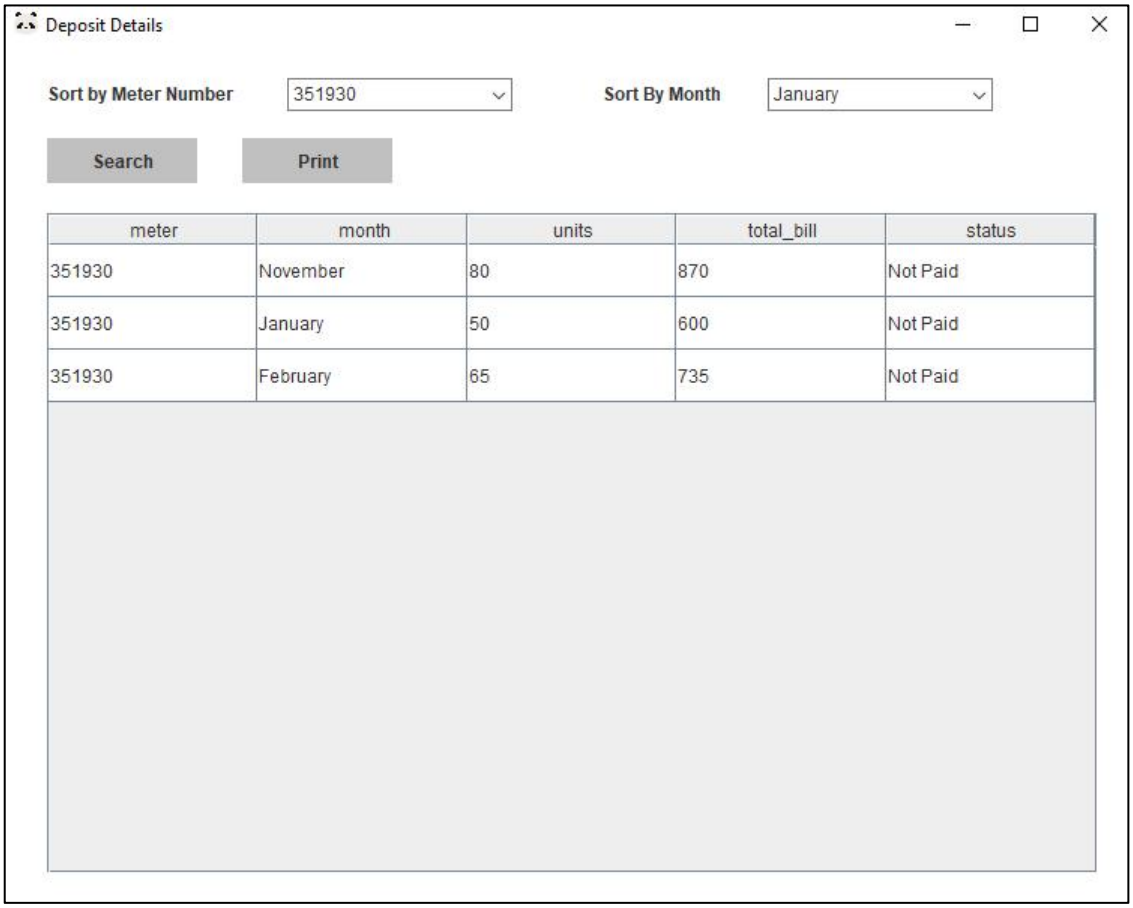
Meter Info Screen

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.



Customer Details Screen

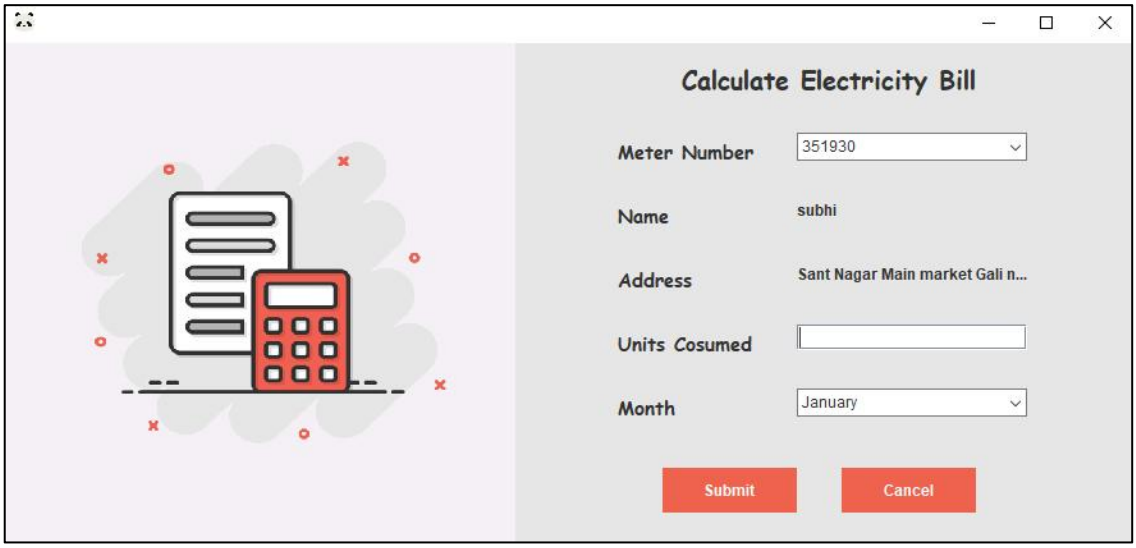
Here Admins can see the details of all registered customers. Admin can print these details in pdf format if the wish.



Deposit Details Screen

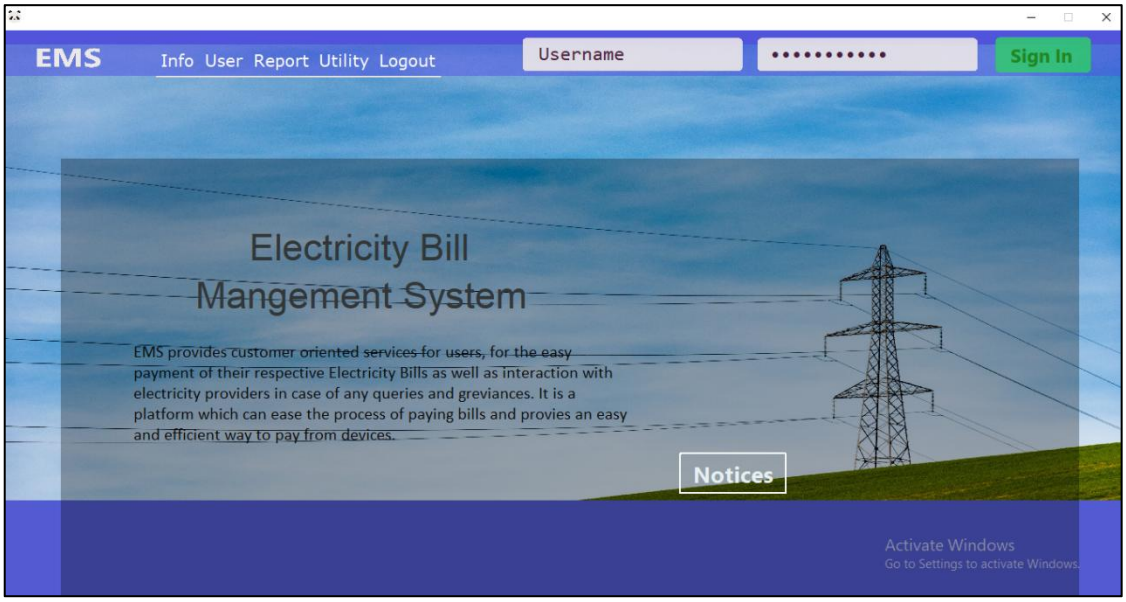
Here Admin can check the status whether customers have paid their bills or not. His list can be sorted according to individual user's meter number or according to month.

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



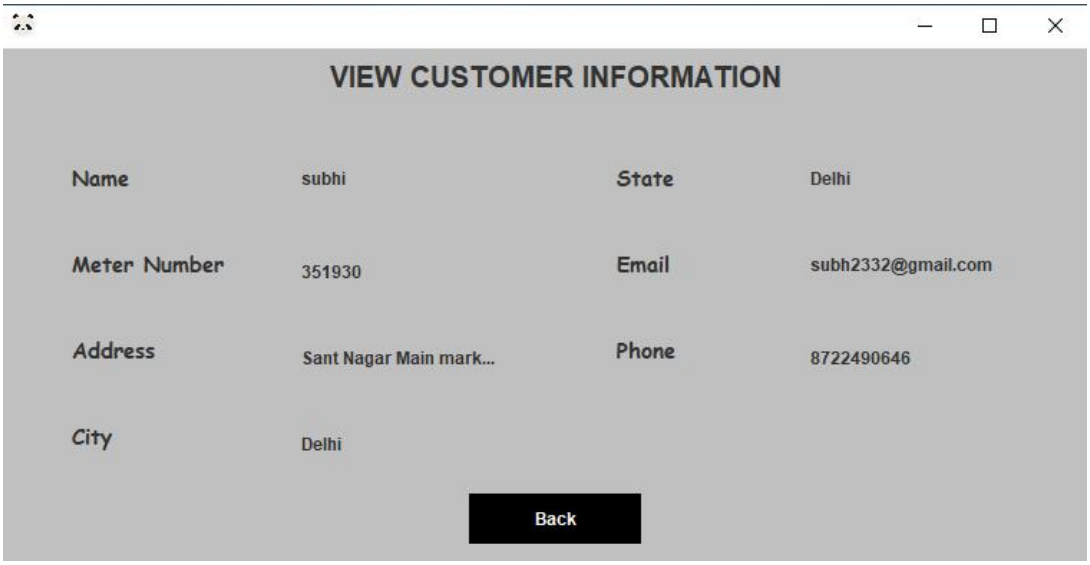
Calculate Bill Screen

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



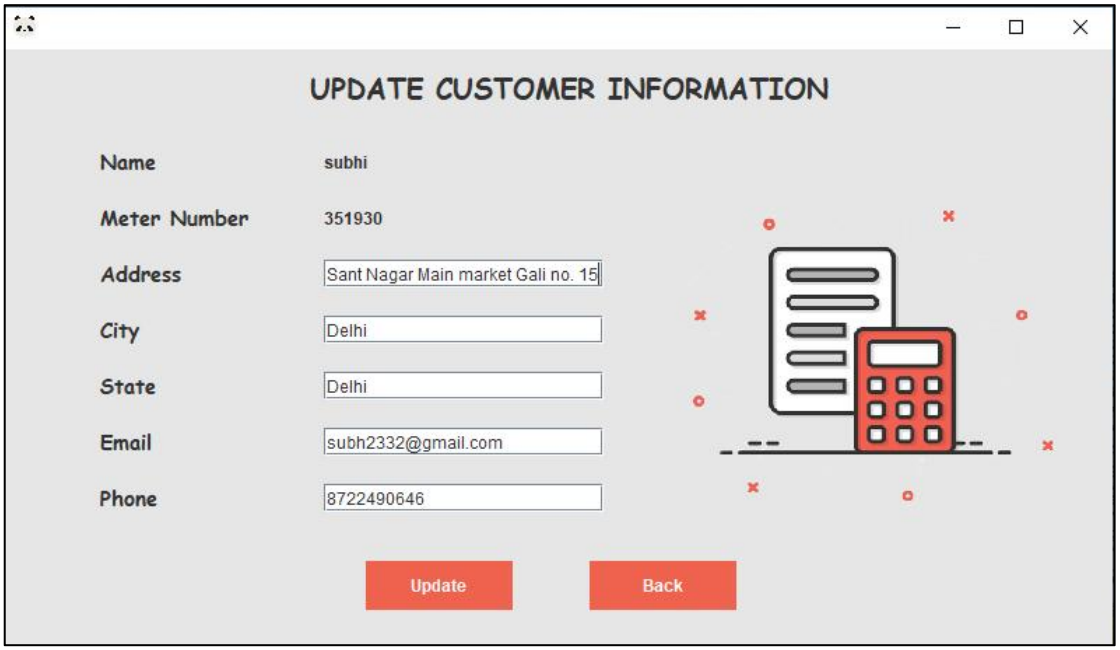
Customer's Home Screen

Customer lands on this page after successful login.



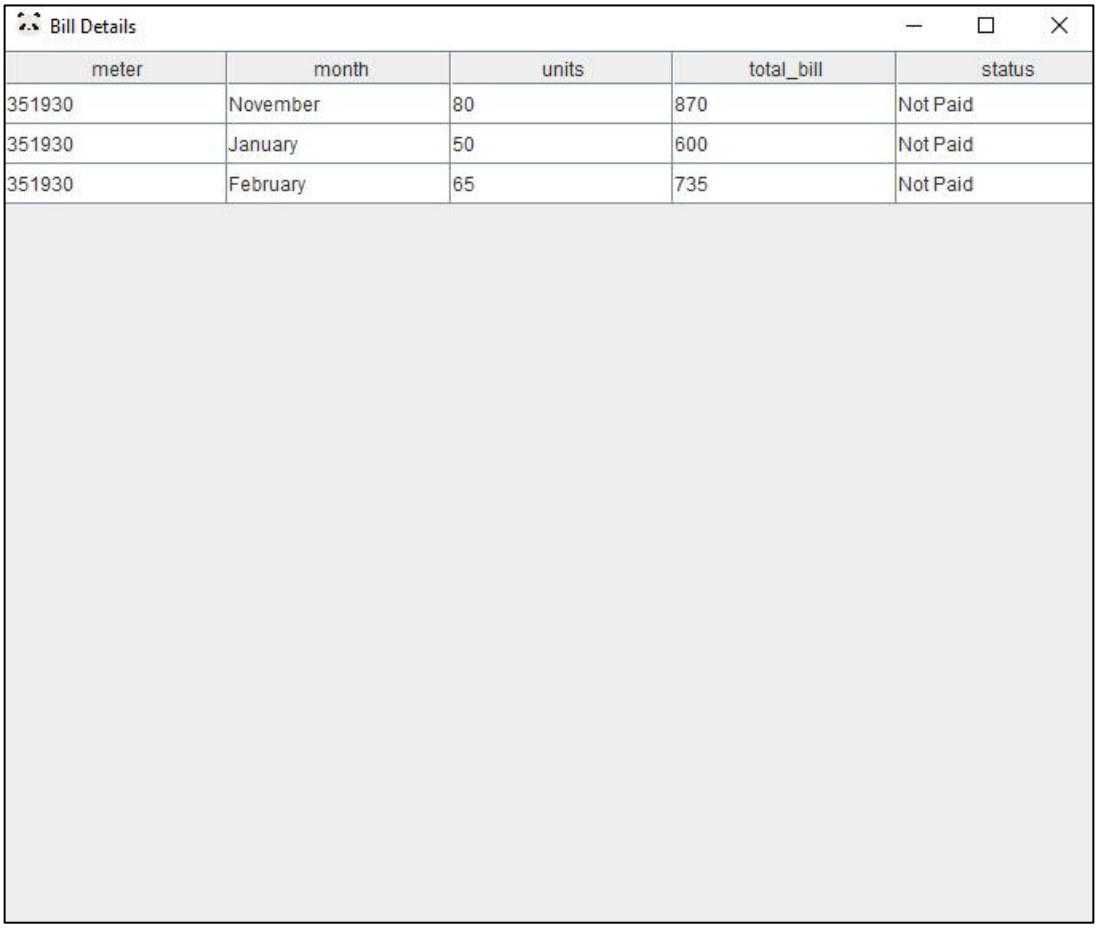
View Customer Info Screen

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



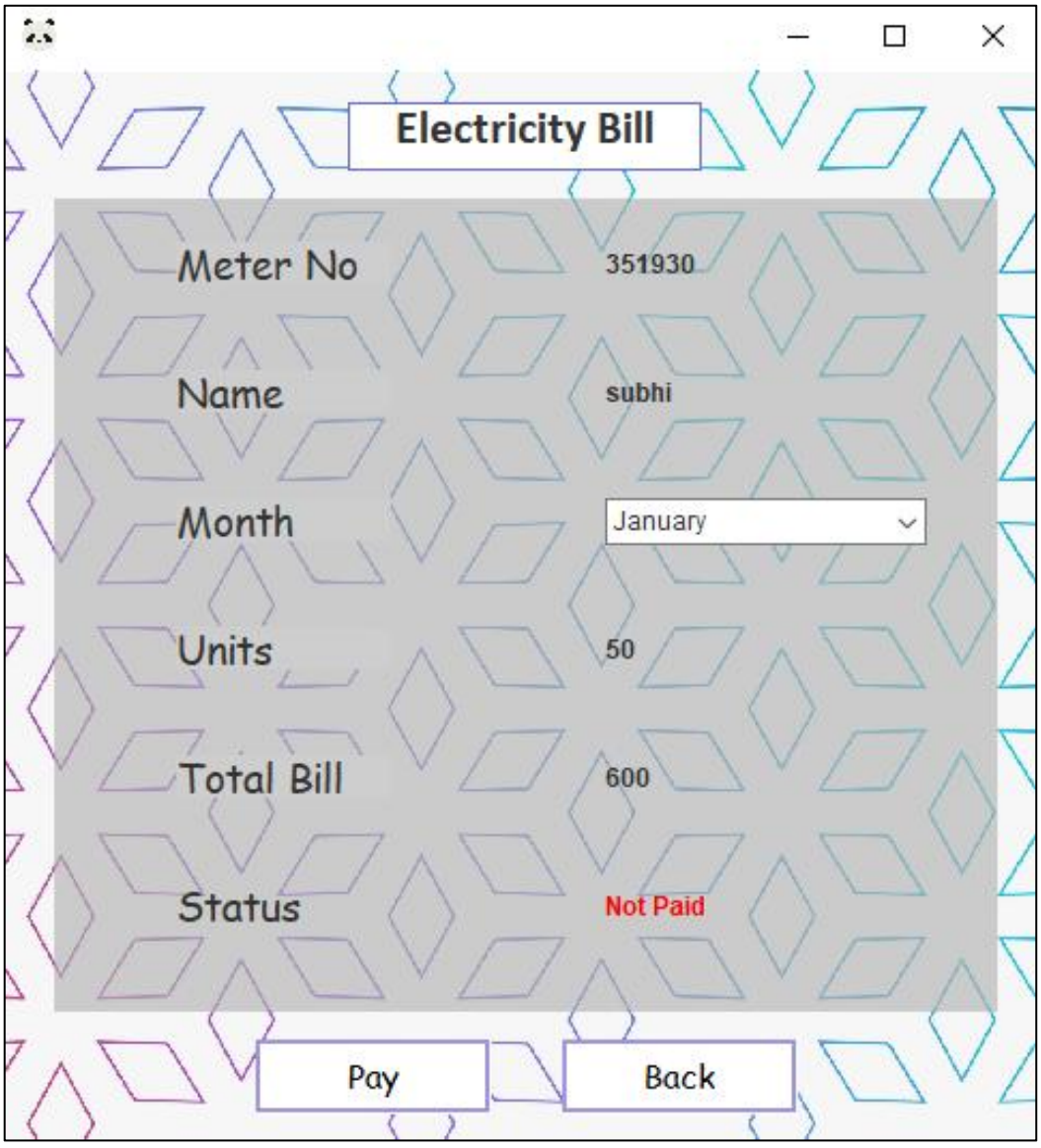
Update Customer Info Screen

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



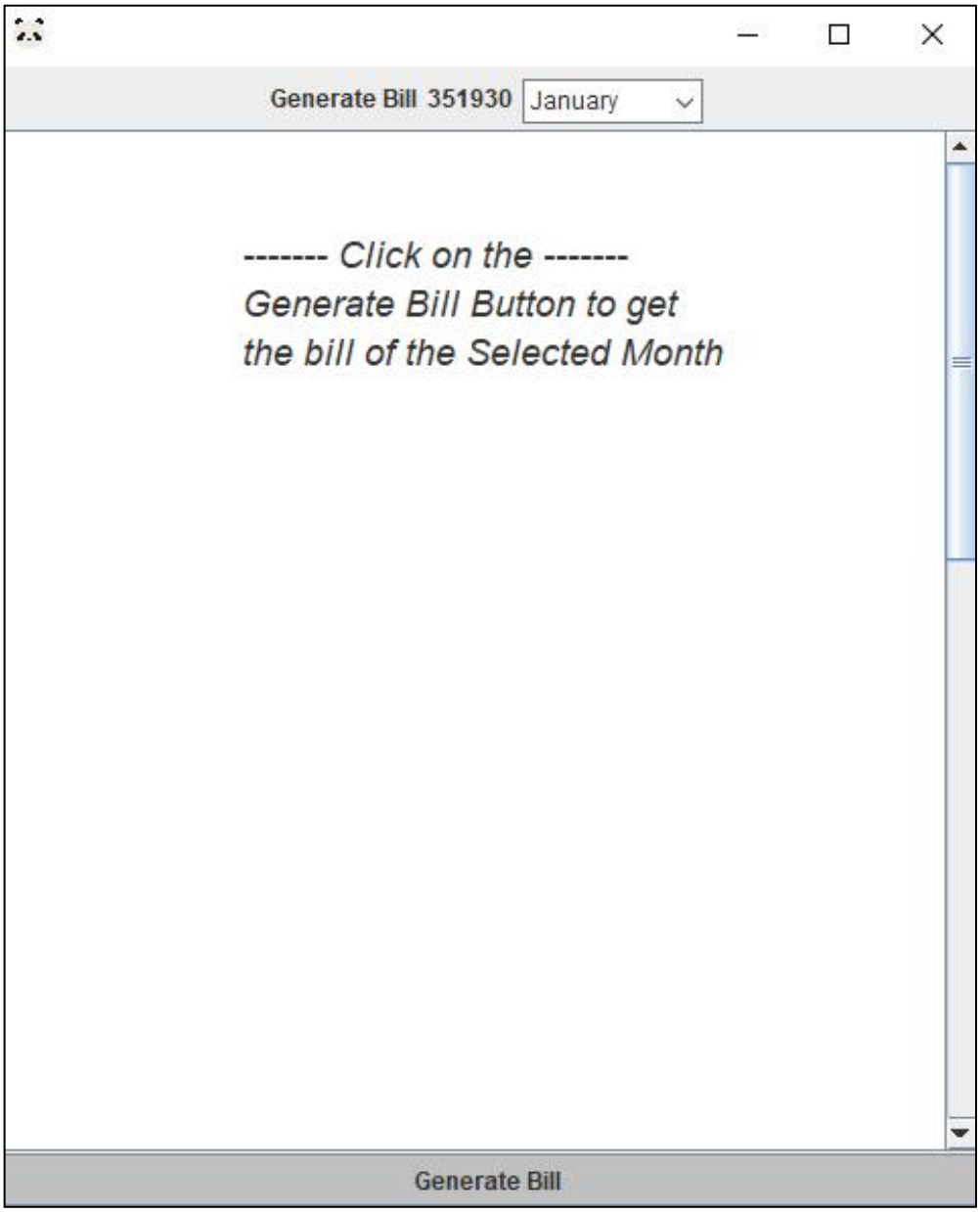
Bill Details Screen for Customers

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



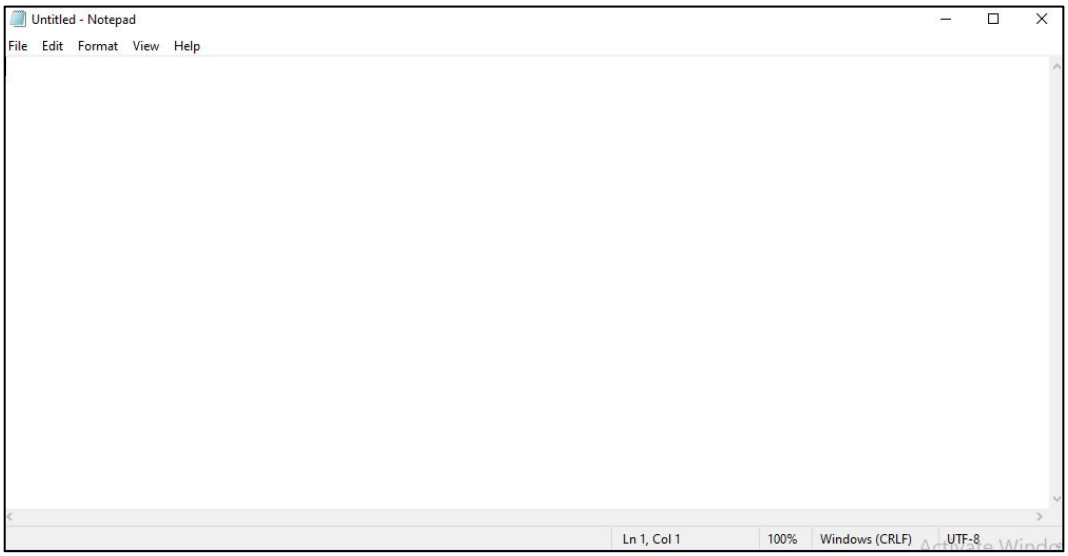
Pay Bill Screen

Here customers pay their bills by selecting appropriate month.



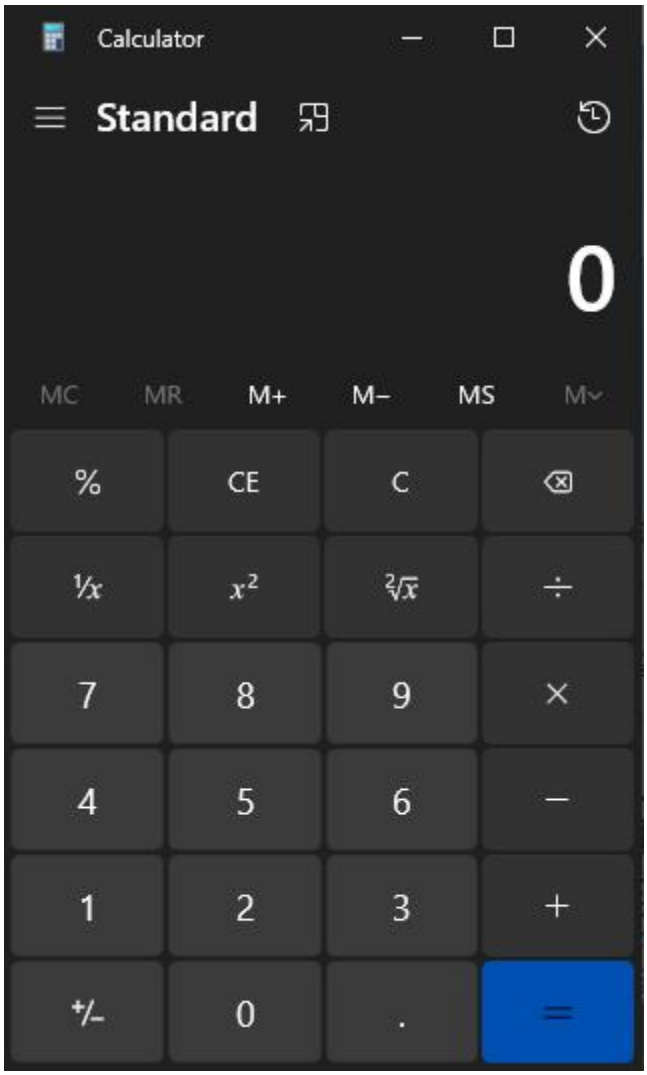
Generate/ Show Bill Screen

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



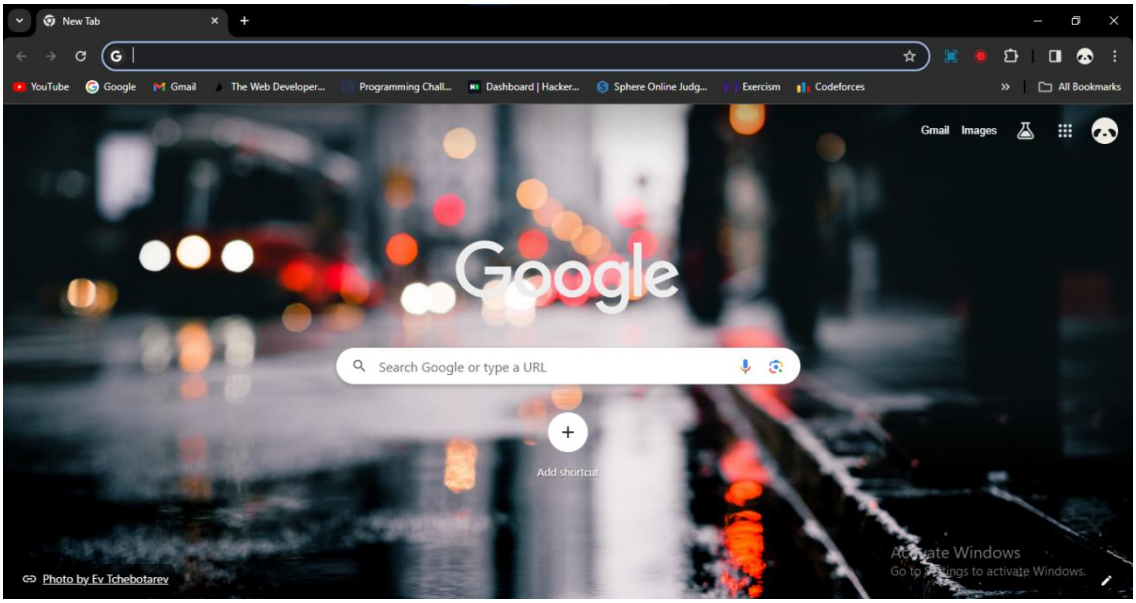
Notepad Screen

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.



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.

CHAPTER 9 - FUTURE SCOPE AND LIMITATIONS

9.1 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.

✧ **Re-usability**

Re-usability 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.

9.2 LIMITATIONS

This application cannot be accessed remotely.

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

CHAPTER 10 - 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.

Usability testing was part of the post implementation review and performance evaluation for the Electricity Online Bill Payment System, in order to ensure that the intended users of the newly developed system can carry out the intended task effectively using real data so as to ascertain the acceptance of the system and operational efficiency. It caters for consumers' bills and also enables the administrator to generate monthly reports. It is possible for the administrator to know the consumers have made payment in respect of their bills for the current month, thereby improving the billing accuracy, reduce the consumption and workload on the Electricity Board employees or designated staff., increase the velocity of electricity distribution, connection, tariff scheduling and eliminates variation in bills based on market demand. The conceptual framework allows necessary adjustments and enhancement maintenance to integrate future demands according to the technological or environmental changes with time. It manages the consumers' data and validates their input with immediate notification centralized in Electricity Board offices across the nation.

CHAPTER 11 - CONCLUSION

REFERENCES

- Arimoro, T. A., Oyetunji, A. K., &Odugboye, O. E. (2019). Analysis of Electricity Billing System in Corporate Buildings in Lagos, Nigeria. Studies, 1(6), 10-20.
- Panthala, S., Islam, N., & Habib, S. A. (2015). Automated industrial load measurement system.'
- Adegboyega, A., Gabriel, A. A., Ademola, A. J., Victor, A. I., &Nigeri, K. (2013). Design and Implementation of an Enhanced Power Billing System for Electricity Consumers in Nigeria. African Journal of Computing & ICT, 6(1).
- Database Management Systems 3rd Edition by Raghu Ramakrishnan.