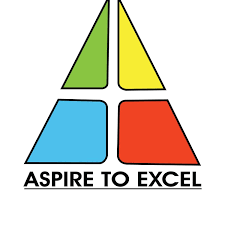
**Electricity Billing System**

**A project submitted to**

****

**BHILAI INSTITUTE OF TECHNOLOGY, DURG**

**An Autonomous Institute**

**Affiliated to**

**Chhattisgarh Swami Vivekanand Technical University, Bhilai (CG)**

**in the partial fulfillment of the requirements for the award** **of**

**MASTER OF COMPUTER APPLICATIONS**

**(MCA)**

***By***

**Esha Sharma, Khivanshu Choudhary**

**Roll No.: 500102122022,500102122046**

**Enrollment No.: CC2735,CC2756**

**Under the Guidance of**

**Mr. B. Varghese**

**DEPARTMENT OF COMPUTER APPLICATIONS,**

Session: 2022-2024

**CERTIFICATE OF SUPERVISOR(S) /GUIDE**

This is to certify that the work incorporated in the project **Electricity Billing System** is a project work assigned by our Institute, successfully carried out by **Esha Sharma, Khivanshu Choudhary** bearing Roll No: **500102122022,500102122046** ,Enrollment No: **CC2735, CC2756**. Under my guidance and supervision for the award of Degree of Master of Computer Applications (MCA) from **Bhilai Institute of Technology, Durg, C.G**., affiliated to Chhattisgarh Swami Vivekanand Technical University (CSVTU), Bhilai, C.G., India. To the best of my knowledge and belief the report embodies the work of the candidate him/herself and has duly been successfully completed.

##### Signature of the Supervisor/Guide

Name:

Designation:

Date:

**DELCLARATION BY THE CANDIDATE**

I, **Esha Sharma** ,Student of 3rd Semester MCA, **Bhilai Institute of Technology**, Durg, C.G., India, bearing Enrolment Number  **CC2735** here by declare that the project entitled **Electricity Billing System** has been carried out by me under the Guidance / Supervision of External Guide

, submitted in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications (MCA) by the Chhattisgarh Swami Vivekanand Technical University during the academic year 2022 - 2024. This report has not been submitted to any other Organization/ University for any award of Degree/ Diploma.

(Signature of Candidate)

Date:

Place:

**CERTIFICATE OF FORWARDING**

This is to Certify that **Esha Sharma, Khivanshu Choudhary**, a bonafide Student of Master of Computer Applications (M.C.A) at **Bhilai Institute of Technology**, Durg, C.G., India, has carried out his project work as mentioned in this project entitled **“Electricity Billing System”** at “**Bhilai Institute of Technology**”, during his third semester of studies in M.C.A as a part of a curriculum for obtaining the degree of M.C.A from Chhattisgarh Swami Vivekanand Technical University (CSVTU), Bhilai, C.G., India to which the institute is affiliated.

This Certificate Issued by the undersigned does not cover any responsibility regarding the statements made and work carried out by the concerned student.

The current dissertation is hereby being forwarded for evaluation for the purpose for which it has been submitted

**Signature of Project Coordinator**  **Signature of Head of Department**

**Name:** Mr. B. Varghese **Name:** Dr. Jyothi Pillai

**Designation:** Associate Professor **Designation:** Head Of Department

**Institution:** Bhilai Institute of Technology **Institution:** Bhilai Institute of Technology

**Date: Date:**

**CERTIFICATE OF APPROVAL**

This is to Certify that the project the entitled “**Electricity Billing System**”, carried out by **“Esha Sharma, Khivanshu Choudhary”** a student of 3rd semester, M.C.A. at **Bhilai Institute of Technology, Durg, C.G., India**, is here by approved after proper examination and evaluation as a creditable work for the partial fulfillment of the requirement for awarding the degree of Master of Computer Applications (M.C.A) from Chhattisgarh Swami Vivekanand Technical University (CSVTU), Bhilai C.G. India

(Internal Examiner) (External Examiner)

Name: Name:

Designation: Designation:

College Name: College Name:

Date: Date:

**ACKNOWLEGEMENT**

I have great pleasure in the submission of this project report entitled “**Electricity Billing System**” for **Bhilai Institute of Technology** in partial fulfillment the degree of the Master of Computer Applications. While submitting this Project report, I take this opportunity to thank those directly or indirectly related to project work.

I would like to thank my external guide in Organization who has provided the opportunity and organizing project for me. Without his active co-operation and guidance, it would have become very difficult to complete task in time.

I would like to express sincere thanks to **Dr. Arun Arora**, Principal, **Dr. Jyothi Pillai,** Head of Department, (Computer Applications).

While Submission of the project, I also like to thanks to **Mr. B. Varghese** Project Coordinator and the all Professors of Bhilai Institute of Technology, Durg, C.G., India, for their continuous help and guidance throughout the course of MCA.

Acknowledgement is due to our parents, family members, friends and all those persons who have helped us directly or indirectly in the successful completion of the project work.

Esha Sharma,

Khivanshu Choudhary

**Table of Content**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Heading | | | Page No. |
| 1 | Abstract and Type of Project (Software). | | | 1 |
| 2 | List of Tables. | | | 1 |
| 3 | List of Figures. | | | 2 |
| 4 | Project Development Process Model Used (Methodology). | | | 2 |
| 5 | Project’s Process Documentation. | | | 3 |
| 5.1. | **INCEPTION PHASE.** | | | 3 |
| 5.1.1 | Initial Description of Problem | | 4 |
| 5.1.2. | Software Requirement Specification-SRS. | | 4 |
|  | 5.1.2.1. | **Non-Functional Requirement**   * Hardware requirement. * Software requirement. * Communication interface | 4 |
|  | 5.1.2.2. | **Functional Requirement**   * Accept New User * Bill Calculate * Customer Details * Deposit Details * Update Customer Information * View Information * Generate Bill * Bill Details | 4 |
|  | 5.1.2.3. | **Performance Requirement**   * Static performance * Dynamic performance | 5 |
| 5.1.3. | **Security features** | | 5 |
| 5.1.4. | **Reliability.** | | 5 |
| 5.1.5. | Cost estimation using DFP. | | 6 |
| 5.2. | **ELABORATION PHASE (DETAIL DESIGN)** | | | 8 |
|  | 5.2.1 | **Object Model.** | | 9 |
|  | 5.2.1.1 | Object Classes. | 9 |
|  | 5.2.1.2 | Data dictionary containing description of class attributes (data members, and methods) . | 9 |
|  | 5.2.1.3 | Association between classes. | 11 |
|  | 5.2.1.4 | Simplifying objects classes using Inheritance. | 11 |
|  | 5.2.1.5 | Group classes into module. | 11 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 5.2.1.6 | Object Diagram. | 12 |
| 5.2.2. | **Dynamic Model** | | 13 |
|  | 5.2.2.1 | Scenario (Project Interaction Sequence) | 13 |
| 5.2.2.2 | Event Flow Diagram (EFD) | 15 |
| 5.2.3 | **Functional Model** | | 16 |
| 5.2.3.1 | Identification of Input / Output values. | 16 |
| 5.2.3.2 | DFD as needed to show functional dependencies. | 18 |
|  | 5.2.4. | **Database design** | | 19 |
|  |  | 5.2.4.1 | Schema definitions with Candidate key/Primary key | 20 |
|  |  | 5.2.4.2 | Normalization | 21 |
|  |  | 5.2.4.3 | E-R Diagram | 21 |
| 6. | **CONSTRUCTION PHASE** | | | 22 |
|  | 6.1. | CASE Tools used to design | | 23 |
|  | 6.2 | Coding Language and Operating System (OS) used (Including explanation). | | 23 |
|  | 6.3. | Database connectivity procedure | | 23 |
|  | 6.4. | Code description. | | 24 |
|  | 6.5. | I/O Interface. | | 25 |
| 7. | **TRANSITION PHASE** | | | 32 |
|  | 7.1. | Report of Alpha testing | | 33 |
|  | 7.2. | DSLOC and Cost estimation using DSLOC | | 33 |
|  | 7.3. | Cost variation between DFP and DSLOC | | 34 |
| 8. | Limitations & Future Enhancement. | | | 35 |
| 9. | Conclusions. | | | 35 |

1. **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 swings as the base programming language which can be used to develop Standalone Software Application. The Microsoft Structured Query Language (SQL) server is also used for creating back-end 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.

The complete schedule of project was decided Six months and total DSLOC calculated 3.507 KLOC with the help of 2 members. The project type is organic and details are presented through this Document.

**2. List of Tables**

|  |  |  |
| --- | --- | --- |
| **No.** | **Heading** | **Page No.** |
| 1 | Functional point complexity weight | 7 |
| 2 | Complexity factor | 7 |
| 3 | Coefficient of Basic COCOMO | 8 |
| 4 | Object Class | 10 |
| 5 | Data Dictionary | 10 |
| 6 | Admin Group Class | 13 |
| 7 | Super Admin Group Class | 13 |
| 8 | I/O Table | 19 |
| 9 | Accept New User Schema | 22 |
| 10 | Notice Schema | 22 |
| 11 | Hall Schema | 23 |
| 12 | Events Schema | 23 |
| 13 | Manage Head Schema | 23 |
| 14 | Remove User Schema | 24 |
| 15 | Admin Schema | 24 |
| 16 | ActivityLog Schema | 24 |
| 17 | Code Description | 28 |
| 18 | Effort Adjustment factor | 37 |
| 19 | Coefficient of Intermediate COCOMO model | 37 |

**3. List of Figures**

|  |  |  |
| --- | --- | --- |
| **No.** | **Heading** | **Page No.** |
| 1 | The phase of life cycle process | 2 |
| 2 | Association between classes | 12 |
| 3 | Object diagram | 13 |
| 4 | User Scenario | 14 |
| 5 | Admin Scenario | 15 |
| 6 | Super Admin Scenario | 16 |
| 7 | Event flow diagram | 17 |
| 8 | DFD | 21 |
| 9 | E-R Diagram | 25 |

1. **Project Development Process Model Used (Methodology).**

In this Project the modern software development has been used where it is divided into two stages

Engineering stage and Production stage. Engineering stage is further divided into Inception phase and Elaboration phase. Production stage is divided into Construction phase and Transition phase

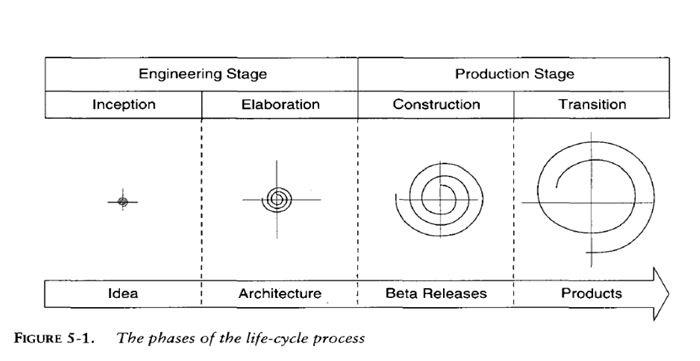
All phase of development based on Iteration. The outcome of the Inception phase is SRS, The outcome of Elaboration phase is detailed architecture, The outcome of the Construction phase is Beta release, The outcome of the Transition phase is final product.

Figure 1: *The Phase of life cycle process*

**INCEPTION PHASE**

**5.1.1. Initial Description of Problem**

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.

**5.1.2. Software Requirement Specification - SRS.**

**5.1.2.1. Non-Functional Requirements**

* **Hardware Requirement**

Processor Intel Pentium or higher

RAM: Minimum of 1GB RAM

Storage: Minimum of 500MB

Monitor LCD Monitor

Keyboard Standard Keyboard

Mouse Compatible Mouse.

* **Software Requirement**

Operating System :- Window 7, 8, 10 or Above.

Front End :- Java Core/ Swing

Back End :- My SQL

**5.1.2.1. Functional Requirements**

**Accept New User :-**

This subsystem provides the feature to allow activation of new registered user. The users can register and then after admin approval they can log in to the system.

**Bill Calculate :-**

The customer’s unit consumed is necessary to calculate the electricity bill by the admin.

**Customer Details :-**

The admin can see the all customer’s details.

**Deposit Details :-**

The Admin can able to search the deposit details of the user according to the meter number and month.

**Update Customer Information :-**

Customer can update their own information. System stores customer details including Address, City, State, Phone No. , Email.

**View Information :-**

The system enables the Customer to view their own information. The module includes viewing a

Name, Meter No, City, Address, State, Email, Phone.

**Generate Bill :-**

The Customer enable to generate electricity bill according to month.

**Bill Details :-**

In Bill details, it show the customer bill is paid or unpaid.

**5.1.2.2. Performance Requirements**

* **Static Performance:-**

There can be 1crore numbers of user who can access the web application

* **Dynamic Performance:-**

This Application is made with PHP programming language and all the exceptions are handled by throw catch exception.

**5.1.3. Security Features**

**MySQL Features:-**

* **User Authentication:-**

MySQL uses a username and password-based authentication system, which allows database administrators to control access to the database.

* **Backup and Recovery:-**

MySQL provides backup and recovery mechanisms that help administrators to recover data in the event of a security breach, data loss, or hardware failure.

* **Security Plugins:-**

MySQL has several security plugins, such as the MySQL Enterprise Firewall and MySQL Enterprise Audit, which provide additional security features, such as intrusion detection and prevention and enhanced auditing capabilities.

**5.1.4. Reliability**

* User Interface : Reliability of user interface is High
* Output Interface : Reliability of output interface is High
* Result : Result Reliability is High

**5.1.5. Cost Estimation Using DFP**

The cost estimation is calculated by delivered function point and uses following complexity factor and weight:

Table No 1: Functional Point Complexity Weight

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Type** | **Simple** | **Average** | **Complex** |
| External Input | 3 | 4 | 6 |
| External Output | 4 | 5 | 7 |
| External Inquiries | 3 | 4 | 6 |
| Internal Logical Files | 7 | 10 | 15 |
| External Interface File | 5 | 7 | 10 |

Table No 2: Complexity Factor

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Factor** | **Influence** | **Value** |
| 1 | Data communications | Moderate | 2 |
| 2 | Distributed processing | Not Present | 0 |
| 3 | Performance objectives | Moderate | 2 |
| 4 | Operation Configuration load | Moderate | 2 |
| 5 | Transaction rate | Not Present | 0 |
| 6 | On-line data entry | Not Present | 0 |
| 7 | End user efficiency | Moderate | 2 |
| 8 | On-line update | Not Present | 0 |
| 9 | Complex processing logic | Moderate | 2 |
| 10 | Re-Usability | Moderate | 2 |
| 11 | Installation case | Moderate | 2 |
| 12 | Operational case | Insignificant | 1 |
| 13 | Desire to facilitate change | Insignificant | 1 |
| 14 | Multiple sites | Not Present | 0 |
|  |  | **Total (N) =** | **16** |

External Inputs: 4

External Output: 4

External Inquiries: 7

Internal logical Files: 8

External Interface Files: 0

The functions use **simple** type factor

DFP = Count-total \* [0.65 + 0.01 \* ∑(fi)] = UFP \* CAF

UFP= 4\*3 + 4\*4 + 8\*7 + 0\*5 + 7\*3 =105

**Unadjusted Function Point = 105**

CAF= 0.65 + 0.01\*N N= Complexity factor=16

= 0.65 + 0.01\*16 = 0.81

**Complexity Adjustment Factor (CAF) = 0.81**

DFP= UFP \* CAF = 105 \* 0.81 = 85.05

**Delivered Function point = 85.05**

To convert function points into expected Source line of code we require language factor.

The Language factor of **Java is 53 per function point**

**Expected SLOC** = Language Factor \* Total Function points

= 53 \* 85.05 = 3507 LOC = **3.507 KLOC Expected**

Table no:3 Coefficient of Intermediate COCOMO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software Product** | **a** | **b** | **c** | **d** |
| Organic | 3.2 | 1.05 | 2.5 | 0.38 |
| Semi-Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 2.8 | 1.20 | 2.5 | 0.32 |

By basic COCOMO model for organic project

**Expected Effort** = a(KLOC)^b = 3.2(3.507)^1.05 = **11.9 PM**

**Expected Duration =** c(Effort)^d = 2.5(11.9)^0.38 = **6.4 Months**

**Expected Person =** Effort / Duration **=** 11.9 / 6.4 = **1.8 ~ 2 Person**

**5.2. ELABORATION PHASE**

**5.2.1 Object Model**

**5.2.1.1 Object Classes**

As per the requirement described in SRS we have following class object. Name and description purpose as given as following table no:2

Table No 4: Object Classes

|  |  |  |
| --- | --- | --- |
| **SNo.** | **Name of Object/Class** | **Description** |
| 1 | Login | To Accept Member |
| 2 | SignUp | To Register New Member |
| 3 | Admin | To Login as Admin |
| 4 | Add New Customer | Admin Register a new Customer |
| 5 | Electricity Bill | To add total unit consumed by customer |
| 6 | Meter Information | To Select type of meter |
| 7 | Customer | To Accept Customer |
| 8 | Update Information | To Update Customer information |
| 9 | Generate Bill | To Generate Electricity Bill |
| 10 | Pay Bill | For pay the electricity Bill |

**5.2.1.2 Data dictionary containing description of class attributes (data members, and methods) .**

As per the defined object given in the table one the dictionary of each object is shown in following table:

Table No 5: Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SNo** | **Name of Class** | **Data Member** | **Valid Range** | **Methods** | **Method Description** |
| 1 | Login | varchar username | Varchar(30): username | login() | This method is used to accept a user with the details |
| varchar password | Varchar(20): password |
| 2 | Signup | varchar username | Varchar(30): username | signup() | This method is used to register a new user |
| varchar password | Varchar(20) password |
| Varchar meter\_no | Varchar(20) meter\_no |
| Varchar name | Varchar(30) name |
| Varchar user | Varchar(20) user |
| 3 | NewCustomer | Varchar name | Varchar(20) name | newcustomer() | This method is used to add a new customer |
| Varchar meter\_no | Varchar(20) meter\_no |
| Varchar address | Varchar(20) address |
| Varchar city | Varchar(30) city |
| Varchar state | Varchar(30) state |
| Varchar phone | Varchar(20) phone |
| Varchar email | Varchar(40) email |
| 4 | MeterInfo | Varchar Meter\_no | Varchar(20) Meter\_no | meterinfo() | This method is used to retrive the meter information |
| Varchar Meter\_location | Varchar(20) Meter\_location |
| Varchar Meter\_type | Varchar(20) Meter\_type |
| Varchar phase\_code | Varchar(20) phase\_code |
| Varchar bill\_type | Varchar(20) bill\_type |
| Varchar days | Varchar(20) days |
| 5 | BillDetails | Varchar Meter\_no | Varchar(20) Meter\_no | billdetails() | This method is used to show the status of the bill |
| Varchar month | Varchar(20) month |
| Varchar units | Varchar(20) units |
| Varchar status | Varchar(20) status |
| Varchar totalbill | Varchar(20) totalbill |
| 6 | CalculateBill | Varchar cost\_per\_unit | Varchar(20) cost\_per\_unit | calculatebill() | This method is used to calculate the bill |
| Varchar meter\_rent | Varchar(20) meter\_rent |
| Varchar service\_charge | Varchar(20) service\_charge |
| Varchar swacch\_bharat\_cess | Varchar(20) swacch\_bharat\_cess |
| Varchar fixed\_tax | Varchar(20) fixed\_tax |
| Varchar service\_tax | Varchar(20) service\_tax |
| Varchar name | Varchar(30) name |
| String password | password: char(30) |
| date | date |

**5.2.1.3 Association between classes.**

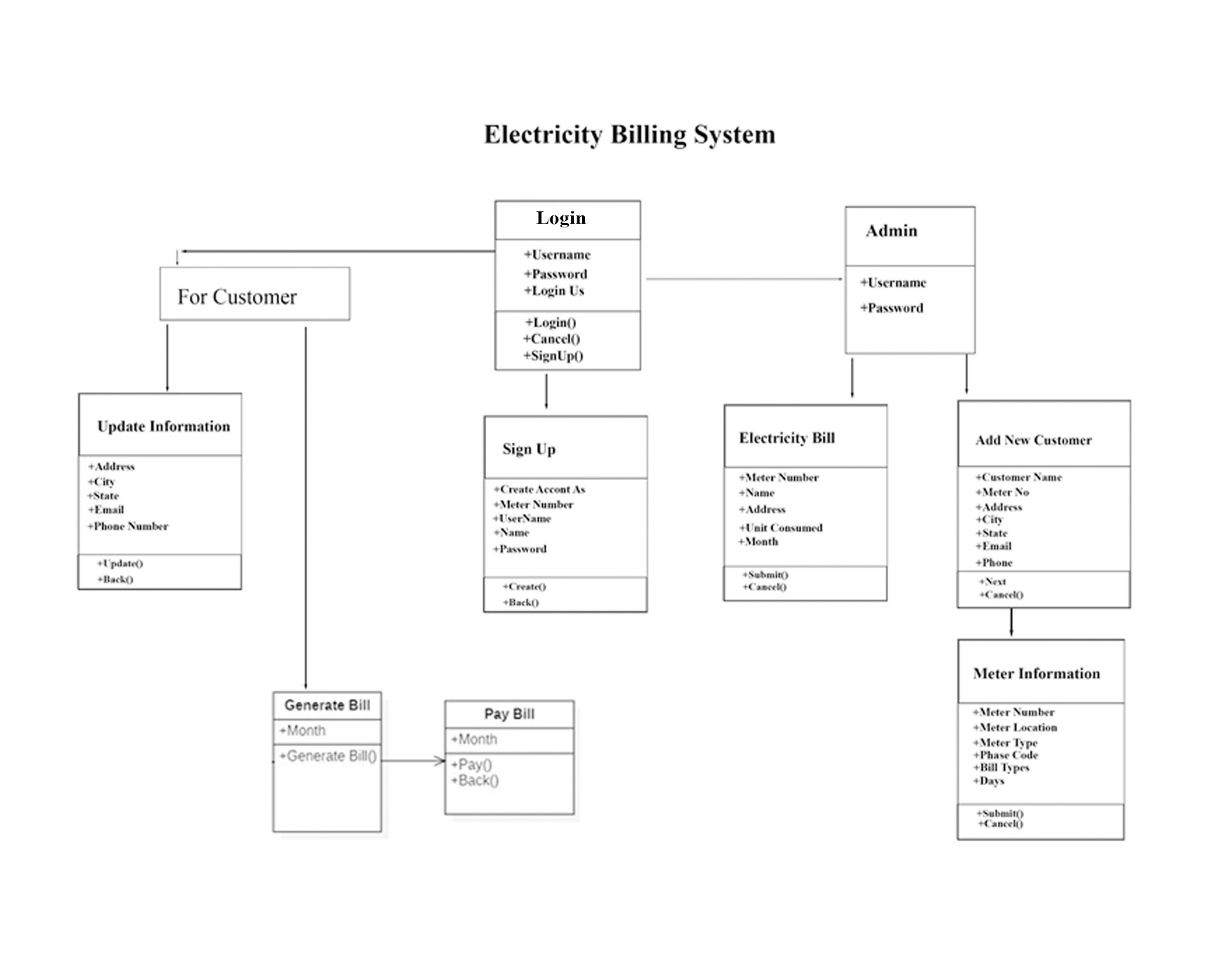
As per the requirement and the data dictionary given in the table no: the class diagram shown as follows. In figure 2

Figure 2: *Association between classes*

**5.2.1.4** **Simplifying objects classes using Inheritance**

No inheritance is used as per simplicity of project.

**5.2.1.5 Group classes into module**

The simplicity of project is single module project has been defined. In which class and object defined as given in the table.

Table No 6: Customer class module

**Admin Module**

|  |  |  |
| --- | --- | --- |
| SNO. | **Name of Class** | **Description** |
| 1 | Customer | To Accept Customer |
| 2 | SignUp | To Register New Member |
| 3 | Update Information | To Update Customer information |
| 4 | Generate Bill | To Generate Electricity Bill |
| 5 | Pay Bill | For pay the electricity Bill |

Table No 7:Admin class module

**Admin**

|  |  |  |
| --- | --- | --- |
| SNO. | **Name of Class** | **Description** |
| 1 | Admin | To Accept Admin |
| 2 | SignUp | To Register New Admin |
| 3 | Electricity Bill | To add total unit consumed by customer |
| 4 | Add New Customer | Admin Register a new Customer |
| 5 | Meter Information | To Select type of meter |

**5.2.1.6 Object Diagram**

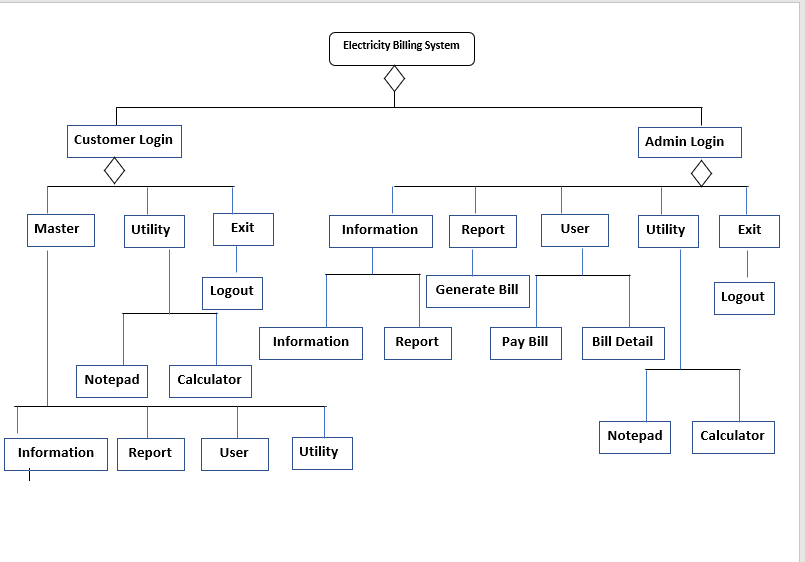
The Object diagram represents a specific instance of a class shown in the figure given below:

Figure 3*: Object Diagram*

**5.2.2 Dynamic Model**

The Dynamic model of system describes the aspect of system those concern with the time and sequence of operation. Even that its change of event, state, that defines context of event and the organization of event and state.

**5.2.2.1 Scenario (Project Interaction Sequence)**

The Scenario describes the specific use of the entire system and the working of the software shown in the figure given below:

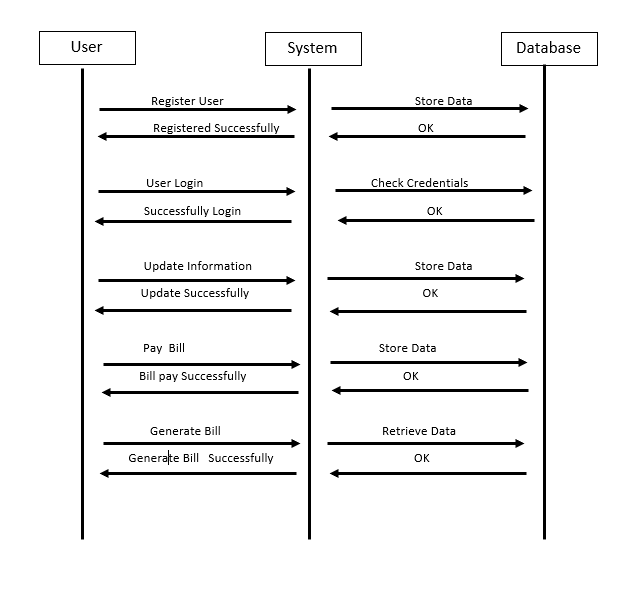
****

Figure 4*: User Scenario*

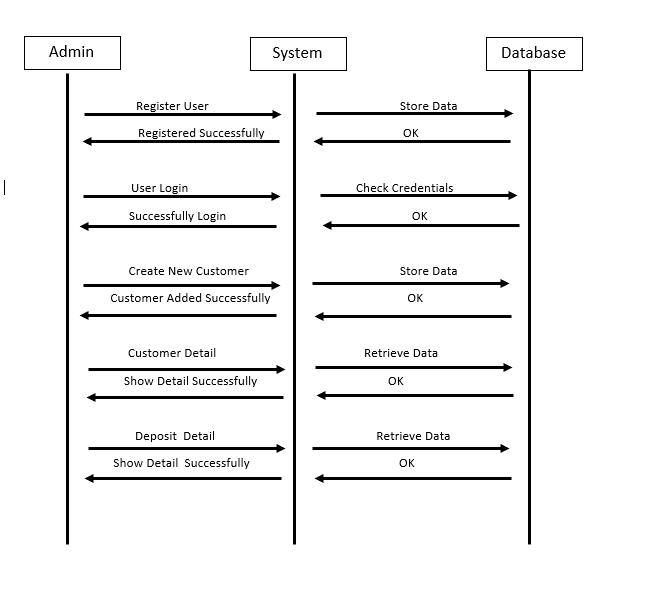
****

Figure 5*: Admin Scenario*

**5.2.2.2 Event Flow Diagram (EFD)**

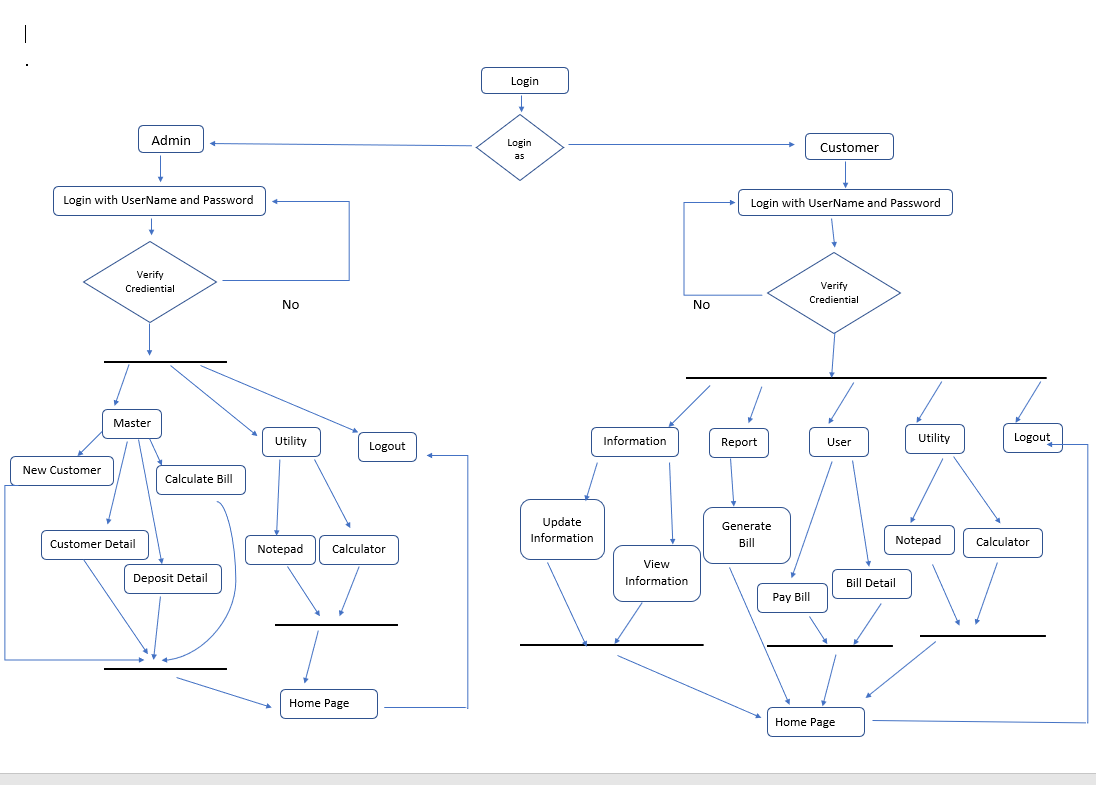
The event flow diagram describes the entire flow of the events which occurs in the software shown in the figure given below:

Figure 7*: Event Flow*

**5.2.3 Functional Model**

It describes aspect concerns with transformation of data values, functions, mapping constraints and functional dependencies. This modeling describes identification of input and output values, DFD for functional dependencies and identification of constraints.

**5.2.3.1 Identification of Input / Output values**

The System has following object and their input and output given in the table.

Table No 8: I/O Table

| **S.no.** | **Object Info** | **Input Details** | **Output Details** |
| --- | --- | --- | --- |
| 1 | Admin Login | Username | Login successful and display Admin Home Page. |
| Password |
| Login Us |
| 2 | Sign Up | Create Account As | Register Successful of New Admin or New Customer. |
| Meter Number |
| UserName |
| Name |
| Password |
| 3 | Customer Admin | Username | Login successful and display Customer Home Page. |
| Password |
| 4 | Electricity Bill | Meter Number | update unit consumed after entering the details. |
| Name |
| Address |
| Unit Consumed |
| Month |
| 5 | Add New Customer | Customer Name | Show Response Message on  Successful registration of new customer and enter into the Meter information. |
| Meter No |
| Address |
| City |
| State |
| Email |
| Phase |
| 6 | Meter Information | Meter Number | After entered all details Show Response Message on  Successful registration of new customer |
| Meter Location |
| Meter Type |
| Phase Code |
| Bill Types |
| Days |
| 7 | Update Information | Address | Show response message on entered new update information. |
| City |
| State |
| Email |
| Phone Number |
| 8 | Generate Bill | Month | After enter month generate electricity bill. |
| 9 | Pay Bill | Month | Display paid or unpaid message of Electricity Bill. |

**5.2.3.2 DFD as needed to show functional dependencies.**

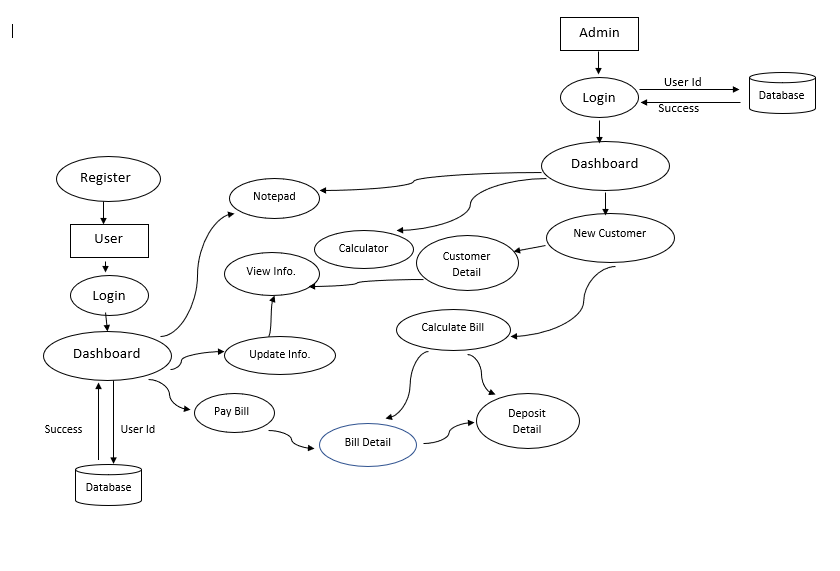
The DFD describes the data movement in the software which is shown in the diagram given below:

Figure 9*: DFD Diagram*

**5.2.4 Database Design**

**5.2.4.1 Schema definitions with Candidate key/Primary key**

The System use following schema with corresponding candidate key.

Table No 9: Login Schema Table

**login**

|  |  |  |
| --- | --- | --- |
| **Title** | **Type** | **key** |
| meterno | Varchar(20) | Primary key |
| username | varchar(30) |  |
| name | varchar(30) |  |
| password | varchar(20) |  |
| user | varchar(20) |  |

login = (meter\_no, user\_name, name, password, user)

Table No 10: Customer Schema Table

**customer**

|  |  |  |
| --- | --- | --- |
| **Title** | **Type** | **Key** |
| name | varchar(20) | Primary key |
| Meterno | varchar(20) |  |
| address | varchar(50) |  |
| city | varchar(30) |  |
| state | varchar(30) |  |
| email | varchar(40) |  |
| phone | Varchar(20) |  |

customer= (name, meter\_no, address, city, state, email, phone)

Table No 11: MeterInfo Schema Table

**Meter\_info**

|  |  |  |
| --- | --- | --- |
| **Title** | **Type** | **key** |
| meterno | Varchar(20) | Primary key |
| meterlocation | Varchar(20) |  |
| metertype | Varchar(20) |  |
| phasecode | Varchar(20) |  |
| billtype | Varchar(20) |  |
| days | Varchar(20) |  |

MeterInfo = (meterno, meter\_location, meter\_type, phase\_code, bill\_type, days)

Table No 12: Bill Schema Table

**bill**

|  |  |  |
| --- | --- | --- |
| **Title** | **Type** | **key** |
| meterno | Varchar(20) | Primary key |
| month | Varchar(20) |  |
| units | Varchar(20) |  |
| totalbill | Varchar(20) |  |
| status | Varchar(20) |  |

bill = (meter\_no, month, units, total\_bill, status)

Table No 13: Tax Schema Table

**tax**

|  |  |  |
| --- | --- | --- |
| **Title** | **Type** | **key** |
| cost per unit | Varchar(20) | Primary key |
| meter rent | Varchar(20) |  |
| service charge | Varchar(20) |  |
| service tax | Varchar(20) |  |
| swacch bharat cess | Varchar(20) |  |
| fixed tax | Varchar(20) |  |

tax = (cost\_per\_unit, meter\_rent, service\_charge, service\_tax, swacch\_bharat\_cess, fixed\_tax)

**5.2.4.2 Normalization**

In this Project Second Normal Form (2NF) database design has been applied to ensure efficient and normalized data storage and retrieval. By adhering to the principles of database normalization, specifically the 2NF, the project team has organized the database schema to minimize data redundancy and maintain data integrity.

**5.2.4.3 E-R Diagram**

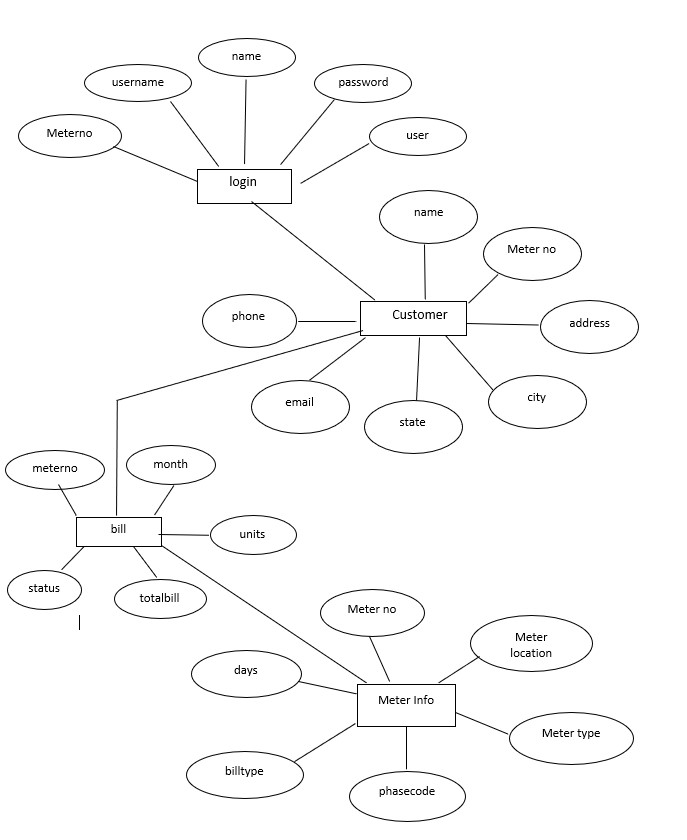
The ER Diagram has been desired in the project given below.

Figure 10*: E-R Diagram*

**6. CONSTRUCTION PHASE**

**6.1. CASE Tools used to design**

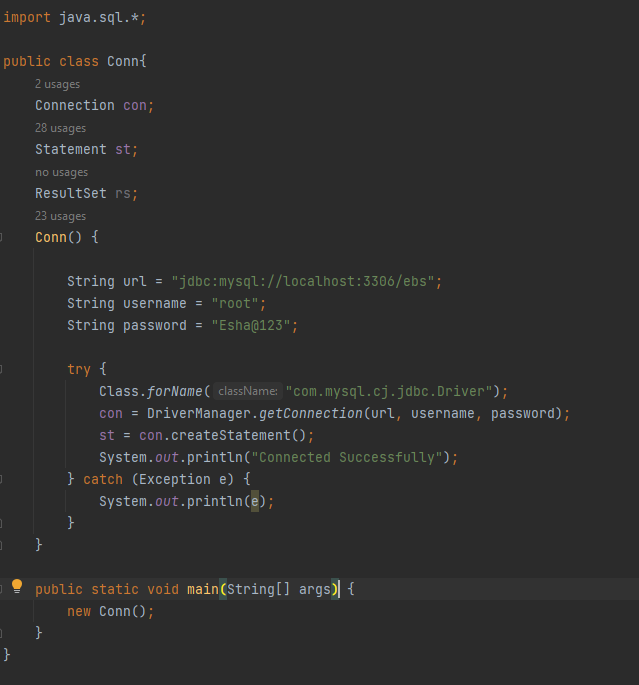
There was no case tools used for designing but for coding Visual Studio Code is used

**6.2. Coding Language and Operating System (OS) and (Including explanation)**

In this project Windows OS is used as operating system far as language is concern Java is used because of its property and facility given.

**6.3. Database Connectivity procedure**

MYSQL Database connectivity is used and required code segment as follows.

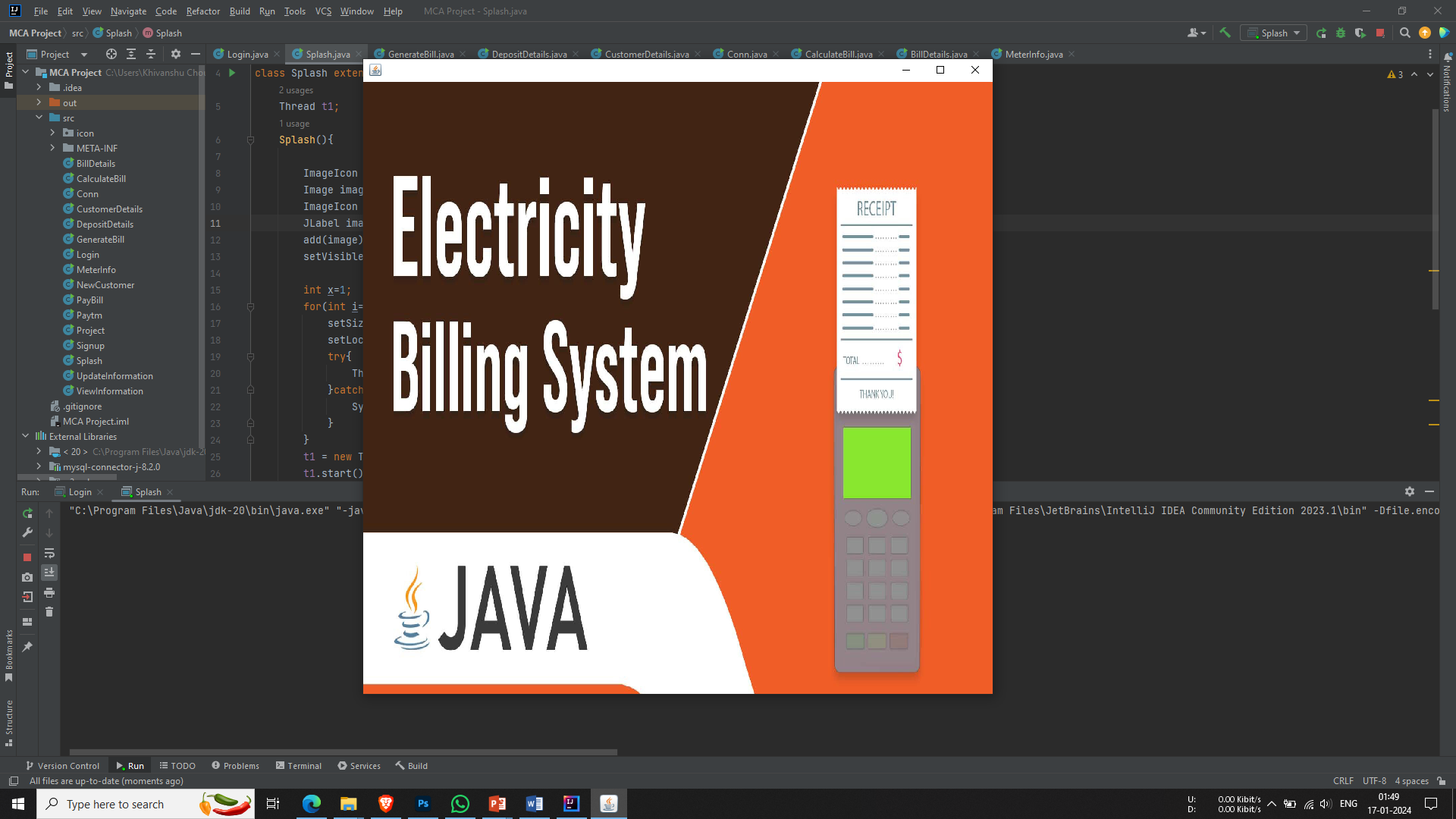


**6.4. Code Description**

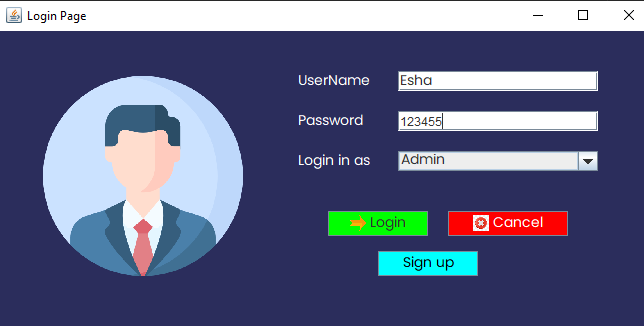
Entire system code is seen in following table.

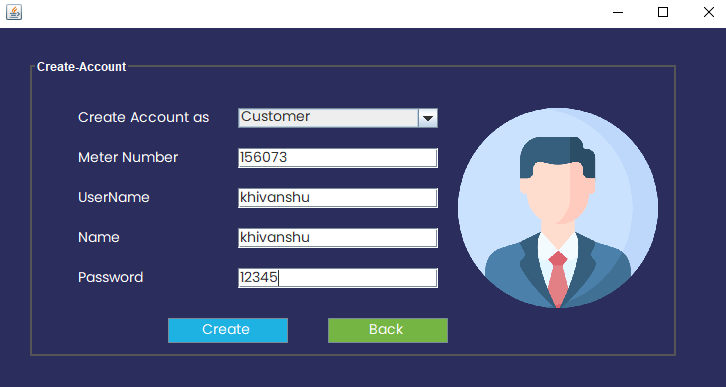
Table No 17: Code Description

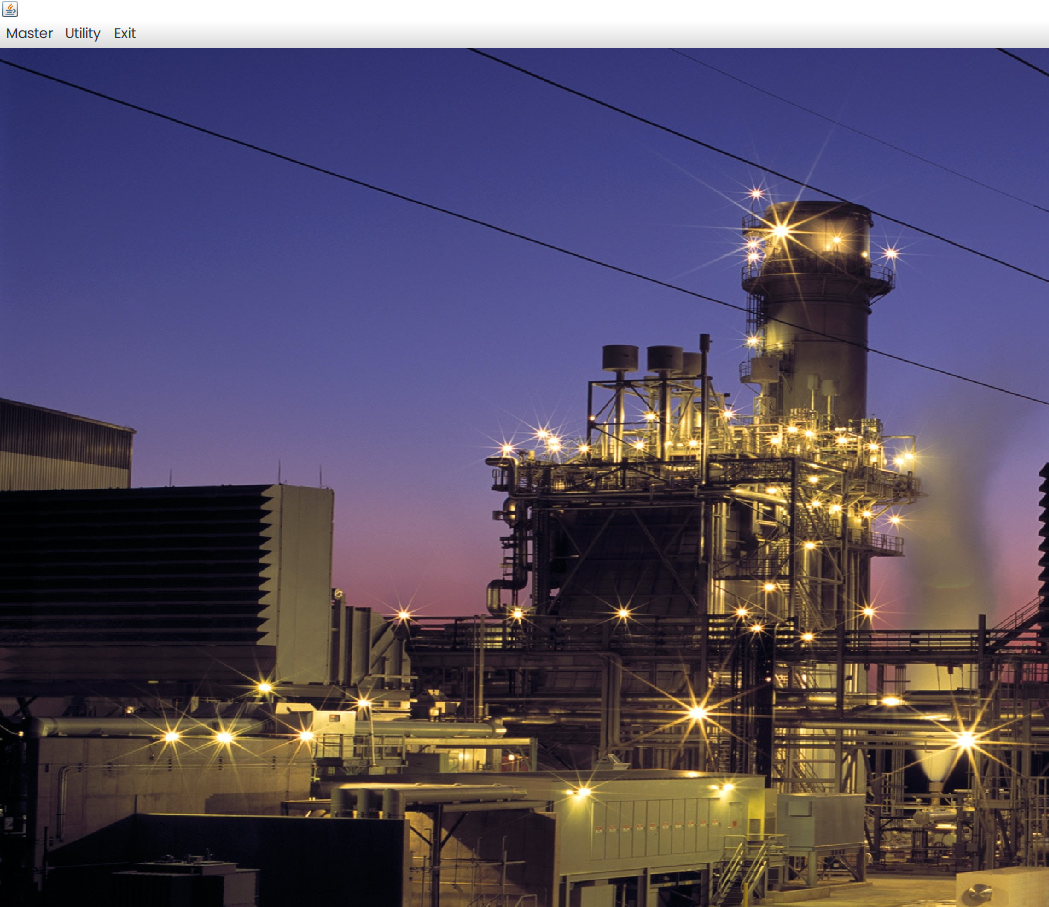
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **File name** | **File size** | **Purpose** | **Constructor** | **LOC** |
| 1 | Login.java | 5kb | Accept User Request | 1 | 126 |
| 2 | NewCustomer.java | 6kb | Add New Customer | 1 | 158 |
| 3 | CustomerDetail.java | 2kb | Show the details of Customer | 1 | 54 |
| 4 | Project.java | 11kb | Dashboard Screen | 1 | 251 |
| 5 | Signup.java | 7kb | Register New Customer | 1 | 189 |
| 6 | BillDetails.java | 1kbs | Bill Details | 1 | 38 |
| 7 | CalculateBill.java | 8kb | Calculate Bill | 1 | 198 |
| 8 | Conn.java | 1kb | Create database Connection | 1 | 28 |
| 9 | GenerateBill.java | 6kb | Generate Electricity Bill | 1 | 115 |
| 10 | DepositDetail.java | 4kb | Show Electricity Deposit Details | 1 | 109 |
| 11 | MeterInfo.java | 6kb | Show to Meter Information | 1 | 167 |
| 12 | PayBill.java | 6kb | Pay Electricity Bill | 1 | 167 |
| 13 | Paytm.java | 2kb |  | 1 | 44 |
| 14 | UpdateInformation.java | 6kb | Update Customer information | 1 | 163 |
| 15 | ViewInformation.java | 5kb | View Customer Details | 1 | 133 |
| 16 | Splash | 2kb | Front Screen of Project | 1 | 44 |
| **TOTAL** | | | | | **1974** |

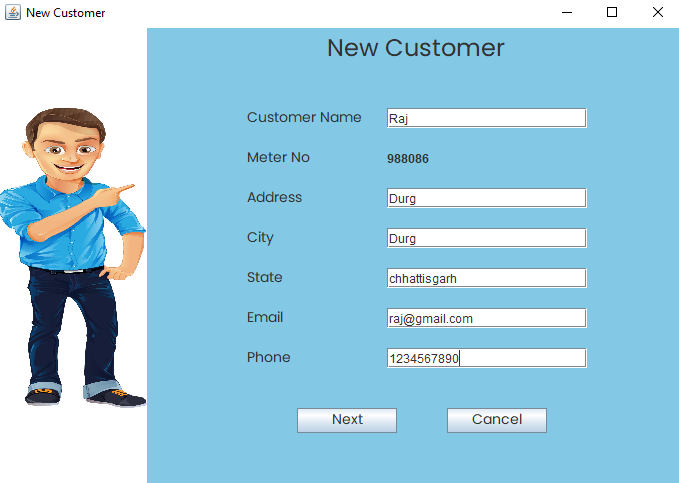
**6.5. Splash**

**Login**

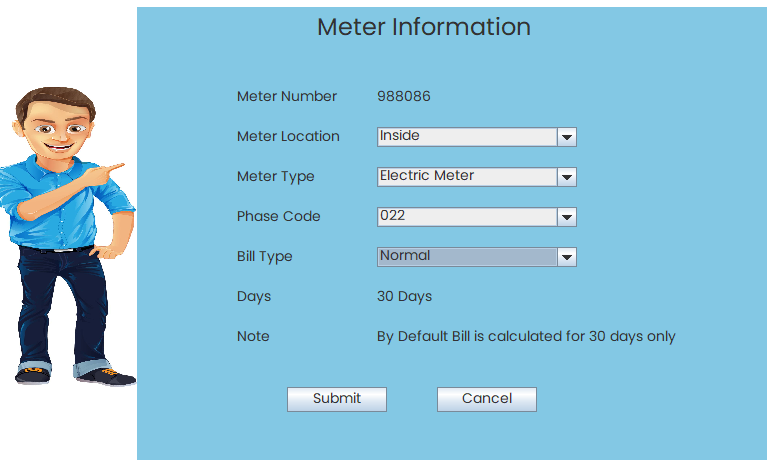


**Sign up**

**Admin Home**

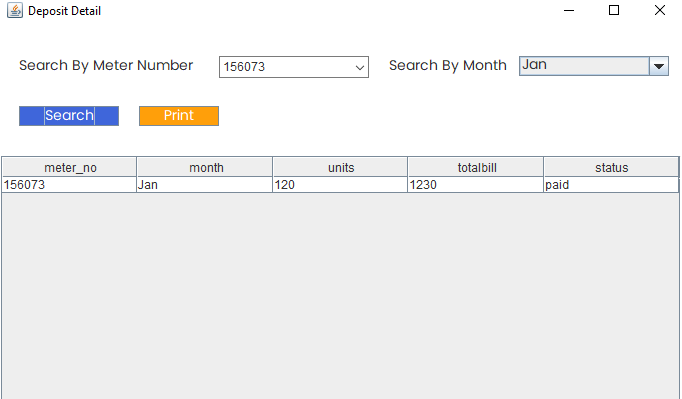
**New Customer**

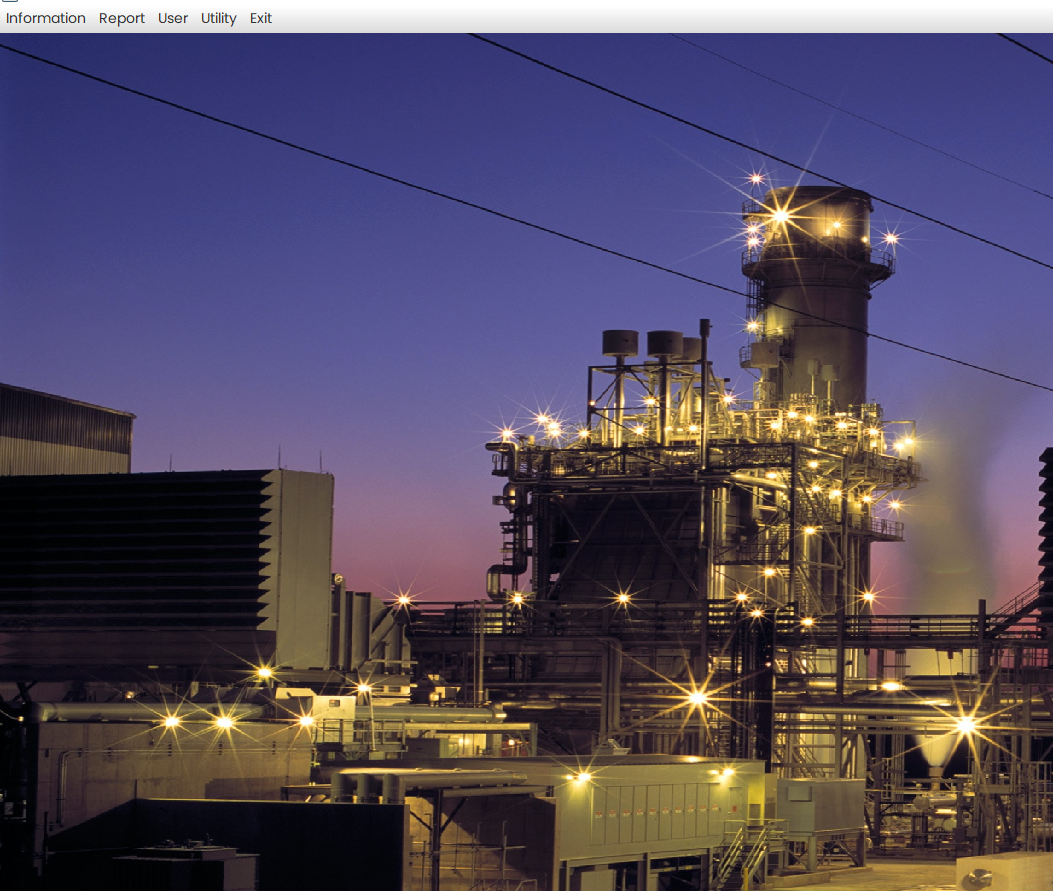
**Meter Information**

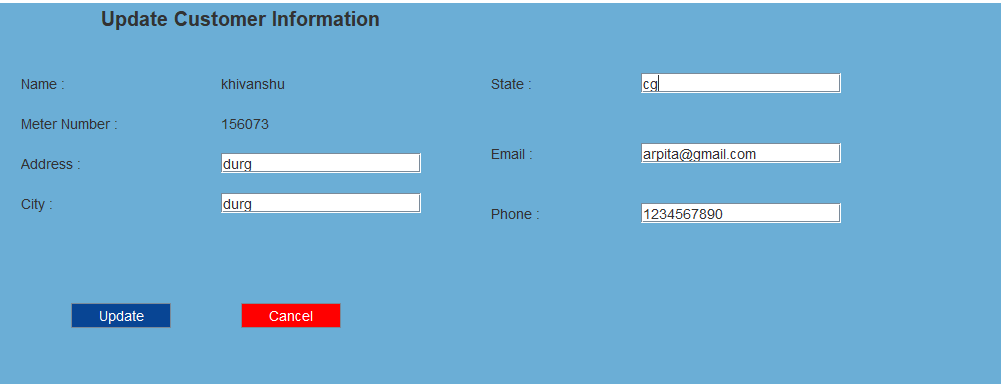


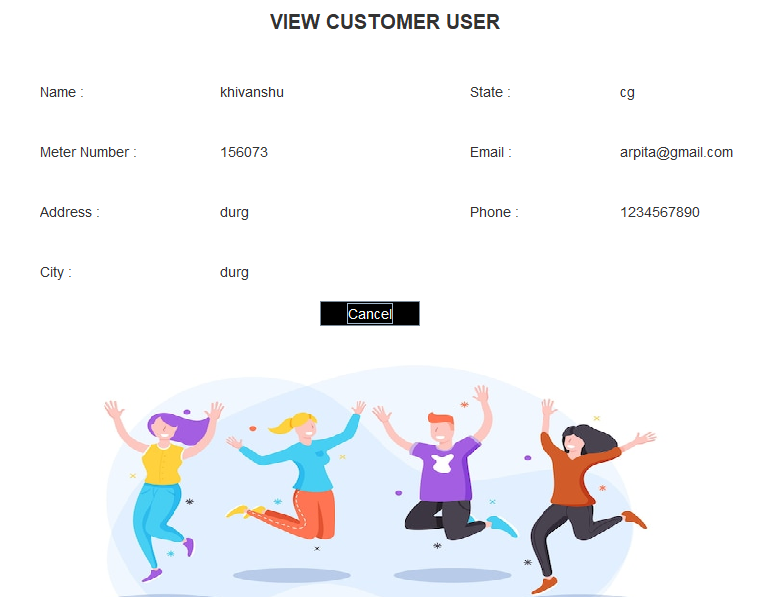
**Customer Detail**

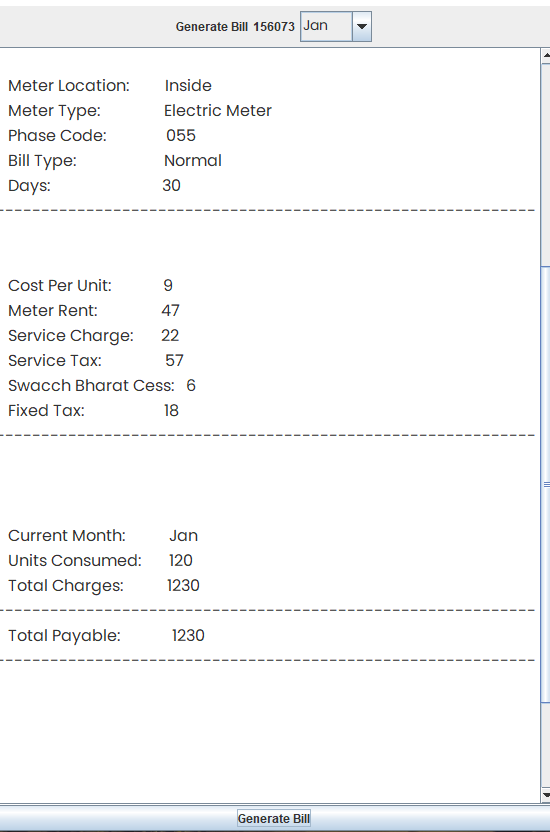
**Deposit Detail**



 **Customer Home Page**

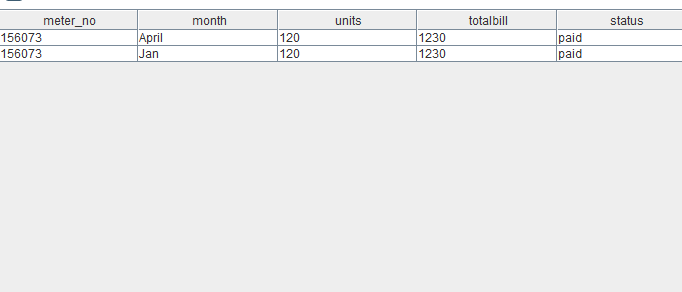
 **Update Information**

**View Customer Information**



**Generate Bill**

 **Pay Bill**

 **Bill Detail**

**7.TRANSITION PHASE**

**7.1. Report of Alpha testing**

1) *Blackbox unit testing* for each object input are given and required output is obtained successfully**.**

2) *Whitebox unit testing* for each object for given input entire program flow including loops and conditional statements are tested successfully.

3) *Integrated system testing* All objects are linked integrated and tested by given input and desired output

4) *Whitebox integrated system testing* In this test all links between object and association has been tested successfully and obtained desired output.

**7.2. DSLOC and Cost estimate using DSLOC**

The cost estimation is calculated using Intermediate COCOMO model:

Table 18: Effort adjustment factor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cost**  **Driver** | **Ratings** | | | | | |
| **Very low** | **Low** | **Normal** | **High** | **Very high** | **Extra high** |
| **RELY** | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |  |
| **DATA** |  | 0.94 | 1.00 | 1.08 | 1.16 |  |
| **CPLX** | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| **TIME** |  |  | 1.00 | 1.11 | 1.30 | 1.66 |
| **STOR** |  |  | 1.00 | 1.06 | 1.21 | 1.56 |
| **VIRT** |  | 0.87 | 1.00 | 1.15 | 1.30 |  |
| **TURN** |  | 0.87 | 1.00 | 1.07 | 1.15 |  |
| **ACAP** | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |  |
| **PCAP** | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |  |
| **AEXP** | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |  |
| **VEXP** | 1.21 | 1.10 | 1.00 | 0.90 |  |  |
| **LEXP** | 1.14 | 1.07 | 1.00 | 0.95 |  |  |
| **TOOL** | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 |  |
| **MODP** | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 |  |
| **SCED** | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 |  |

Table 19: Coefficient of Intermediate COCOMO model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software project** | **a** | **b** | **c** | **d** |
| Organic | 3.2 | 1.05 | 2.5 | 0.38 |
| Semi Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 2.8 | 1.20 | 2.5 | 0.32 |

On the basis of KDLOC and intermediate COCOMO model the cost is calculated as follows

**Total Delivered SLOC of whole project is= 1.974 KLOC**

* Required software reliability= Very Low = 0.75
* Size of application database= Low = 0.94
* Complexity of the product= Very Low = 0.70
* Use of software tools= Nominal = 1.00
* Required development schedule= Nominal = 1.00

**EAF**= 0.75\*0.94\*0.70\*1.00\*1.00 = 0.49

**Effort** = a(KLOC)^b \* EAF = 3.2 \* (1.974^1.05) \* 0.49 = **3.19PM**

**Duration** = c(Effort)^d= 2.5(3.19^0.38) = **3.87 Months**

**Person** = Effort / Duration= 3.19 / 3.87 = **0.92  ~  1 person**

**7.3. Cost variation between DFP and DSLOC**

The cost variation is the difference between Actual value and estimated value and cost variation is given below:

**Effort Expected by DFP = 11.9 PM**

**Effort By Delivered SLOC = 3.19PM**

Variance = Expected cost – Actual Cost

**= 11.9 – 3.19 = 8.71 PM**

**Expected KLOC = 3.507 KLOC**

**Delivered KLOC = 1.974 KLOC**

Variance in KLOC = Expected KLOC – Delivered KLOC

**= 3.507 – 1.974 = 1.533 KLOC**

**8. Limitation and Future Enhancement**

The Limitation of the project are as follows:

* No recovery of the Password
* Not server side billing system

These limitations can be overcome in the future by giving update in the application and providing better user experience

**9. Conclusions**

It is software that helps the user to work with the billing cycles, paying bills etc. This software reduces the amount of manual data entry and gives greater efficiency.. This project is under the organic type project. For complete error fixing and during the SRS development and writing complete SRS, modern software practice has been utilized instead of conventional waterfall technique. In which 4 development phase was used are Inception, Elaboration, Construction, Transition phase. During the SRS Development in Inception phase, the Delivered function point is 85.05 identified. During the Elaboration phase the complete detail design of system-object model, dynamic and functional model is developed. The MYSQL is used for the database system For Backend side Java language is used for the implementation of the object. Total DSLOC in terms of KLOC 1.974 Total effort calculated is 3.19 PM. The Complete System has been tested successfully