**CIS 634 Object-Oriented Software Engineering**

**Software Requirements Specification**

**Project Title: Power Management System**

Logo, company name

Description automatically generated

Instructor: Prof. Weidong Xiong

Group Number: 7

Group Members:

Tapan Desai CSU ID: 2777437

Shaishav Shah CSU ID: 2835285

Kinjal Patel CSU ID: 2794458

Mitee Patel CSU ID: 2817313

1. **Introduction**

* Power Management System is a web application project which will monitor and manage the electricity usage of customers, calculate real-time units used & due billable amount, provide administration functionalities for admins to handle the management of bills, and provide payment gateway for final payment for the bills.

**1.1 Goals and objectives**

* The major goal of this project is to provide a desktop web application capable of managing and paying bills considering both the customer and administration perspectives.
* To Achieve the goal, we will be using Java Language as its platform Independent Java is object-oriented. This allows you to create modular programs and reusable code.
* Additionally, OOPs concepts are also being kept in mind while developing this web application. Considering this, the project will use Java programming language for back-end development of the project.
* The Front end would be designed in HTML+CSS. Using HTML would be easy as well as compatible to the system as we can convert the Designed UI from CorelDraw or any other designing platform easily.
* For database, we would be using MY SQL.
* Under Java, we would be using Spring Boot framework as it is s build applications with ease and with far less toil than other competing paradigms.

**1.2 Statement of Scope**

* For the customer side, the customer will be able to log in into the system, view the customer panel, look up his/her electricity usage, pay the pending bills, and look up previous bills.
* For the admin side, the administrative personnel will be able to log in into the system, view the admin panel, locate all the customers and their billing history, view all payment deadlines, and manage all bill payment activities.
* The admin can also keep track of customer records and alter units used.

**1.3 Major Constraints**

* The main constraint for this project will be that as we don’t have access to get all the actual bill data from an existing database. So, we will instead work with dummy data inputted by ourselves for this project.
* We also would have only the payment gateway, but no actual payment would be processed.

1. **Usage Scenario**

* To log-in for both Customers as well as Admins, they need username and password. The customer ID should have their house number for tracking information easily. Passwords can be changed by the individual but for initial version it can include ethe last name and their DOB.
  1. **User Profiles**
* Our Project only has local user profiles.
* The user’s profiles would have address, phone number, Customer ID, etc information that is manually entered by us.
* The Unique key is the customer ID which has the last name of the user and the house number.
  1. **Use Case Diagram**

**Diagram

Description automatically generated**

* 1. **Special usage considerations**
* This is a prototype project.
* There is only a payment-gateway, but no payment is processed.
* The Units are manually entered directly into database.

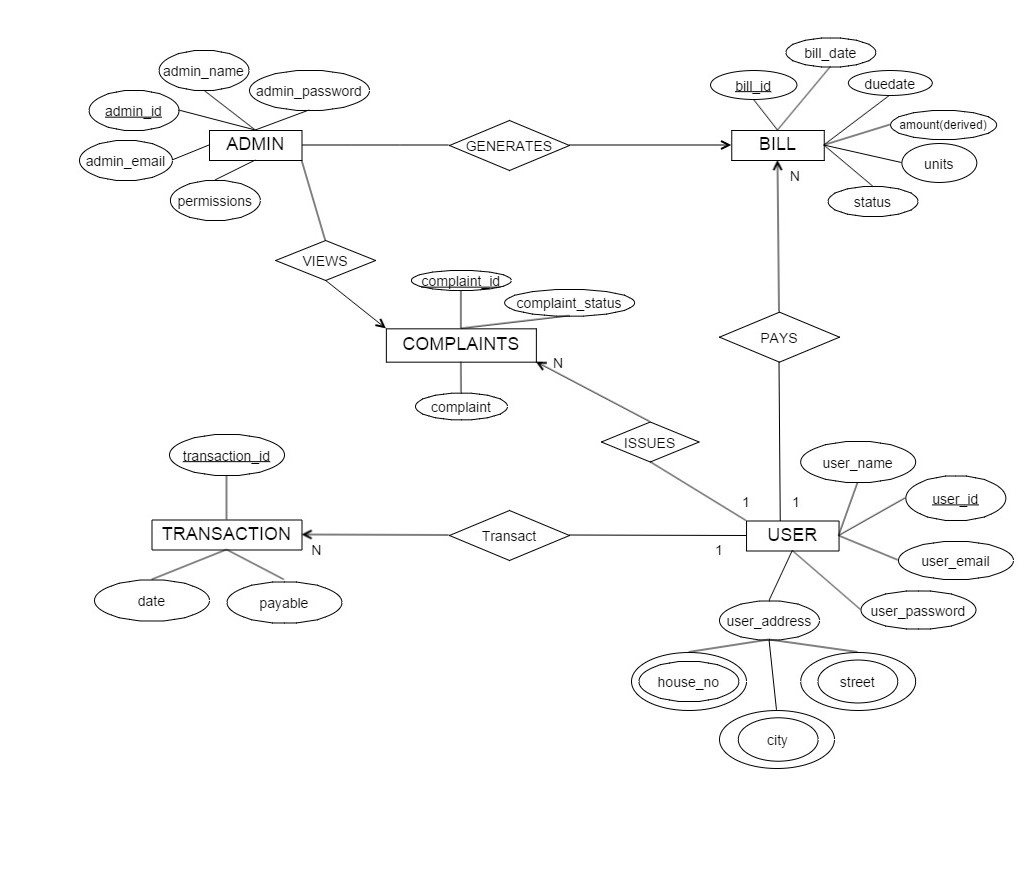
1. **Data Model and Description**

**3.1 Data Description**

- The fields that will be manipulated by the administrator are Units used, Customer I information and Bill payment confirmation.

- The fields that will be manipulated by the customer will be Bill Payment mode and viewing previous bills.

**3.2 Complete Data Model**



**3.3 Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Customer |  | Table | Login |
| Field Name | Type |  | Field Name | Type |
| id | integer |  | id | int |
| meterno | integer |  | userid | varchar() |
| consumerno | biginteger |  | branch | text |
| consumername | varchar() |  | session\_in | datetime |
| load\_con | varchar() |  | session\_out | datetime |
| month | varchar() |  |  |  |
| year | integer |  |  |  |
| email | varchar() |  | Table | Bill |
| address | text |  | Field Name | Type |
| amountgen | decimal |  | bill\_id | int |
|  |  |  | bill\_date | datetime |
|  |  |  | duedate | datetime |
| Table | Admin |  | amount | Int |
| Field Name | Type |  | units | Int |
| Id | Int |  | status | varchar() |
| Name | varchar() |  |  |  |
| Password | varchar() |  |  |  |
| email | varchar() |  |  |  |

**4.0 Functional Model and Description**

**4.1 Requirement**

**1.** All the bills should be able to retrieve back when required by the customer.

**2.** Customer should be able to see the units used by them over a period.

**3.** All the passwords and username should be pre-decided by the administrator according to his/her and house number.

**4.** The Administrator can alter the units used by each customer.

**4.2 Software Interface Description**

**-** The Software main purpose is to deliver a simple yet efficient model to not only serve the customer but also simplify the administrators work.

- So, to do that we need the web app to have very simple interface. The Log-in page should be common for customer and as well as administrators.

**5.0 Restrictions, Limitations, and Constraints**