



IT3030

Programming Applications and Frameworks

3rd Year, 1st Semester

Assignment

Group Project

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the
Bachelor of Science Special Honors Degree in Information Technology

Group No:Y3.S1.WE.IT.01.02(IT)_ Group 29

Submitted By :

Vidanagamage C.S : IT20198886

Semini J.P.D.L : IT20241346

Wickramasinghe W.A.P.M : IT20189976

D.M.T.D Bandara :IT20219802

Table of Contents

1. Cover Page
2. Table of Content
3. Introduction
4. Member's and Workload
5. Git Link
6. SE Methodologies/Methods
7. Gantt Chart
8. Requirement's analysis (Functional, Non-functional, Technical requirements)
9. Use case Diagram/ Activity Diagram
10. Overall Architecture
11. ER Diagram
12. Individual Section

Introduction

Electro Grid (EG) is the company in control of the country's power system. They have a system to monitor the customers' energy consumption, generates monthly bills and sends them to them automatically, and allows online payments from the users. We used java JAX-RS Jersey, tomcat and MySQL as our tools to build the platform. Using this system customer can pay for electricity bills online. This platform is simply and easy to understand.

Group Member's Details

IT Number	Name	Function	Description of the Web Service
IT20198886	Vidanagamage C.S	Payment Management	Insert Payment Details View Payment Details Update Payment Details Delete Payment Details
IT20241346	Semini J.P.D.L	Employee Management	Insert Employee Details View Employee Details Update Employee Details Delete Employee Details
IT20189976	Wickramasinghe W.A.P.M	Customer Management	Insert Customer Details View Customer Details Update Customer Details Delete Customer Details
IT20219802	D.M.T.D Bandara	Complain Management	Add Complain View Complain Update Complain Delete Complain

Git Link

https://github.com/chami1999/ElectroGrid_PAF.git

SE Methodology

- Agile methodology can be used for development purposes.

Agile methodology refers to the combination of iterative and incremental models. Complex projects are easily managed due from its adaptability. The project team should be able to break the task down into small stages using this methodology.

Pros

- Although people interaction occurs often here, it is very advantageous to people communication.
- Adaption can happen anytime and in any phase.

Cons

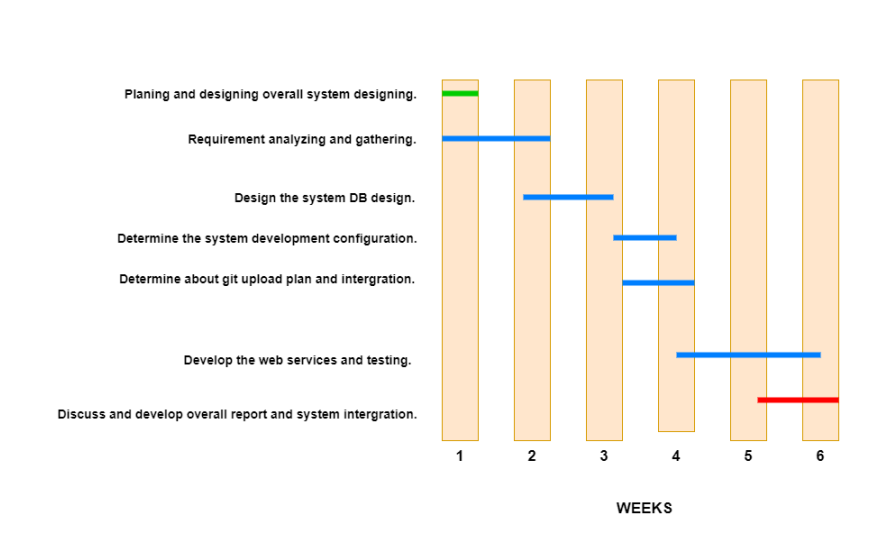
- When a debate is going awry, the project's direction can be thrown off.

We have used integrated testing and unit testing as our testing purposes

Unit testing can be used for unit testing services such as individual elements or units. Every startup awards that undergoes unit testing is tested as a unit.

Integrate testing can describe as all services being tested as a unit after unit testing. The tested micro service will be tested as a particular service during integration testing, and the referencing will be selected.

Gantt



Chart

System's overall design

Onion diagram



Requirement's analysis

Functional Requirements

Payment Management –Add, Update, View and Delete Payment details
Employee Management –Add, Update, View and Delete Employee details
Customer Management –Add, Update ,View and Delete Customer
Complaint Management –Add ,Update ,View and Delete Complaint

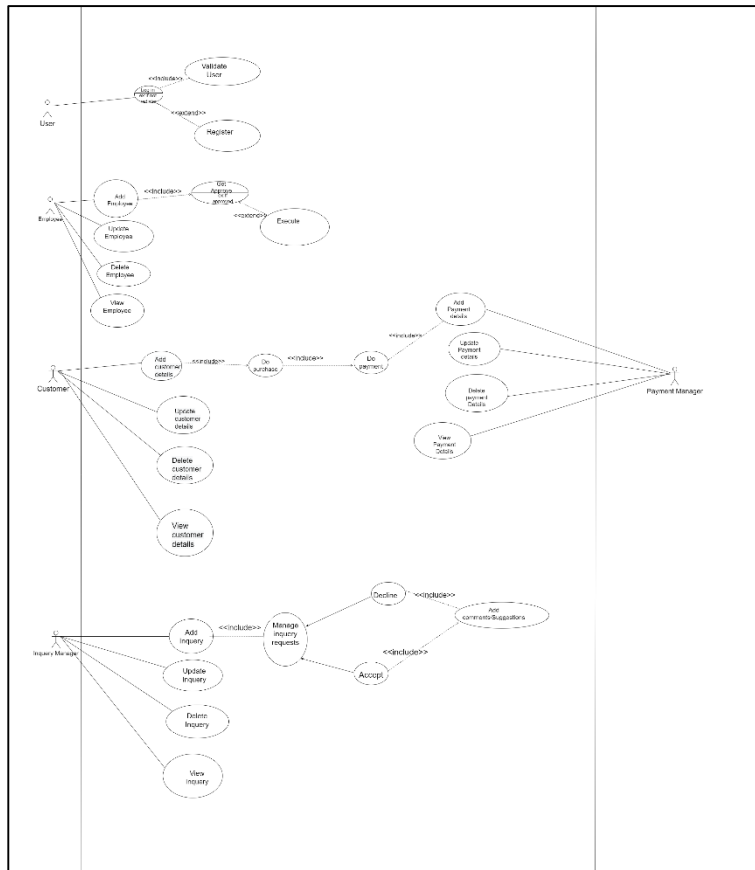
Non-Functional Requirements

Efficiency ,Maintainability,Privacy,Campatibility,Quality

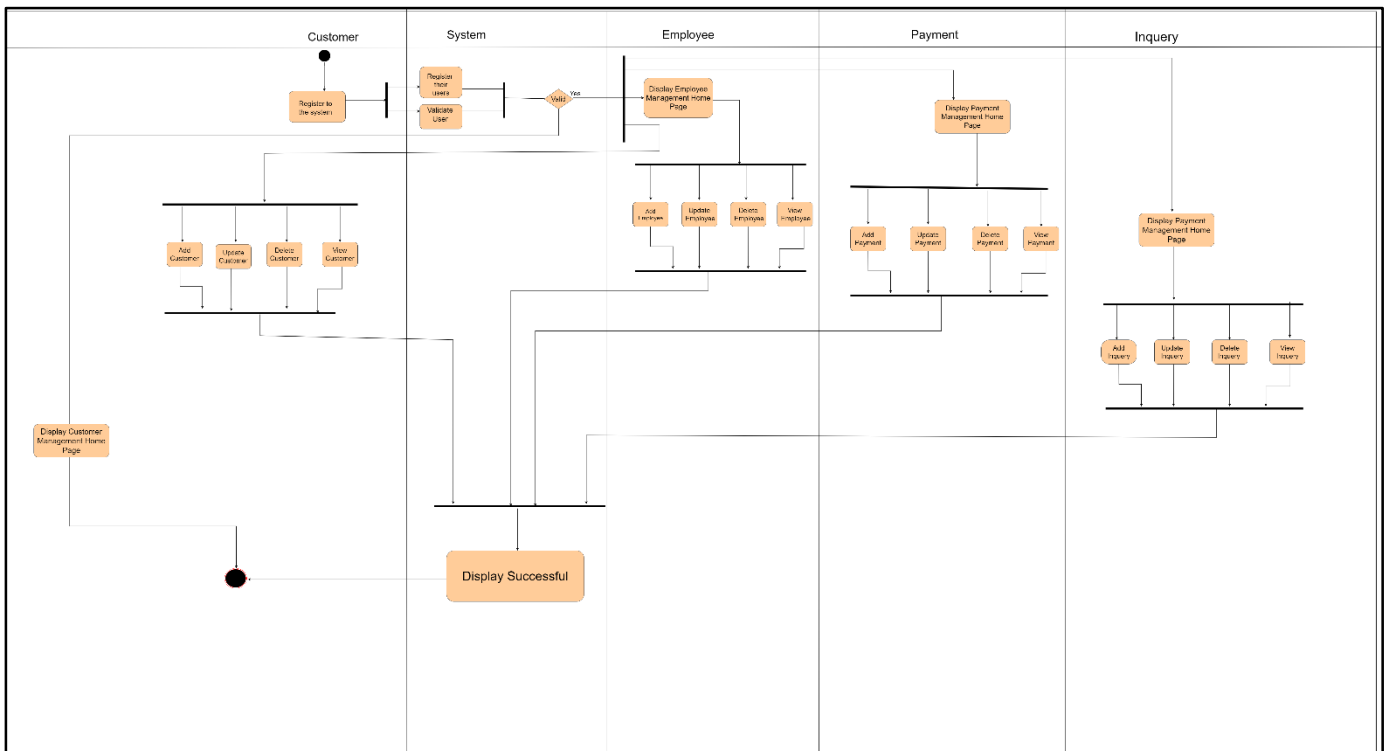
Technical Requirements

The backend have developed using Restful web services JAX-RS, Jersey, Java on Tomcat on Eclipse IDE in a windows OS. Database is MYSQL and we have used postman API client for testing purposes.

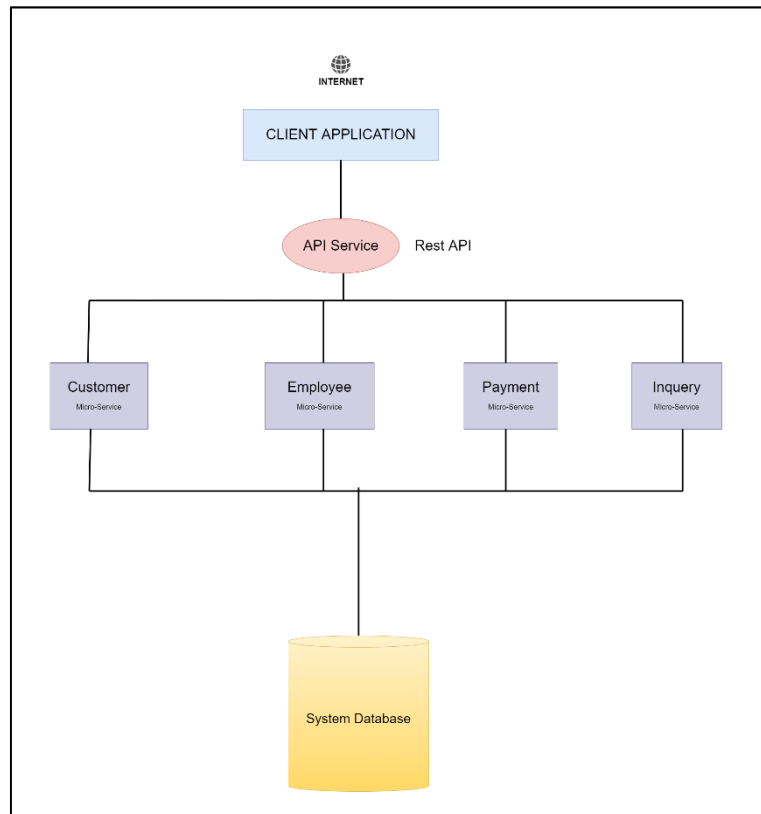
System Use Case Diagram



Activity Diagram

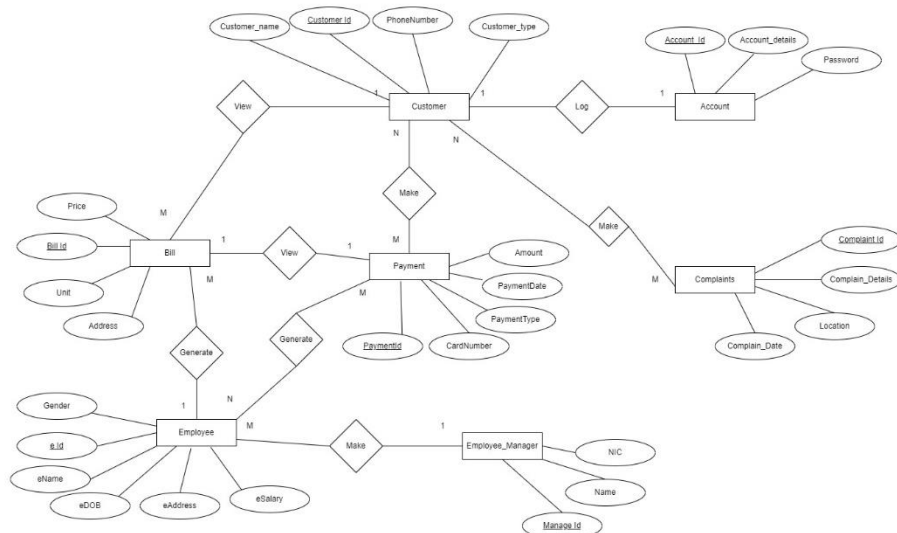


Overall Architecture

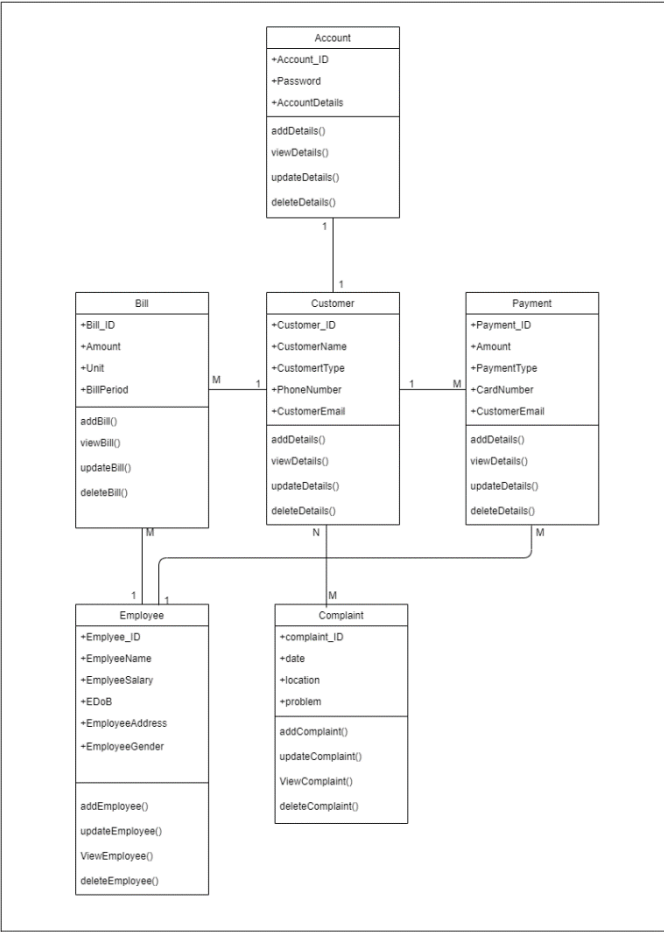


Our task to create a system to monitor power consumption of the users and generate monthly bill using this system and automatically send bills to users through this system and also customer can do payments through this system.

ER Diagram



Class Diagram



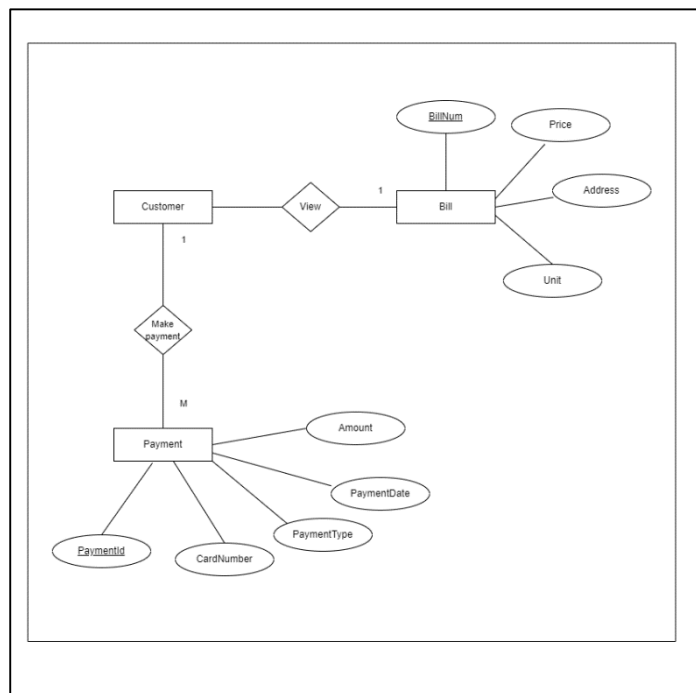
1 Member : IT20198886 – Vidanagamage C.S

URL: <http://localhost8082/ElectroGrid/PaymentService/Payments>

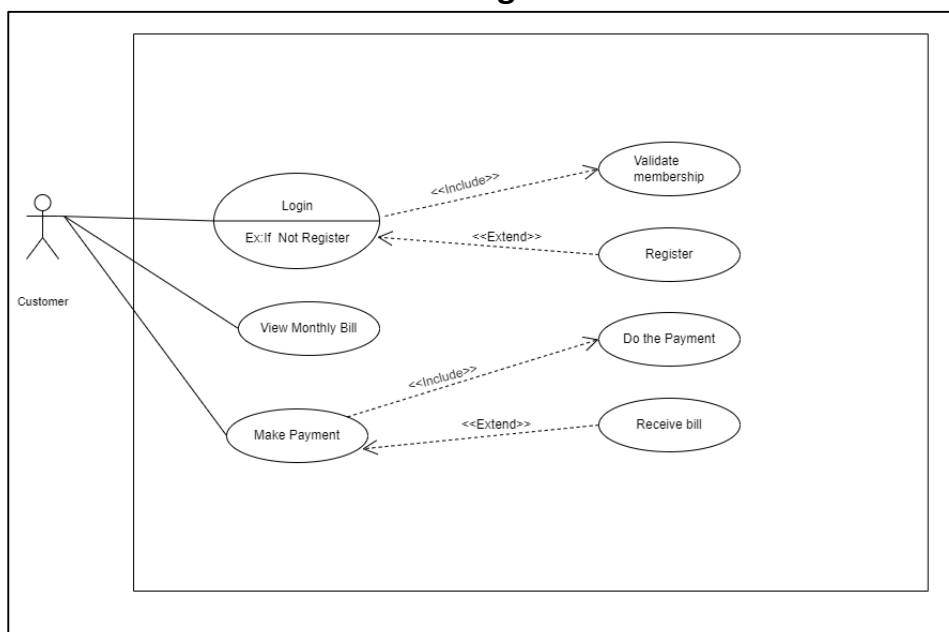
Payment Management

Here the full scope of the Payment management is completed. The respected parties can perform their function successfully through this platform. In payment management the parties can insert, update, delete and view the payment details.

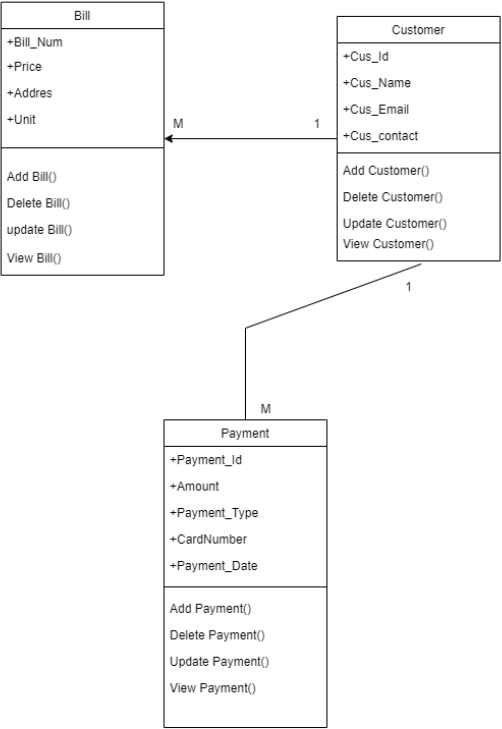
ER Diagram



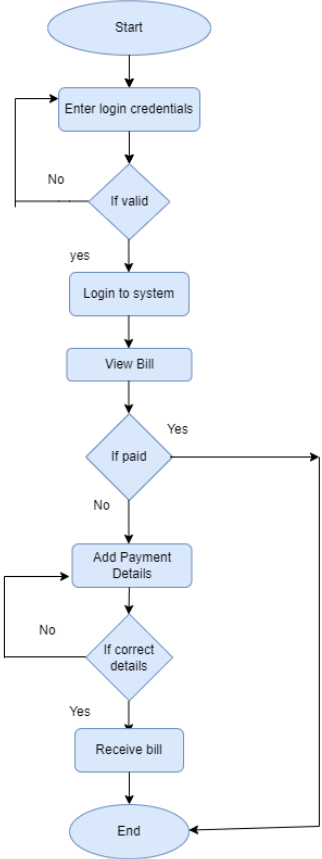
Use Case Diagram



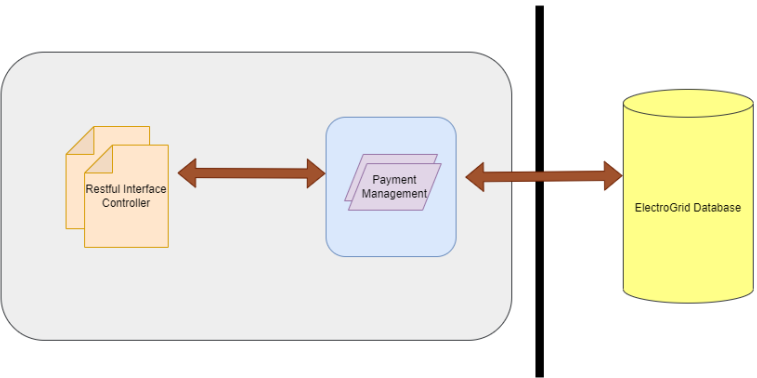
Class Diagram



Flow Chart



Internal workflow



API Design Relational

Payment management is one of a main part in this system. The Customers can view their electricity bill and pay payment through system and employee can generates bills through system. Using this customer can do easily their payments. As well as employee can generates bill easily using this system.

Service Development :-

- Technologies Used: Java - JAX-RS (Jersey) on Tomcat
- IDE : Eclipse
- Database : MySQL
- Testing : POSTMAN

API for get all the Employee details in the database (GET Request)

URL: <http://localhost8082/ElectroGrid/PaymentService/Payments>

Request : - { }

Response : - {PaymentId : "Auto generated integer value"}

{ PaymentDate ="22/02/2021"}

{ CardNumber : " 43636754744343546 "}

{ Amount = "14000.00 "}

{ PaymentType = "Cash" }

API for insert a new Employee to the database (POST Request)

URL: <http://localhost8082/ElectroGrid/PaymentService/Payments/insert>

Request : - {PaymentId : "Auto generated integer value"}

{ PaymentDate ="22/04/2022"}

{ CardNumber : " 446367547443123546 "}

{ Amount = "13000.00 "}

{ PaymentType = "Cash" }

Response :- { Result = "Inserted successfully" }

{ PaymentID = "Auto generated integer value" }

{ Error= "Error while inserting"}

API for insert a new Employee to the database (PUT Request)

URL: <http://localhost8082/ElectroGrid/PaymentService/Payments/update>

Request : - {PaymentId : "Auto generated integer value"}

{ PaymentDate ="22/02/2022"}

{ CardNumber : " 446367547443123546 "}

{ Amount = "11000.00 "}

{ PaymentType = "Cheque" }

Response :- { Result = "Updated successfully" }

{ PaymentID = "Auto generated integer value" }

{ Error= "Error while updating"}

API for insert a new Employee to the database (DELETE Request)

URL: <http://localhost8082/ElectroGrid/PaymentService/Payments/delete>

Request : - {PaymentId : "Select PaymentId from the service"}

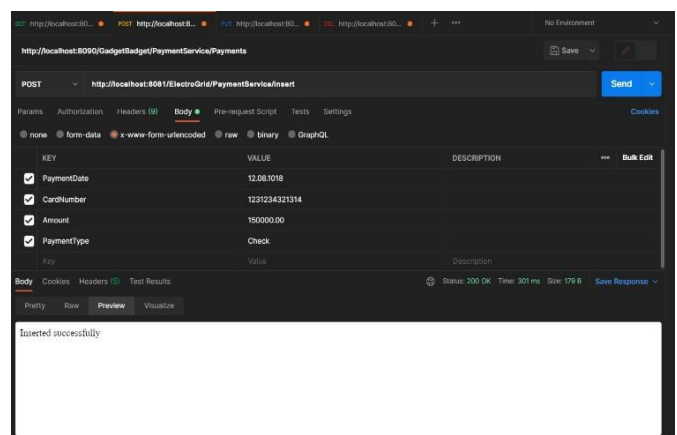
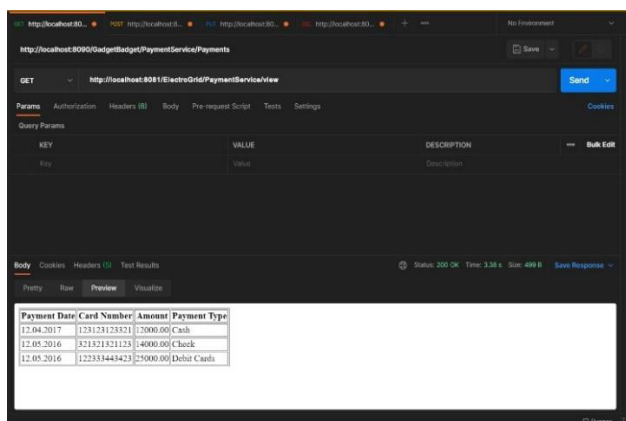
Response :- { Result = "deleted successfully" }

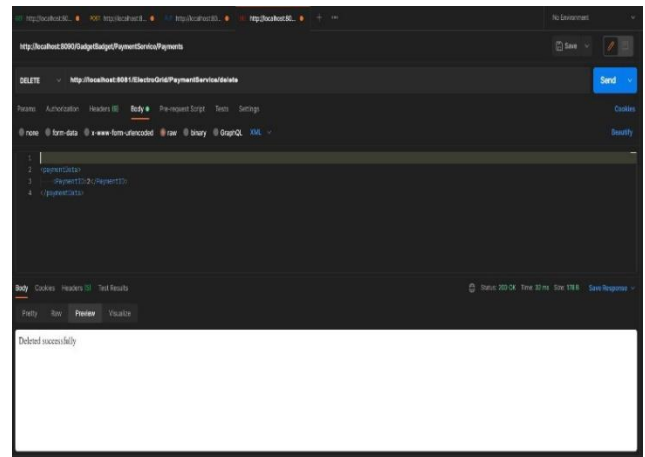
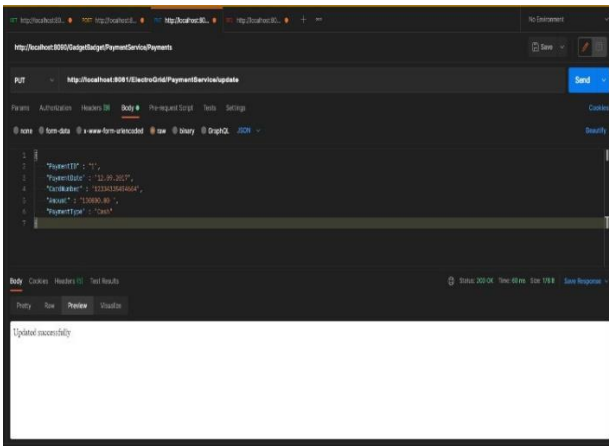
{ Error= "Error while deleting the payment details"}

Test Cases

TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Payment Details	{PaymentId: "Auto generated integer value"} { PaymentDate="22/04/2022"} { CardNumber : "446367547443123546 "} { Amount = "13000.00 "} { PaymentType = "Cash" }	New Payment details are added to the database. Show message as "Inserted successfully".	New Payment details are added to the database. Show message as "Inserted successfully"	Pass
2	Update Payment Details	{PaymentId : "Auto generated integer value"} { PaymentDate ="22/02/2022"} { CardNumber : "446367547443123546 "} { Amount = "11000.00 "} { PaymentType = "Cheque" }	Update payment details are added to the database and show message as "Update details successfully"	Updated details added and show message as "Update details successfully"	Pass
3	Delete Payment Details	<paymentData> <PaymentId>2</PaymentId> </PatmentData>	Delete payment details and show message as "Payment details deleted successfully"	Payment details deleted and showed message "Details deleted succussfully"	Pass

Testing





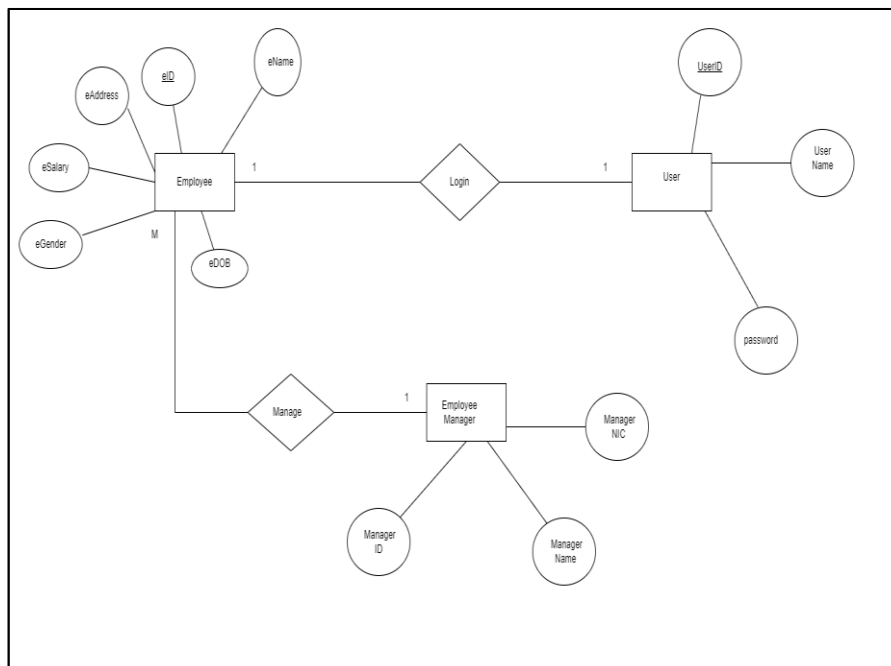
2 Member :IT20241346 (Semini J.P.D.L.)

URL: <http://localhost8082/ElectroGrid/EmployeeService/Employee>

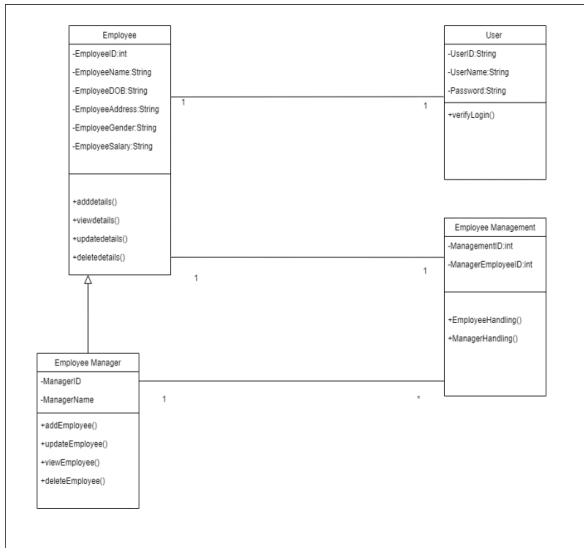
Employee Management

The Employee's full scope is accomplished here. Through this platform, respectable parties can properly carry out their functions. An employee can insert, update, delete, and view Employee information in Employee management.

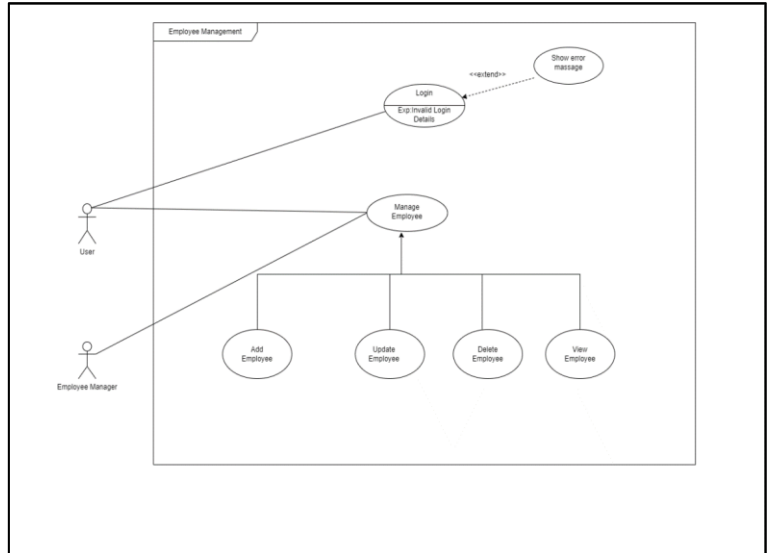
ER Diagram



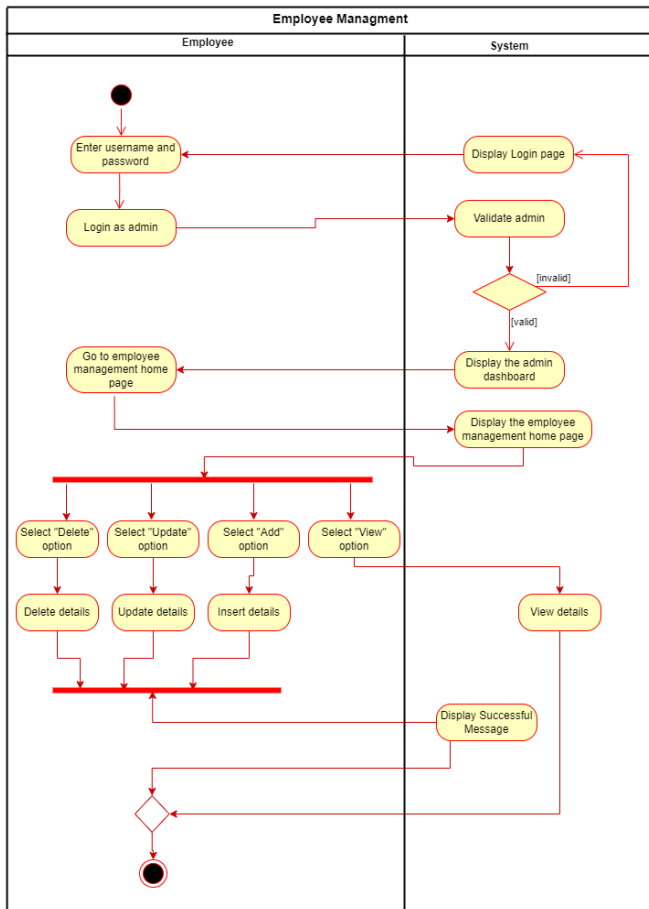
Class Diagram



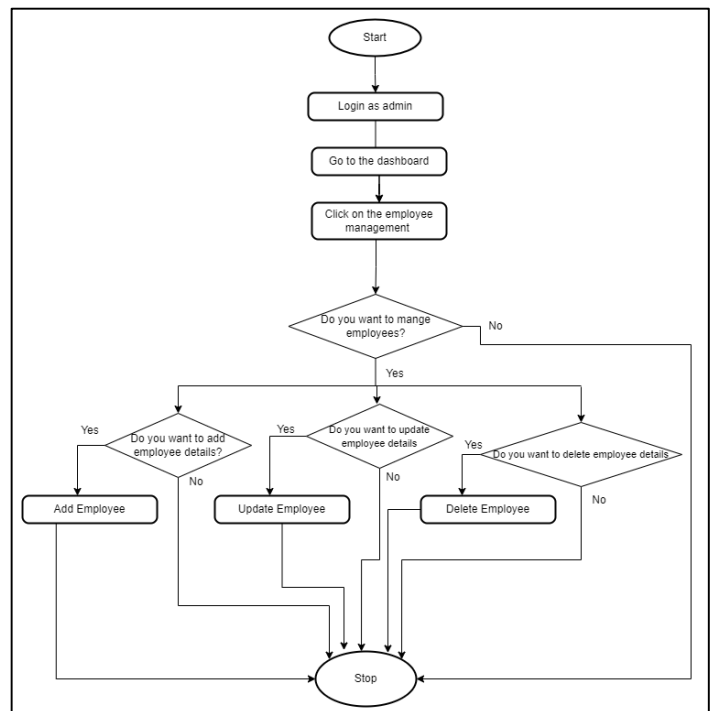
Usecase Diagram



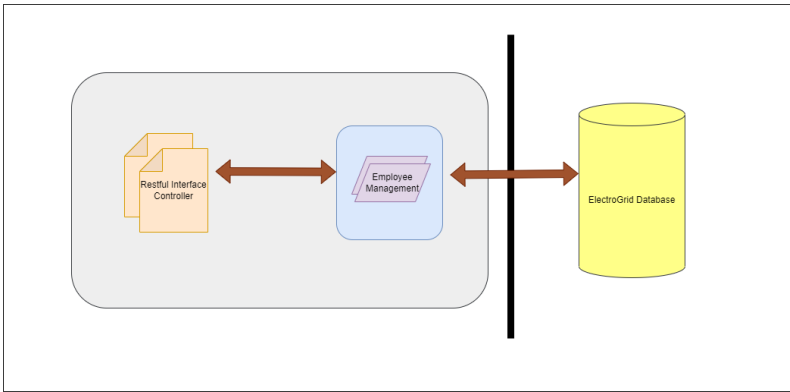
Activity Diagram



Flow Chart



Internal workflow



API Design Relational

Employee management is one of the main part in this system. It is a user friendly one that anybody can understand . Employee Manager plays a main role besides of employees . An employee can insert, delete, update and view the employee details.

Service Development :-

- Technologies Used: Java - JAX-RS (Jersey) on Tomcat
- IDE : Eclipse
- Database : MySQL
- Testing : POSTMAN

API for get all the Employee details in the database (GET Request)

URL:- <http://localhost8082/ElectroGrid/EmployeeService/Employee>

Request : - { }

Response : - {Employee ID : "Auto generated integer value"}

{Employee Name ="Kavishka"}

{Address : "Galle"}

{Email : kavishkal@gamil.com}

{Reg API for insert a new Employee to the database (POST Request)

URL:- <http://localhost8082/ElectroGrid/EmployeeService/Employee>

Request : - {Employee ID : "Auto generated integer value"}

{Employee Name ="Ravindu"}

{Address : "Mathara"}

{Email : ravindul@gamil.com}

{Reg Date: "2022.04.22"}

{Phone No: 0712345678"}

Response :- { Result = "Inserted successfully" }

Date: "2022.04.23"

{Phone No: 0786543213"}

API for update an Employee who is existing in the database (PUT Request)

URL:- <http://localhost8082/ElectroGrid/EmployeeService/Employee>

Request : - {Employee ID : "Auto generated integer value"}

{Employee Name ="Ravindu"}

{Address : "Kandy"}

{Email : ravindul@gamil.com}

{Reg Date: "2022.04.22"}

{Phone No: 0756789065"}

Response :- { Result = "Updated successfully" }

API for delete an Employee who is existing in the database (DELETE Request)

URL:- <http://localhost8082/ElectroGrid/EmployeeService/Employee>

Request :- { Employee ID = " Selected EmployeeID from the table "}

Response :-

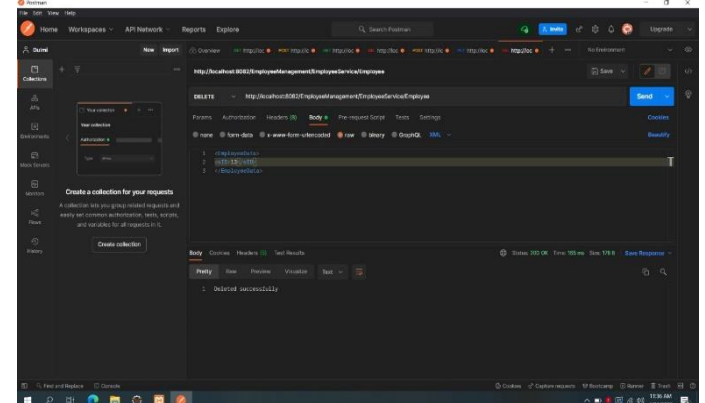
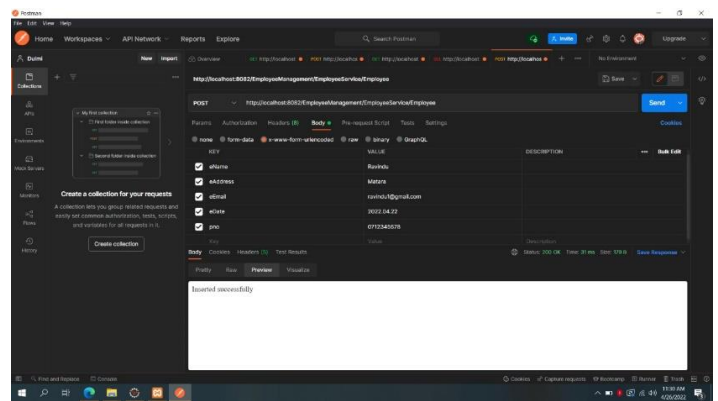
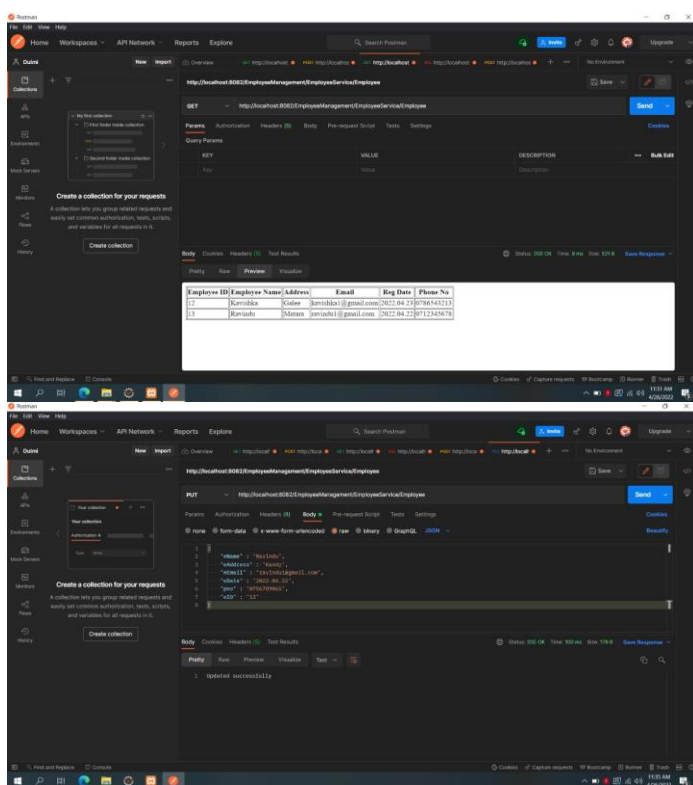
{ Result = "Deleted successfully" }

Test Cases:-

TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Employee Details	{Employee ID : "Auto generated integer value"} {Employee Name ="Kavishka"} {Address : "Galle"} {Email : kavishkal@gamil.com} {Reg Date: "2022.04.23"} {Phone No: 0786543213"}	View Employee details table.	View Employee details table.	Pass
2	Add Employee Details	{Employee ID : "Auto generated integer value"} {Employee Name ="Ravindu"} {Address : "Mathara"} {Email : ravindul@gamil.com}	New Employee details are added to the database. Show message as	New Employee details are added to the database. Show message as	Pass

		{Reg Date: "2022.04.22"} {Phone No: 0712345678"}	"Inserted successfully".	"Inserted successfully".	
3	Update Employee Details	Employee ID : "Auto generated integer value" {Employee Name = "Ravindu"} {Address : "Kandy"} {Email : ravindul@gamil.com} {Reg Date: "2022.04.22"} {Phone No: 0756789065"}	Employee details are updated according to relevant input Employee details. Show message as "Updated successfully".	Employee details are updated according to relevant input Employee details. Show message as "Updated successfully".	Pass
4	Delete Employee Details	<EmployeeData> <eID>13</eID> </EmployeeData>	Show message as "Deleted Successfully	Show message as "Deleted Successfully	Pass

Testing:-



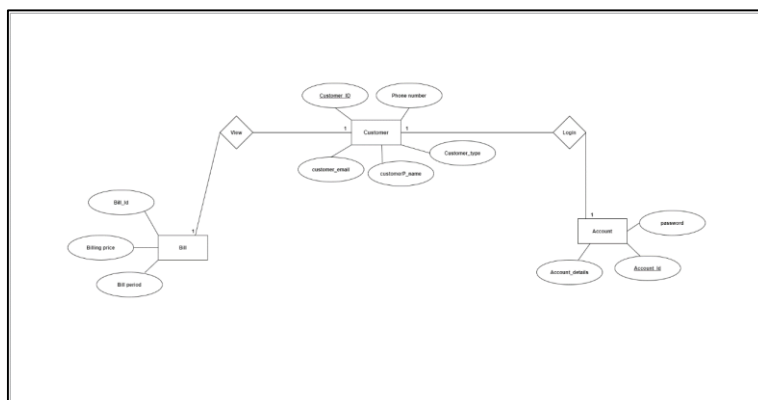
3 Member :IT20189976 - Wickramasinghe W.A.P.M

URL: <http://localhost:8082/ElecroGrid/CustomerService/Customer>

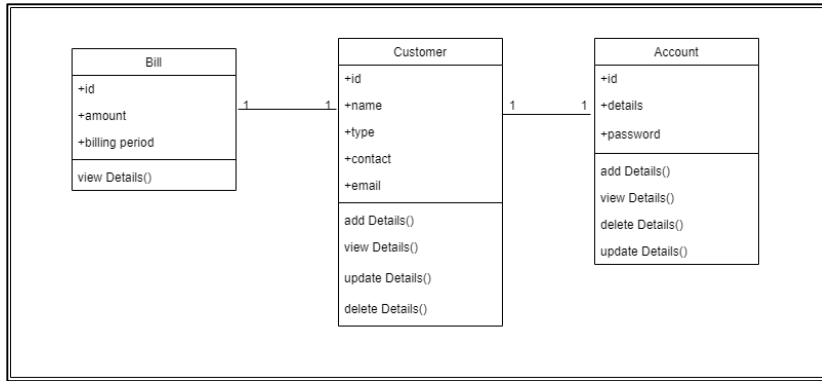
Customer Management

Here the full scope of the customer management is completed. The respected party can perform it function successfully though this platform. In customer management the customers can insert, update, delete and view the customer details

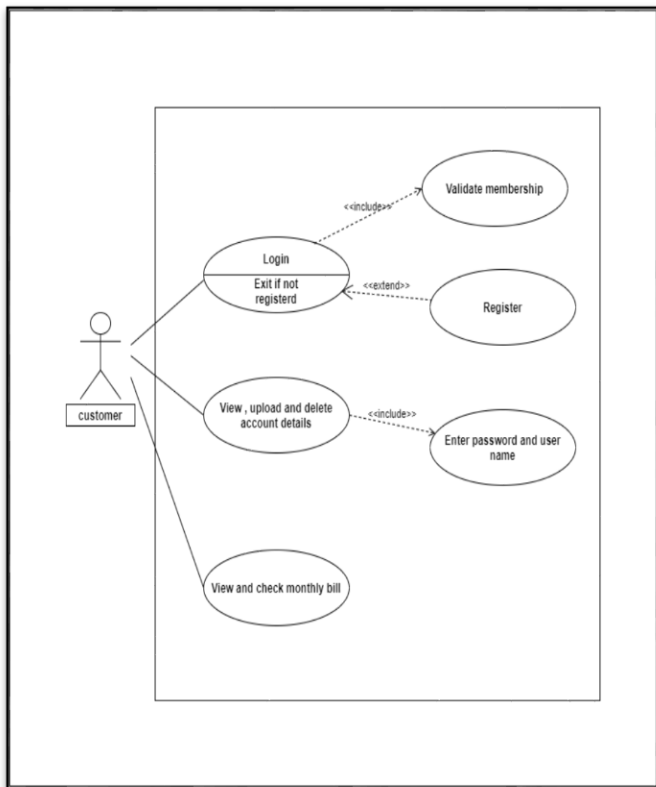
ER Diagram



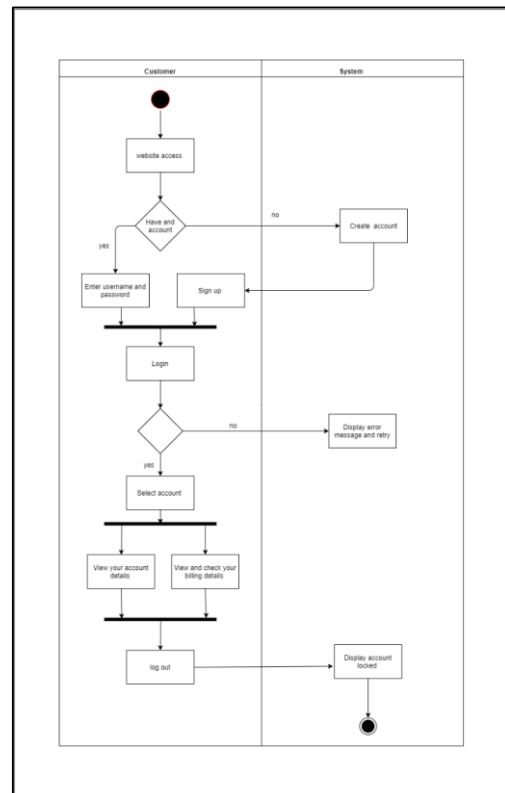
Class Diagram



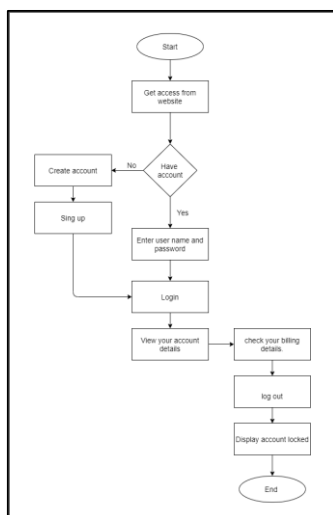
Use case Diagram



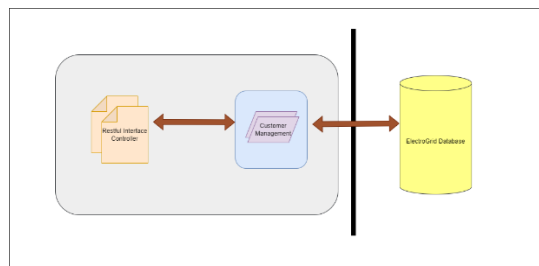
Activity Diagram



Flow Chart



Internal work flow



API Design Relational

Customer management is one of the main parts in this system. The Customers must create an account to view or check their monthly billing details and their account details . The system is user friendly and low complicated.

- Customers can insert, delete, update, and view the customer details
- The billing details also can be viewed by the respected party.

Service Development

- Technologies Used: Java - JAX-RS (Jersey) on Tomcat
- IDE : Eclipse
- Database : MySQL
- Testing : POSTMAN

API for get all the Customer details in the database (GET Request)

URL: - <http://localhost8082/ElectroGrid/CustomerService/Customers>

Request : - { }

Response : - {CustomerID : "Auto generated integer value"}
{CustomerName =:"anushka"}
{CustomerEmail :anushka@gmail.com }
{CustomerType : Loyal customer"}
{CustomerContact : "0765842136"}

API for insert a new Customers to the database (POST Request)

URL: - <http://localhost8082/ElectroGrid/CustomerService/Customers>

Request : - {CustomerID : "Auto generated integer value"}
{CustomerName =:"kalpana anushka"}
{CustomerEmail : kalpanaanushka@gmail.com }
{CustomerType : Need-based customer"}
{CustomerContact : "0765842136"}

Response :- { Result = "Inserted successfully" }
{ CustomerID = "Auto generated integer value"
{ Error= "Error while inserting"}

API for update a customers which is existing in database (PUT Request)

URL: - <http://localhost8082/ElectroGrid/CustomerService/Customers>

Request : - {CustomerID : "Auto generated integer value"}
{CustomerName =:" kalpana anushka "}

{CustomerEmail : kalpanaanushka@gmail.com }
{CustomerType : Need-based customer"}
{CustomerContact : "0765842136"}

Response :- { Result = "Updated successfully" }
{ CustomerID = "Auto generated integer value"
{ Error= "Error while Updating"}

API for delete a Customers which is existing in database (DELETE Request)

URL:- <http://localhost8082/ElectroGrid/CustomerService/Customers>

Request :- { CustomerID = " Selected Customer ID from the service "}

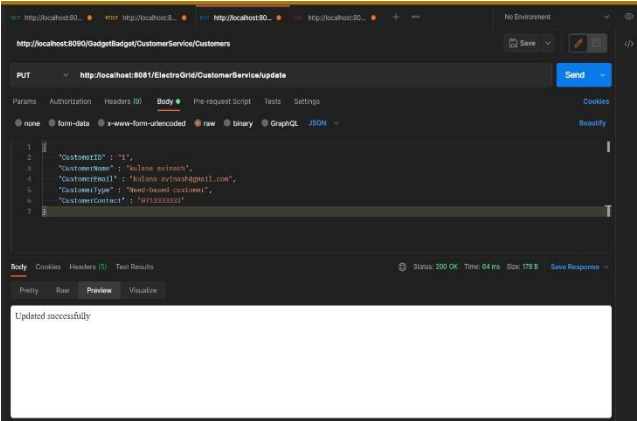
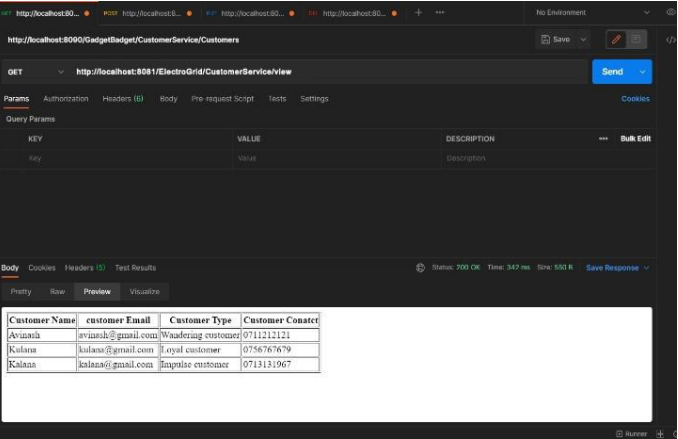
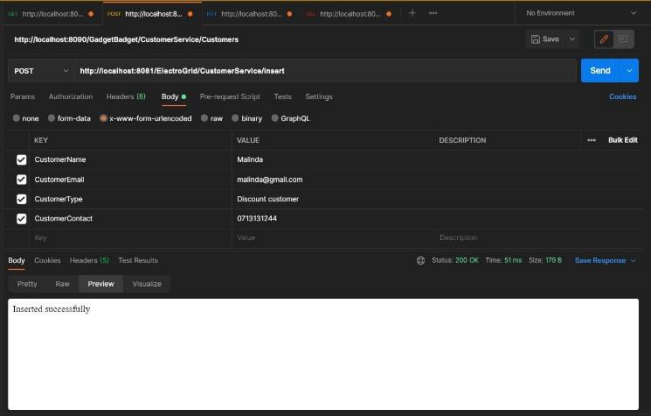
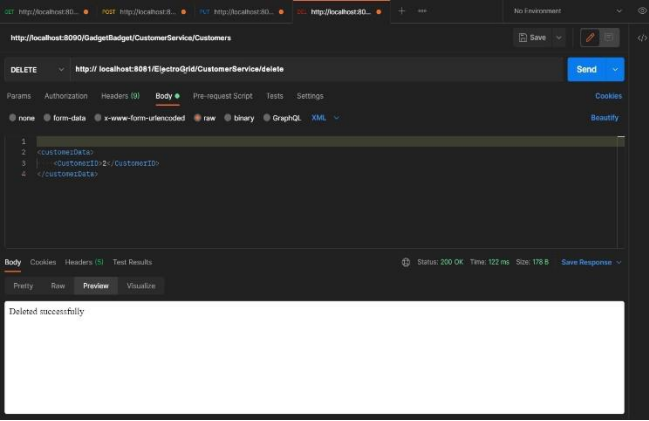
Response :- { Result = "Deleted successfully" }
{Error = "Error while deleting the account" }

Testing

TestID	Test Description / Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add customer details	{CustomerName =:"anushka"} {CustomerEmail :anushka@gmail.com } {CustomerType : Loyal customer"} {CustomerContact : "0765842136"}	New customer details are added to the database. Display message as "data successfully added	New customer details are added to the database. Display message as "data successfully added".	Pass
2	Update customer details	{CustomerID : "1" CustomerName =:" kalpana anushka" CustomerEmail : kalpanaanushka@gmail.com CustomerType : Need-based customer" CustomerContact : "0765842136"}	Customer details are updated according to the relevant input. Customer details are displaying the message "Update Successfully".	Customer details are updated according to the relevant input. Customer details are displaying the message "Update Successfully".	Pass
3	Delete customer details	<customerData>	Show message as	Customer details are deleted	Pass

		<pre><customer_ID>1<customer_Id > </customerData></pre>	“Delete Successfully Details”	successfully. Show message as “Delete Successfully Details	
--	--	--	-------------------------------------	---	--

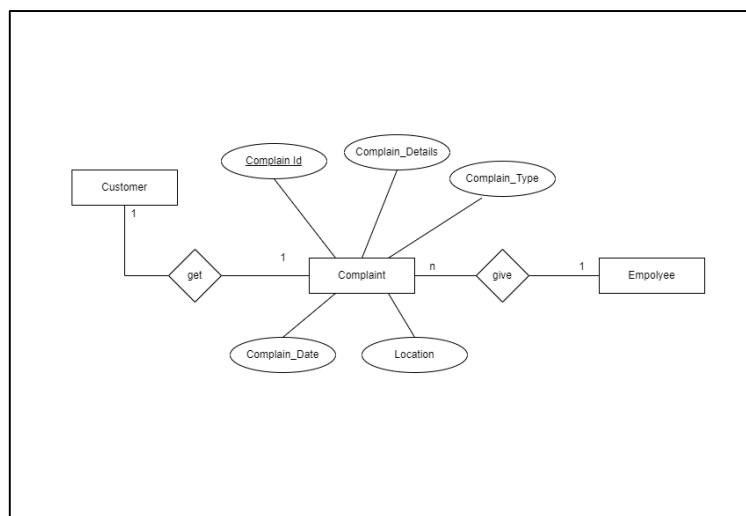
Testing



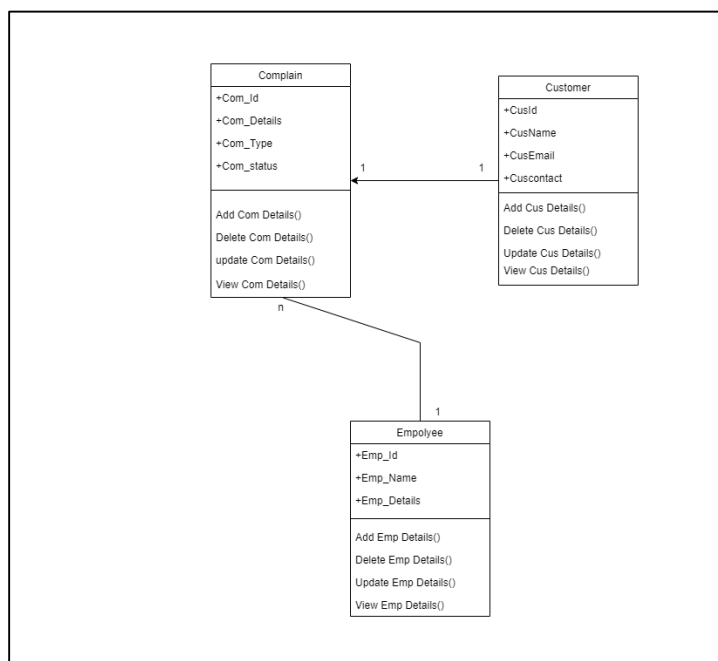
Complain Management

Here the full scope of the complain management is completed. The respected party can perform it function successfully though this platform. In Complain management the customers can add, update, delete and view the complaint.

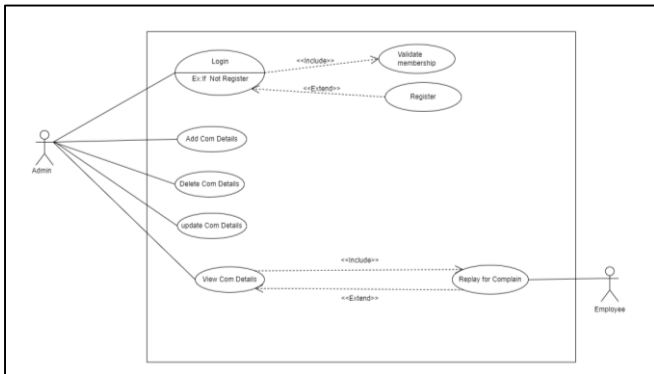
ER Diagram



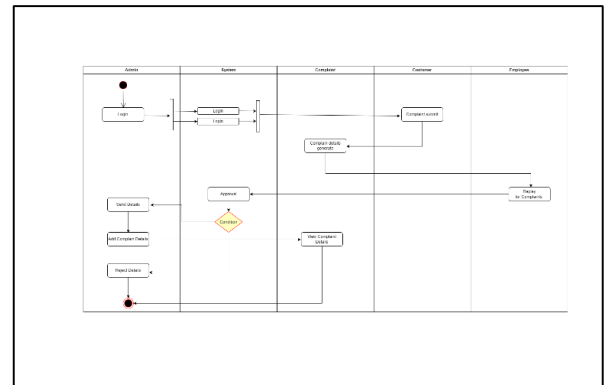
Class Diagram



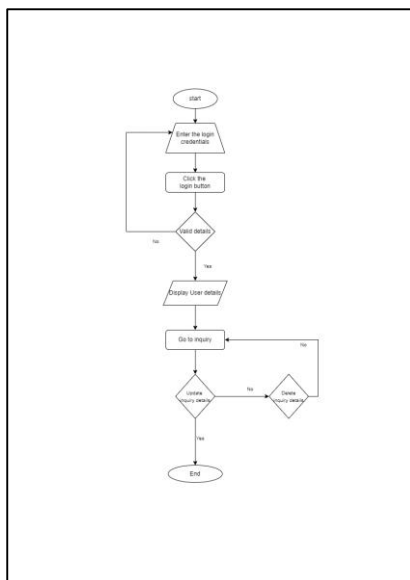
Use case Diagram



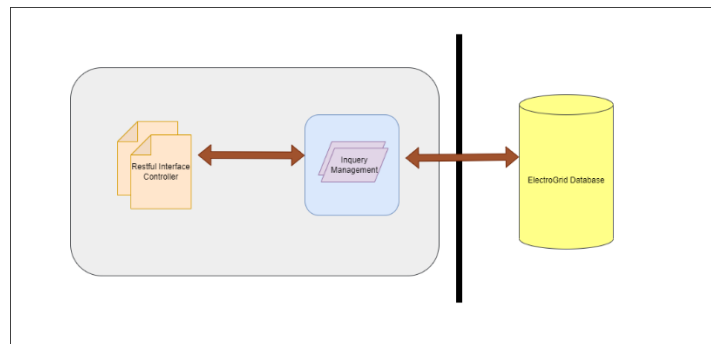
Activity Diagram



Flow Chart



Internal work flow



API Design Relational

Complaint management is one of the main parts in this system. The Customers can add complaint through this system.

- Customers can insert, delete, update, and view the Complaint.

Service Development

- Technologies Used: Java - JAX-RS (Jersey) on Tomcat
- IDE : Eclipse
- Database : MySQL
- Testing : POSTMAN

API for get all the Customer details in the database (GET Request)

URL: - <http://localhost8090/ElectroGrid/ComplaintService/Complaints>

Request : - { }

Response : - {ComplaintId: "Auto generated integer value"}
{ComplaintName =:"Thilina"}
{Date : "02/01/2022" }
{ComplaintDetails : "Sudden power failure"}
{ComplaintCategory : "power Failure"}

API for insert a new Customers to the database (POST Request)

URL: - <http://localhost8090/ElectroGrid/ComplaintService/Complaints>

Request : -

- {ComplaintId: "Auto generated integer value"}
 {ComplaintName =:"Kaveen"}
 {Date : "22/04/2022" }
 {ComplaintDetails : "Sudden power failure"}
 {ComplaintCategory : "power Failure"}

Response:-

{ Result = "Inserted successfully" }
{ ComplaintId = "Auto generated integer value"}
{ Error= "Error while inserting"}

API for update a customer which is existing in database (PUT Request)

URL: - <http://localhost8090/ElectroGrid/ComplaintService/Complaints>

Request:

```
- {ComplaintId: "Auto generated integer value"}
{ComplaintName =:"Kaveen"}
{Date : "22/04/2022" }
{ComplaintDetails : "Sudden power failure"}
{ComplaintCategory : "Because heavy rain power failure"}
```

Response :-

```
{ Result = "Updated successfully" }
{ ComplaintId = "Auto generated integer value"
{ Error= "Error while Updating"}
```

API for delete a Customers which is existing in database (DELETE Request)

URL: - <http://localhost8090/ElectroGrid/ComplaintService/Complaints>

Request :- { ComplaintId= " Selected Complaint Id from the service "}

Response :- { Result = "Deleted successfully" }
{Error = "Error while deleting the account" }

Test

TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Complaint	{ComplaintId: "Auto generated integer value"} {ComplaintName =:"Kaveen"} {Date : "22/04/2022"} {ComplaintDetails : "Sudden power failure"} {ComplaintCategory : "power Failure"}	New complaints are added to the database. Show message as "Inserted successfully".	New Complaints are added to the database. Show message as "Inserted successfully"	Pass
2	Update complaint	{ComplaintId: "Auto generated integer value"} {ComplaintName =:"Kaveen"} {Date : "22/04/2022"} {ComplaintDetails : "Sudden power failure"} {ComplaintCategory : "Because heavy rain power failure"}	Update Complaints are added to the database and show message as "Update details successfully"	Updated Complaints added and show message as "Update details successfully"	Pass
3	Delete complaint	<Complaint> <ComplaintId>13</ComplaintId> </Complaint>	Delete Complaint and show message as "Payment details deleted successfully"	Complaint deleted and showed message "Details deleted successfully"	Pass

Testing

