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: Y3S1.WE.IT.02.02_Group211

Table of Contents

1.Introduction

ElectroGrid (EG) is the company who maintains the power grid of the country. They have a system to monitor the power consumption of the users, generate the
Monthly bills and automatically send to the users and accept the online payments from the users.
Our task was to create the online platform by covering the whole scope of the company.We used java JAX-RS Jersey, tomcat and MySQL as our tools to build the platform.

2.Member's Details

IT Number	Name	Web Service	Description of the Web Service
IT20178154	Dilshan P.A.D.S.D	Customer Management Payment Management	Insert, Update, Delete and View Customer Details Insert, Update, Delete and ViewPayment Details
IT20140366	Tharushika Devindi M.K.S	Bill Management Feedback Management	Insert, Update, Delete and ViewBill Details. Insert, Update, Delete and View Feedback Details.
IT20139544	Kapukotuwa S.A.A.H	Power Consumption Management	Insert, Update, Delete and view Power Consumption Details.
IT20139858	Gunasinghe P.S.L	Electricity Board Management	Insert, Update, Delete and View Electricity Board Details.
IT20235024	Abishaalini.S	Employee Management	Insert, Update, Delete and View Employee Details.

Clickable link

https://github.com/SaChInD99/PAF-Group-Project-2022.git

SE Methodology

Agile Methodology is used for the development purposes

The combination of iterative and the incremental models is referred to agile methodology. Due to its adaptiveness the complex projects are very easily to handle. This model will provide chance for the project team to break the task and put into small iteration.

Pros:

- People interaction is frequently happening here, so it very help for people communication.
- Adaption can have any time and any phase

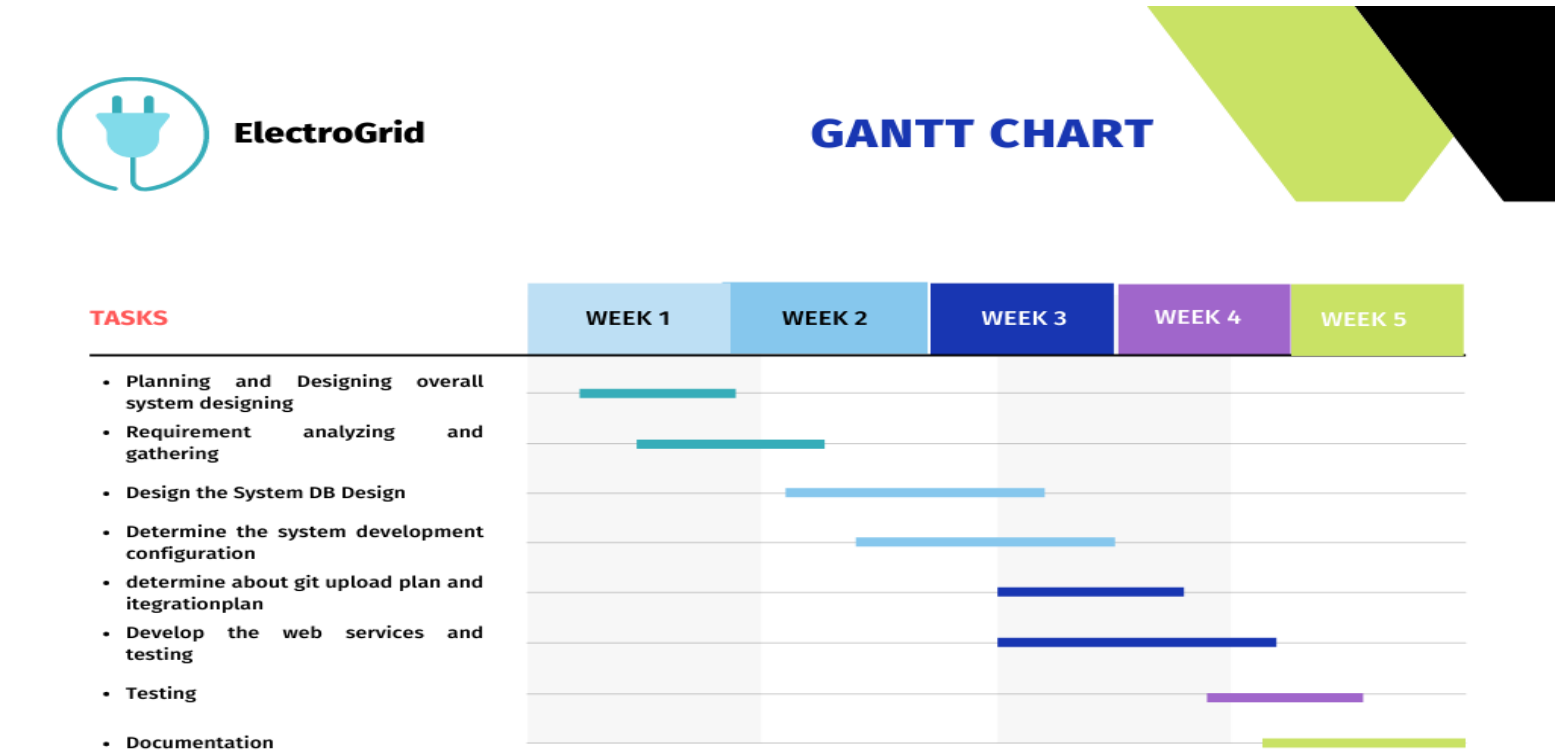
Cons:

- Lack of emphasize designing and documentation.
- Project direction can off track when the discussion gone wrong

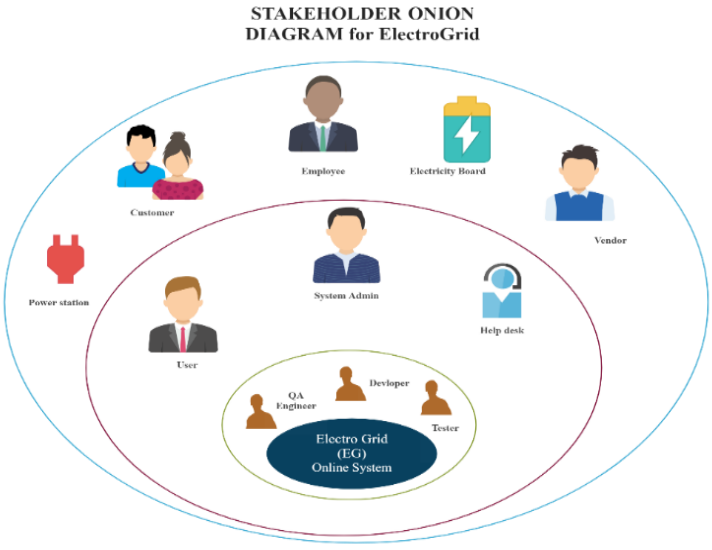
We used the unit testing and integrated testing methodologies for testing purposes.

1. Unit testing
   - We use this for Unit-testing to test the services like individual component or unit. Inevery Unit-testing micro service tested as unit.
2. Integrate testing
   - After Unit-testing all the services are tested as unit. In integration testing the testedmicroservice will tested as one service and the service order will decided.

03.Time schedule (Gantt chart)



System's overall design



- Stakeholder analysis (Onion di agram)

Requirement’s analysis (Functional, Non-functional, Technical requirements)

Functional Requirements

1. Customer Management: Add, Update, Delete and View Customer Details
2. Payment Management: Add, Update, Delete and View Payment Details
3. Bill Management: Add, Update, Delete and View Details
4. Consumption Management: Add, Update, Delete and View details
5. Employee Management: Add, Update, Delete and View Employee Details
6. Feedback Management: Add, Update, Delete and View Feedback Management Details
7. Electricity Board Management: Add, Update, Delete and View Electricity Management Details

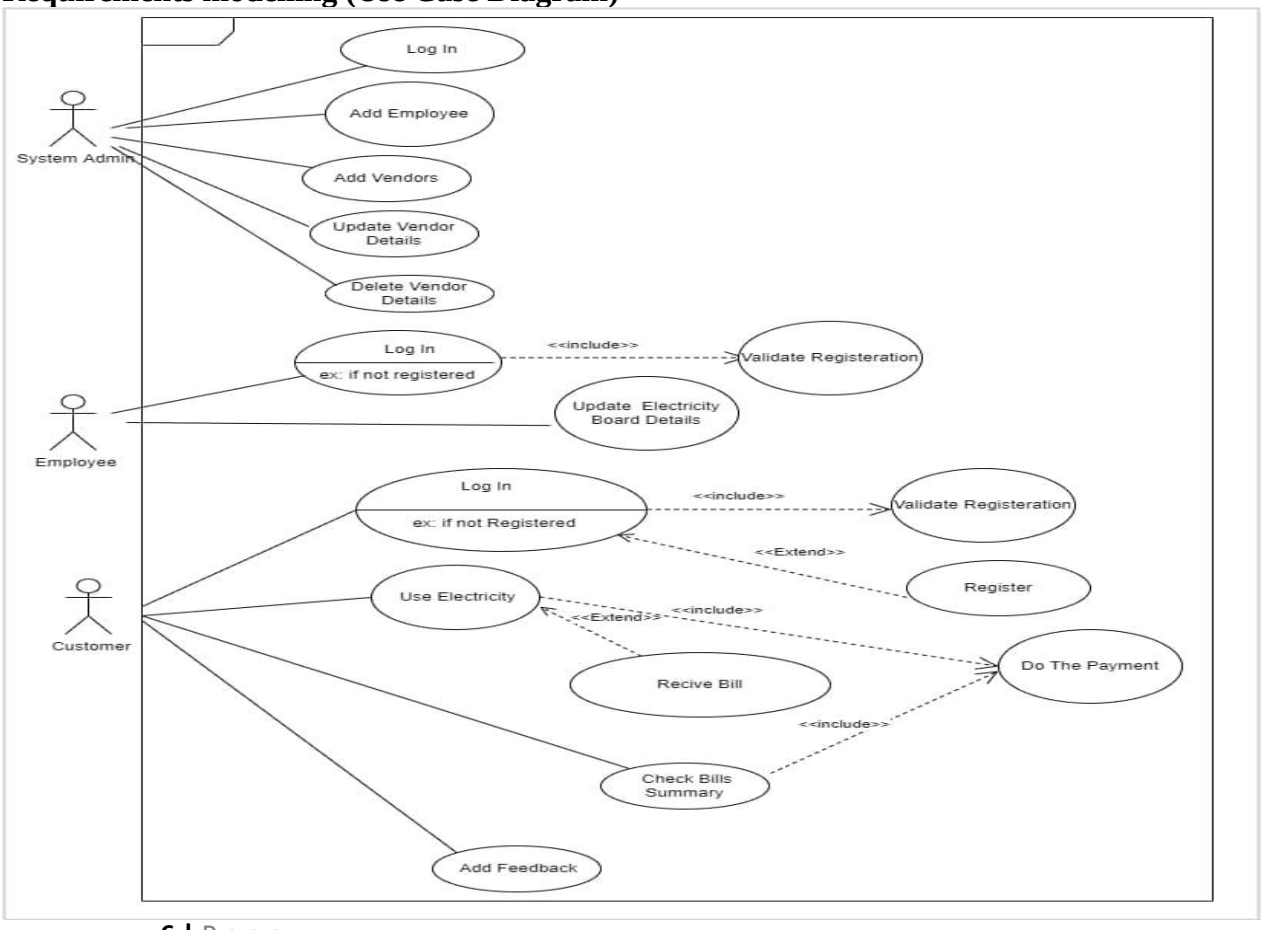
Nonfunctional Requirements

1. Quality, Efficiency, Maintainability, Reliability, Privacy, Accessibility, Compatibility, Extensibility

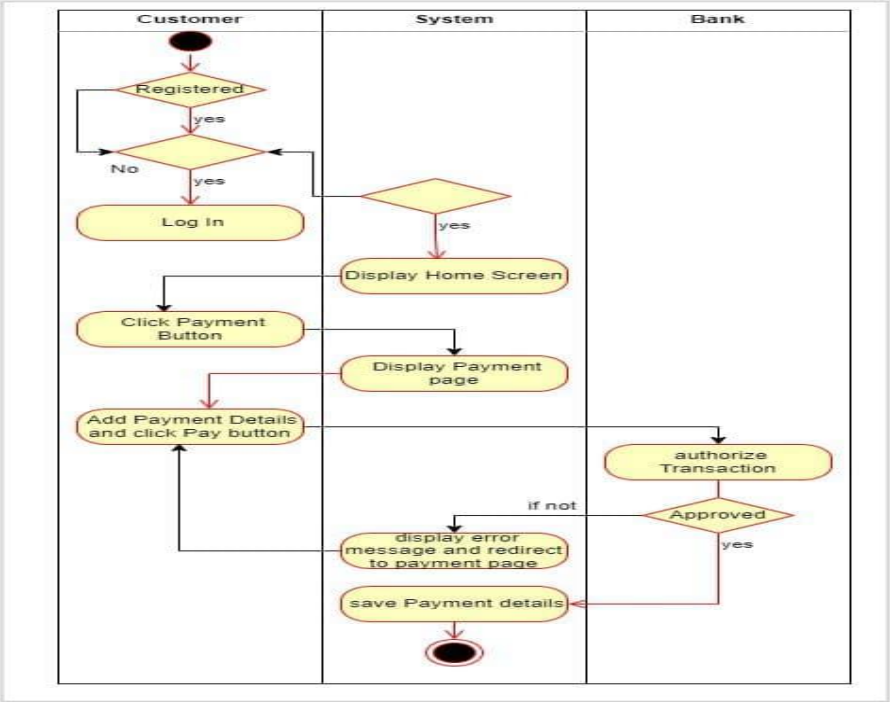
Technical Requirements

The backend is developed using Restful web services **JAX-RS, Jersey, Java on Tomcat** via Eclipse IDE in a windows operating system. My SQL is used in creating database and Postman is an API Clientwhich uses for testing.

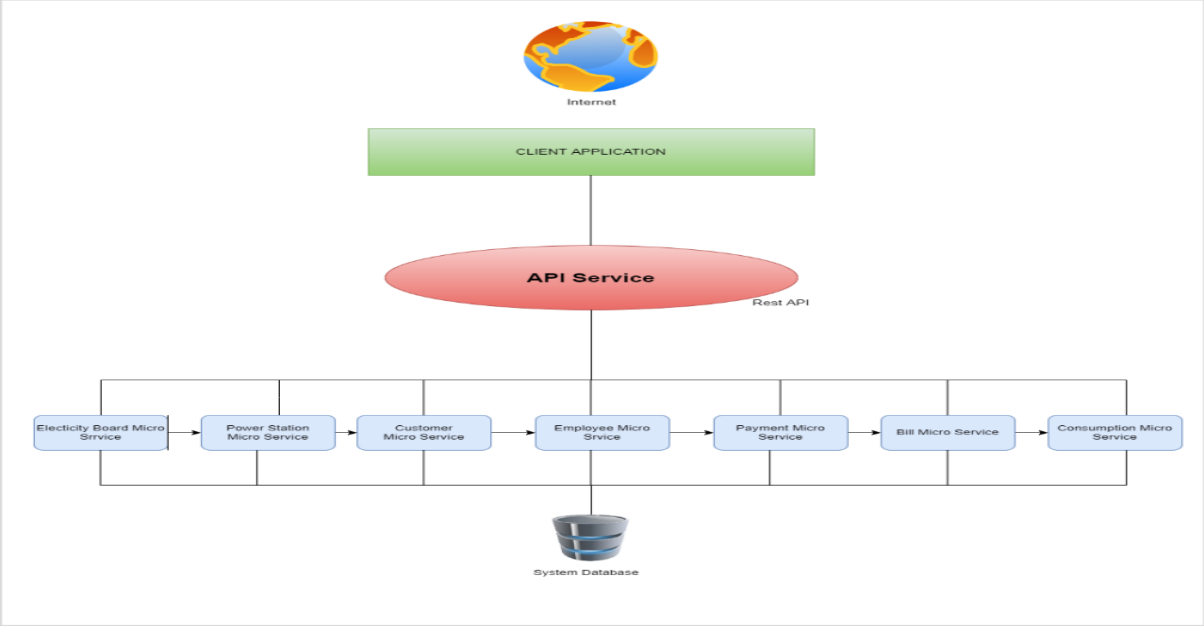
Requirements modelling (Use Case Diagram)



Activity Diagram

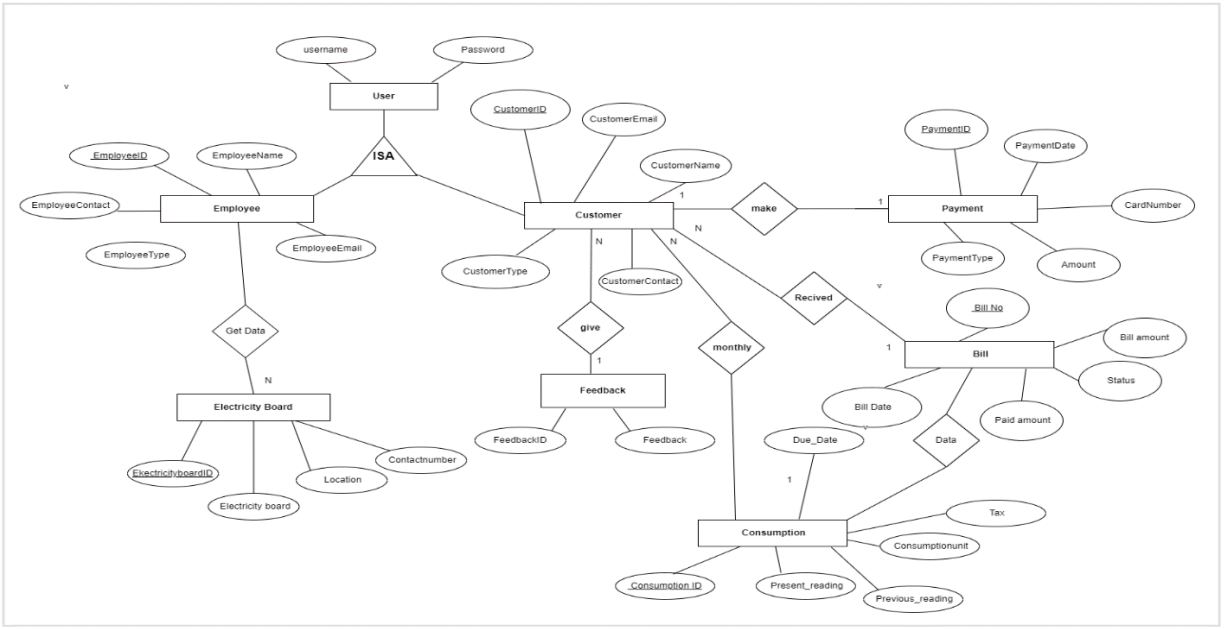


Overall Architecture

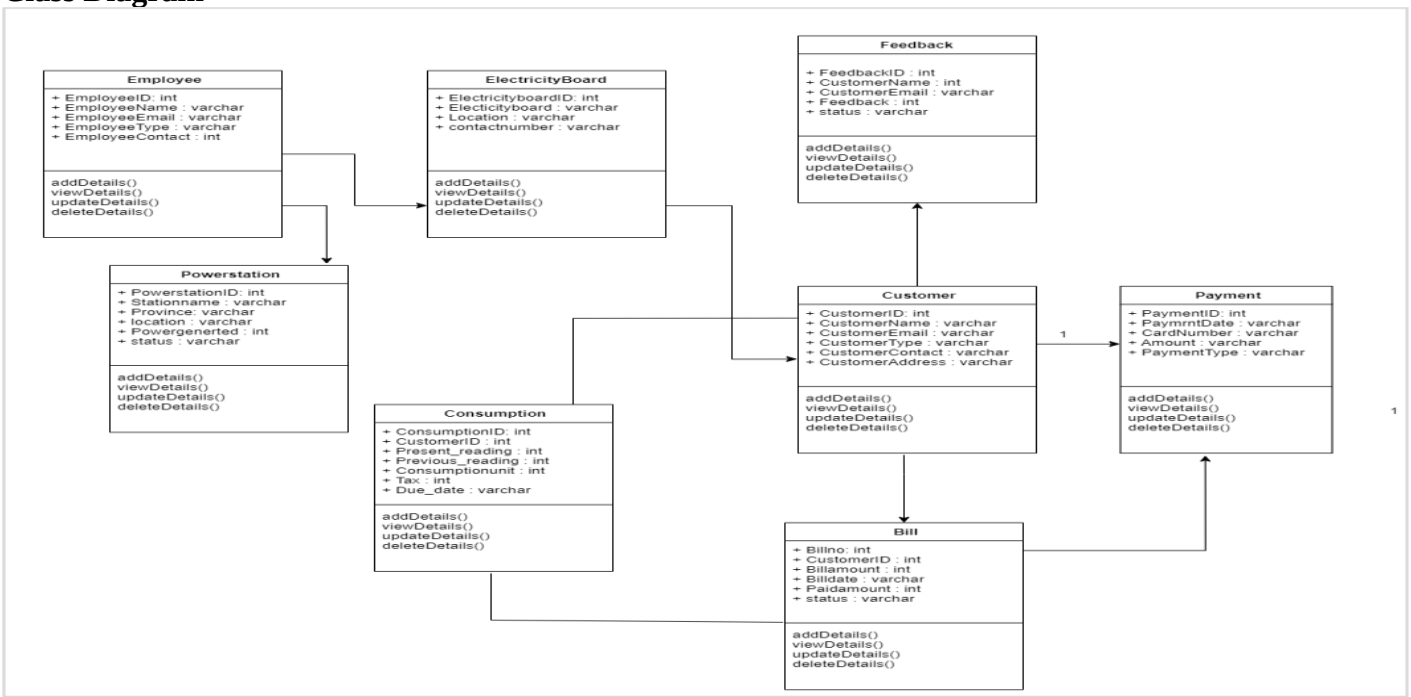


Our task is to create an online platform for a company which maintains the power grid of the country. They have a system to monitor the power consumption of the users, generate the Monthly bills and automatically send to the users and accept the online payments from the users. The number of stakeholders of EG is growing and the current system is not scalable and they asked to built a highly scalable platform

ER Diagram



Class Diagram

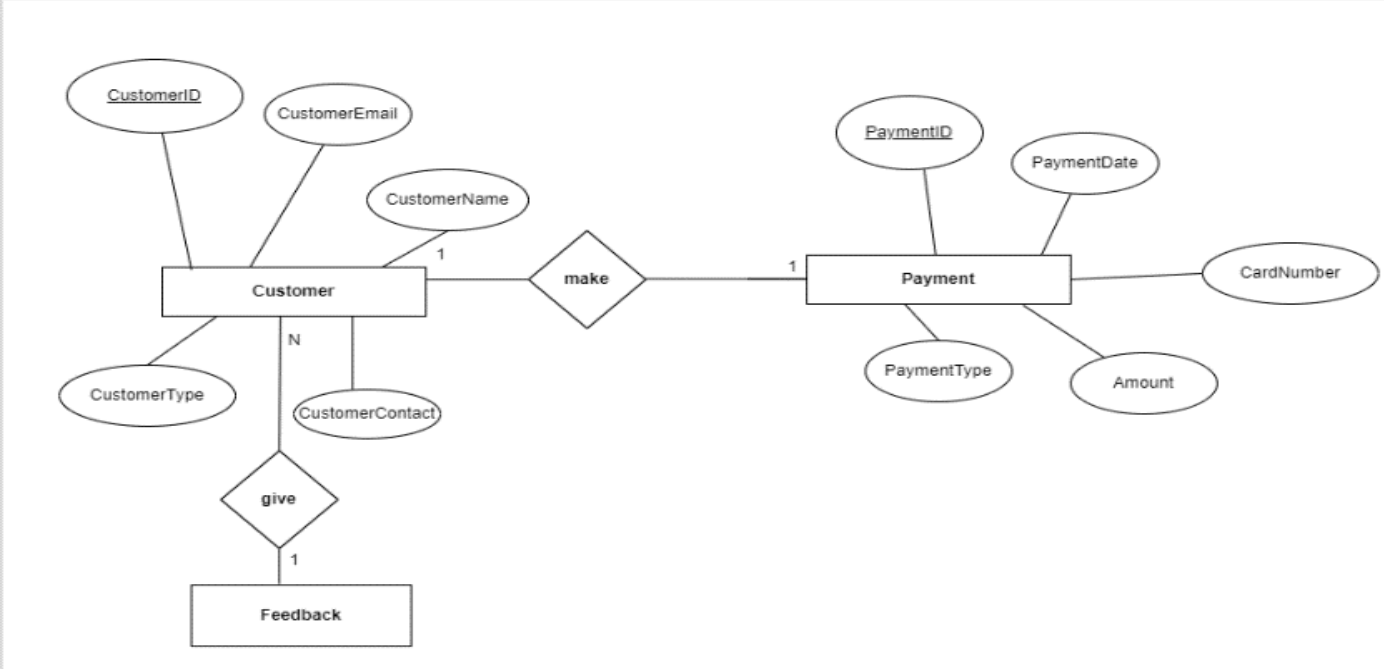


IT20178154(Dilshan P.A.D.S.D)
URL http://localhost:8085/ElectroGrid/CustomerService/Customers
http://localhost:8085/ElectroGrid/PaymentService/Payments

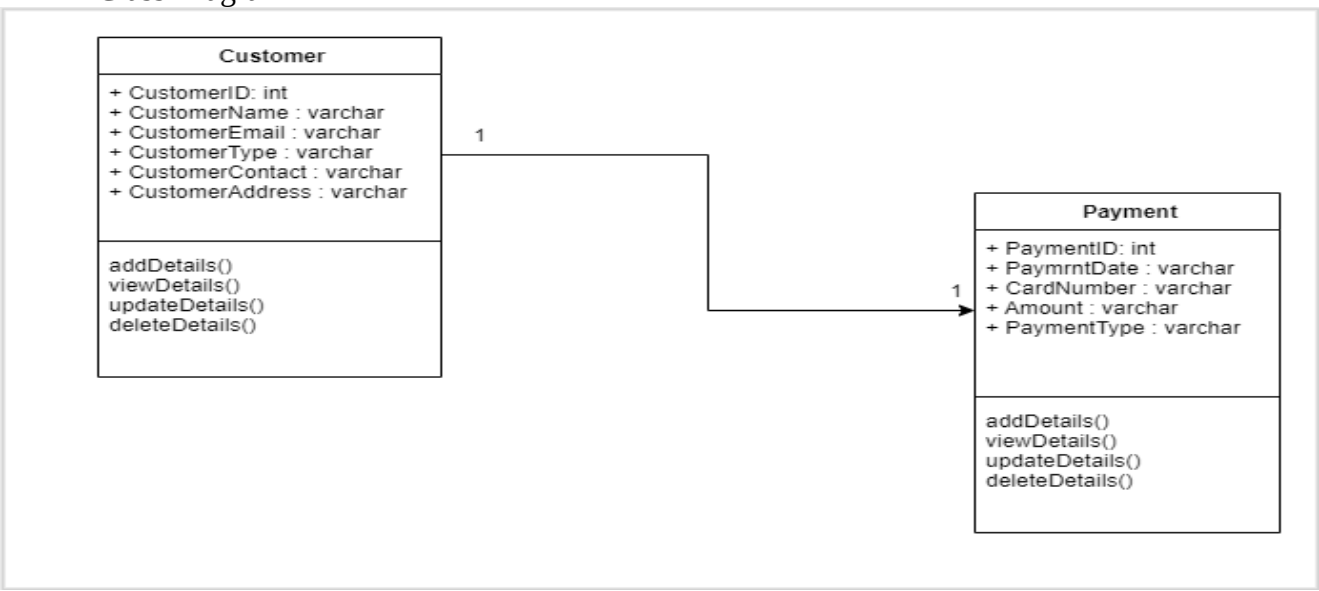
Customer Management and Payment Management

Here is the full scope of the customer and payment management. The requested parties can perform their function successfully through the system. In customer management customer can insert, update delete and view customer details and in payment management the parties can insert, update, delete and view the payment details.

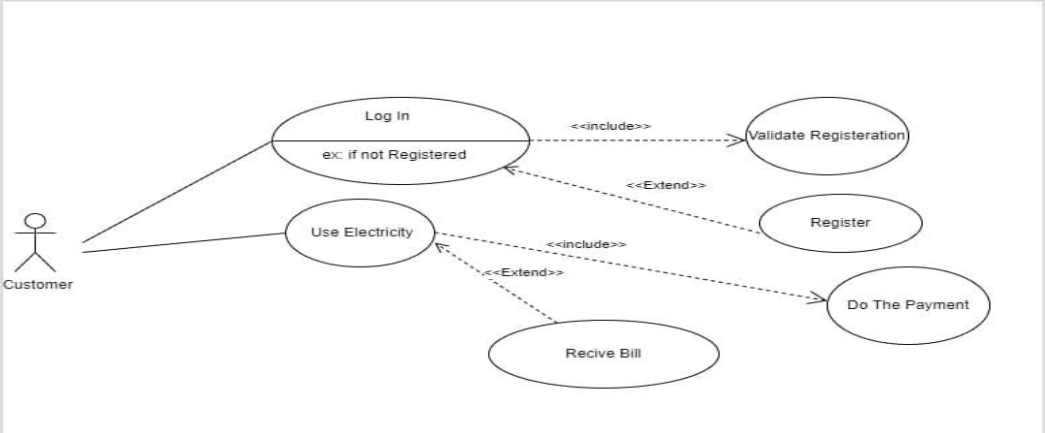
- Er Diagram



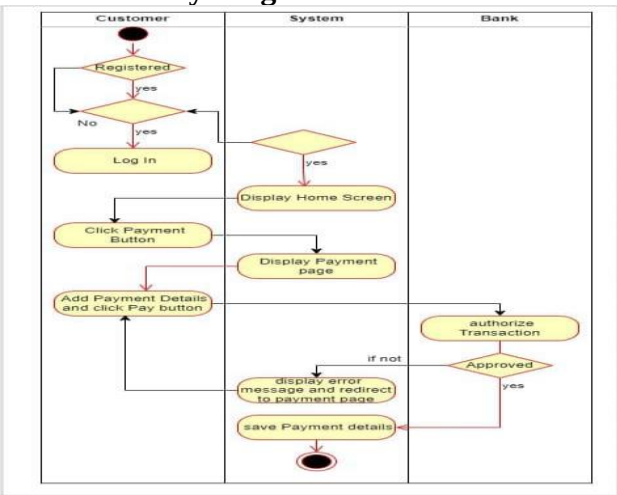
- Class Diagram



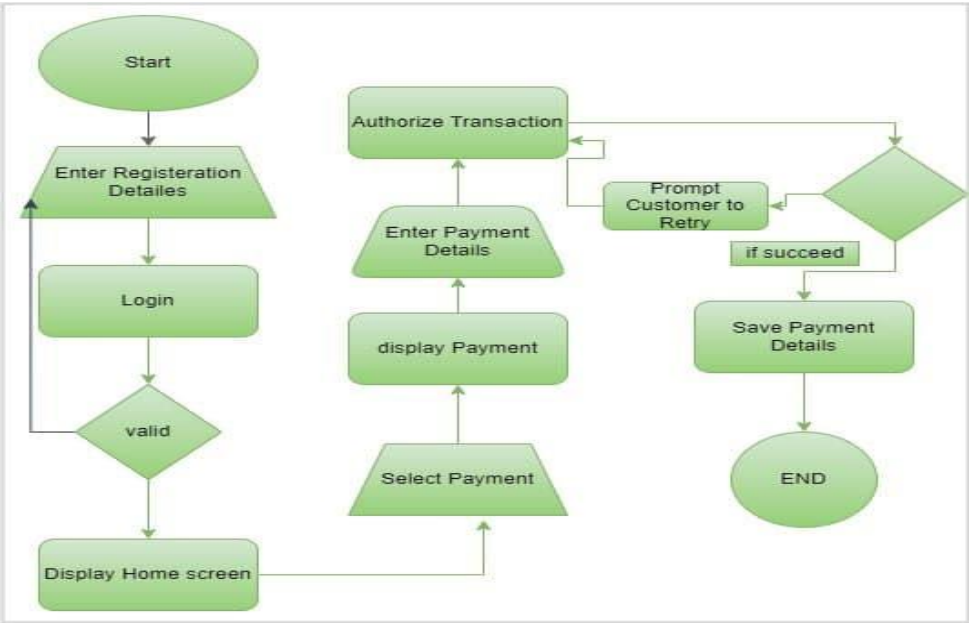
- **Usecase Diagram**



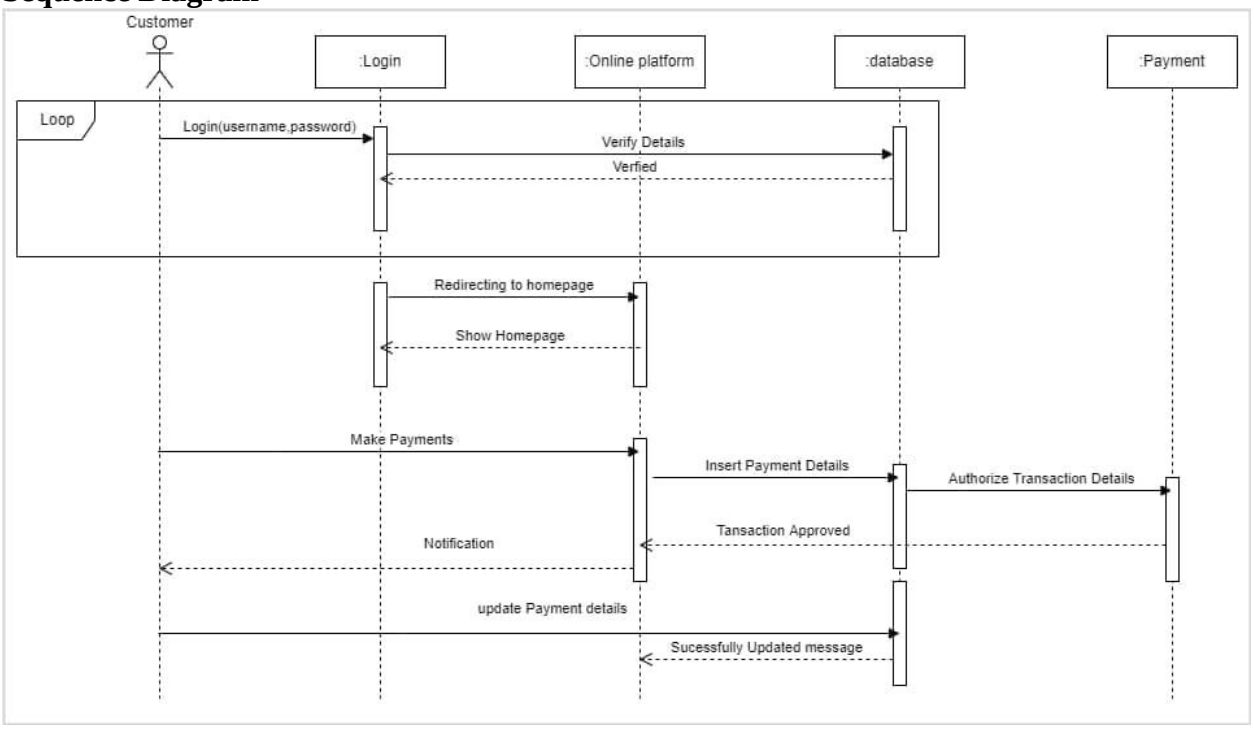
Activity Diagram



- **Flow Chart**

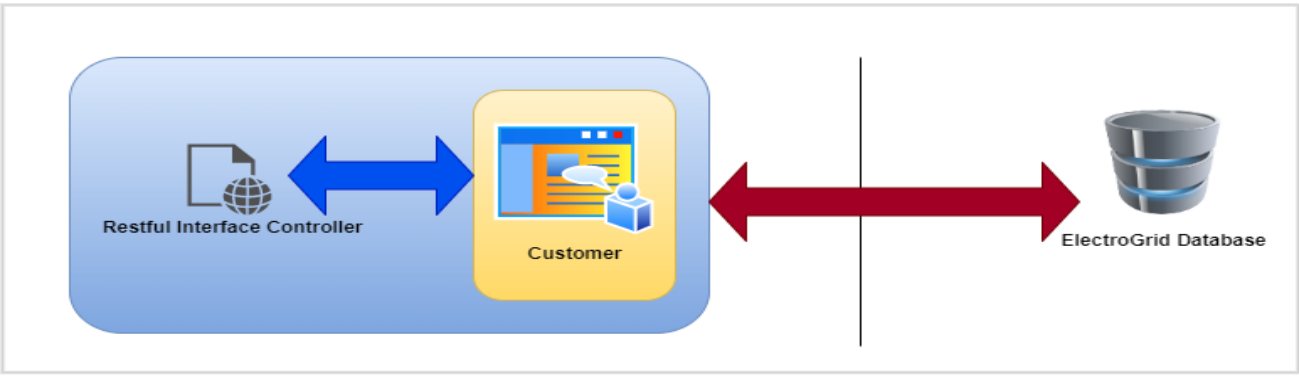


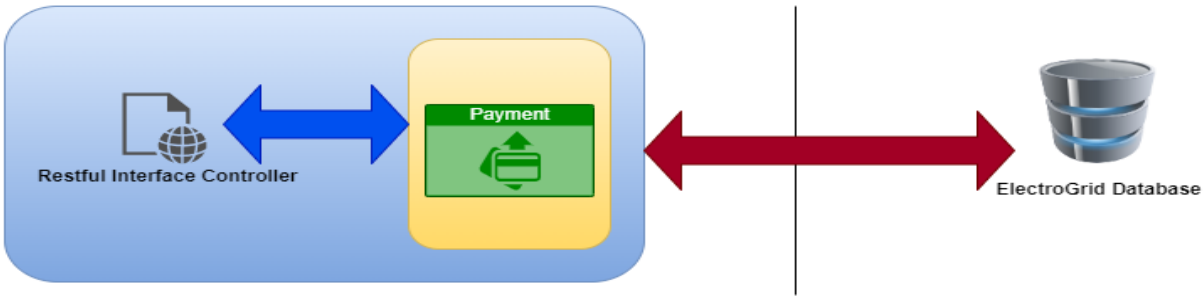
Sequence Diagram



Internal workflow

Customer and Payment





API Design Relational

Customer management and payment management is main part of this project. Customer can easily get the bills from the electrogrid system and they can easily get bills and pay their bills from the system. The system is user friendly and low complicated**.**

- **Customer can insert, delete, update, view the customer details.**
- **Payment details can be inserted, viewed, deleted, and updated 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://localhost:8085/ElectroGrid/CustomerService/Customers

Request : - { }

Response : - {CustomerID : "Auto generated integer value"}
{CustomerName =:"sachin"}
{CustomerEmail :sachin@gmail.com }
{CustomerType : Old customer"}
{CustomerContact : "071513456"}
{CustomerAddress:"No 40, Malabe"}

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

URL:- http://localhost:8085/ElectroGrid/CustomerService/Customers

Request : - {CustomerID : "Auto generated integer value"}
{CustomerName =:"sachin d"}
{CustomerEmail : Sachind@gmail.com }
{CustomerType : old customer"}
{CustomerContact : "075123456"}
{CustomerAddress:"No 40, Malabe"}

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

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

URL:- http://localhost:8085/ElectroGrid/CustomerService/Customers

Request : - {CustomerID : "Auto generated integer value"}
{CustomerName =:"sachin d"}
{CustomerEmail : Sachind@gmail.com }
{CustomerType : old customer"}
{CustomerContact : "075123456"}
{CustomerAddress:"No 40, Malabe"}

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://localhost:8085/ElectroGrid/CustomerService/Customers

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

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

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

URL: - http://localhost:8085/ElectroGrid/PaymentService/Payments
Request: - { }
Response: - {PaymentID = "Auto generated integer value"}
{ PaymentDate" = "12.09.2017"}
{CardNumber = "12334335454664"}
{Amount = "130000.00 "}
{PaymentType = "Cash”}

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

URL: - http://localhost:8085/ElectroGrid/PaymentService/Payments

Request: - { PaymentID: "Auto generated integer value"}
{ PaymentDate" = "13.11.2021"}
{CardNumber = "1213123434"}
{Amount = "1234.00 "}
{PaymentType = "Credit”}

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

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

URL: - http://localhost:8085/ElectroGrid/PaymentService/Payments

Request : - { PaymentID: "Auto generated integer value"}
{ PaymentDate" = "12.03.2022"}
{CardNumber = "1121233123"}
{Amount = "123450.00 "}
{PaymentType = "Debit Card”}
Response: - {Result = "Updated successfully" }
{ PaymentID = "Auto generated integer value" }
{Error= "Error while Updating "}

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

URL: - http://localhost:8085/ElectroGrid/PaymentService/Payments

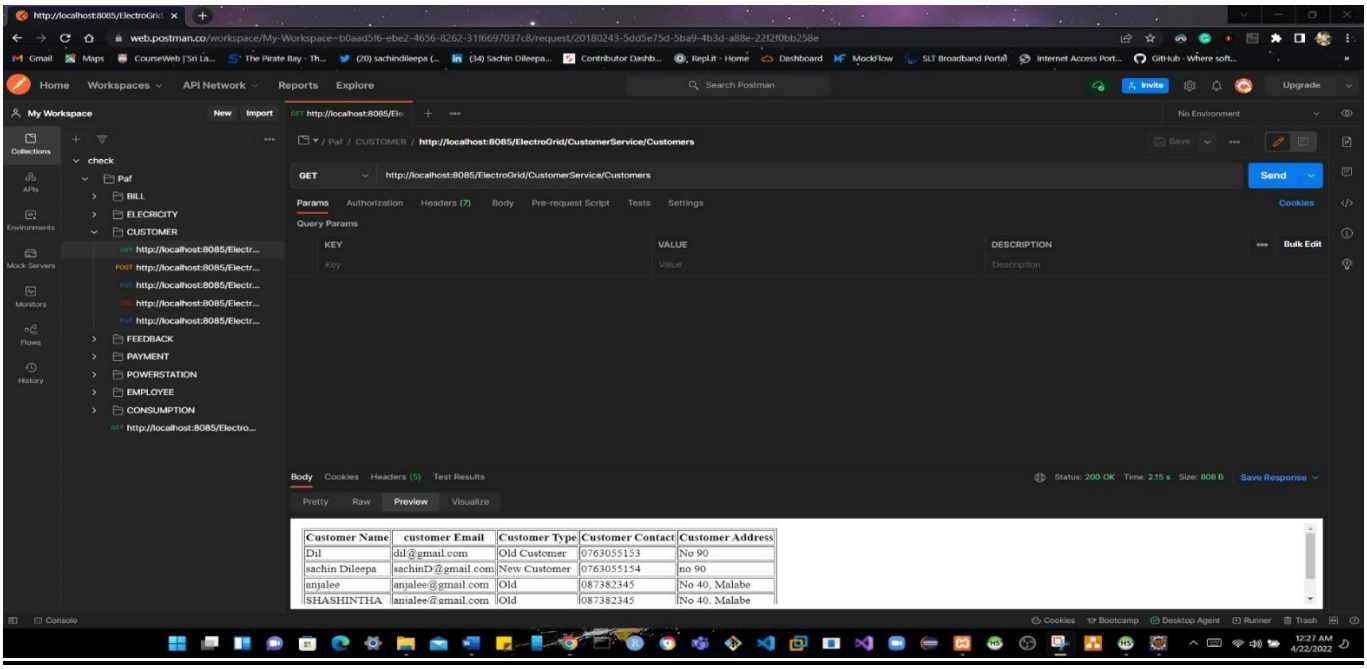
Request :- { PaymentID = " Selected PaymentID from the service "}
Response :- {Result = "Deleted successfully" }
{Error = "Error while deleting the Payment Details" }

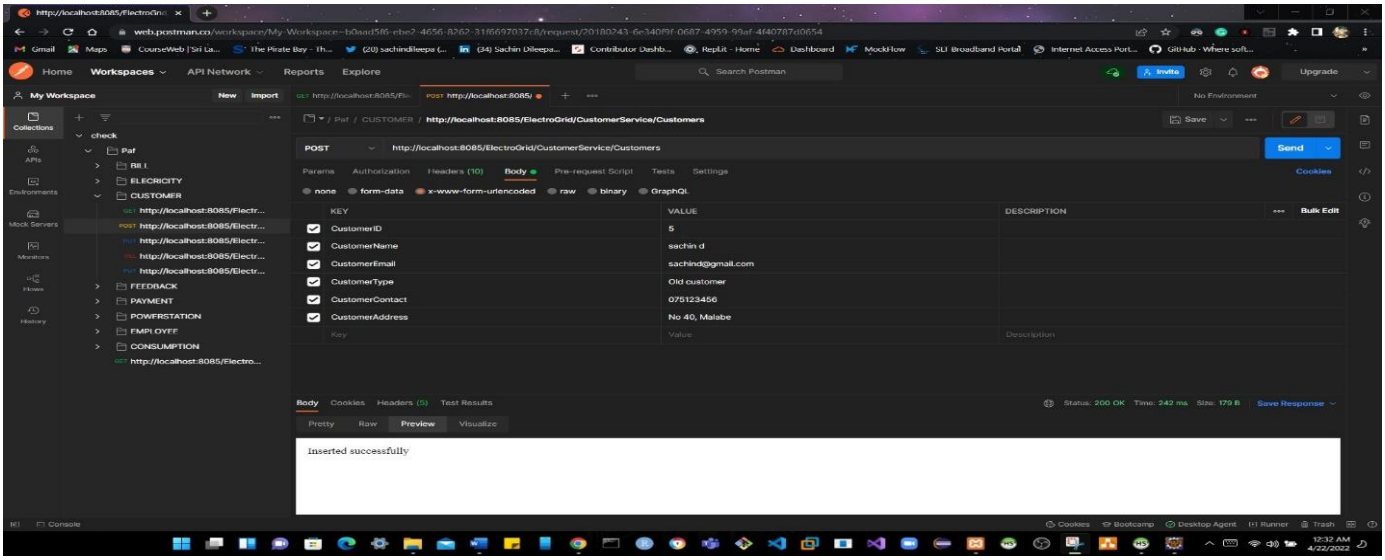
Test Cases TestID	**Test Description/ Test Steps**	Test Input(s)	**Expected Output(s)**	Actual Output(s)	Result (Pass/Fail)
1	Add Customer Details	{CustomerName =:"sachin d"} {CustomerEmail : Sachind@gmail.com } {CustomerType : old customer"} {CustomerContact : "075123456"} {CustomerAddress:"No 40, Malabe"}	New Customer details are added to the database. Show message as "Inserted successfully".	New Customer details are added to the database. Show message as "Inserted successfully".	PASS
2	Update Customer Details	{CustomerName =:"sachin d"} {CustomerEmail : Sachind@gmail.com } {CustomerType : old customer"} {CustomerContact : "075123456"} {CustomerAddress:"No 40, Malabe"}	Customer details are updated according to relevant input Customer details. Show message as "Updated successfully".	Customer details are updated according to relevant input Customer details. Show message as "Updated successfully".	PASS
3	Delete Customer Details	<customerData> <CustomerID>1</CustomerID> </customerData>	Show message as "Deleted Successfully"	Customer Details are deleted successfully. Show message as "Deleted Successfully"	PASS

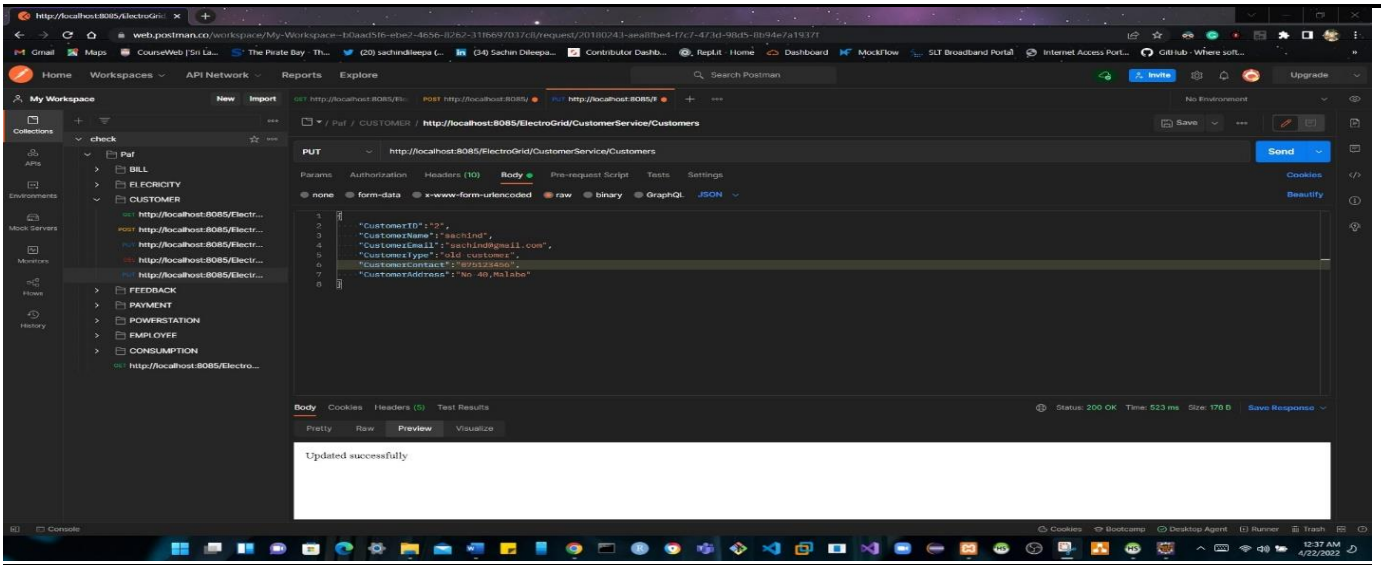
TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Payment Details	{ PaymentDate" = "13.11.2021"} {CardNumber = "1213123434"} {Amount = "1234.00 "} {PaymentType = "Credit"}	New Payment details are added to the database. Show message as "Inserted successfully".	New Payment details are added to the database. Show message as "Inserted successfully".	PASS
2	Update Payment Details	{ PaymentDate" = "12.03.2022"} {CardNumber = "1121233123"} {Amount = "123450.00 "} {PaymentType = "Debit Card"}	Payment details are updated according to relevant input Payment details. Show message as "Updated successfully".	Customer details are updated according to relevant input Payment details. Show message as "Updated successfully".	PASS
3	Delete Payment Details	<paymentData> <PaymentID>1</PaymentID> </paymentData>	Show message as "Deleted Successfully"	Payments Details are deleted successfully. Show message as "Deleted Successfully"	PASS

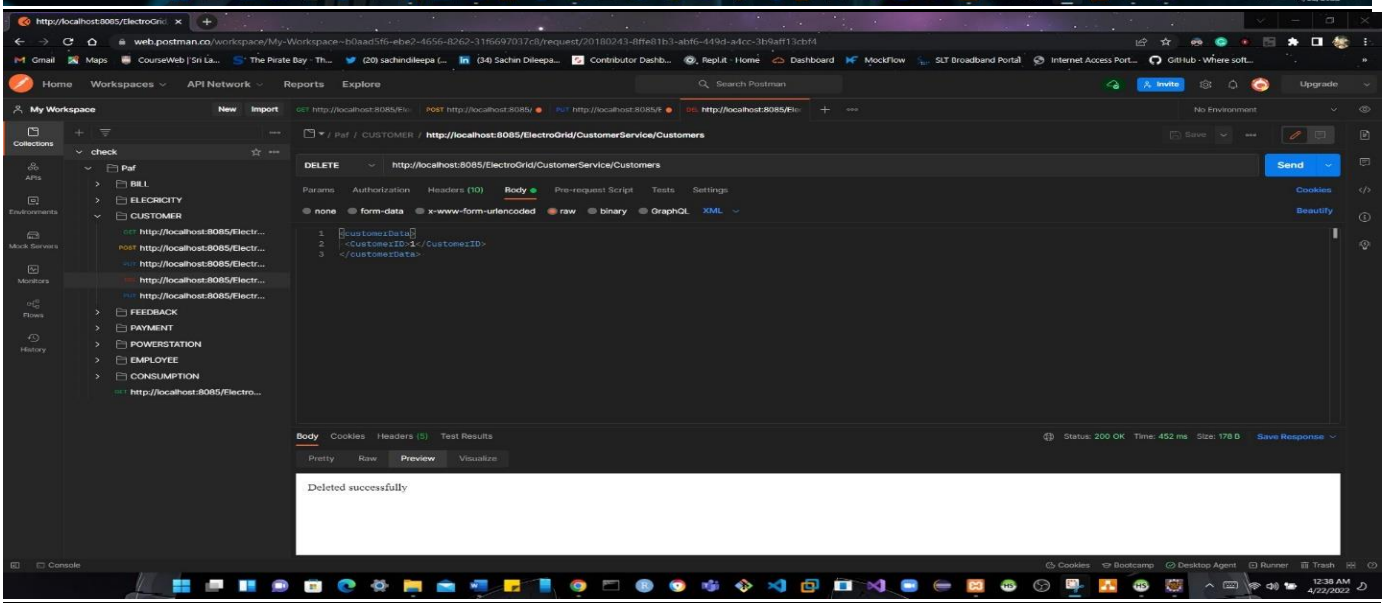
Testing

Customer Management

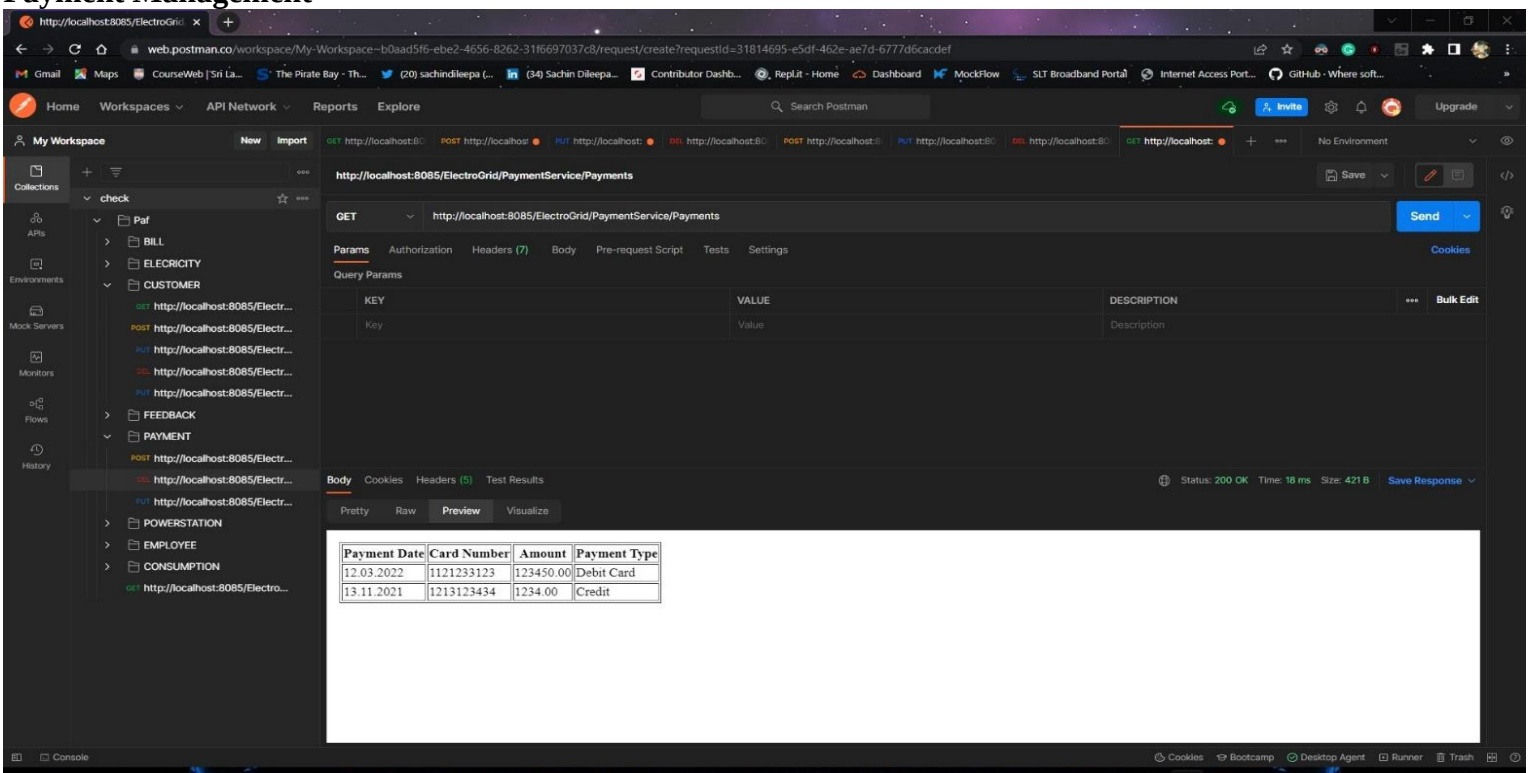


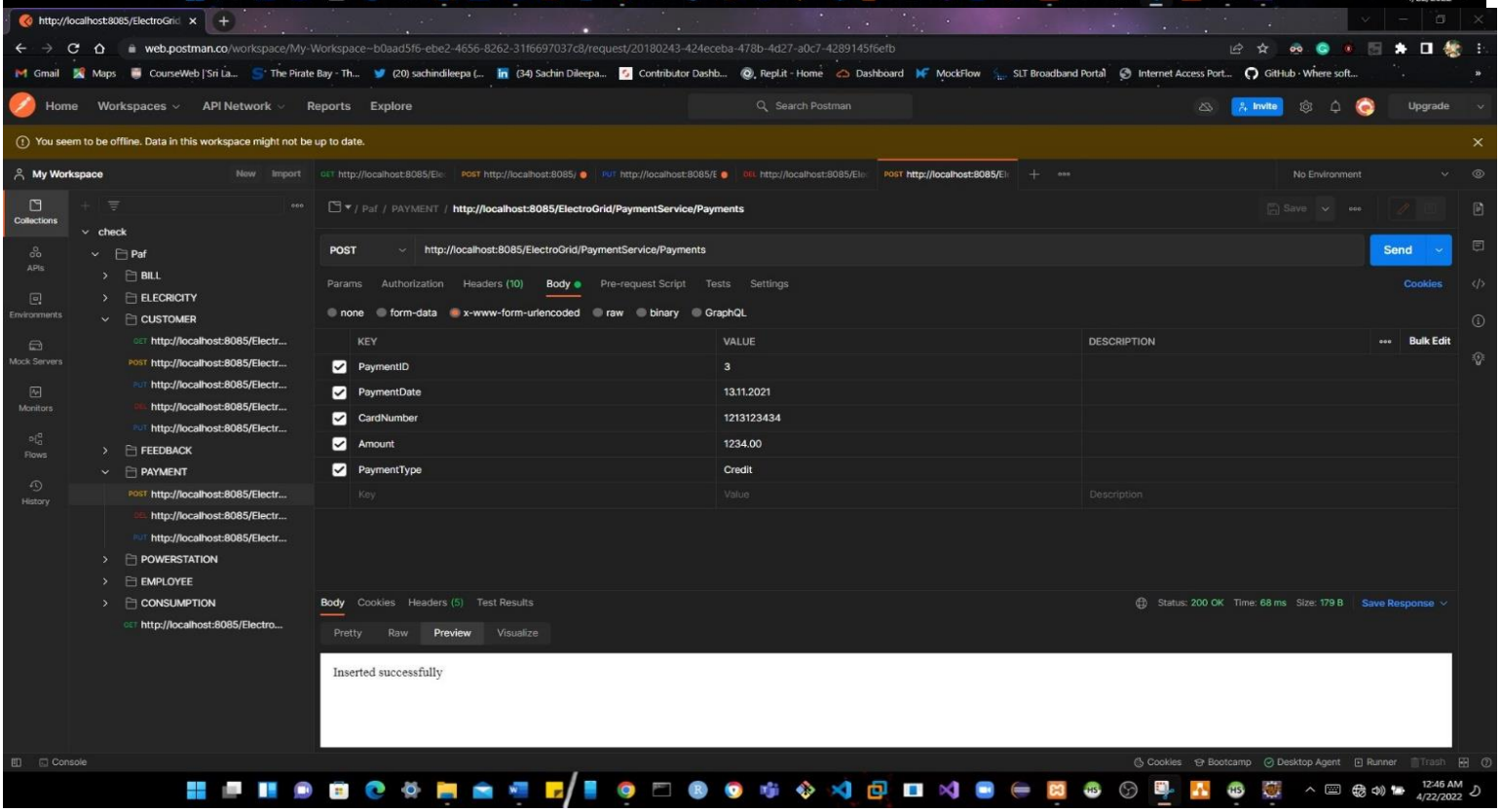

POST
http://localhost:8085/ElectroGrid/CustomerService/Customers
Send
KEY
VALUE
DESCRIPTION
CustomerID
5
CustomerName
sachin d
CustomerEmail
sachind@gmail.com
CustomerType
Old customer
CustomerContact
075123456
CustomerAddress
No 40, Malabe
Inserted successfully

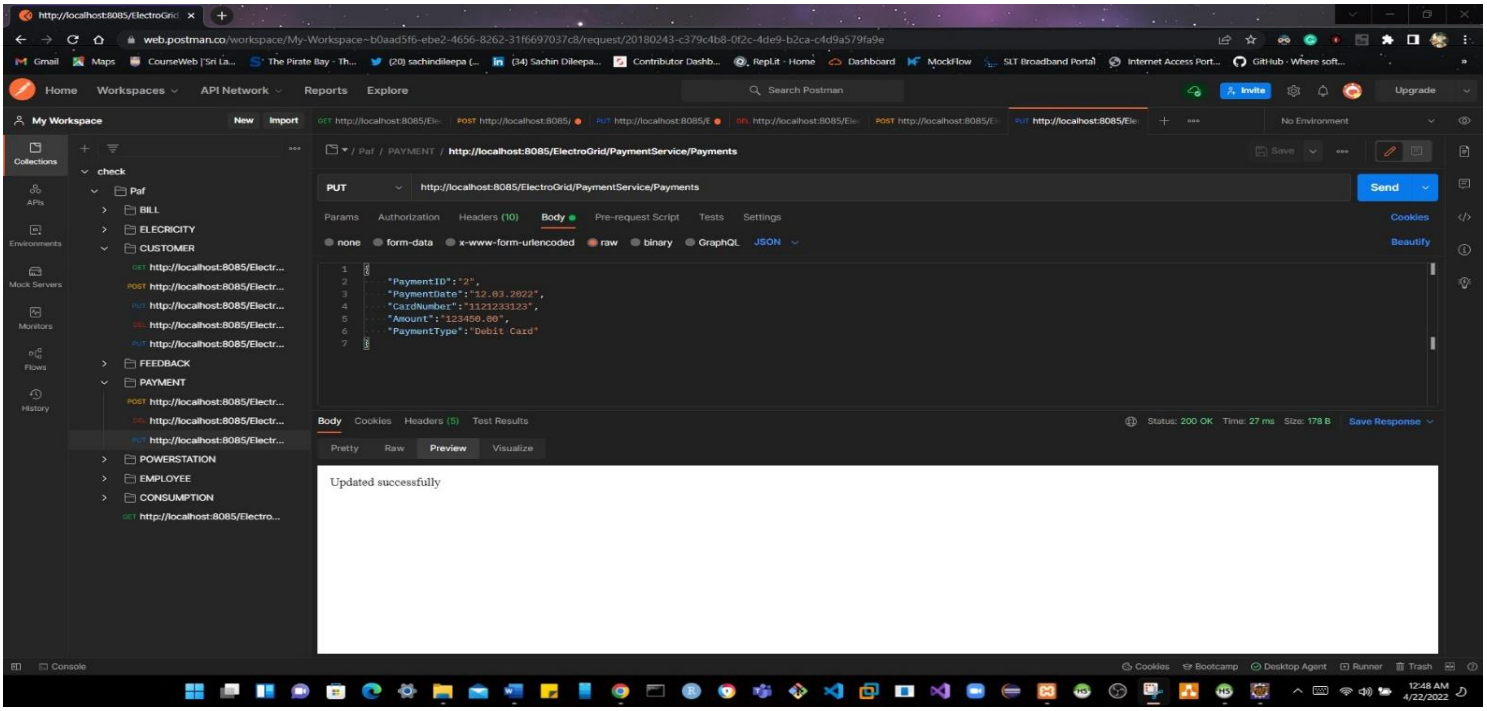

PUT
http://localhost:8085/ElectroGrid/CustomerService/Customers
Send
{
"CustomerID":"2",
"CustomerName":"sachind",
"CustomerEmail":"sachind@gmail.com",
"CustomerType":"old customer",
"CustomerContact":"075123456",
"CustomerAddress":"No 40,Malabe"
}
Updated successfully

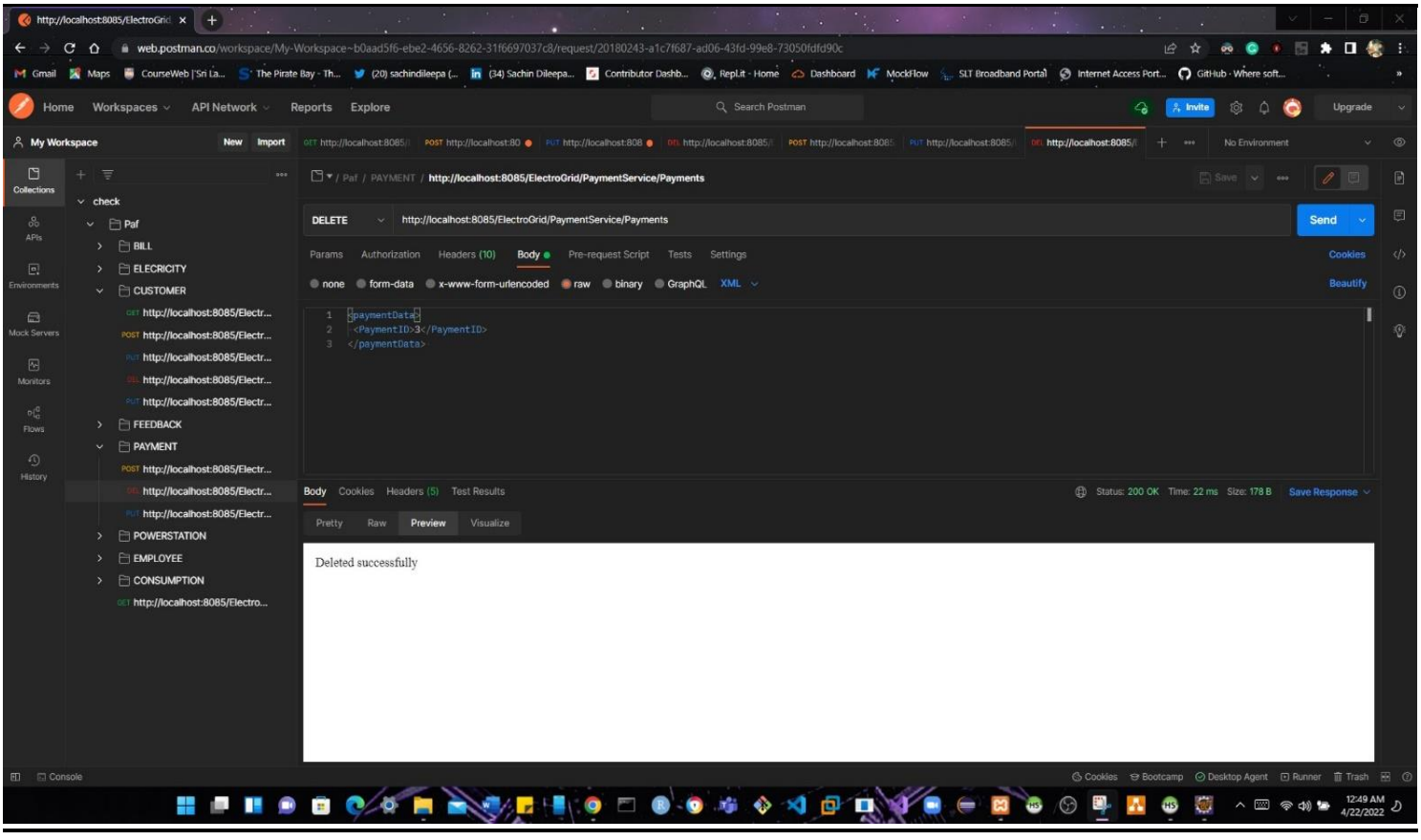

DELETE
http://localhost:8085/ElectroGrid/CustomerService/Customers
Send
<customerData>
<CustomerID>1</CustomerID>
</customerData>
Deleted successfully

Payment Management






PUT
http://localhost:8085/ElectroGrid/PaymentService/Payments
Send
{
"PaymentID":"2",
"PaymentDate":"12.03.2022",
"CardNumber":"1121233123",
"Amount":"123450.00",
"PaymentType":"Debit Card"
}
Status: 200 OK Time: 27 ms Size: 178 B
Updated successfully

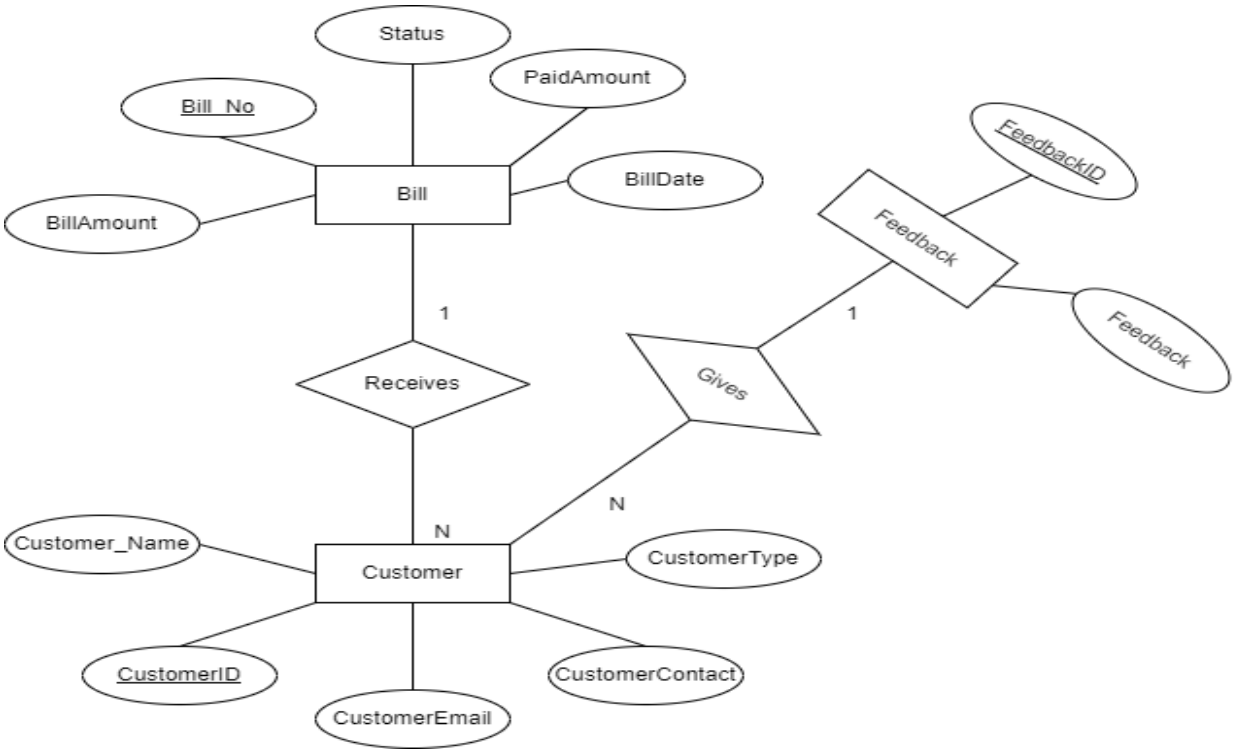

DELETE
http://localhost:8085/ElectroGrid/PaymentService/Payments
Send
<paymentData>
<PaymentID>3</PaymentID>
</paymentData>
Status: 200 OK Time: 22 ms Size: 178 B
Deleted successfully

IT20140366 (Tharushika Devindi M.K.S.)
http://localhost:8085/ElectroGrid/BillService/Bills
http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks

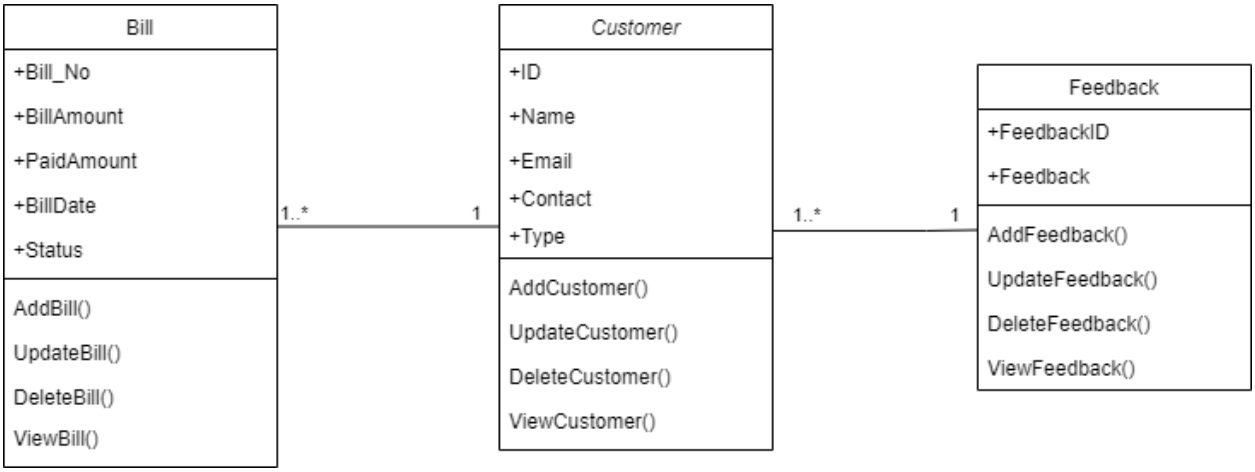
Bill Management and Feedback Management

Energy consumption is the amount of energy used per unit time. ElectroGrid's system calculates how much electricity consumers electricity consume per month and issues bills for it. In addition, those bills are automatically sent to users. Through the system can enter, update, and delete bill details to those bills as well as view those bills.
Customers can provide feedback about the service provided by ElectroGrid, where they can add, update, delete and review feedback.

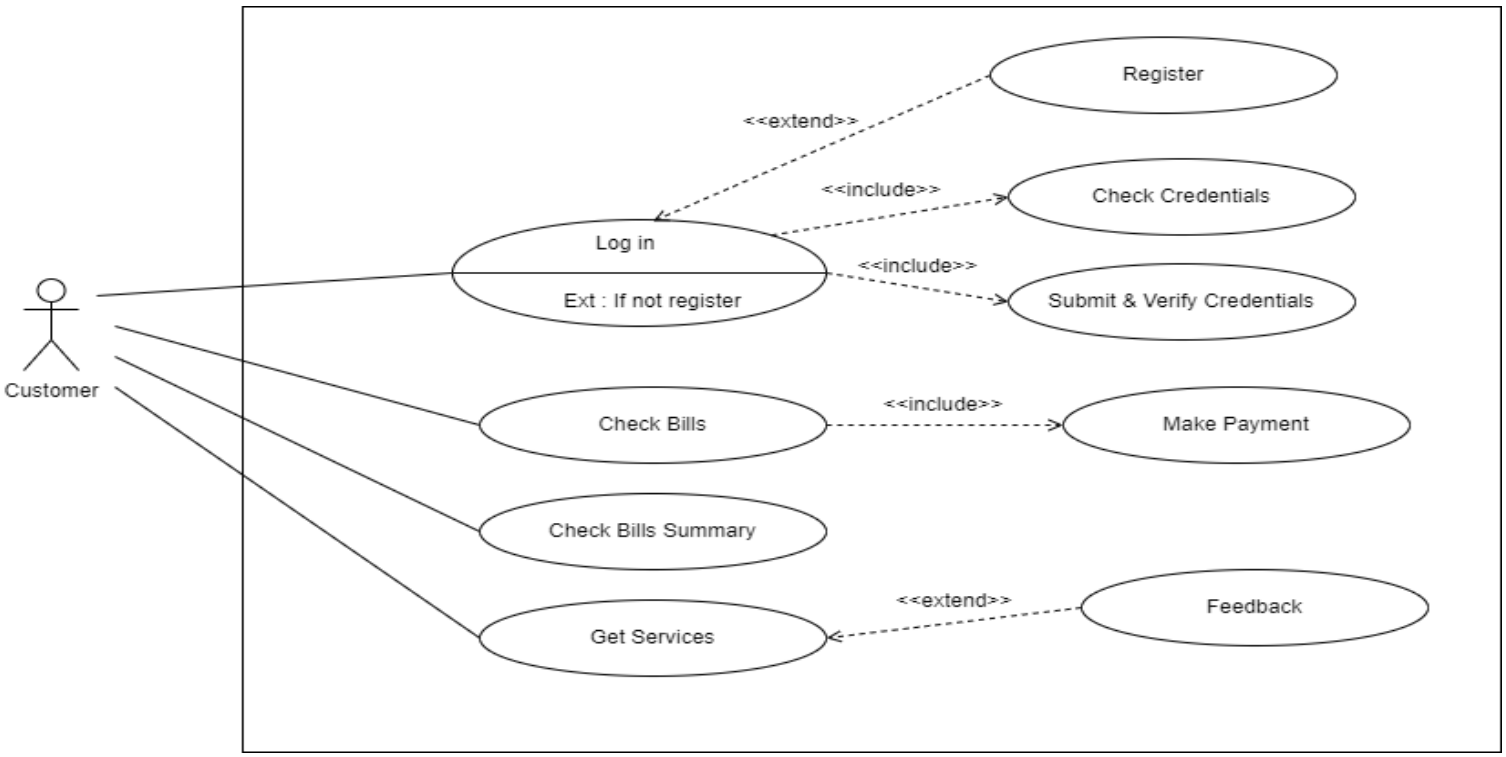
- **ER Diagram**



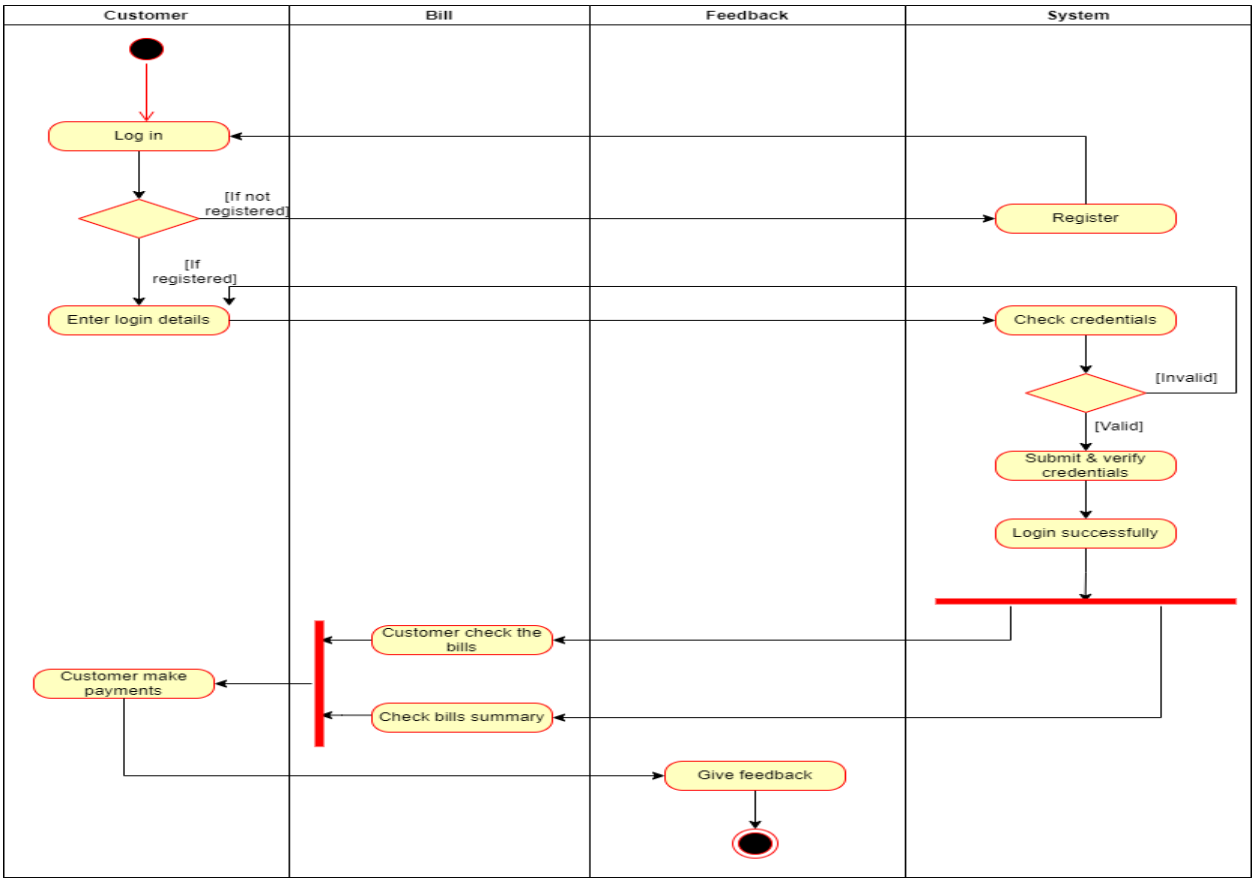
- **Class Diagram**



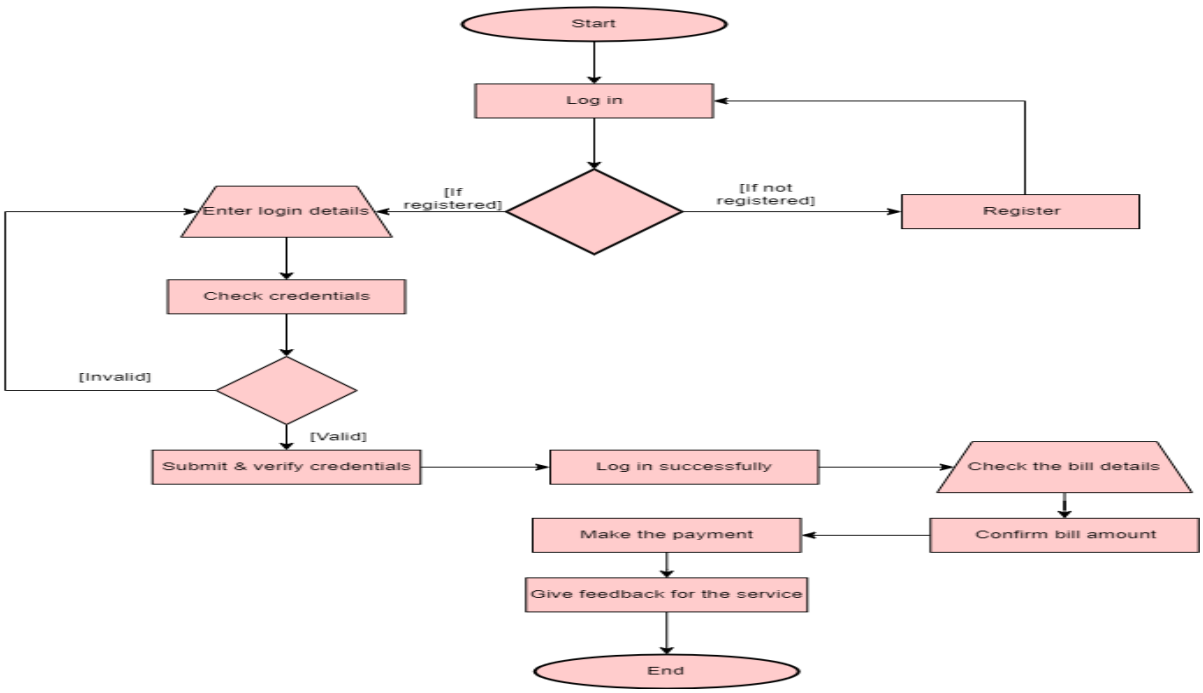
- **Usecase Diagram**



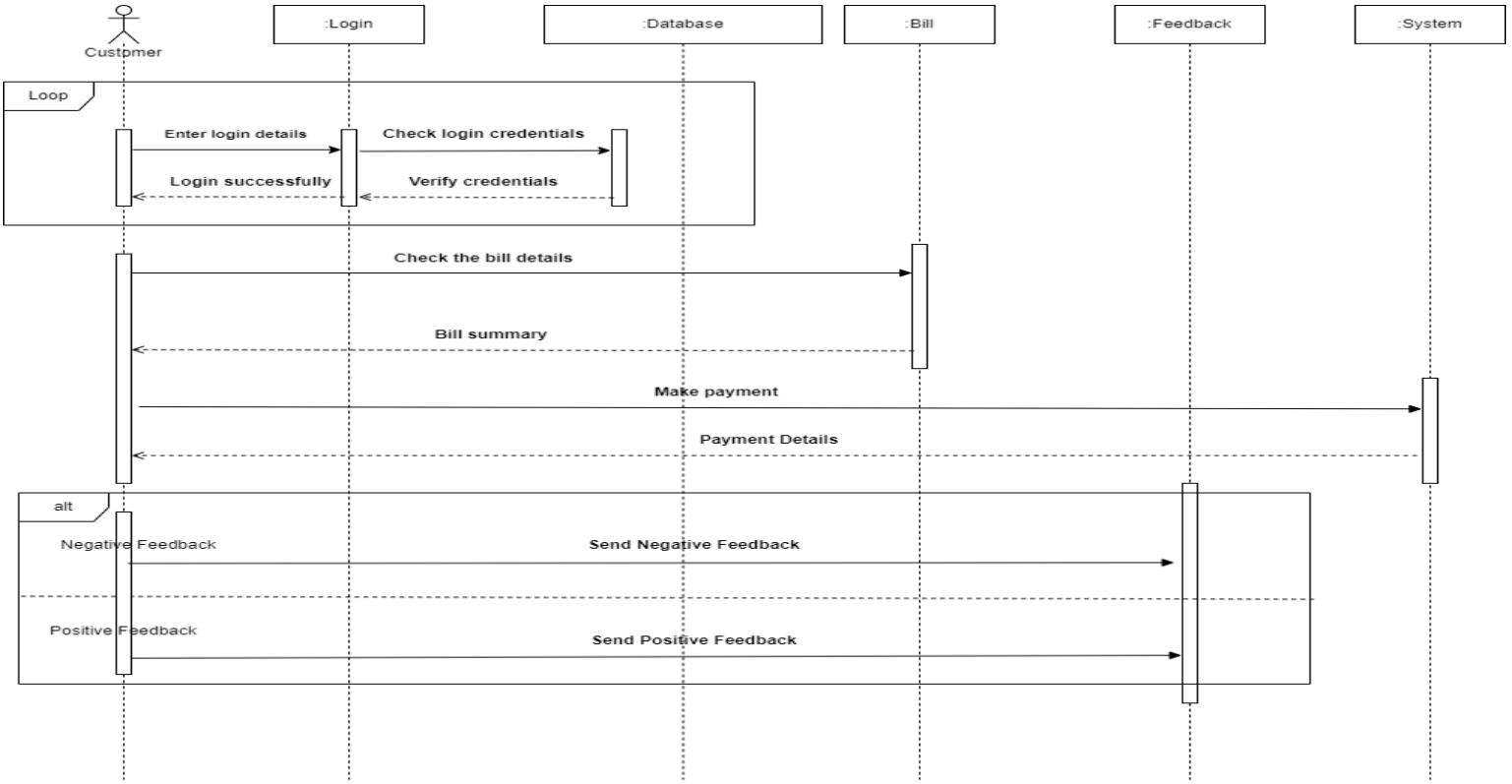
- **Activity Diagram**



- **Flow Chart**

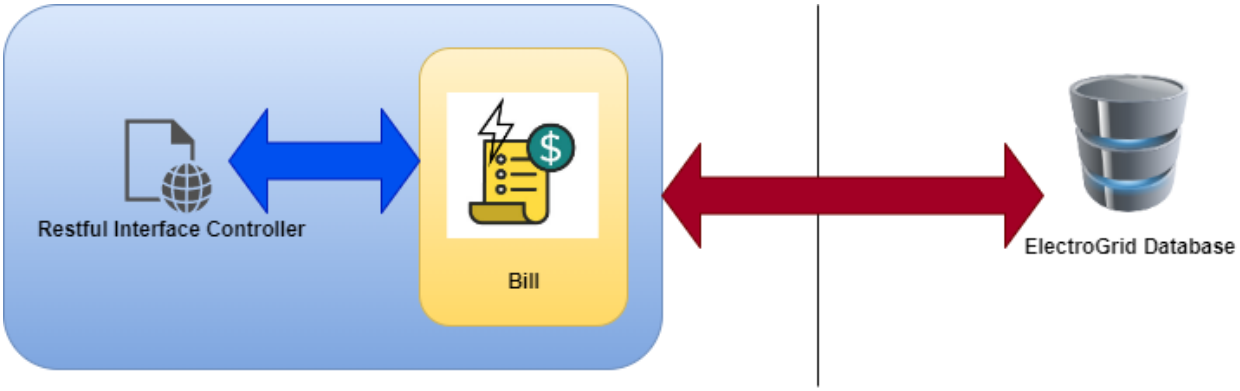


- **Sequence Diagram**

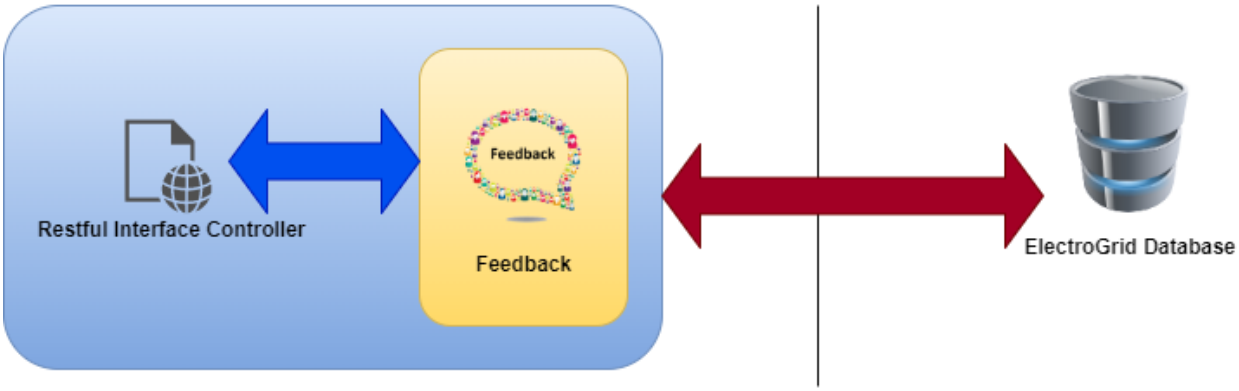


Internal Workflow

Bill:



Feedback:



API Design Relational

ElectroGrid's system calculates how much electricity consumers electricity consume per month and issues bills for it. Customers can provide feedback about the service provided by ElectroGrid. The bill management and feedback management is user friendly and easy to understand with less complication.

- **System admin can enter, update, and delete bill details to those bills as well as view those bills.**
- **Customer can add, update, delete and review feedback.**

Service Development

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

- **Bill Management**

API for get all the Bill Details in the database (GET Request)
URL:- http://localhost:8085/ElectroGrid/BillService/Bills

Request : - { }

Response : - {Billno : "Auto generated integer value"}
{Billamount : 2500.00 }
{Billdate : "12/11/2021"}
{Paidamount : 1500.00 }
{Status : "paid"}

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

URL:- http://localhost:8085/ElectroGrid/BillService/Bills

Request : - {Billno : "Auto generated integer value"}
{Billamount : 2500.00 }
{Billdate : "12/11/2021"}
{Paidamount : 1500.00 }
{Status : " paid "}

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

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

URL:- http://localhost:8085/ElectroGrid/BillService/Bills

Request : - {Billno : "Auto generated integer value"}
{Billamount : 3000.00 }
{Billdate : "10/11/2021"}
{Paidamount : 1500.00 }
{Status : " not paid "}

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

API for delete Bill Details which is existing in database (DELETE Request)

URL:- http://localhost:8085/ElectroGrid/BillService/Bills

Request :- { Billno = " Selected Billno from the service "}

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

- **Feedback Management**

API for get all the Feedback Details in the database (GET Request)

URL:- http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks

Request : - { }

Response : - {FeedbackID : "Auto generated integer value"}
{Feedback : " Good "}

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

URL:- http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks

Request : - {FeedbackID : "Auto generated integer value"}
{Feedback : " Good "}

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

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

URL:- http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks

Request : - {FeedbackID : "Auto generated integer value"}
{Feedback : " bad "}

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

API for delete Feedback Details which is existing in database (DELETE Request)

URL:- http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks

Request :- { FeedbackID = " Selected FeedbackID from the
service "}

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

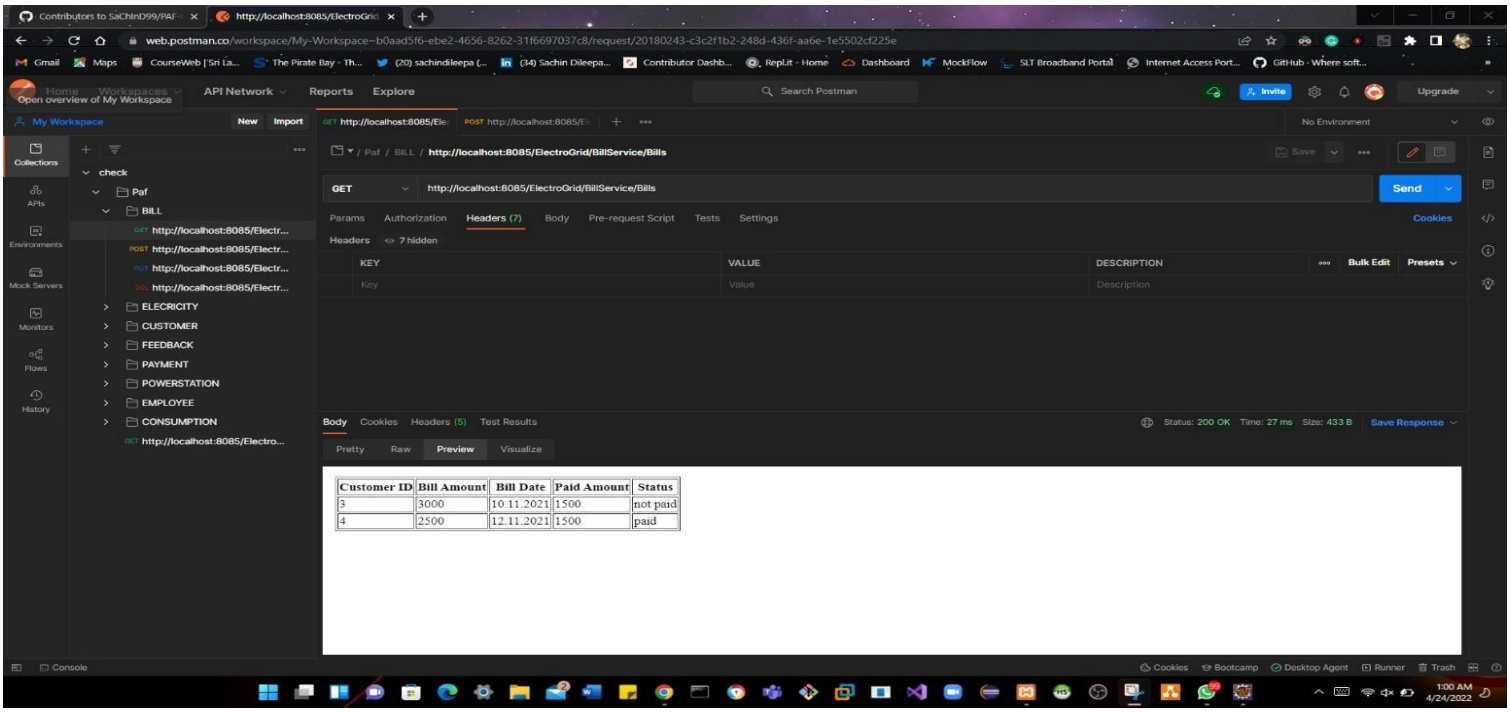
Test Cases TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Bill Details	{BillNo : "Auto generated integer value"} {BillAmount : 960.75 } {PaidAmount : 1000.00 } {BillStatus : "Electricity Bill"} {BillDate : "15/04/2022"}	New Bill details are added to the database. Show message as "Inserted successfully".	New Bill details are added to the database. Show message as “Inserted successfully".	PASS
2	Update Bill Details	{BillNo : "Auto generated integer value"} {BillAmount : 960.75 } {PaidAmount : 1000.00 } {BillStatus : "Internet Bill"} {BillDate : "15/04/2022"}	Bill details are updated according to relevant input Bill details. Show message as "Updated successfully".	Bill details are updated according to relevant input Bill details. Show message as "Updated successfully".	PASS
3	Delete Bill Details	<billData> <BillNo >1</ BillNo > </billData>	Show message as “Deleted Successfully”	Bill Details are deleted successfully. Show message as “Deleted Successfully”	PASS

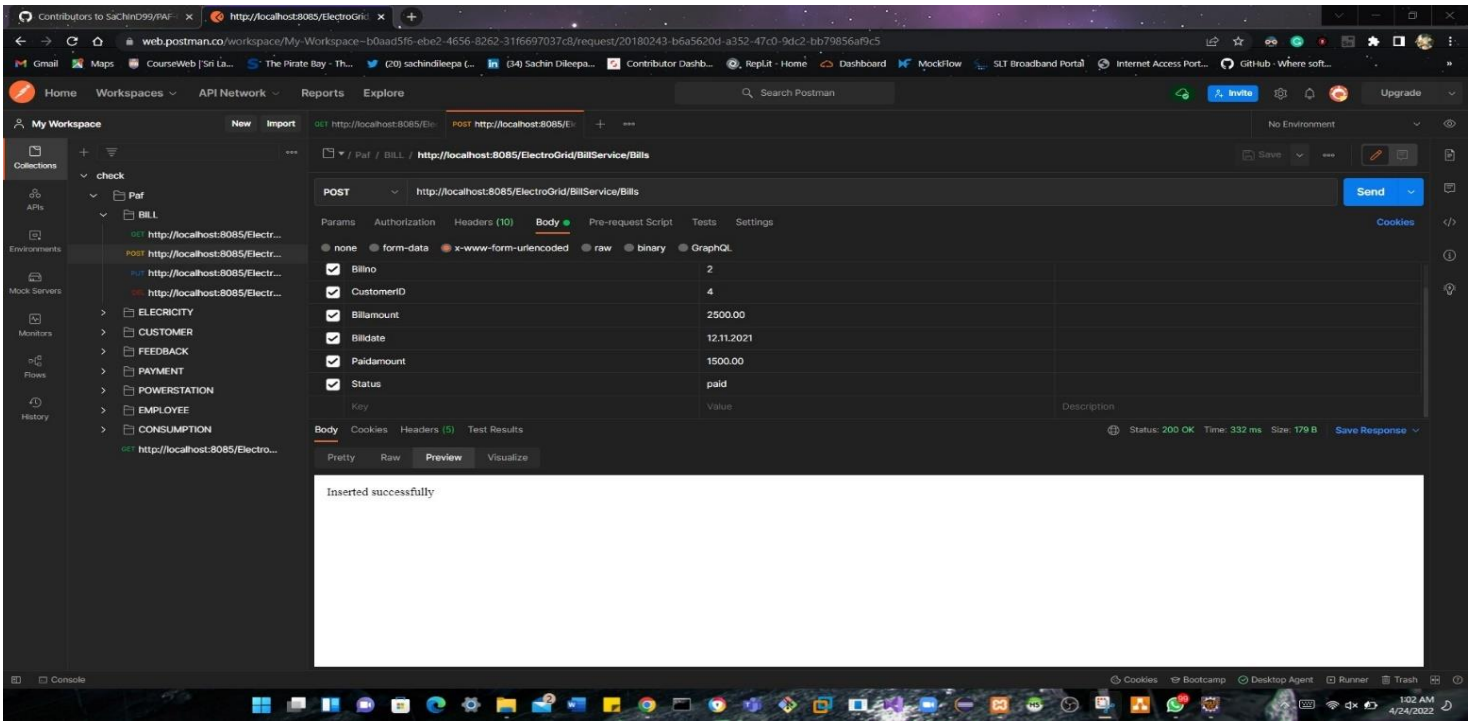
Test Cases TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)

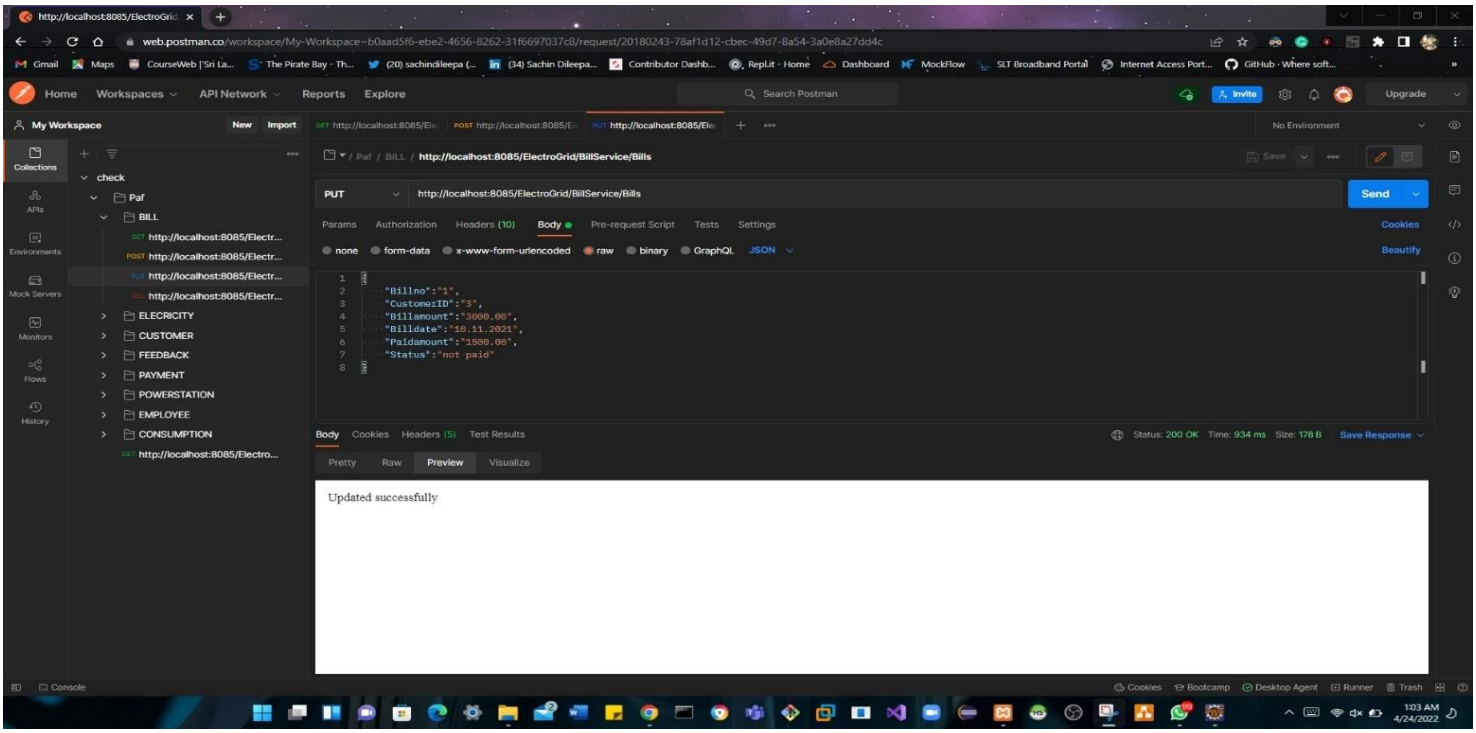
1	Add Feedback Details	{FeedbackID : "Auto generated integer value"} {Feedback : Excellent customer service. Thank you for the amazing support!}	New feedback details are added to the database. Show message as "Inserted Successfully".	New feedback details are added to the database. Show message as “Inserted Successfully".	PASS
2	Update Feedback Details	{FeedbackID : "Auto generated integer value"} {Feedback : Excellent customer service at ElectroGrid! Thank you for the amazing support!}	Feedback details are updated according to relevant input Bill details. Show message as "Updated successfully".	Feedback details are updated according to relevant input Bill details. Show message as "Updated successfully".	PASS
3	Delete Bill Details	< feedbackData> < FeedbackID >1</ FeedbackID > </ feedbackData>	Show message as “Deleted Successfully”	Feedback Details are deleted successfully. Show message as “Deleted Successfully”	PASS

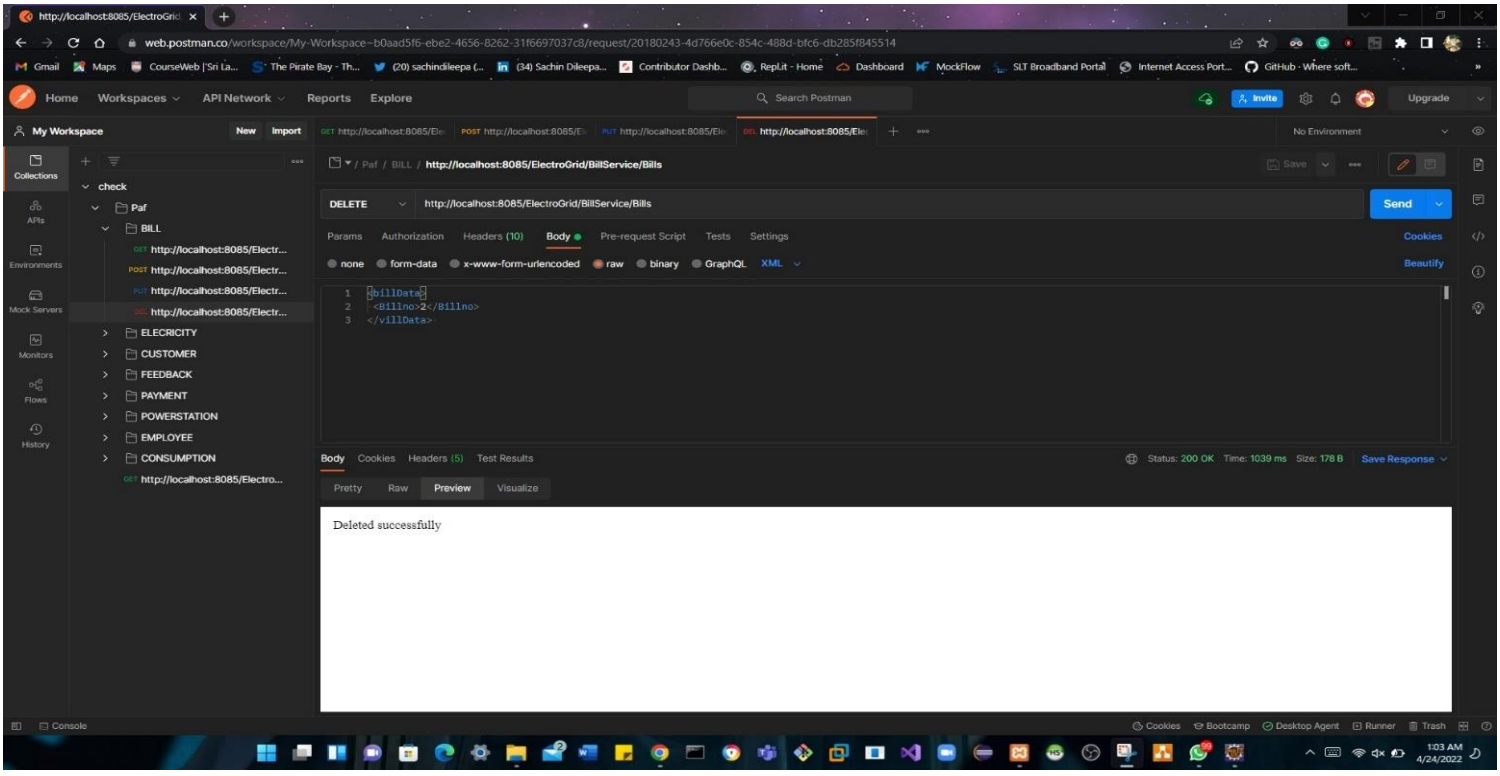
Testing

Bill Management

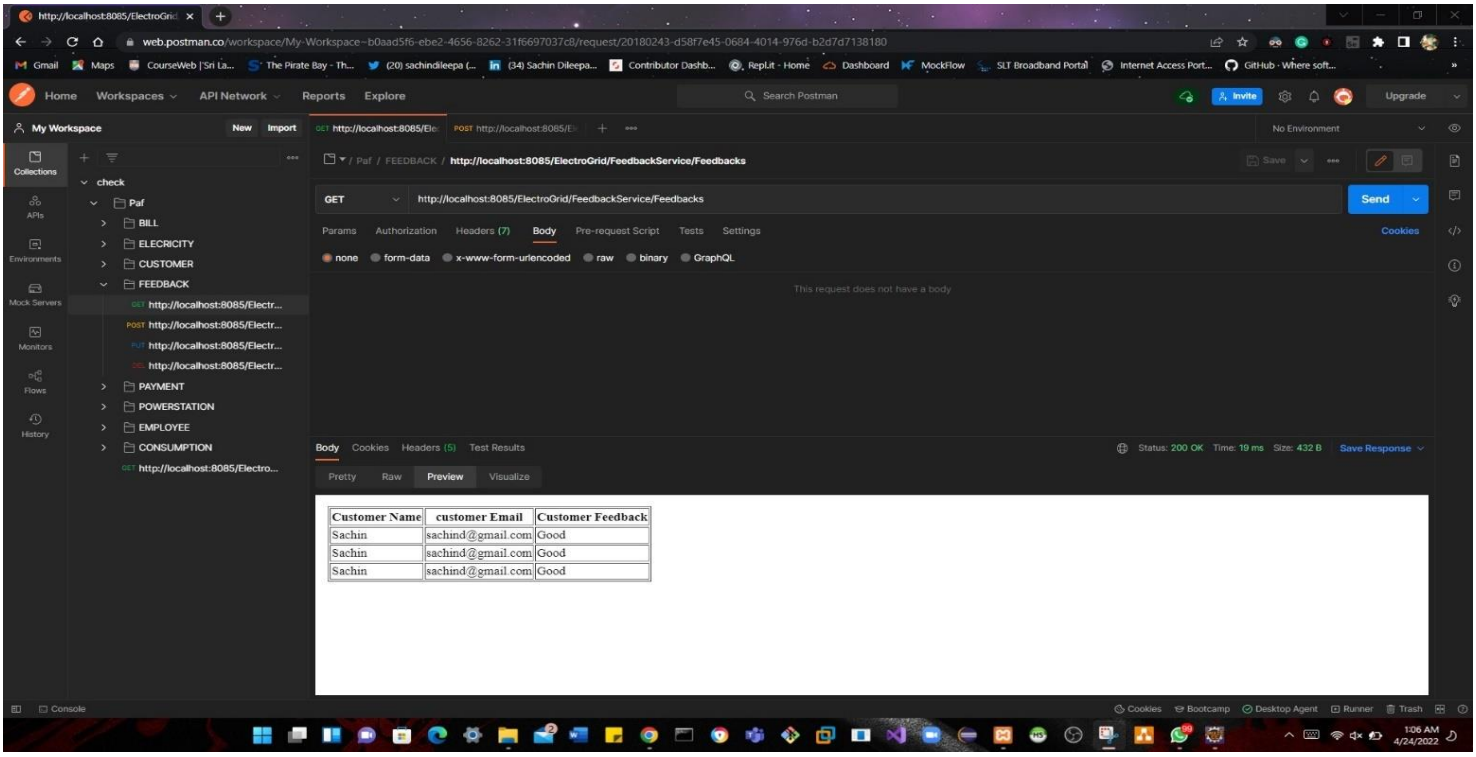

GET http://localhost:8085/ElectroGrid/BillService/Bills
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
KEY VALUE DESCRIPTION
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 27 ms Size: 433 B
Save Response
Pretty Raw Preview Visualize
Customer ID | Bill Amount | Bill Date | Paid Amount | Status
3 | 3000 | 10.11.2021 | 1500 | not paid
4 | 2500 | 12.11.2021 | 1500 | paid

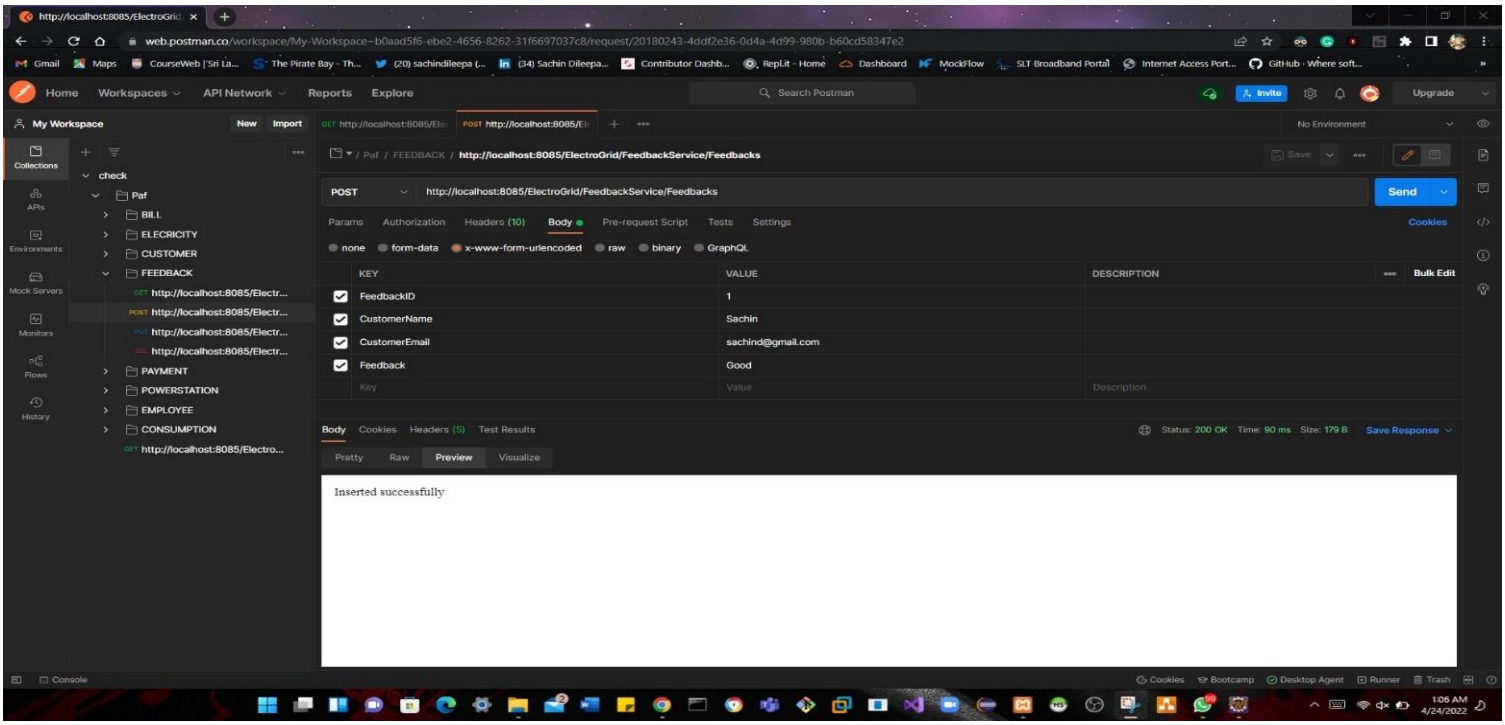

POST http://localhost:8085/ElectroGrid/BillService/Bills
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL
Billno 2
CustomerID 4
Billamount 2500.00
Billdate 12.11.2021
Paidamount 1500.00
Status paid
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 332 ms Size: 179 B
Save Response
Pretty Raw Preview Visualize
Inserted successfully

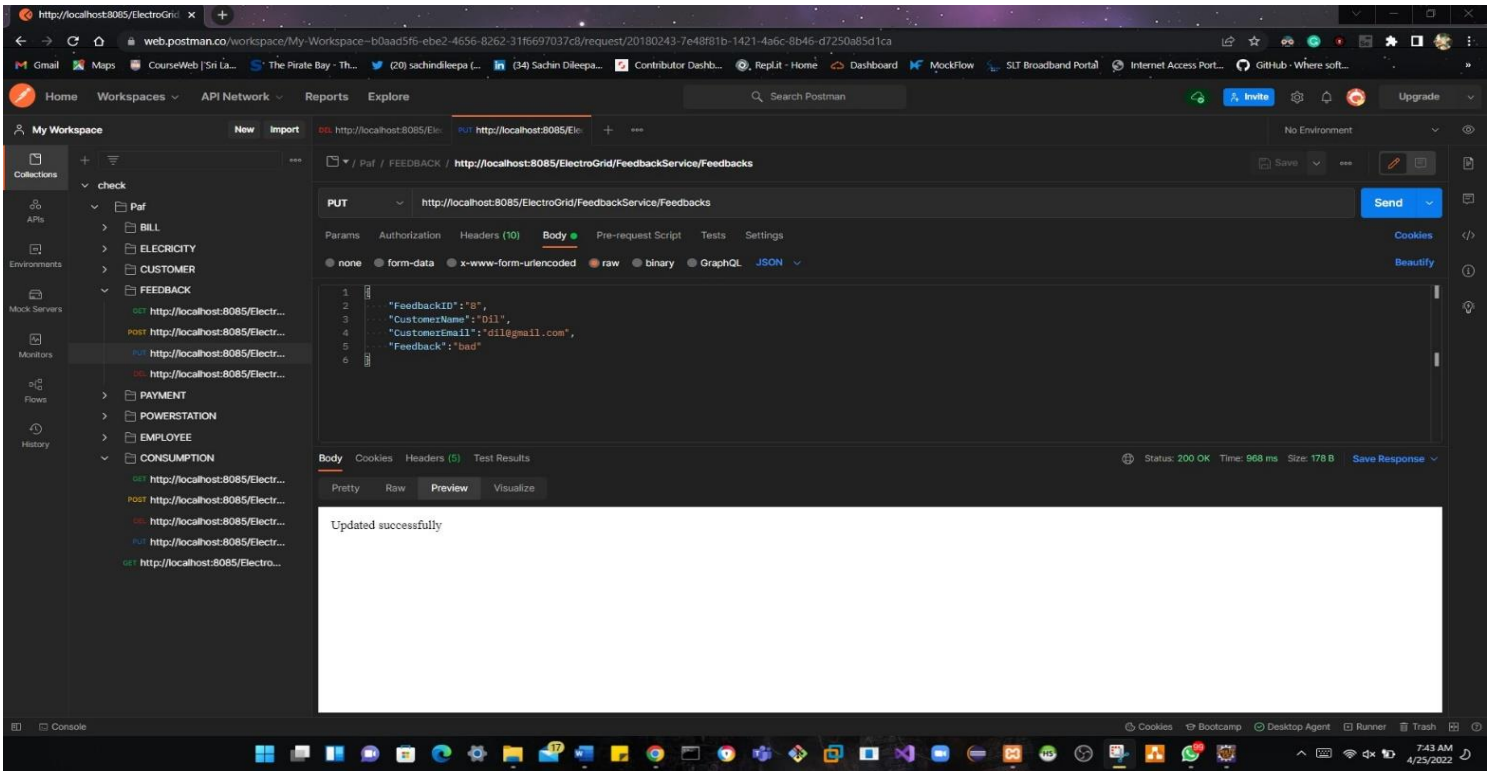


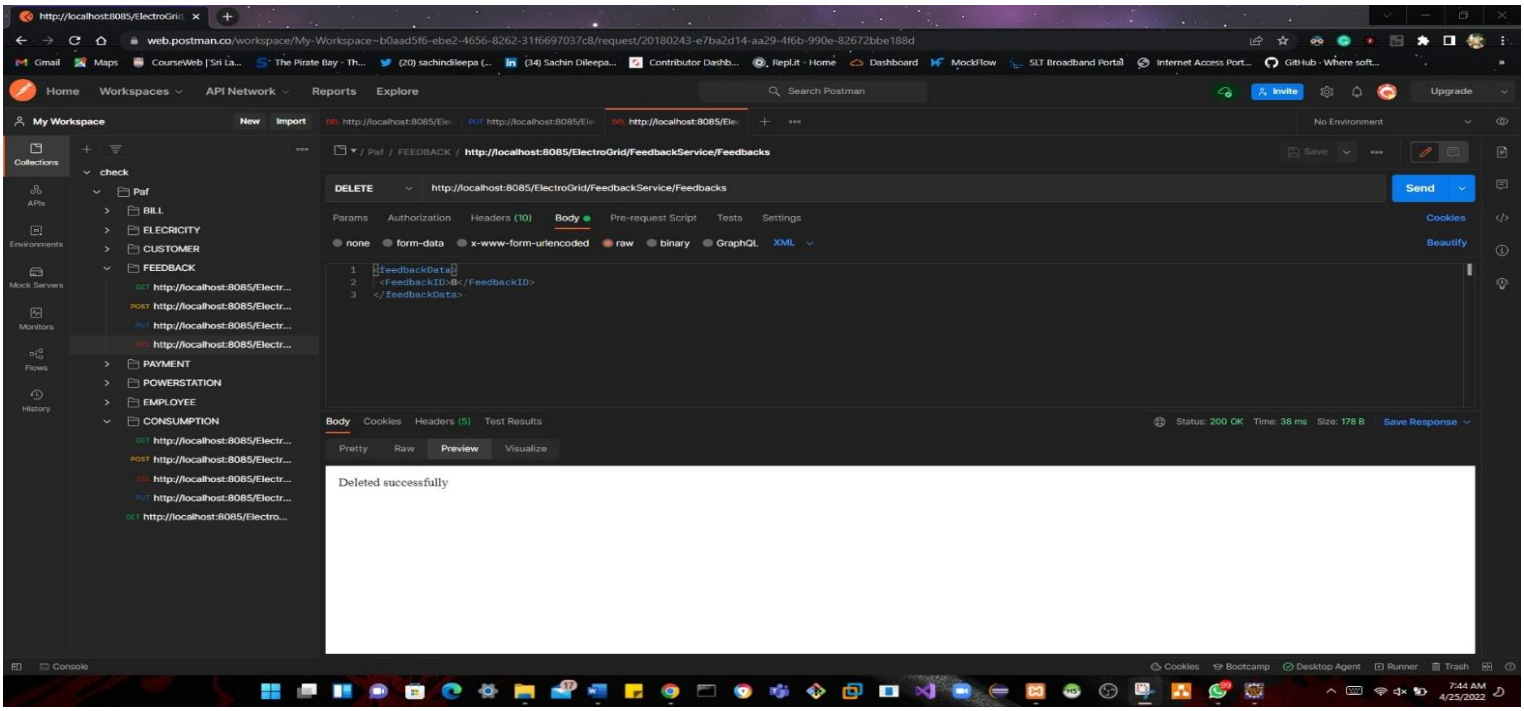


Feedback Management


GET
http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL
This request does not have a body
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 19 ms Size: 432 B
Save Response
Pretty Raw Preview Visualize
Customer Name | customer Email | Customer Feedback
Sachin | sachind@gmail.com | Good
Sachin | sachind@gmail.com | Good
Sachin | sachind@gmail.com | Good


POST
http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL
KEY VALUE DESCRIPTION Bulk Edit
FeedbackID 1
CustomerName Sachin
CustomerEmail sachind@gmail.com
Feedback Good
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 90 ms Size: 179 B
Save Response
Pretty Raw Preview Visualize
Inserted successfully


PUT
http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
{
"FeedbackID":"8",
"CustomerName":"Dil",
"CustomerEmail":"dil@gmail.com",
"Feedback":"bad"
}
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 968 ms Size: 178 B
Pretty Raw Preview Visualize
Updated successfully

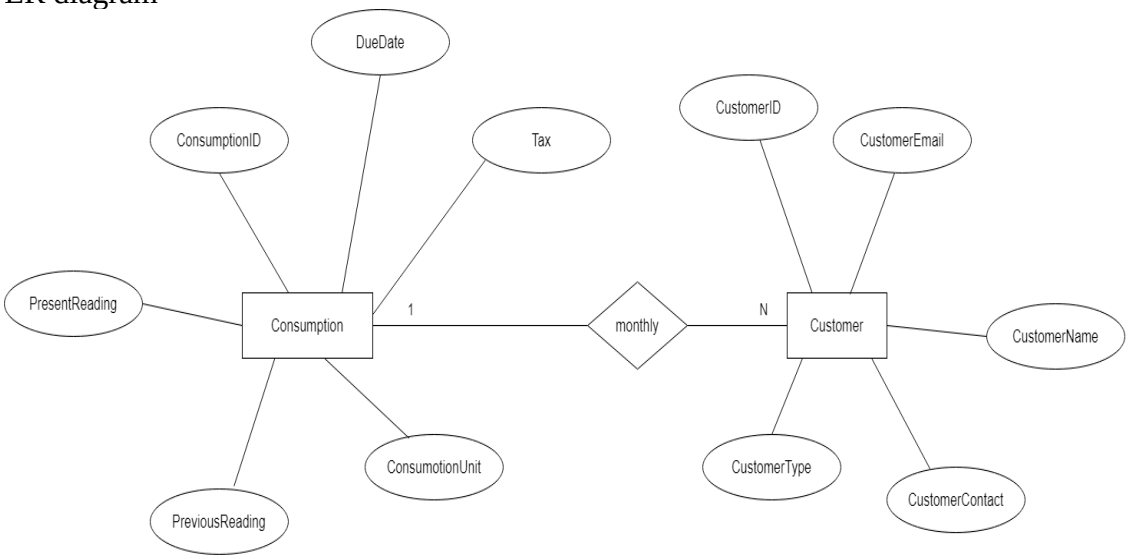

DELETE
http://localhost:8085/ElectroGrid/FeedbackService/Feedbacks
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL XML
<feedbackData>
<FeedbackID>8</FeedbackID>
</feedbackData>
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 38 ms Size: 178 B
Pretty Raw Preview Visualize
Deleted successfully

IT20139544 (Kapukotuwa S.A.A.H)
URL:
http://localhost:8085/ElectroGrid/PowerConsumptionService/Consumptions

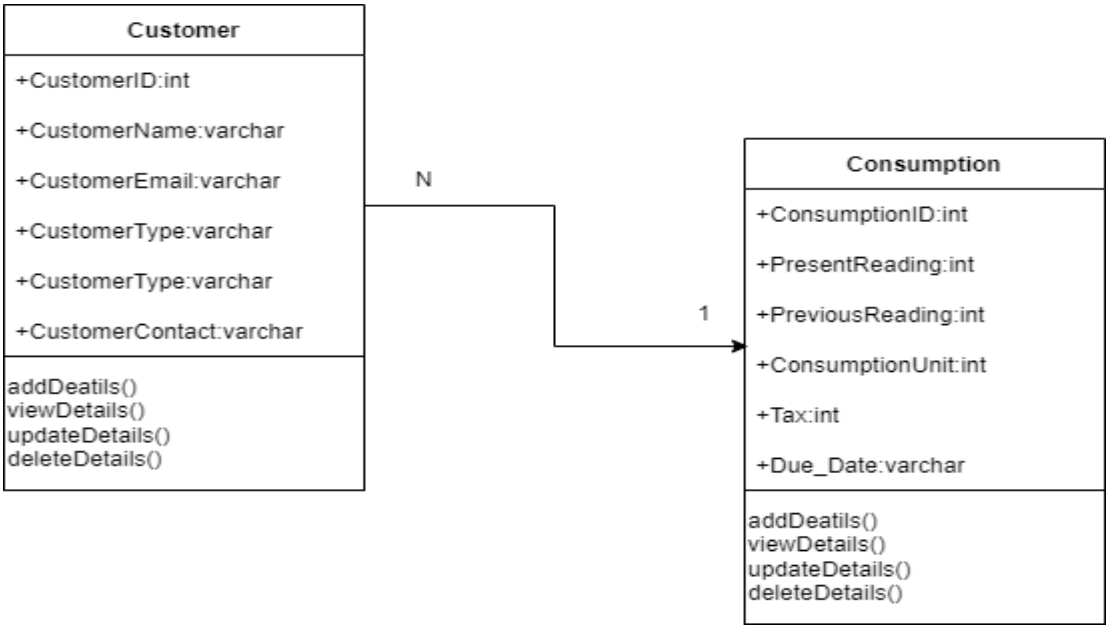
Consumption Management

Energy consumption is the amount of energy used per unit time. So this duty is assigned to consumption management. Here is the full scope of consumption management.Through this system user can enter,update and delete the details.

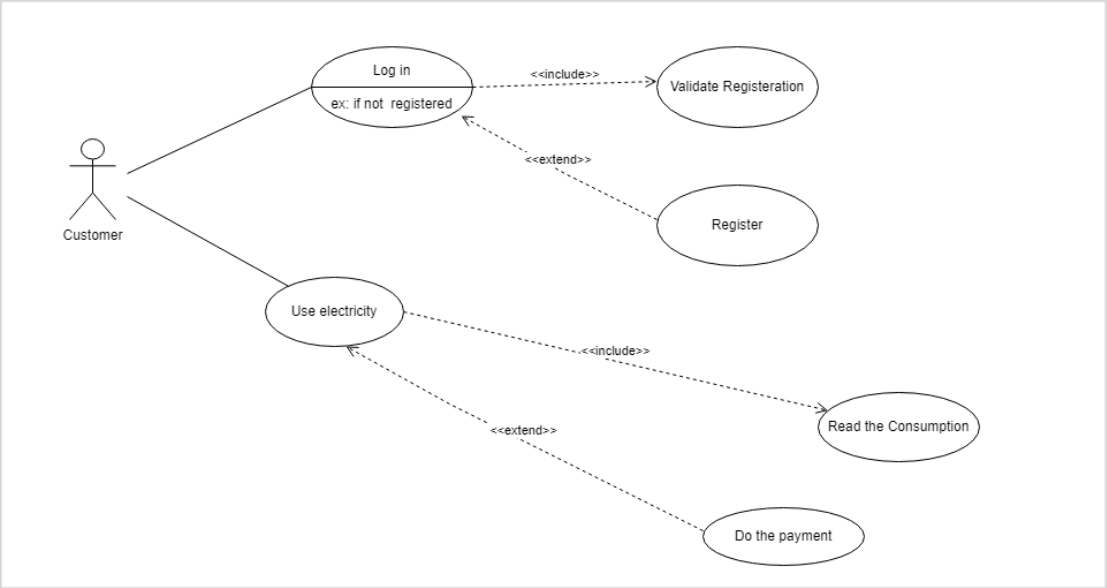
- ER diagram



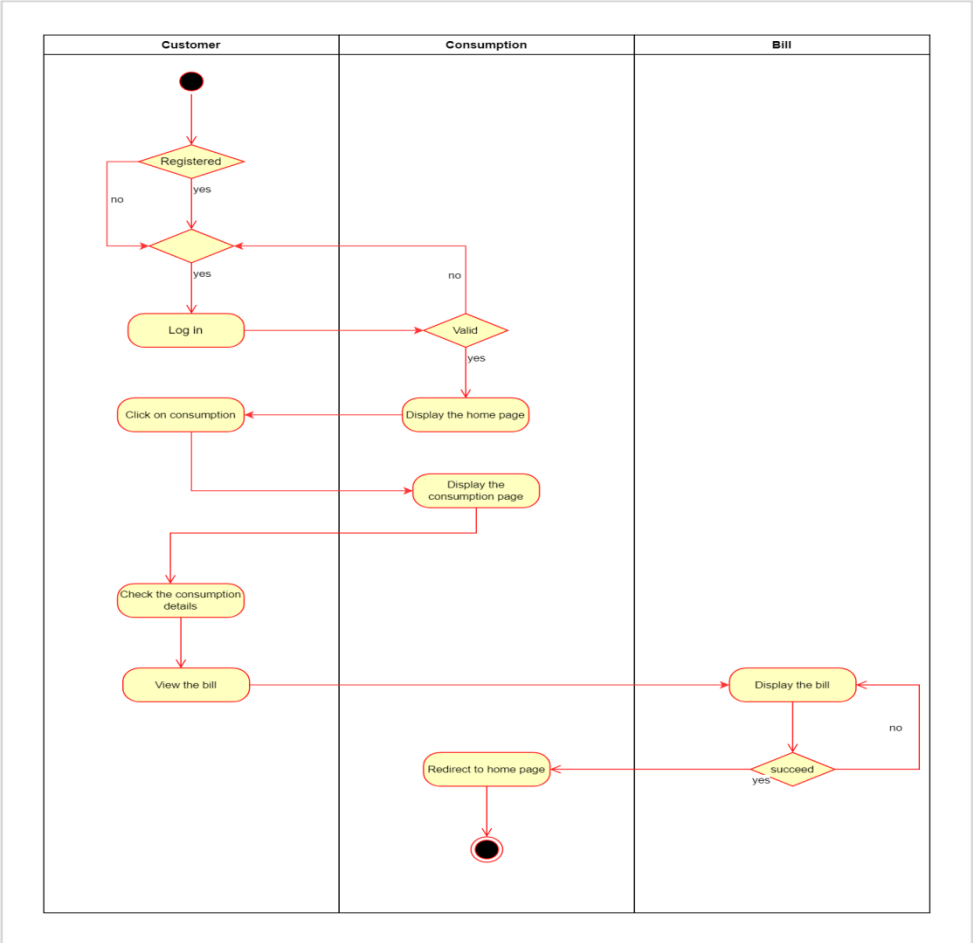
- Class Diagram



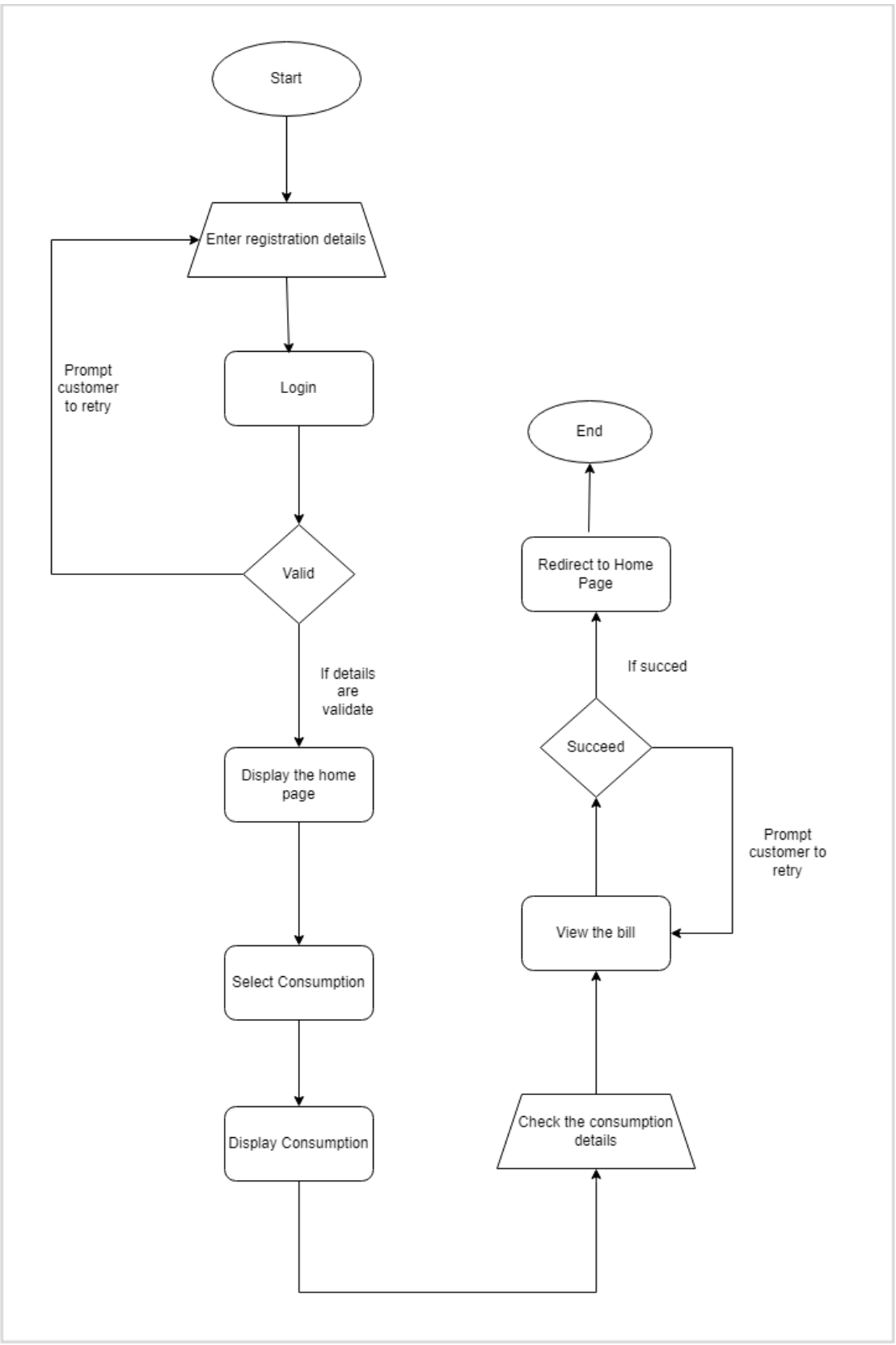
Usecase Diagram

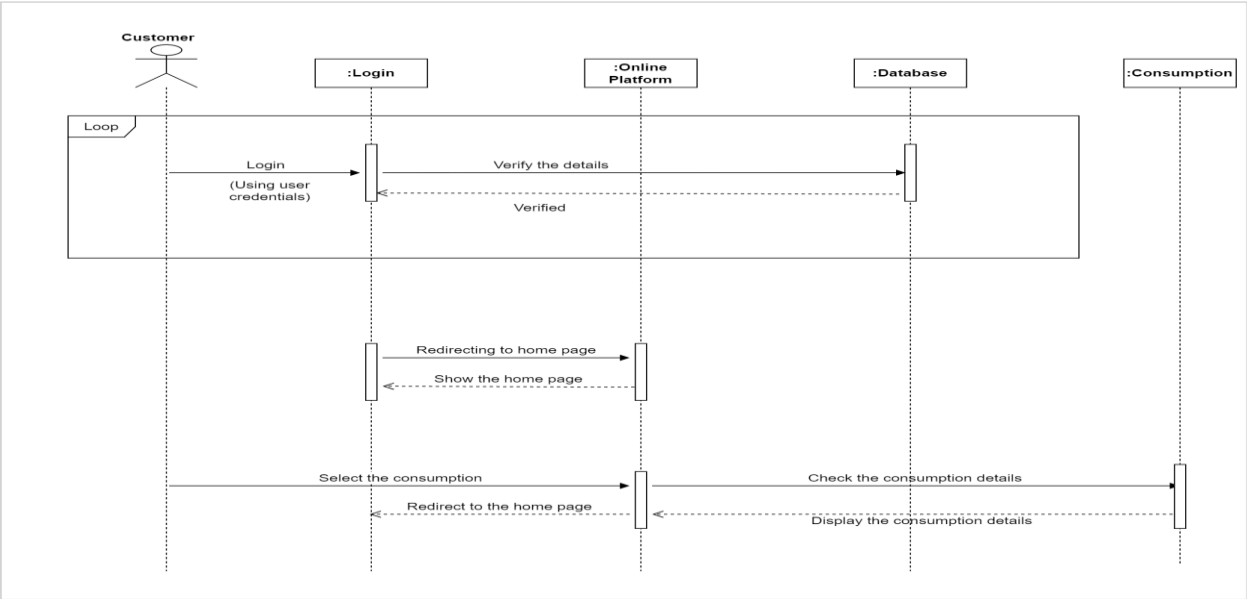


- Activity Diagram



- Flow Chart

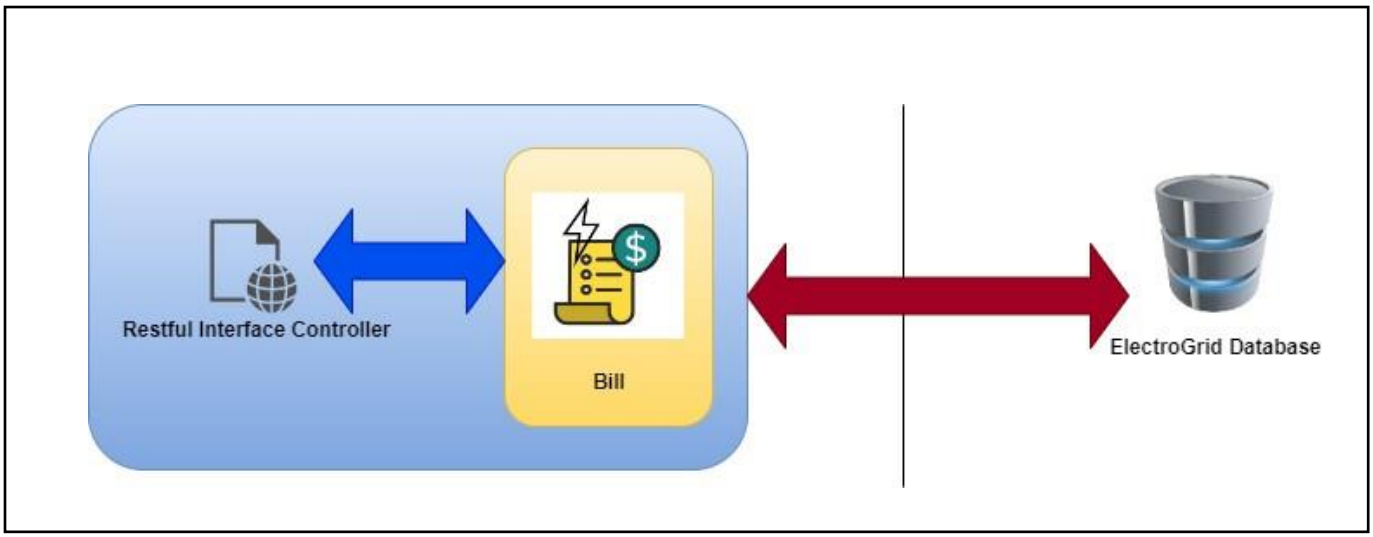

Start
Enter registration details
Prompt customer to retry
Login
Valid
If details are validate
Display the home page
Select Consumption
Display Consumption
End
Redirect to Home Page
If succed
Succeed
Prompt customer to retry
View the bill
Check the consumption details



Internal workflow

Consumption:

API



Design Relational

Consumption management system is user friendly and low complicated. Present and previous unit readings, due dates and tax which charged for unit is managed properly by this system.

- **Consumption details can be inserted, viewed, updated and deleted by user.**

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

API for get all the Consumption Details in the database (GET Request)

URL:-
http://localhost:8085/ElectroGrid/PowerConsumptionService/Consumptions

Request : - { }

Response : - {ConsumptionID : "Auto generated integer value"}
{CustomerID : 1}
{PresentReading :1000}
{PreviousReading:855}
{ConsumptionUnit: 235}
{Tax : 2400.00}
{DueDate:"18.02.2021"}

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

URL:- http://localhost:8085/ElectroGrid/PowerConsumptionService/Consumptions

Request : - {ConsumptionID : "Auto generated integer value"}
{CustomerID : 2}
{PresentReading :1343}
{PreviousReading:1000}
{ConsumptionUnit: 235}
{Tax : 2600.00}
{DueDate:"20.03.2021"}

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

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

URL:- http://localhost:8085/ElectroGrid/PowerConsumptionService/Consumptions

Request : - {ConsumptionID : "Auto generated integer value"}
{CustomerID :4}
{PresentReading :1200}
{PreviousReading:1000}
{ConsumptionUnit: 200}
{Tax : 3000.00}
{DueDate:"10.04.2025"}

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

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

URL:- http://localhost:8085/ElectroGrid/PowerConsumptionService/Consumptions

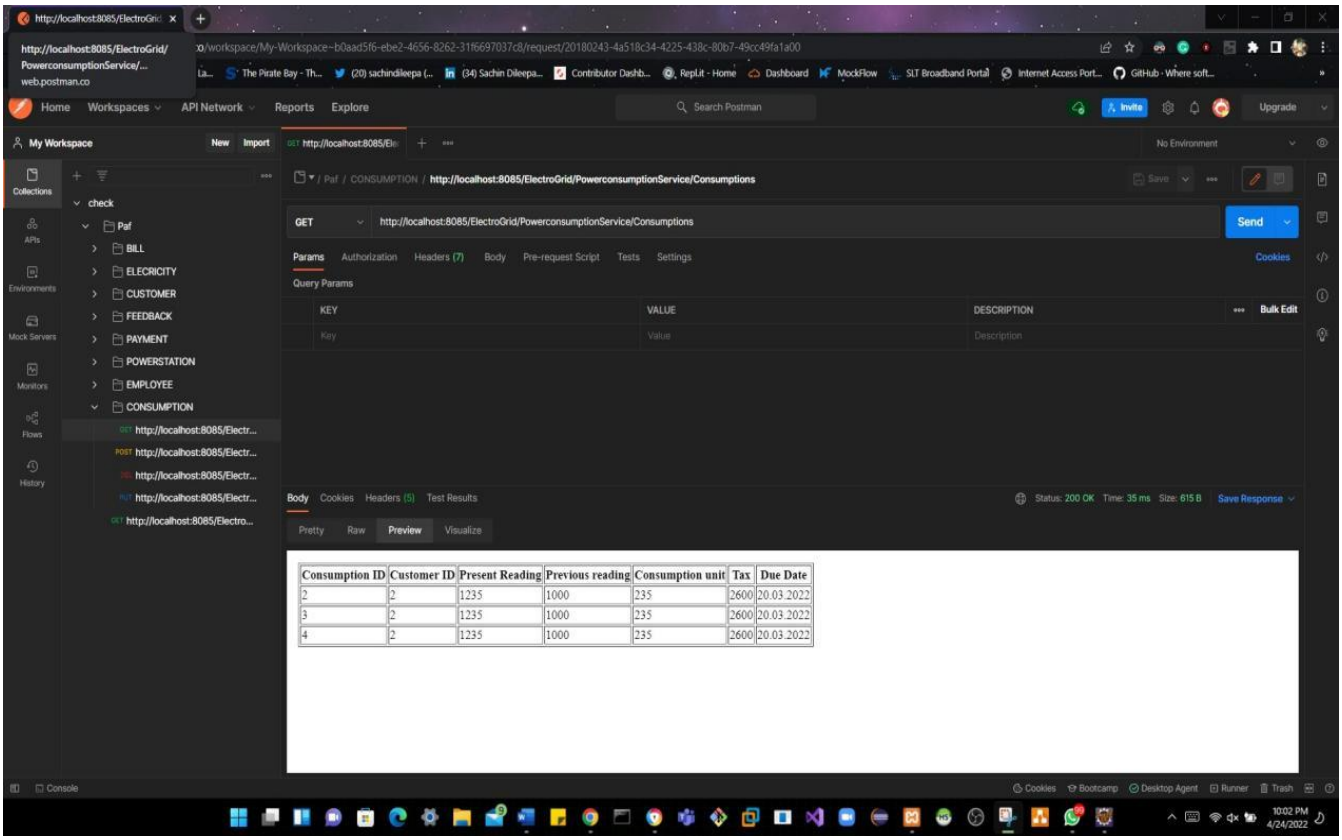
Request :- { ConsumptionID = " Selected Consumption ID from the service "}

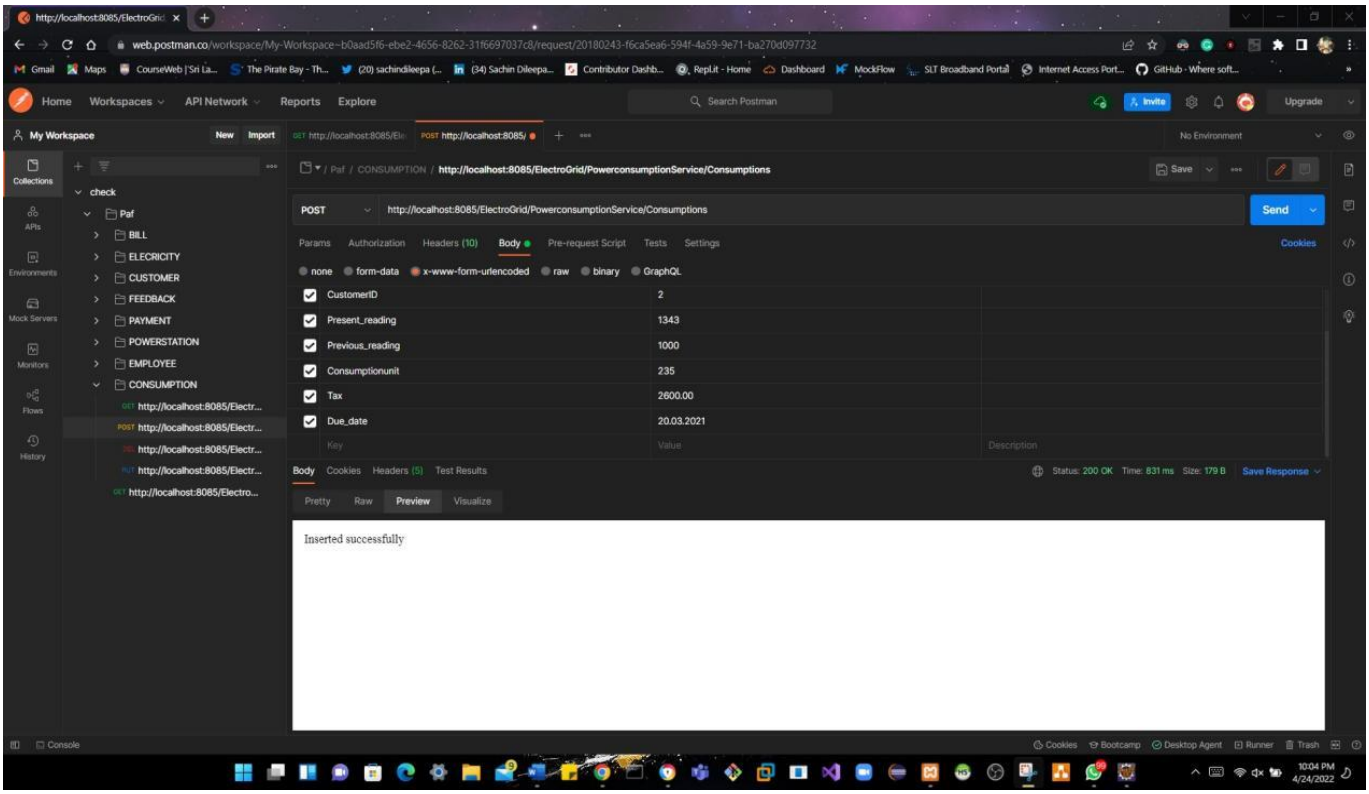
Response :- { Result = "Deleted successfully" }
{Error = "Error while deleting the Consumption details"}

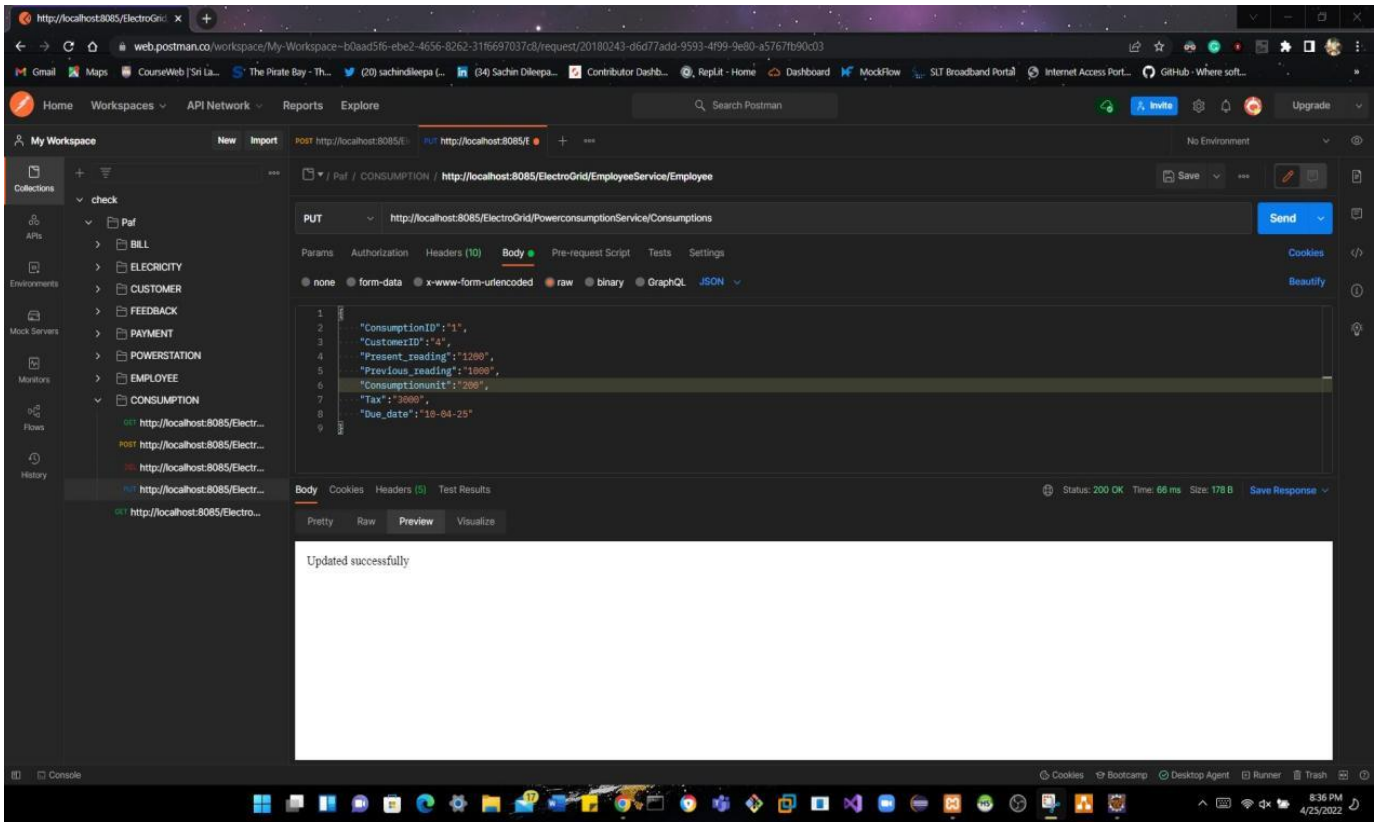
Test Cases TestID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Consumption Details	{CustomerID :2} {PresentReading :1343} {PreviousReading : 1000} {ConsumptionUnit : 235} {Tax:2600.00} {DueDate:"20.03.2021"	New Consumption details are added to the database. Show message as "Inserted successfully".	New Consumption details are added to the database. Show message as "Inserted successfully".	PASS
2	Update Customer Details	{CustomerID :4} {PresentReading :1200} {PreviousReading : 1000} {ConsumptionUnit : 200} {Tax:3000.00} {DueDate:"10.04.2025"	Consumption details are updated according to relevant input Consumption details. Show message as "Updated	Consumption details are updated according to relevant input Consumption details. Show message as "Updated successfully".	PASS

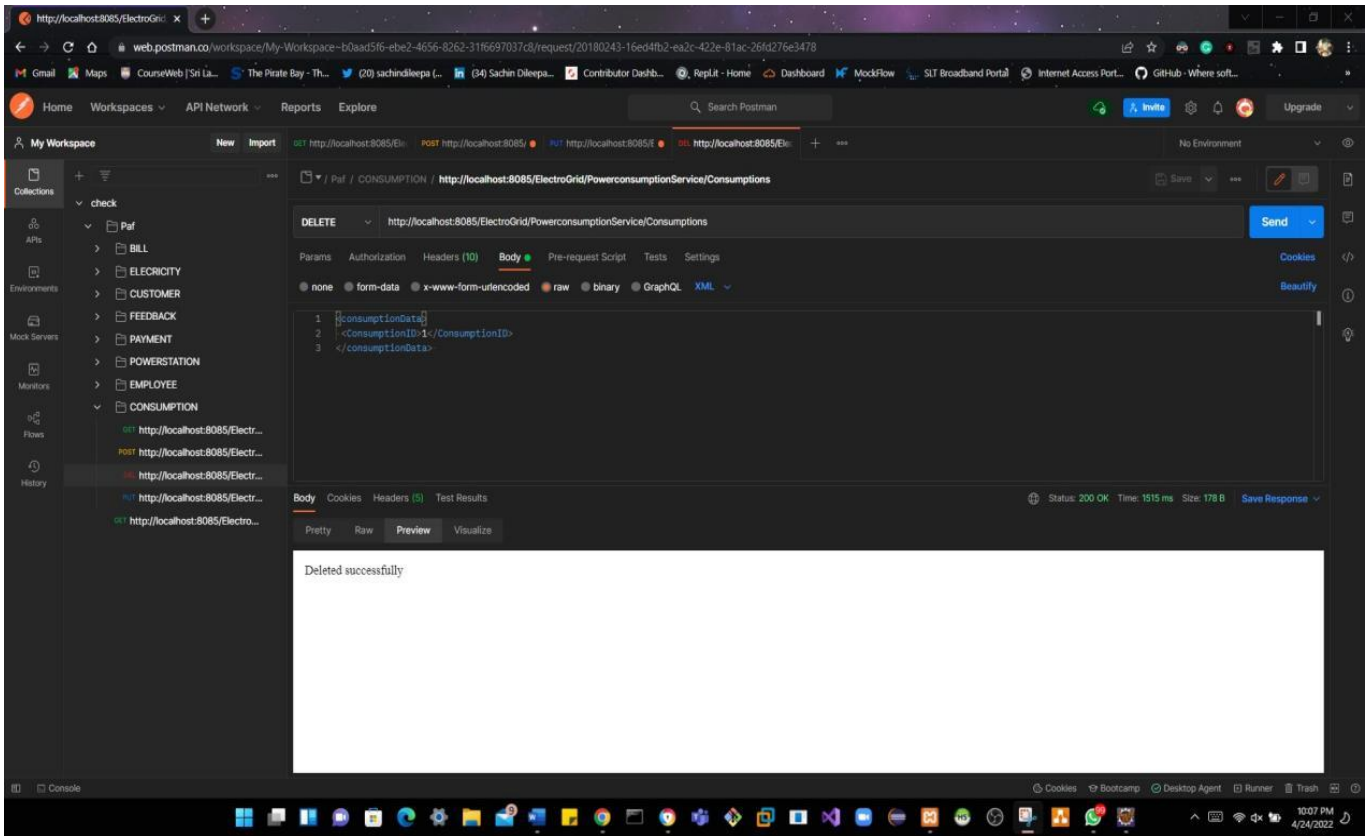
			successfully".		
3	Delete Customer Details	<consumptionData> <ConsumptionID>1</ConsumptionID> </consumptionData>	Show message as “Deleted Successfully”	Consumption details are deleted successfully. Show message as “Deleted Successfully ”	PASS

Testig




POST
http://localhost:8085/ElectroGrid/PowerconsumptionService/Consumptions
Send
CustomerID 2
Present_reading 1343
Previous_reading 1000
Consumptionunit 235
Tax 2600.00
Due_date 20.03.2021
Inserted successfully


PUT
http://localhost:8085/ElectroGrid/PowerconsumptionService/Consumptions
Send
"ConsumptionID":"1",
"CustomerID":"4",
"Present_reading":"1200",
"Previous_reading":"1000",
"Consumptionunit":"200",
"Tax":"3000",
"Due_date":"10-04-25"
Updated successfully


http://localhost:8085/ElectroGrid/
web.postman.co/workspace/My-Workspace~b0aad5f6-ebe2-4656-8262-31f6697037c8/request/20180243-16ed4fb2-ea2c-422e-81ac-26fd276e3478
Home
Workspaces
API Network
Reports
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
check
Paf
BILL
ELECRICITY
CUSTOMER
FEEDBACK
PAYMENT
POWERSTATION
EMPLOYEE
CONSUMPTION
Paf / CONSUMPTION / http://localhost:8085/ElectroGrid/PowerconsumptionService/Consumptions
Save
DELETE
http://localhost:8085/ElectroGrid/PowerconsumptionService/Consumptions
Send
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
XML
Beautify
<consumptionData>
<ConsumptionID>1</ConsumptionID>
</consumptionData>
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 1515 ms
Size: 178 B
Save Response
Pretty
Raw
Preview
Visualize
Deleted successfully
Console
Cookies
Bootcamp
Desktop Agent
Runner
Trash
10:07 PM
4/24/2022

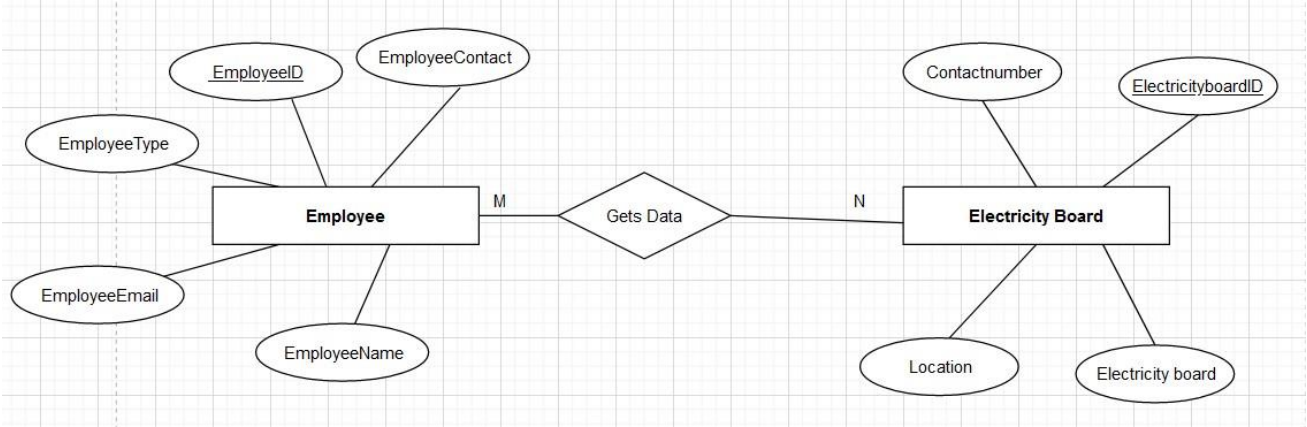
IT20235024(Abishaalini.S) URL

http://localhost:8085/ElectroGrid/EmployeeService/Employees

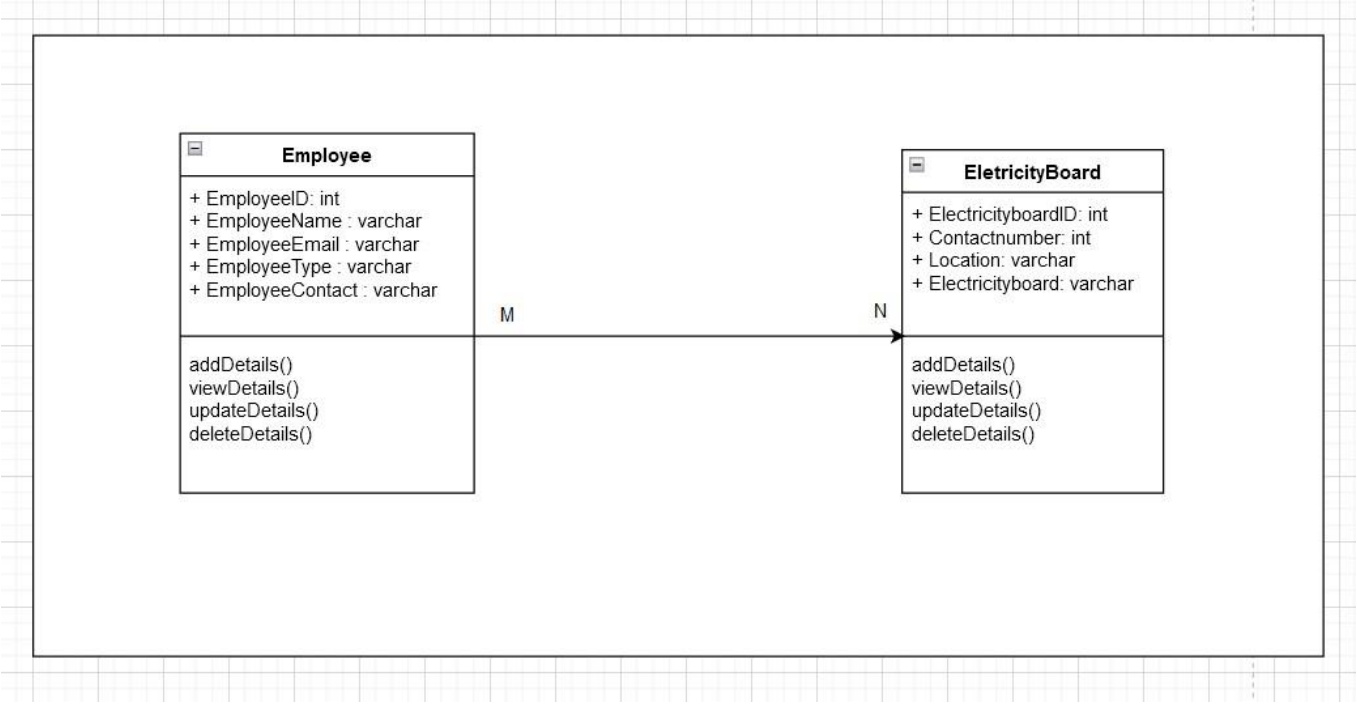
Employee Management

Here is the full scope of the employee management. The requested parties can perform their function successfully through the system. In employee management user can insert, update delete and view employee details.

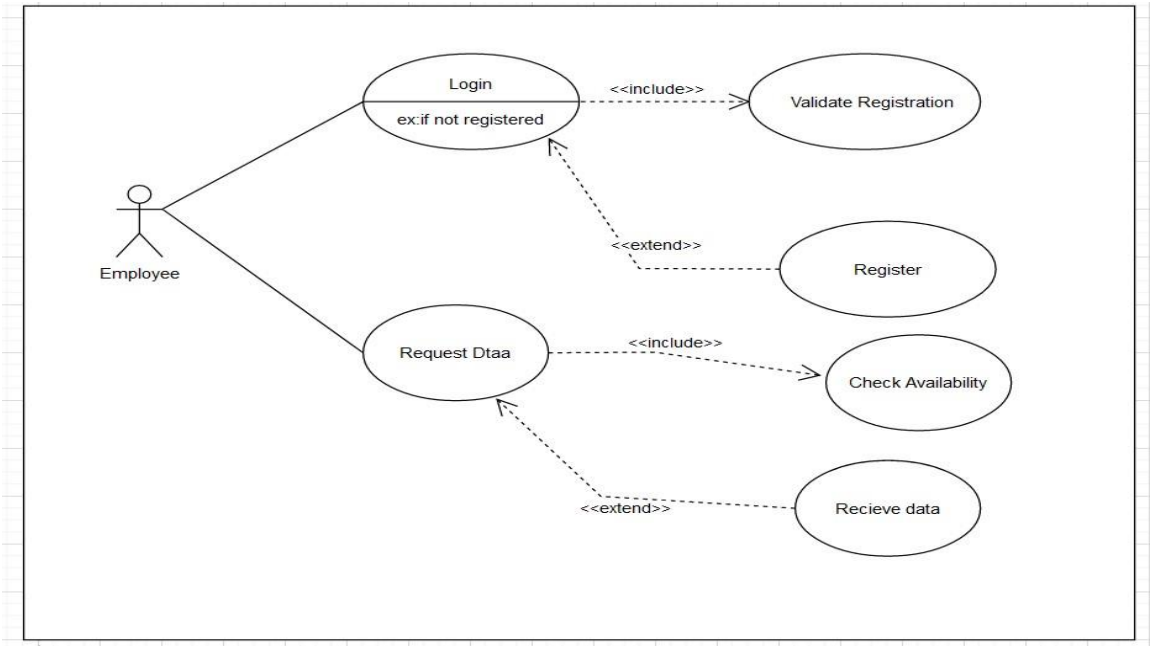
- **Er Diagram**



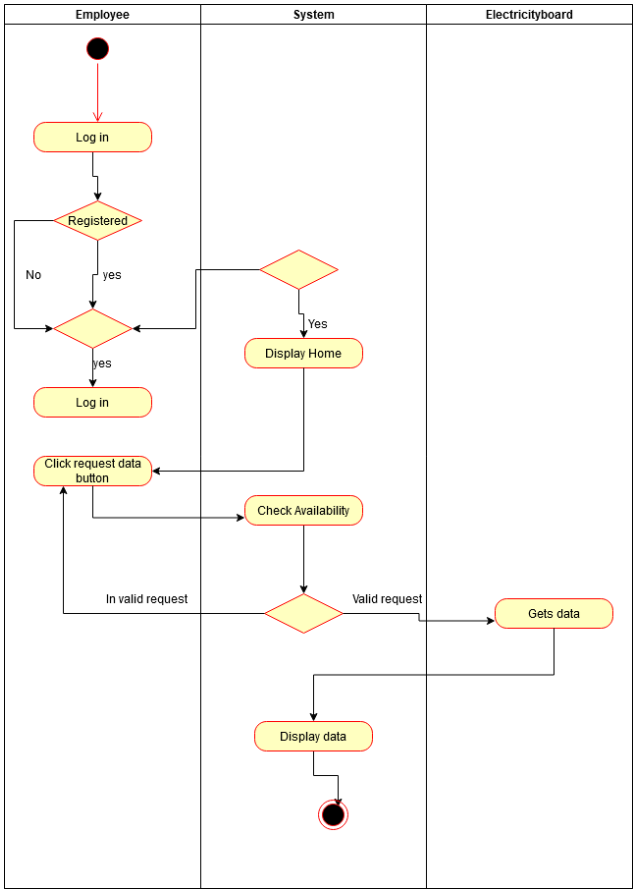
- **Class Diagram**



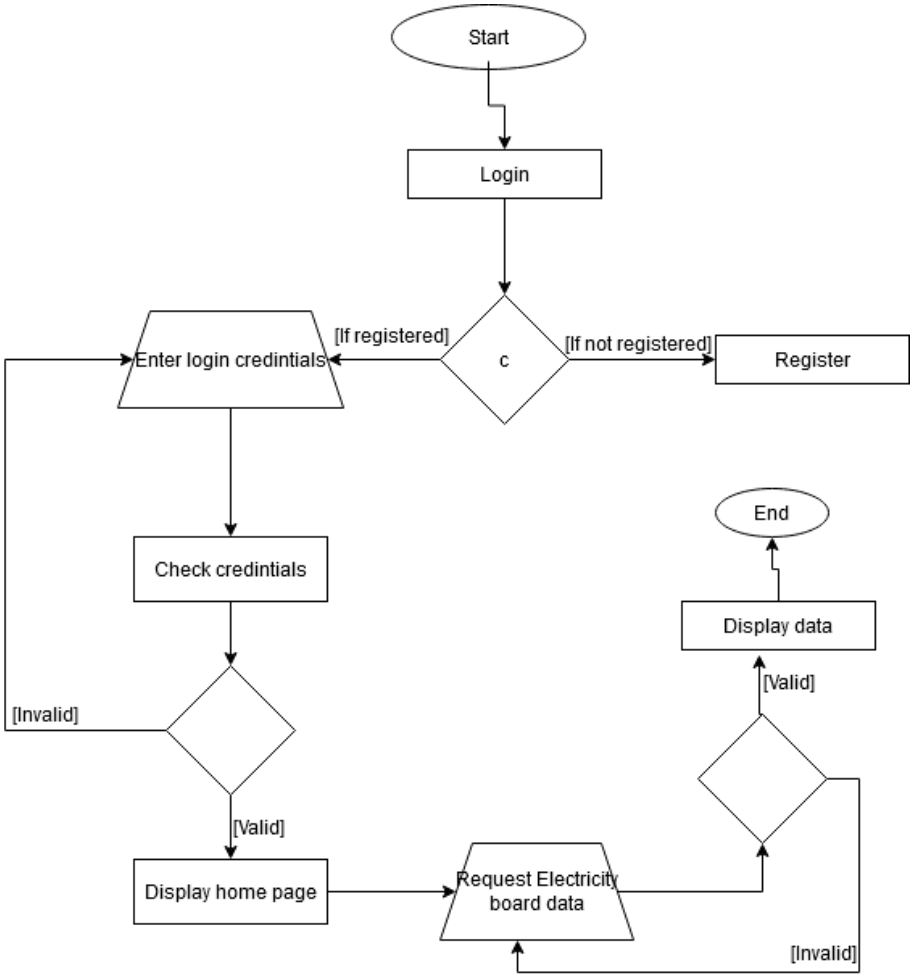
- **Use case Diagram**



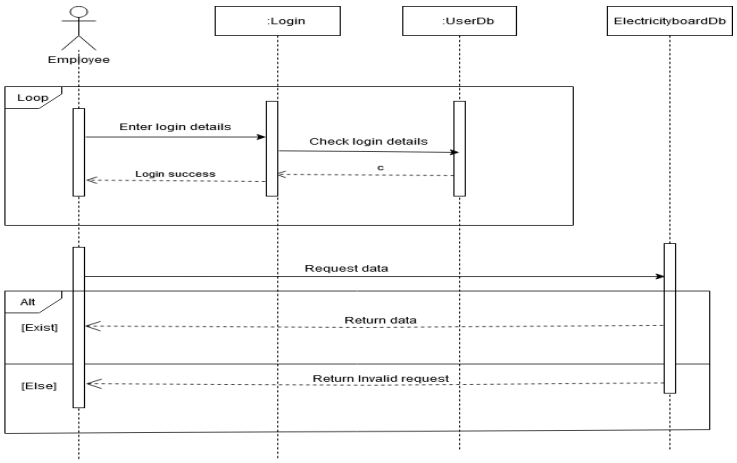
- **Activity Diagram**



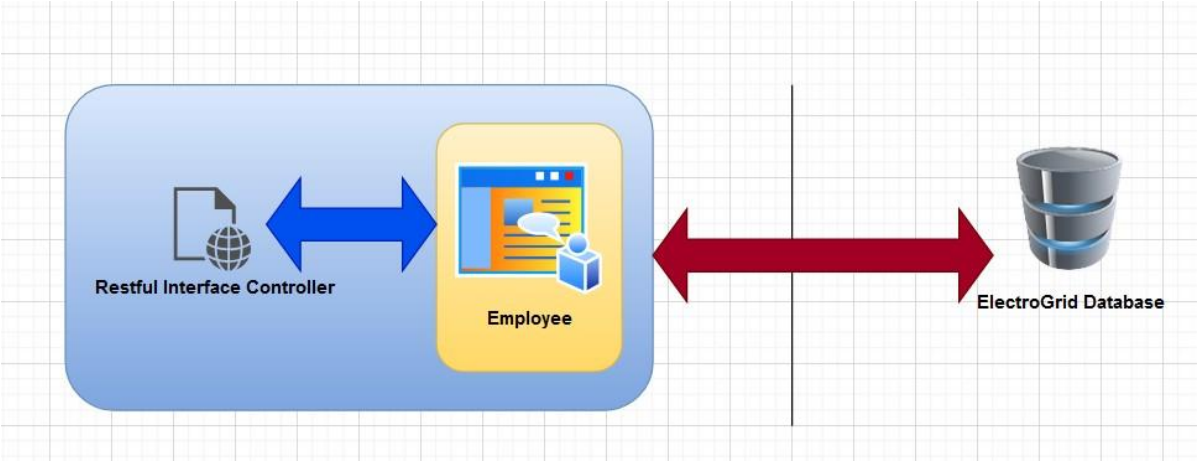
- **Flow Chart**



- **Sequence Diagram**



- **Internal workflow**



API Design Relational

Employee management handles all the information related to employees through insert, delete update, read methods.

- **Employee can insert, delete, update, 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://localhost:8085/ElectroGrid/EmployeeService/Employees

Request : - { }

Response : - {EmployeeID : "Auto generated integer value"}
{EmployeeName =:"Kamal"}
{EmployeeEmail :kamal@gmail.com }
{EmployeeType : Old Employee"}

{EmployeeContact : "071222256"}

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

URL:- http://localhost:8085/ElectroGrid/EmployeeService/Employees

Request : - {EmployeeID : "Auto generated integer value"}
{EmployeeName =:"Nimal"}
{EmployeeEmail :nimal@gmail.com }
{EmployeeType : New Employee"}
{EmployeeContact : "071222256"}

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

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

URL:- http://localhost:8085/ElectroGrid/EmployeeService/Employees

Request : - {EmployeeID : "Auto generated integer value"}
{EmployeeName =:"Nimal"}
{EmployeeEmail :nimal@gmail.com }
{EmployeeType : New Employee"}
{EmployeeContact : "071222256"}

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

API for delete an Employee which is existing in database (DELETE Request)
URL:- http://localhost:8085/ElectroGrid/EmployeeService/Employees

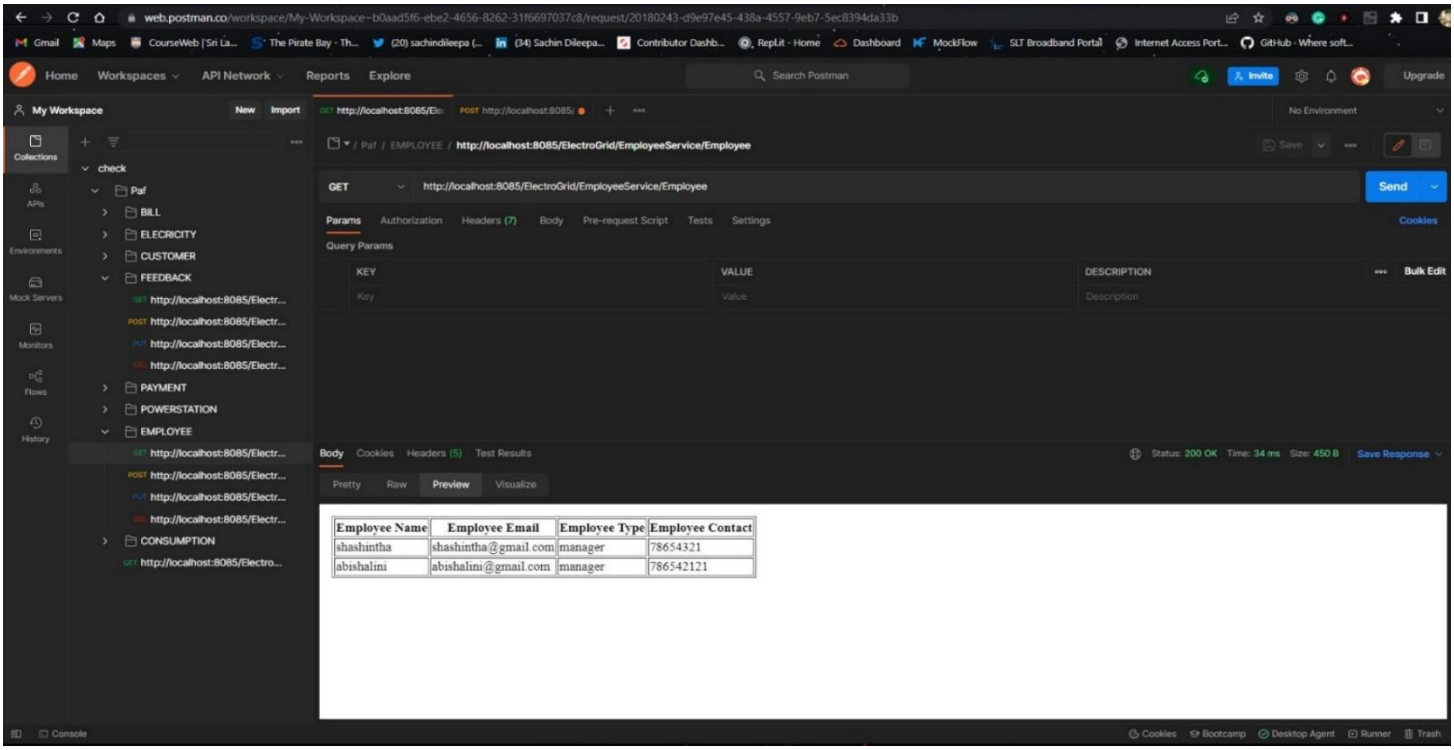
Request :- { EmployeeID = " Selected Employee ID from the service "}

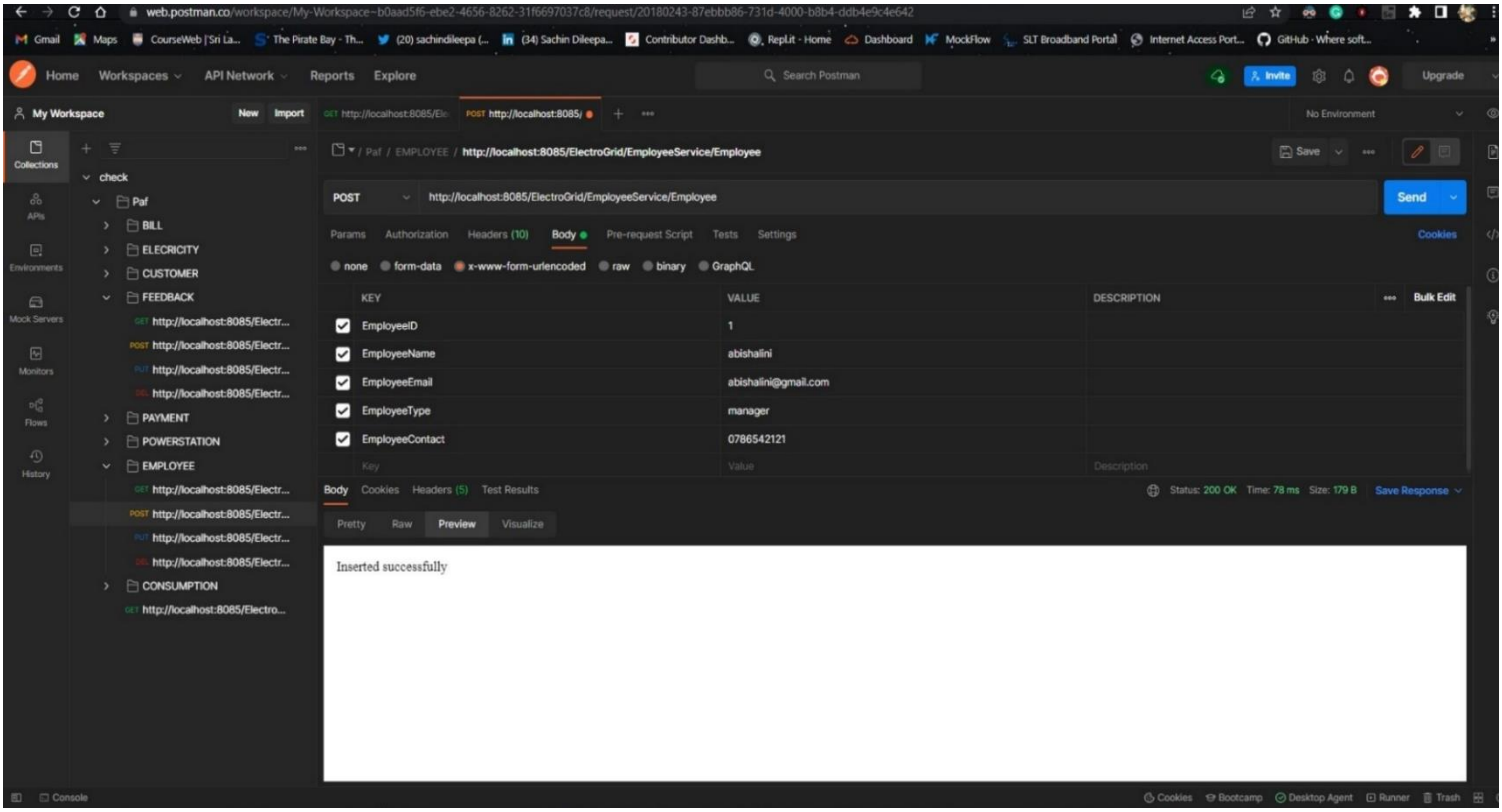
Response :- { Result = "Deleted successfully" }
{Error = "Error while deleting the Employee" }

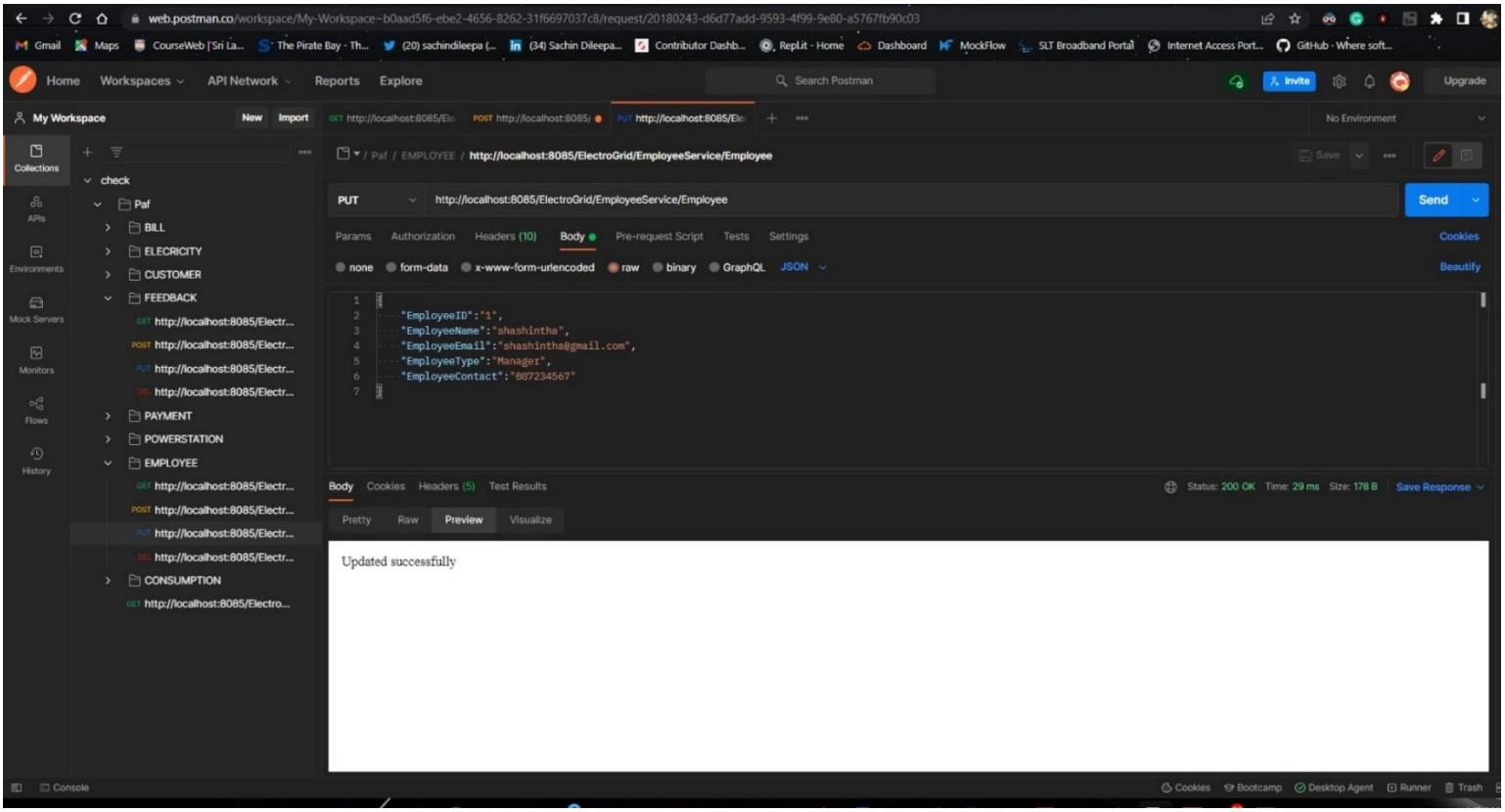
Test Cases TestID	**Test Description/ Test Steps**	Test Input(s)	**Expected Output(s)**	Actual Output(s)	Result (Pass/Fail)

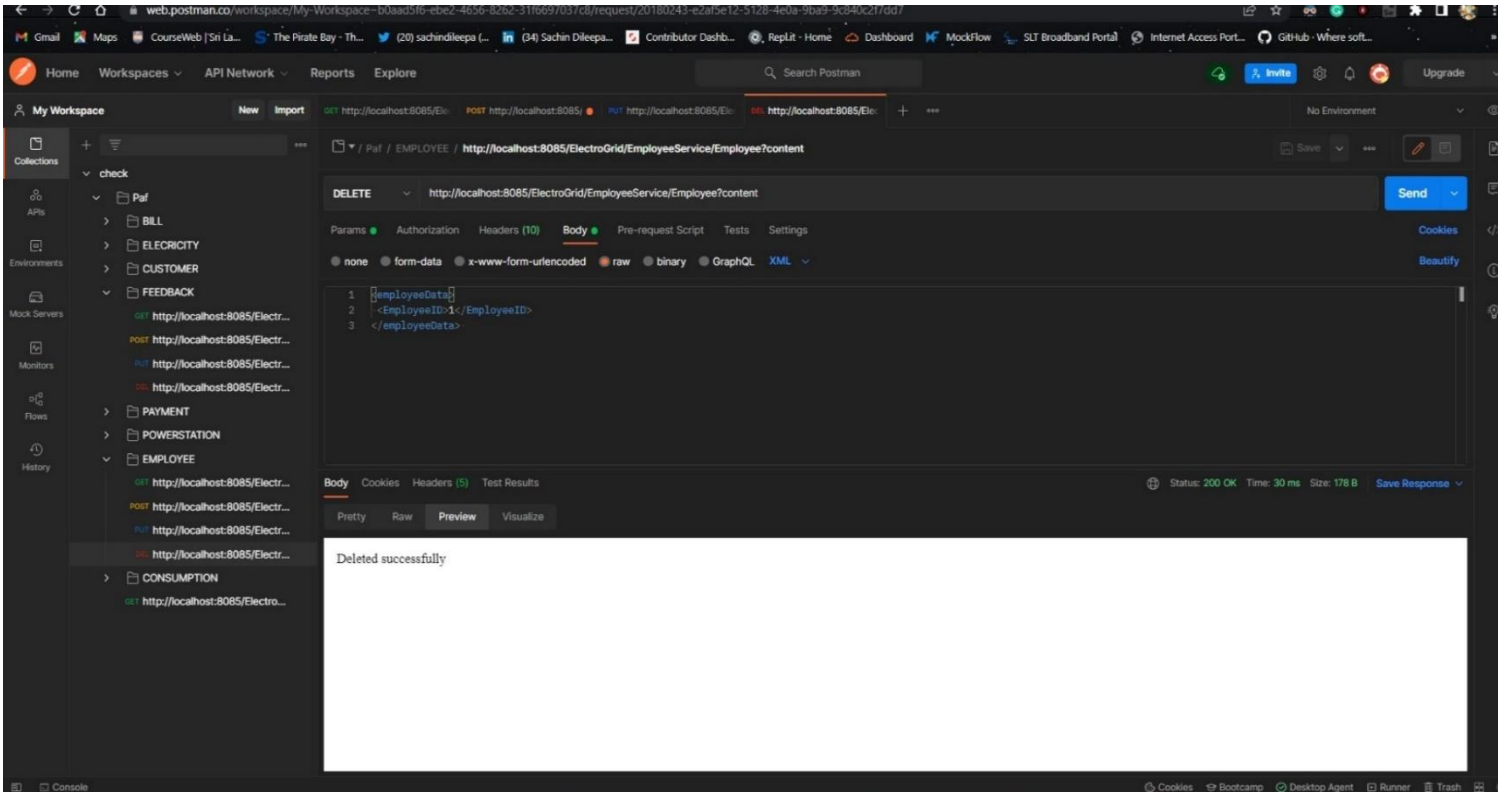
1	Add Employee Details	{EmployeeName =:"Abishaalini"} {EmployeeEmail : abi@gmail.com } {EmployeeType : old employee"} {EmployeeContact : "075123456"}	New Employee details are added to the database. Show message as "Inserted successfully".	New Employee details are added to the database. Show message as "Inserted successfully".	PASS
2	Update Employee Details	{EmployeeName =:"Abishaalini"} {EmployeeEmail : abi@gmail.com } {EmployeeType : old employee"} {EmployeeContact : "075123456"}	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
3	Delete Employee Details	<EmployeeData> <EmployeeID>1</EmployeeID> </EmployeeData>	Show message as “Deleted Successfully”	Employee Details are deleted successfully. Show message as “Deleted Successfully”	PASS

Testing (Employee)




POST
http://localhost:8085/ElectroGrid/EmployeeService/Employee
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL
KEY VALUE DESCRIPTION
EmployeeID 1
EmployeeName abishalini
EmployeeEmail abishalini@gmail.com
EmployeeType manager
EmployeeContact 0786542121
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 78 ms Size: 179 B
Pretty Raw Preview Visualize
Inserted successfully


PUT
http://localhost:8085/ElectroGrid/EmployeeService/Employee
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
{
"EmployeeID":"1",
"EmployeeName":"shashintha",
"EmployeeEmail":"shashintha@gmail.com",
"EmployeeType":"Manager",
"EmployeeContact":"887234567"
}
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 29 ms Size: 178 B
Pretty Raw Preview Visualize
Updated successfully

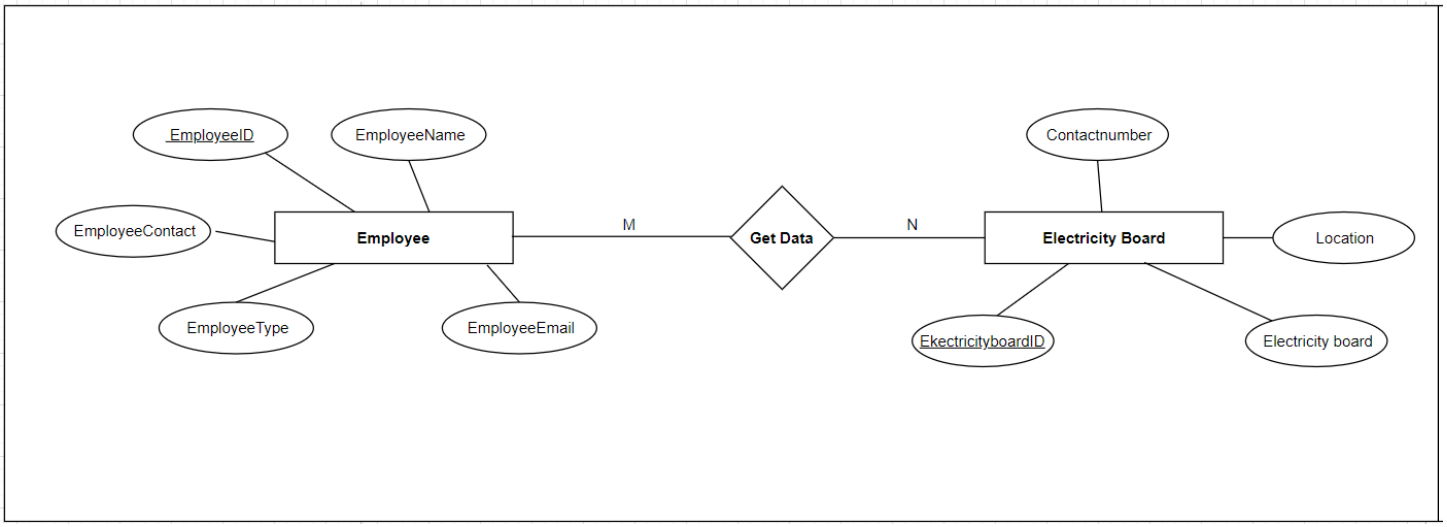

Home
Workspaces
API Network
Reports
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
check
Paf
BILL
ELECRICITY
CUSTOMER
FEEDBACK
PAYMENT
POWERSTATION
EMPLOYEE
CONSUMPTION
Paf / EMPLOYEE / http://localhost:8085/ElectroGrid/EmployeeService/Employee?content
DELETE
http://localhost:8085/ElectroGrid/EmployeeService/Employee?content
Send
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
XML
Beautify
<employeeData>
<EmployeeID>1</EmployeeID>
</employeeData>
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 30 ms
Size: 178 B
Save Response
Pretty
Raw
Preview
Visualize
Deleted successfully
Console
Bootcamp
Desktop Agent
Runner
Trash

IT20139858 (Gunasinghe P.S.L)
URL :- http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards

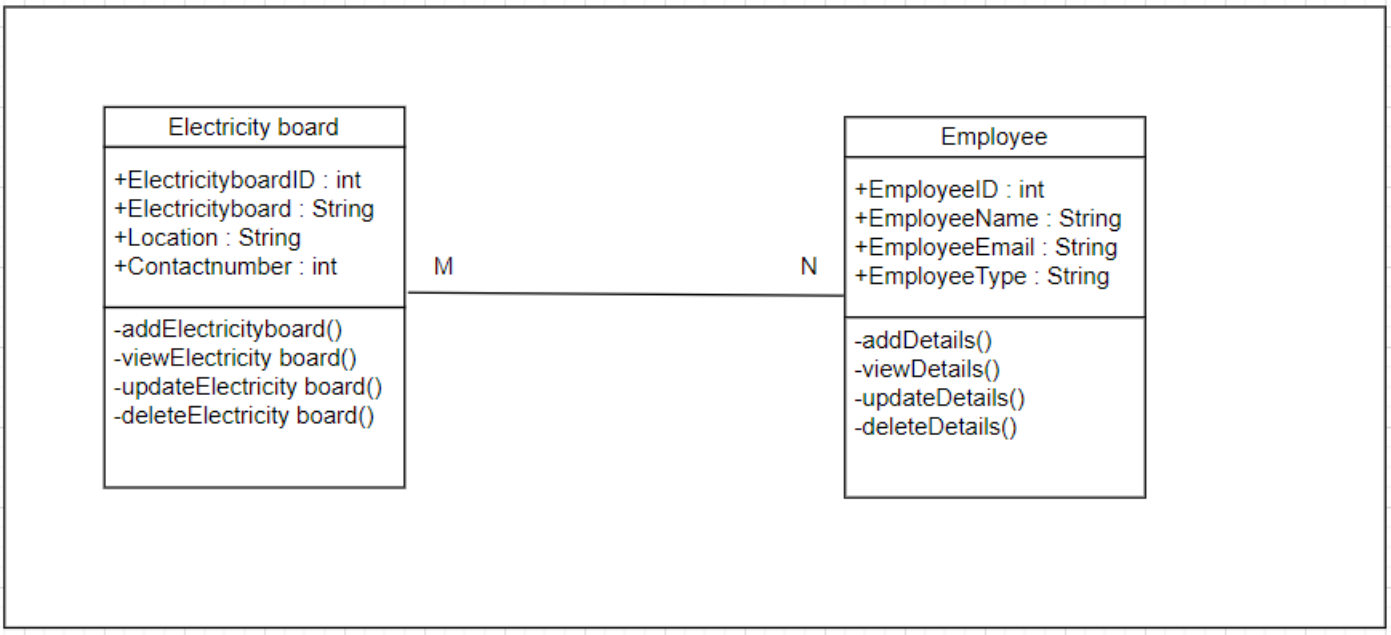
Electricity board Management

Here is the full scope of the electricity board management. The requested parties can perform their function successfully through the system. In electricity board management, user can insert, update delete and view Electricity board branch details.

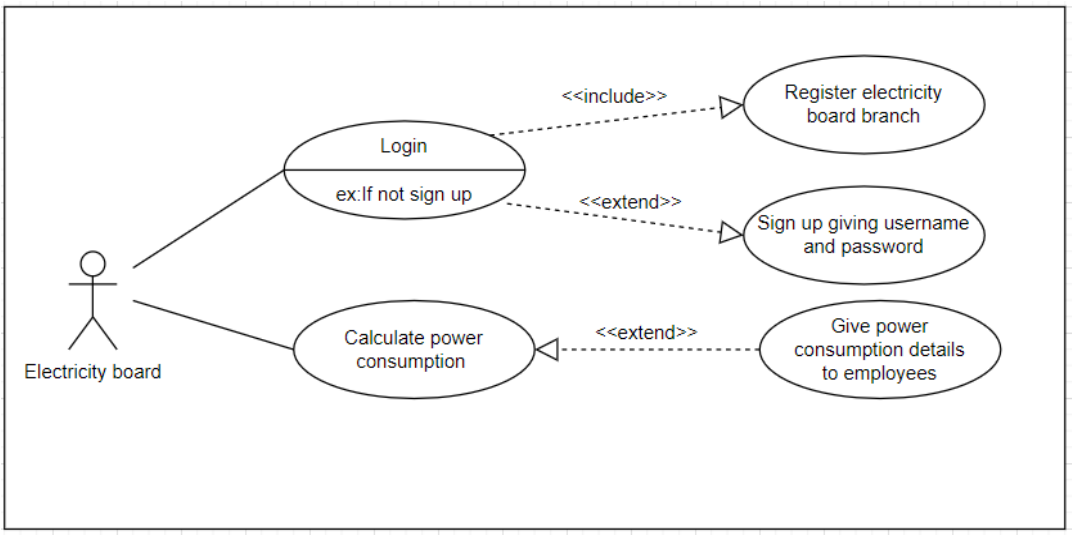
- **Er Diagram**



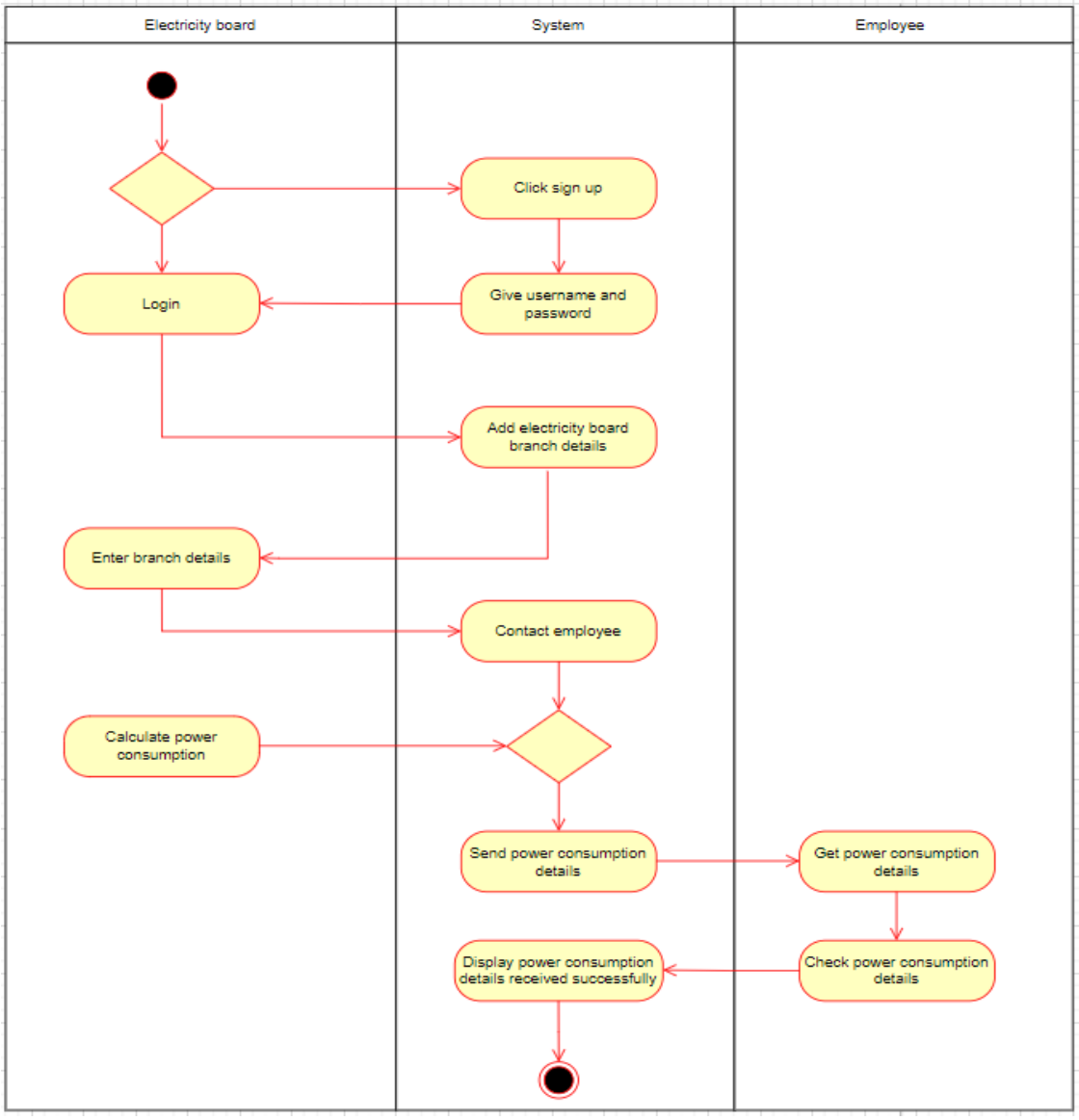
- **Class Diagram**



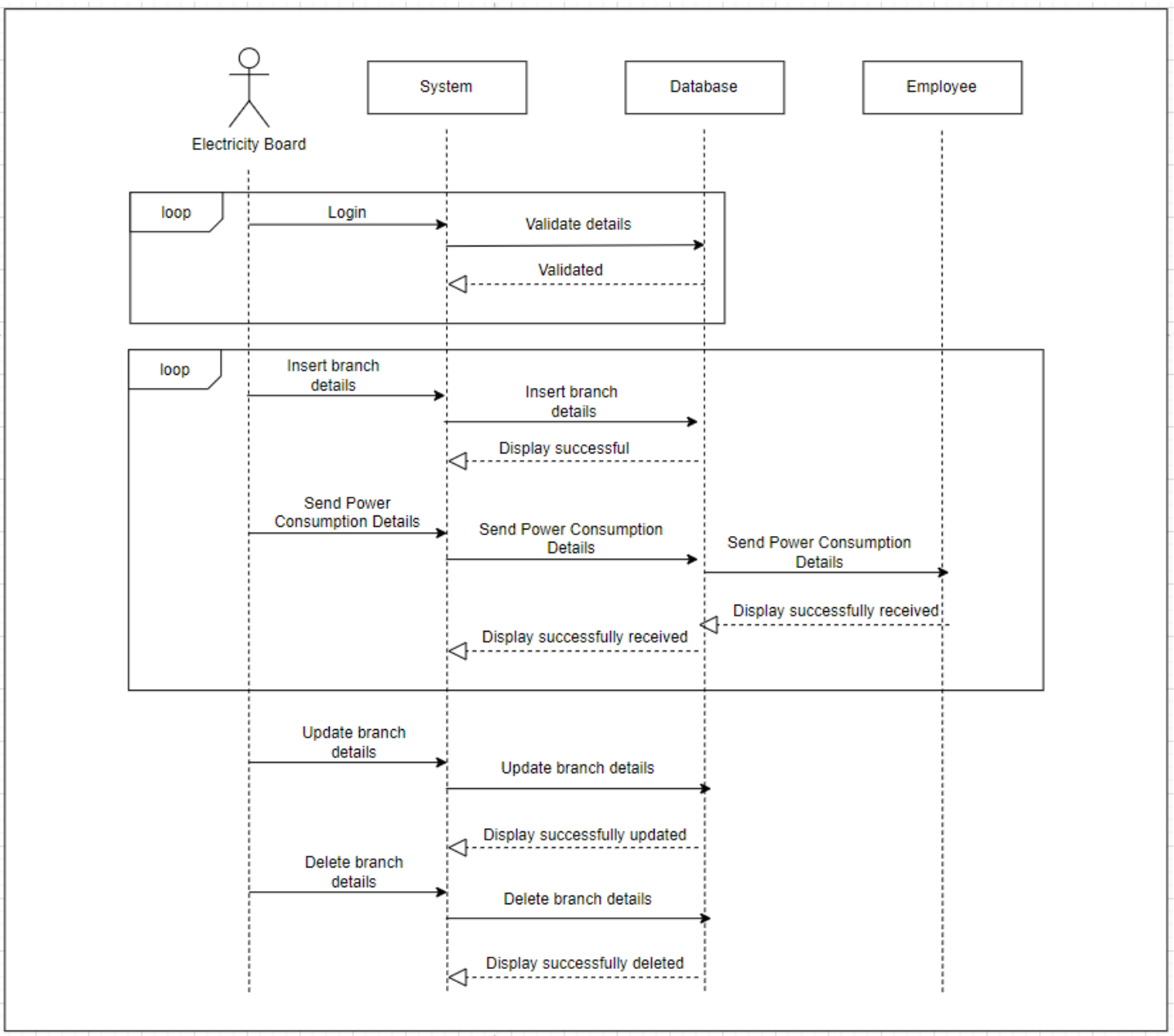
- **Use case Diagram**



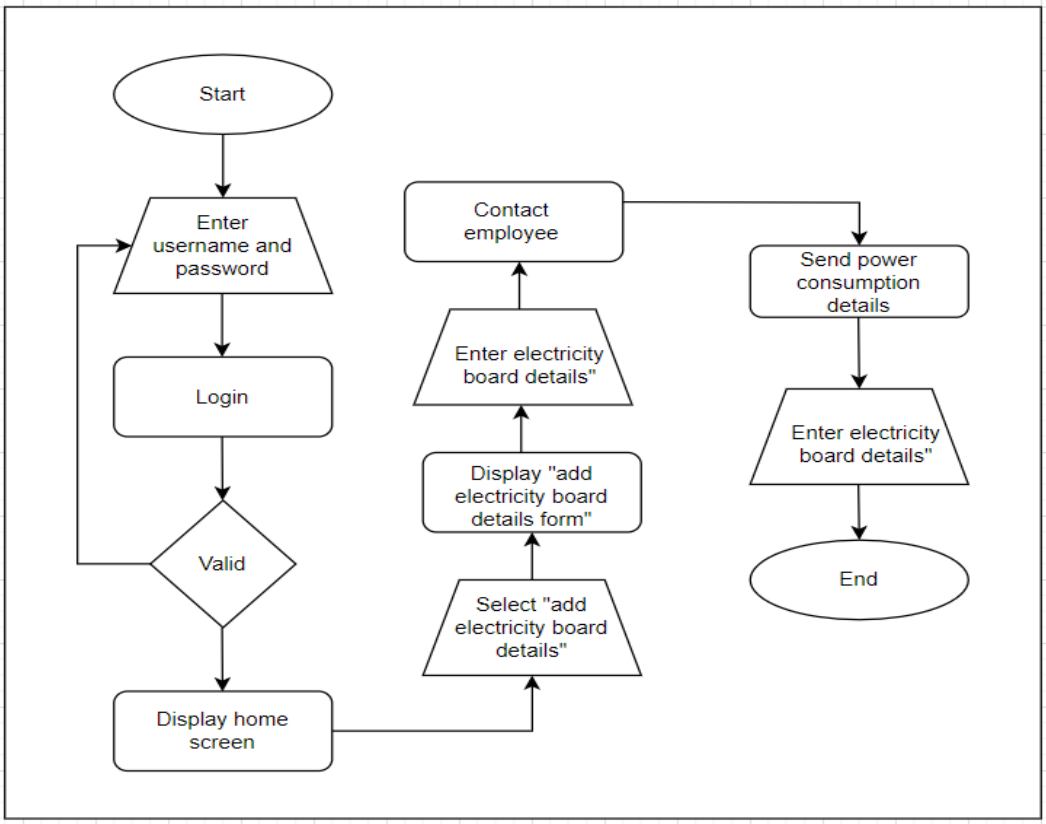
- **Activity Diagram**



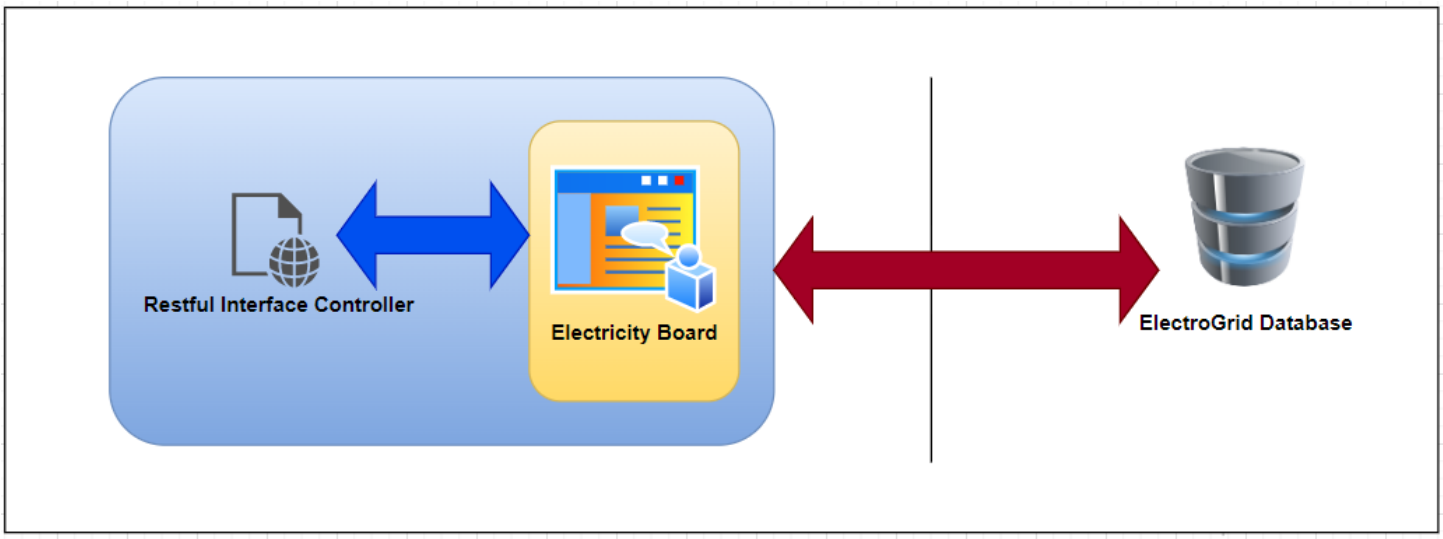
- **Sequence Diagram**



- **Flow chart**



- **Internal workflow**



API Design Relational

Electricity board management handles all the information related to electricity boards

through insert, delete update, read methods.

- **Electricity board can insert, delete, update, view the electricity board branch details.**

Service Development

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

1) API for get all the Electricity board branch details in the database (GET Request)

URL :- http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards

Request : - { }

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

{Electricityboard =:"LECO"}

{Location : "Colombo" }

{Contactnumber : "0814563297"}

2) API for insert a new Electricity board branch details in the database (POST Request)

URL :- http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards

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

{Electricityboard :"CEB"}

{Location : "Kandy" }

{Contactnumber : "0814563297"}

Response :- { "Inserted successfully"}

3) API for update existing Electricity board branch details in the database (PUT Request)

URL :- http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards

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

{Electricityboard :"CEB"}

{Location : "Kurunegala" }

{Contactnumber : "0379629186"}

Response :- { Result = "Updated successfully" }

4) API for delete existing Electricity board branch details in the database (DELETE Request)

URL :- http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards

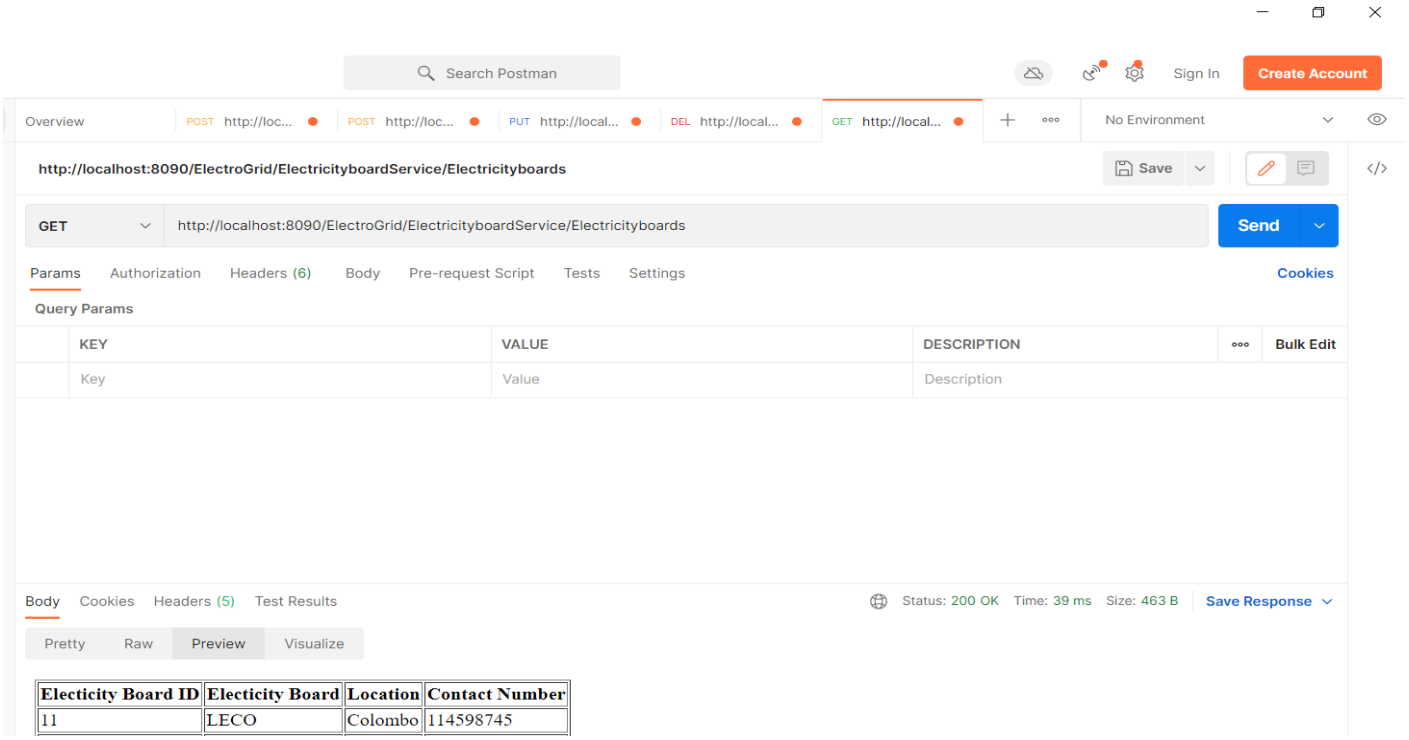
Request : - {ElectricityboardID : "Selected ElectricityboardID from the service"}

Response :- { Result = "Deleted successfully" }

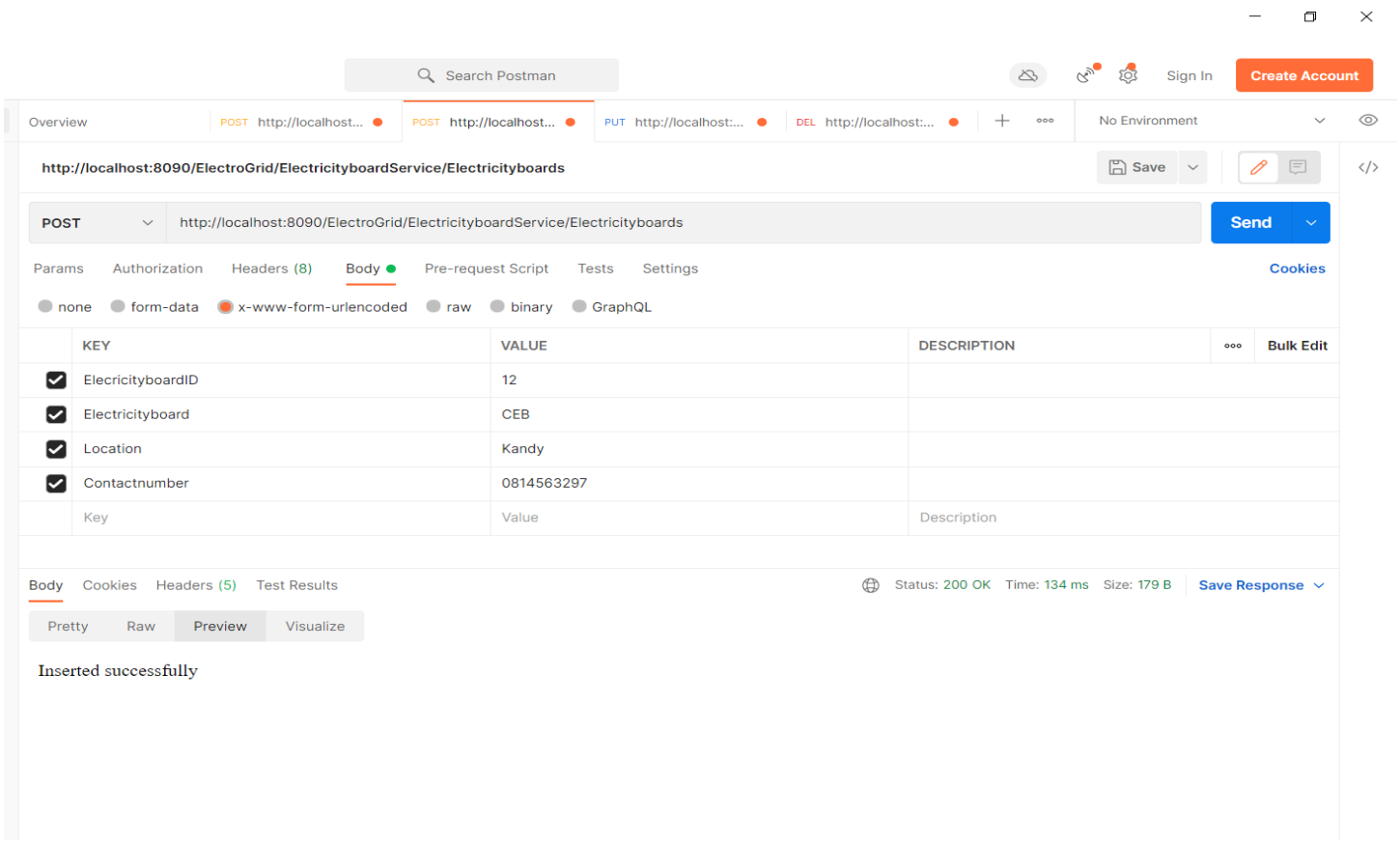
Test Cases

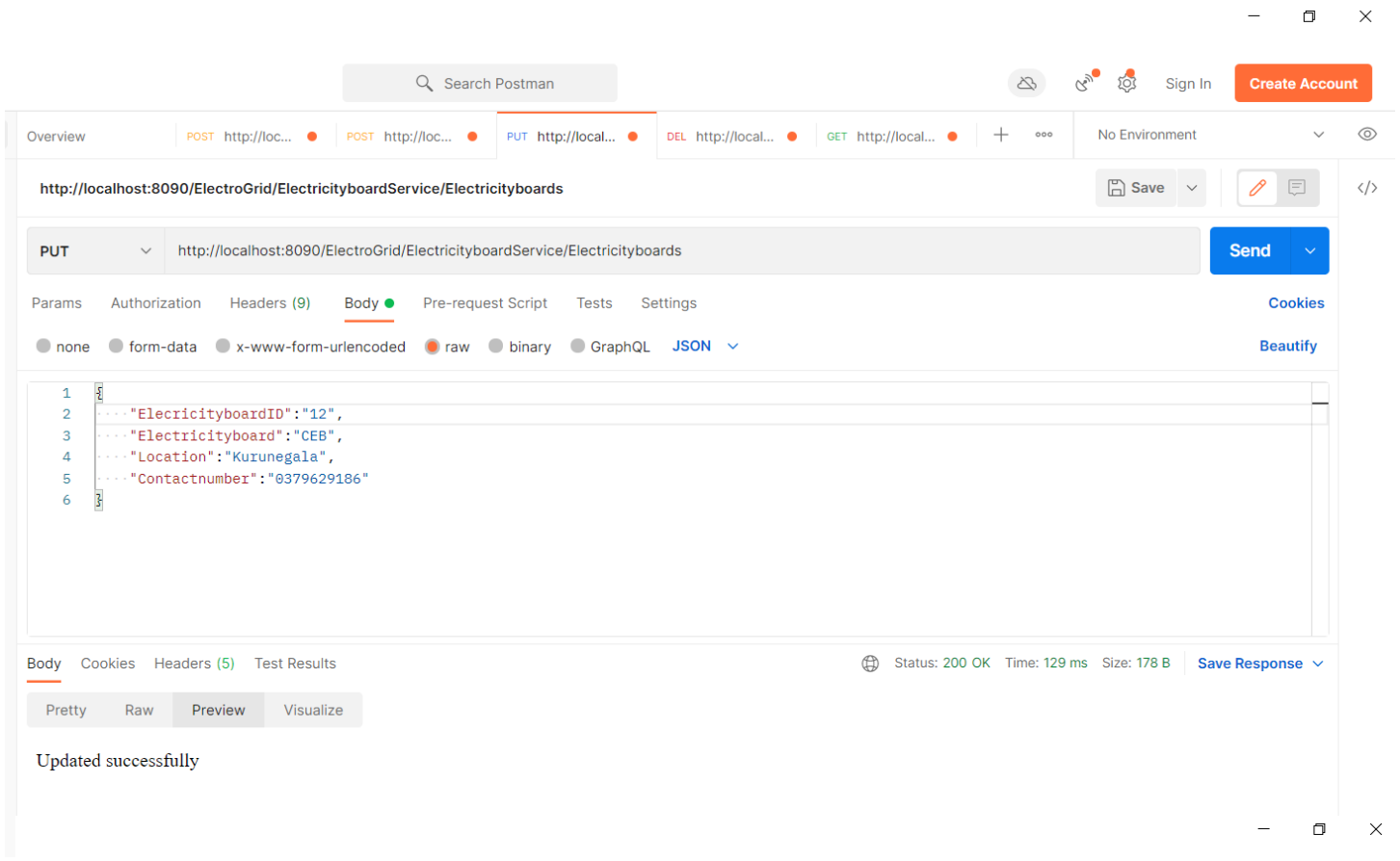
Test ID	Test Description/ Test Steps	Test Input(s)	Expected Output(s)	Actual Output(s)	Result (Pass/Fail)
1	Add Electricity board branch details	{ElectricityboardID : ""} {Electricityboard :"CEB"} {Location : "Kandy" } {Contactnumber : "0814563297"}	New Electricity board branch details are added to the database. Show message as "Inserted successfully".	New Electricity board branch details are added to the database. Show message as "Inserted successfully".	**PASS**
2	Update Electricity board branch details	{ElectricityboardID : ""} {Electricityboard :"CEB"} {Location : "Kurunegala" } {Contactnumber : "0379629186"}	Electricity board branch details are updated according to relevant input details. Show message as "Updated successfully".	Electricity board branch details are updated according to relevant input details. Show message as "Updated successfully".	**PASS**
3	Delete Electricity board branch details	<elecData> <ElectricityboardID>12</ElectricityboardID> </elecData>	Show message as "Deleted Successfully"	Electricity board branch details are deleted successfully. Show message as "Deleted Successfully"	**PASS**

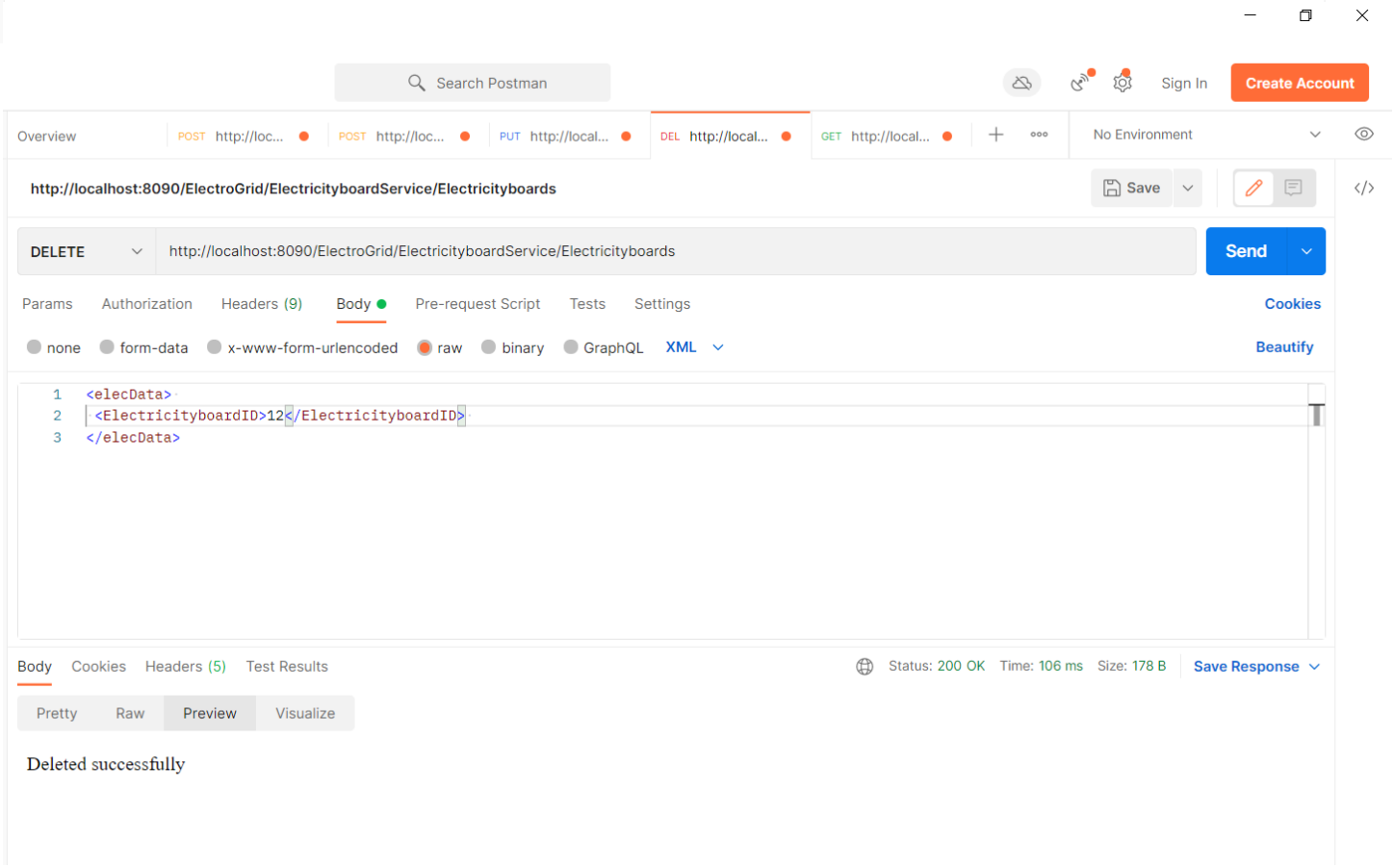
Testing



Electicity Board ID	Electicity Board	Location	Contact Number
11	LECO	Colombo	114598745




Search Postman
Sign In
Create Account
Overview
POST http://loc...
POST http://loc...
PUT http://local...
DEL http://local...
GET http://local...
No Environment
http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards
Save
PUT
http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"ElecricityboardID":"12",
"Electricityboard":"CEB",
"Location":"Kurunegala",
"Contactnumber":"0379629186"
}
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 129 ms
Size: 178 B
Save Response
Pretty
Raw
Preview
Visualize
Updated successfully


Search Postman
Sign In
Create Account
Overview
POST http://loc...
POST http://loc...
PUT http://local...
DEL http://local...
GET http://local...
No Environment
http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards
Save
DELETE
http://localhost:8090/ElectroGrid/ElectricityboardService/Electricityboards
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
XML
Beautify
<elecData>
<ElectricityboardID>12</ElectricityboardID>
</elecData>
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 106 ms
Size: 178 B
Save Response
Pretty
Raw
Preview
Visualize
Deleted successfully

Tool Used

- Java-JAX-RS (Jersey): Used for backend development
- Eclipse: Use as IDE for developing this project
- MVC Architecture: used to easily code our project
- Apache Tomcat V.09: Used as the server
- My SQL Used as our database