Sri Lanka Institute of Information Technology

Programming Applications & Frameworks (IT 3030)

3rd Year, 1st Semester

ElectroGrid (EG) Power Consumption System

Project Report

Project ID: 8

Submitted By: -

1. IT20278380 - User Management
2. IT20276232 - User Assign and Power Plant Management
3. IT20426958 - Billing Management
4. IT20243876 - Meter Reader and Management

2022/04/25

IT20278380 – J P P WICKRAMASINHA

IT20276232 – U K M C ULPATHAKUMBURA

IT20243876 – T J N PRIYADARSHANI

IT20426958 – M K H K RANATHUNGA

Table of contents

2. Member's Details

Introduction –

ElectroGrid (EG) is the company that maintains the power grid in the country. First, users are given a power plant after registering with the system as a valid user. The system also could generate customers' monthly bills and send the bill automatically to users, while customers can view their own consumption and pay the monthly bill online. This system also can accept online payments from users.

Student ID	Student Name	Workload
IT20278380	J P P WICKRAMASINHA	User Management – Once a customer registers, they are put on a pending list. Once the user registers with the system as a valid user, the customer can pay the monthly bill online based on their consumption. The user can also update their data if they need to, as well as delete any unwanted data,
IT20276232	U K M C ULPATHAKUMBURA	User Assign and Power Plant Management – This function selects only those who have registered correctly from the desired list, re-registers and inspects the relevant power plant for them. Also, the relevant power plant can be updated. If a customer is not properly registered or lacks accuracy, they can be removed from the waiting list.
IT20243876	T J N PRIYADARSHANI	Meter Reader and Management – This function is used to Meter readings. You must first select a user. You should then check the old bill of that user. If the bill is not paid, leave a comment and you will receive the meter status. The details are put in the new bill and the old bill is deleted. The bill is created with the number of units used for the new bill.
IT20426958	M K H K RANATHUNGA	Billing Management – This function calculates the monthly bill based on the number of units consumed by the customer and automatically sends the bill to the customer. Also, if a customer does not make payments for two consecutive months, the customer's electricity will be cut off. It is also possible to update the bill as and when required.

3. Git Repository

Link: - https://github.com/IT20278380/PAF_Project2022_SLIIT_GID_8/pulse

4. SE Methodology/Methods (With Justification)

What is Agile?

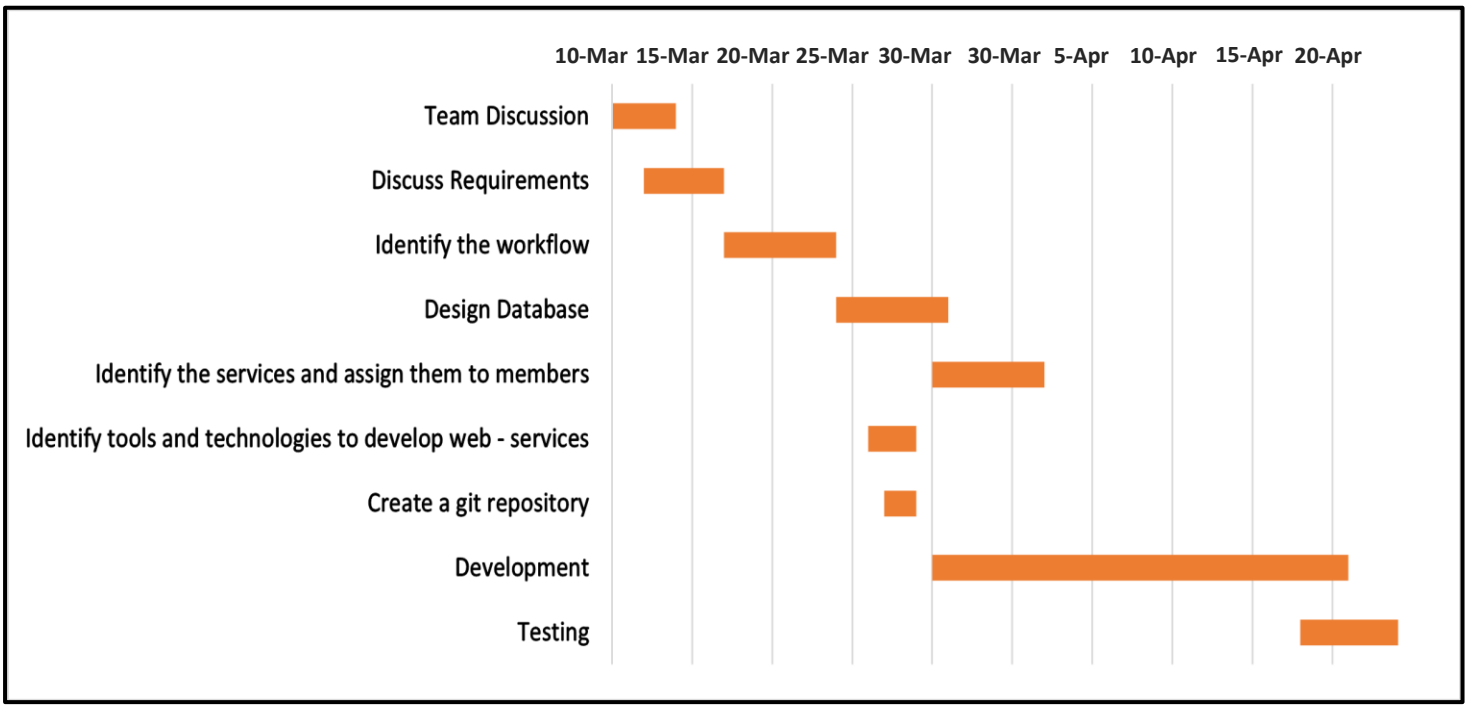
Agile Software Development is a methodology for creating a disciplined software management process that also allows for rapid changes to the development project. This methodology reduces risk by building software in short time periods known as iterations, which typically run one week to one month.

Agile development methodology has a few advantages. Agile Development Methodology's Drawbacks This methodology employs an adaptive strategy that allows it to adjust to changing client needs, among other things. Because this practice concentrates on working software rather than documentation, it may result in documentation gaps and other issues.	Agile Development Methodology's Drawbacks Because this practice concentrates on working software rather than documentation, it may result in documentation gaps and other issues.

How did we structure ourselves as a team to operate with Agile?

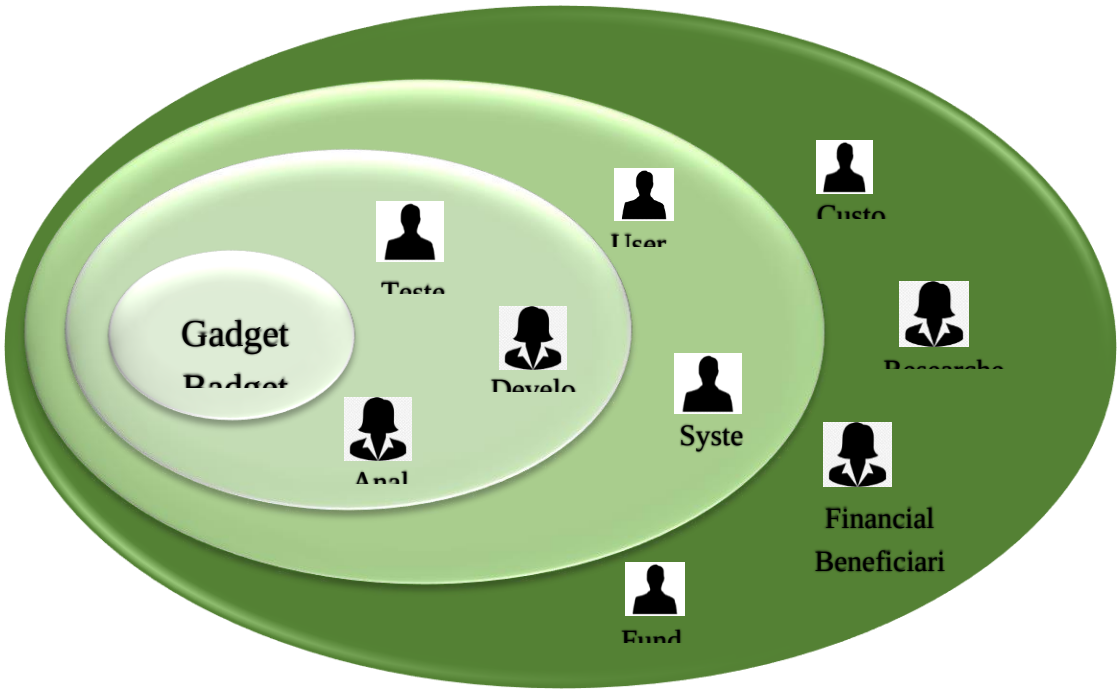
After talks (modify the requirements or accept the team's proposals), the scope of work was occasionally altered to accommodate new requirements. This is referred to as Aiglet's flexibility advantage. We divided the scenario's requirements into five micro services after recognizing them. This is another benefit of using Agile approach, which allows you to divide down work and consider microservices as tiny cycles (known as Sprints in Scrum). Because of the discussions with team members, they work closely together and have a clear understanding of their roles, and the work done within a cycle is frequently reassessed to improve the result.

5. Time Schedule



6. Requirements

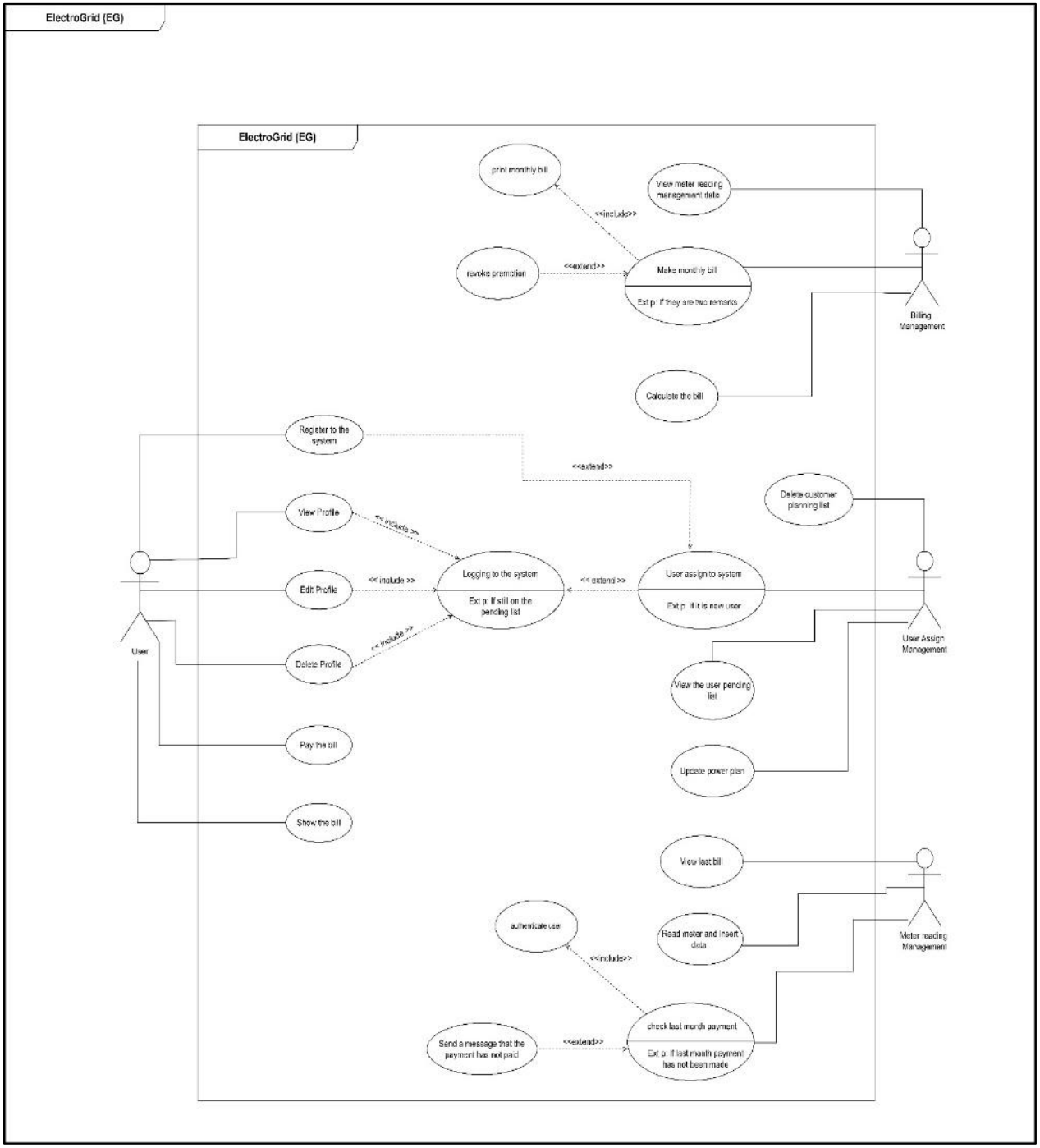
6.1. Stakeholder analysis



6.2. Requirements Analysis

- Functional Requirement
  - User Management – IT20278380
    - Register User
    - View User Profile
    - Update User
    - Delete User
    - View Bill
    - Bill Payment
  - User Assign and Power Plant Management – IT20276232
    - View Pending Users
    - Assign User in Power Plant
    - View All User
    - Update User Power Plant
    - Delete Pending User
  - Billing Management – IT20243876
    - View User All User
    - Add Meter Unit and Remark
    - View User One by One
    - Read All Bills
    - Update Red Notice
  - Meter Reader and Management – IT20426958
    - View Meter Unit in User's
    - Create Bill
    - Update Bill
    - View Non-Payed User
    - Delete Non-Payed User
- Non-Functional Requirement
  - Performance – Response time, Throughput, Utilization, Static Volumetric, User interface, Conformity
  - Security requirements - Security requirements assure that all data inside the system or its part will be protected against malware attacks or unauthorized access.
  - Software Quality Attributes
    - Availability, Maintainability, Usability, Accuracy, Stability, Correctness
- Technical Requirements
  - Technical requirements are the technical issues that must be considered to make the system successful. Table detail can be updated, deleted, and view when needed.

6.3. Requirements modelling (Use Case Diagram)

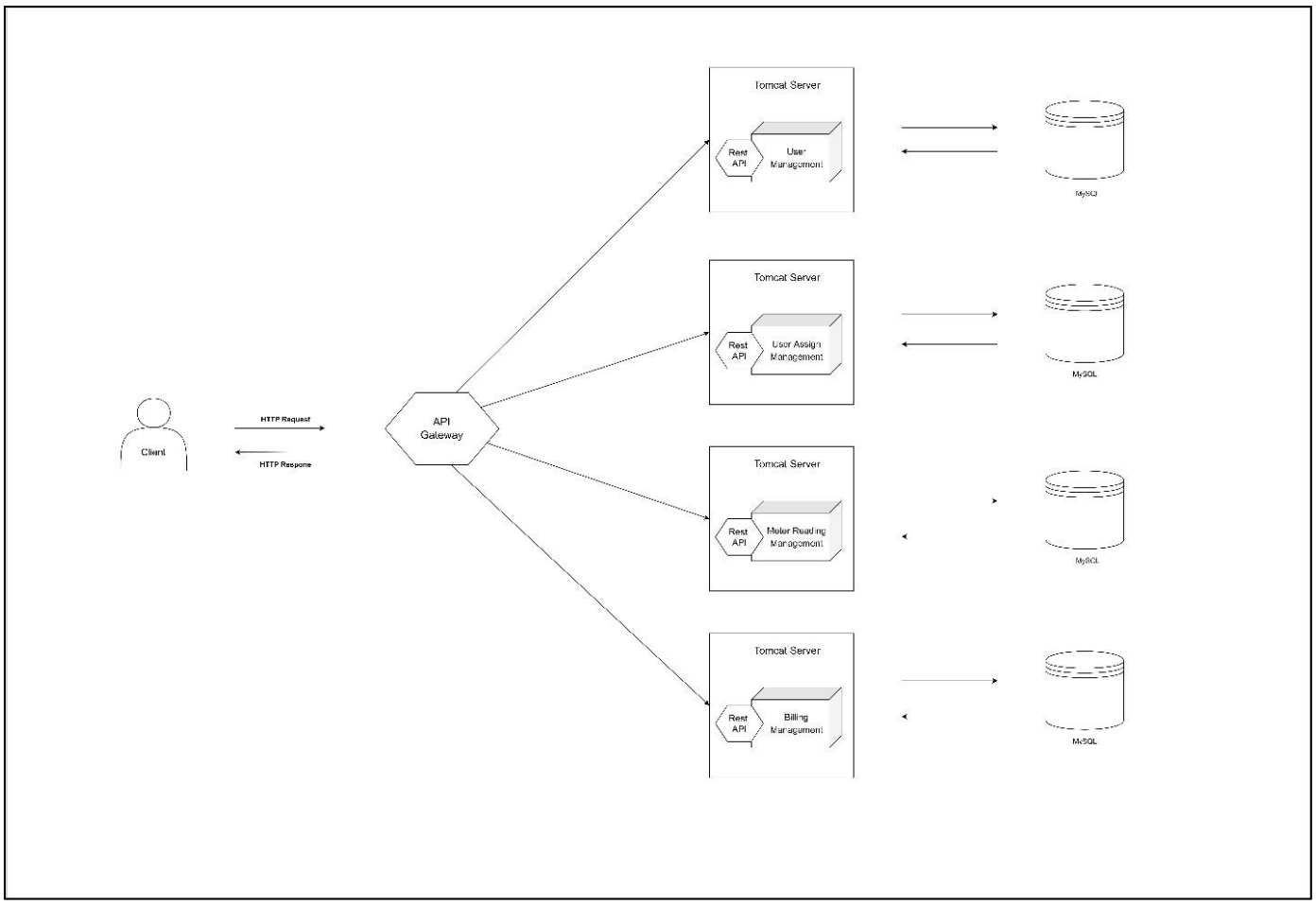


7. System's Overall Design

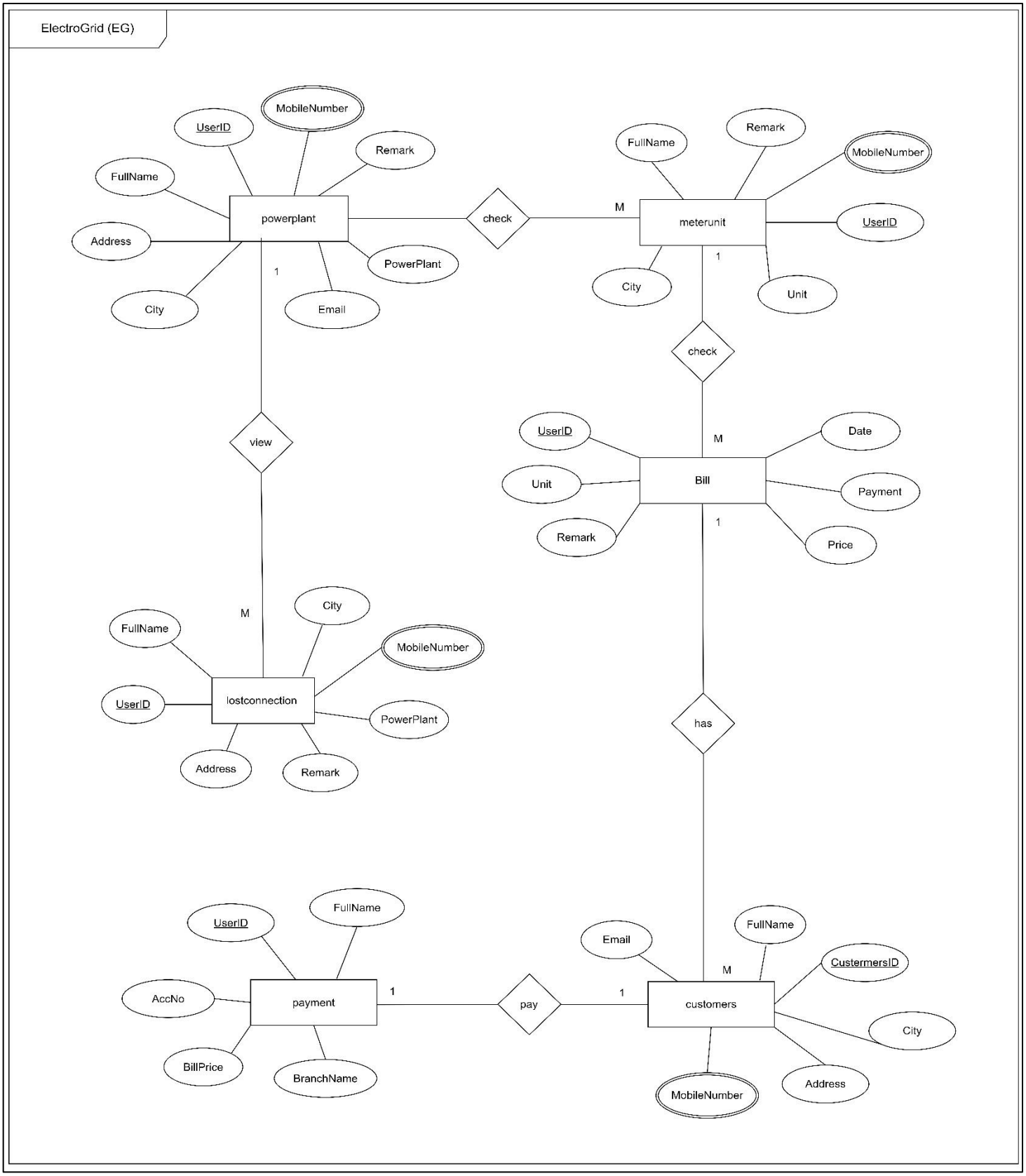
7.1. Overall architecture

ElectroGrid (EG) is the company that maintains the power grid in the country. It has the ability to monitor user's energy consumption, provide new customers with a new power plant, generate customers' monthly bills and send the bill automatically to users, and customers can make online payments. This system also has the ability to accept online payments from users.

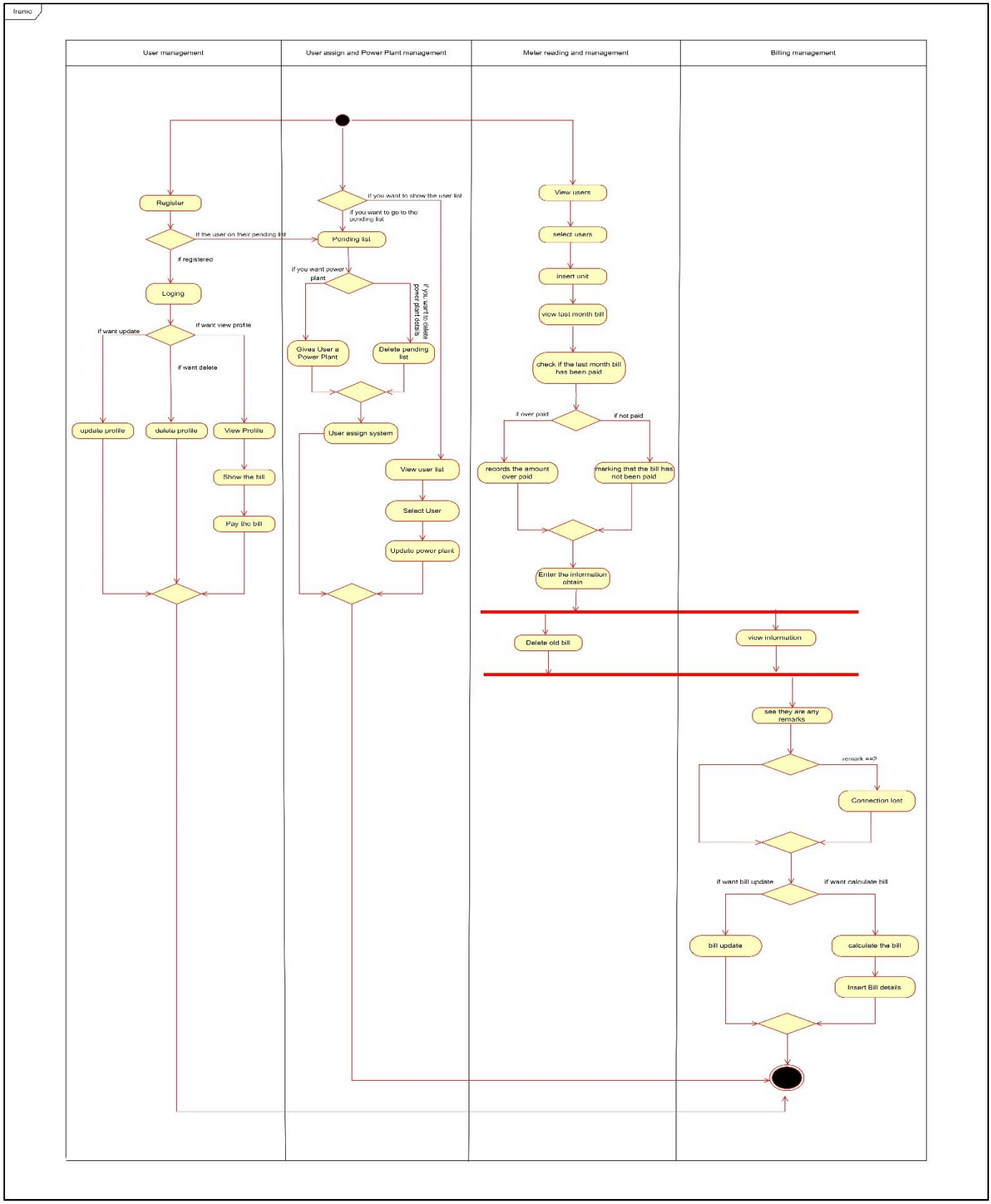
This management system consists of four function as User Management, User Assign and Power Plant Management, Meter Reader and Power Plant Management, Billing Management. Separated databases are used for each service. Also, ıan was used to take the inputs and test the output results. Here the data is passed when a client make a request it filters through a gateway to the relevant service and the respond is sent back in the same way through the gateway to the client. Application Programming Interfaces (APIs) allow these operations like improve existing services, that can work isolate and more efficiently.



7.2. Overall DB design (ER)



7.3. Activity diagrams

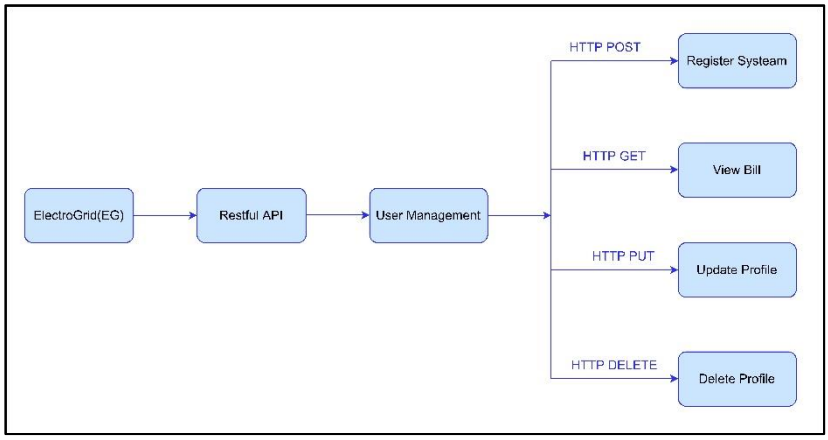


8.1. Individual Sections – IT20278380 – User Management

8.1.1. Service design

Users can register to the system using registration form by providing their details. After that users can login to the system as a valid user by providing valid credentials. The access to the system varies according to the user type.

8.1.1.1. API Of the Service



i. **Register Customer (POST)**
   - **Resource:** Customer
   - **Request:** POST ElectroG/ Customers /Add
   - **Media:** Form data - URL encoded
   - **Data:** CustomerID: “C01”, “FullName: Pamuditha”, Address: “No,10/8 Colombo”, City: “Colombo”, MobileNumber: “0711584935”, Email: “wickramasinha1219@gmail.com”
   - **Response:** Inserted successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers /Add

ii. **Update User Profile (PUT)**
   - **Resource:** PowerPlant
   - **Request:** PUT ElectroG/Customers /Update
   - **Media:** Form data – Application JSON

```
UserData {
    "UserID": "P1_05",
    "FullName": "Chathuranga",
    "Address": "10/85, Colombo",
    "City": "Colombo",
    "MobileNumber": "0711584935",
    "Email": "wickramasinha1219@gmail.com"
}
```

   - **Response:** Update successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers /Update

iii. View User Profile (GET)

- **Resource:** PowerPlant
- **Request:** GET ElectroG/Customers/One

```
<CustomerData>
    </UserID>P4_05</UserID>
</CustomerData>
```

- **Media:** Form Data
- **Response:** HTML table with all attributes in the User table
- **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers /One

iv. Delete User Profile (DELETE)

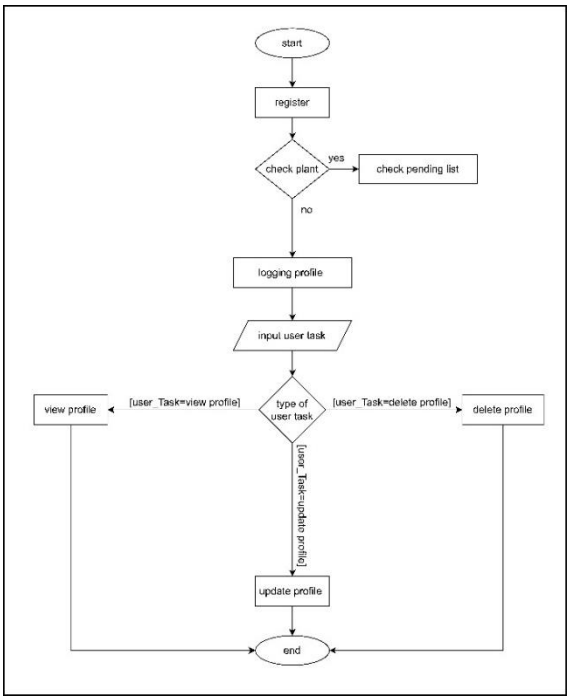
- **Resource:** PowerPlant
- **Request:** DELETE
- **Media:** Application XML

```
<CustomerData>
    </UserID>P4_05</UserID>
</CustomerData>
```

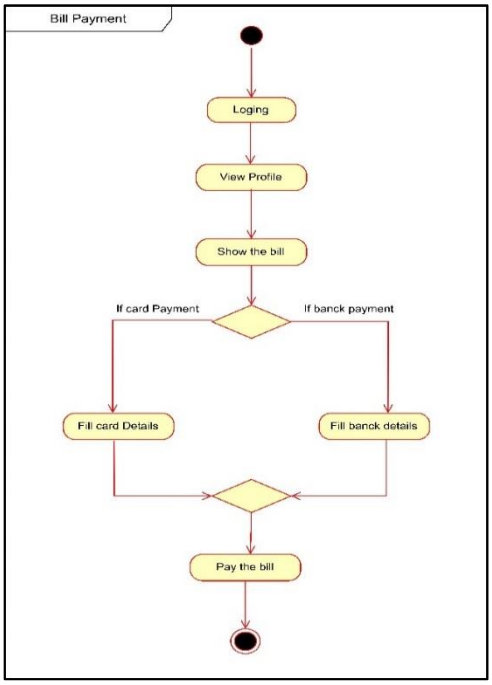
- **Response:** Deleted successfully
- **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers /Delete

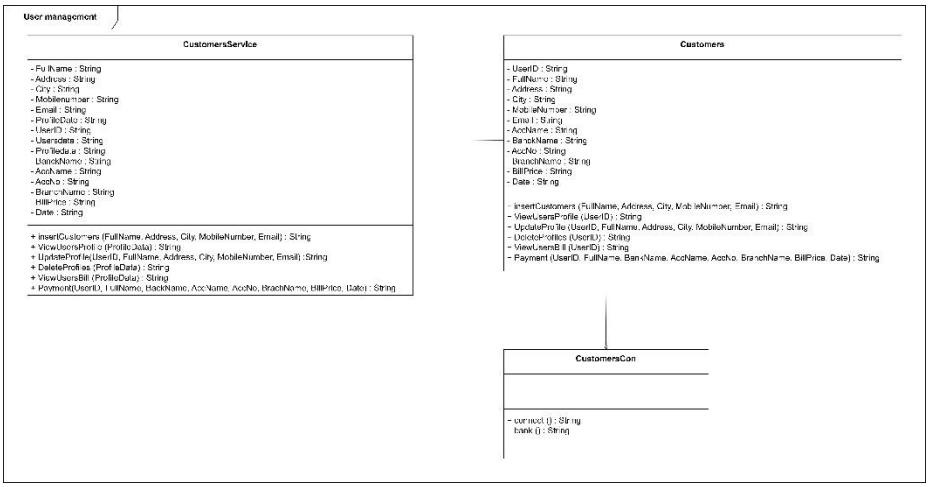
8.1.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)

- Flow Chart

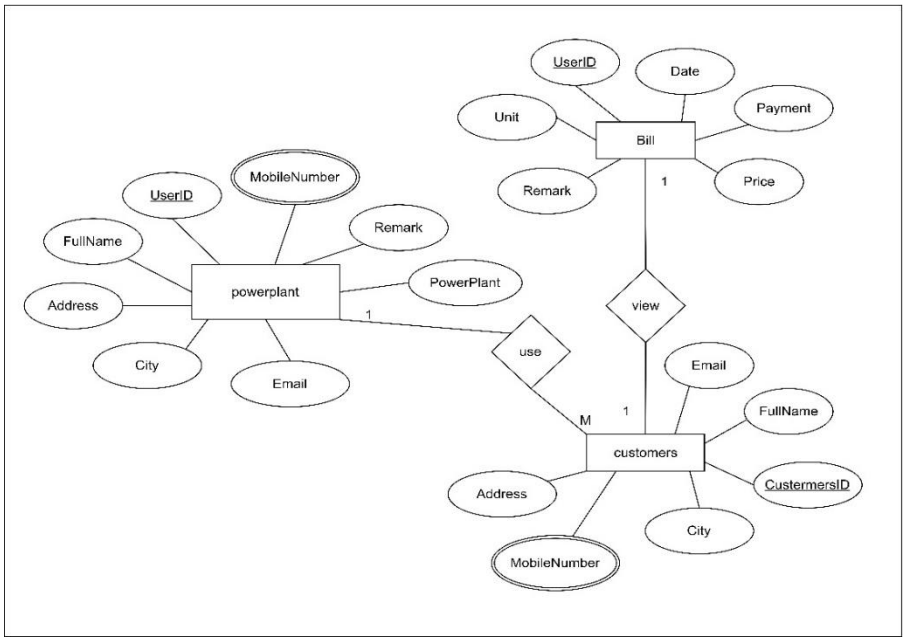


- Activity Diagram / Class Diagram





8.1.1.3. Database for the service (ER)



8.1.2. Service development and testing.

8.1.2.1. Tools used

- **Dependency Management Tool:** Maven
- **Testing Tool:** Postman
- **Version Control System:** Git
- **IDE:** eclipse
- **Programming Language:** Jersey framework (JAX-RS)
- **Programming Language:** Java
- **Database:** phpMyAdmin (MySQL)
- **Server:** Apache Tomcat Server

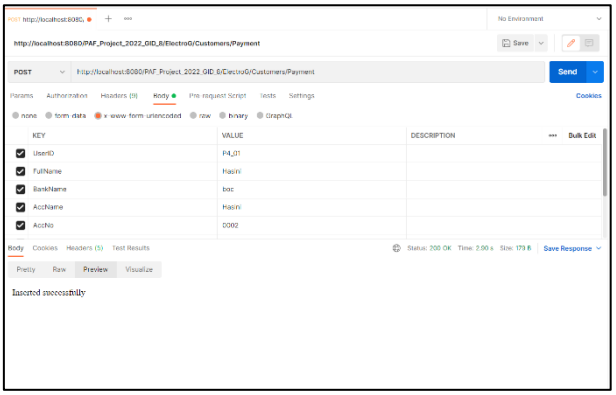
8.1.2.2. Testing Methodology and Results.

Test ID	Description	Input	Excepted Output	Actual Output	Result
1	Register	UserID: “P1_05”, FullName: “Chathuranga”, Address: “10/85, Colombo”, City: “Colombo”, MobileNumber: “0711584935”, Email: “wickramasinha1219@gmail.com”	Inserted Successfully	Inserted Successfully	Pass
2	View User Profile	UserID: “P1_05”	Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass
3	Update User Profile	UserID: “P1_05”, FullName: “Pamuditha”, Address: “10/85, Colombo”, City: “Colombo”, MobileNumber: “0711584935”, Email: “wickramasinha1219@gmail.com”	Update Successfully	Update Successfully	Pass
4	Delete User Profile	UserID: “P1_05”	Delete Successfully	Delete Successfully	Pass
5	View User Bill	UserID: “P1_05”	Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass
6	Bill Payment	UserID: “P1_05”, FullName: “Pamuditha”, Address: “10/85, Colombo”, City: “Colombo”, MobileNumber: “0711584935”, Email: “wickramasinha1219@gmail.com”	Payment Successfully	Payment Successfully	Pass

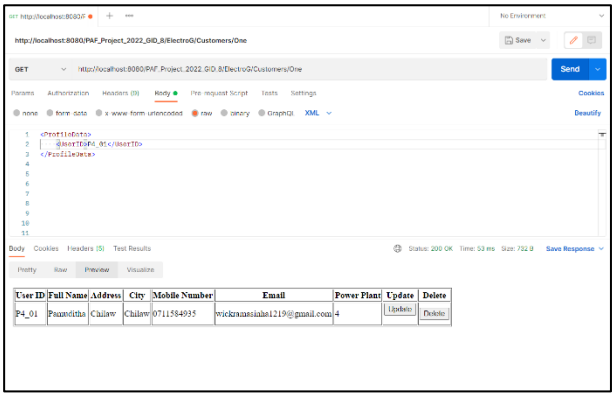
8.1.3. Assumptions and any other details

Postman screenshots

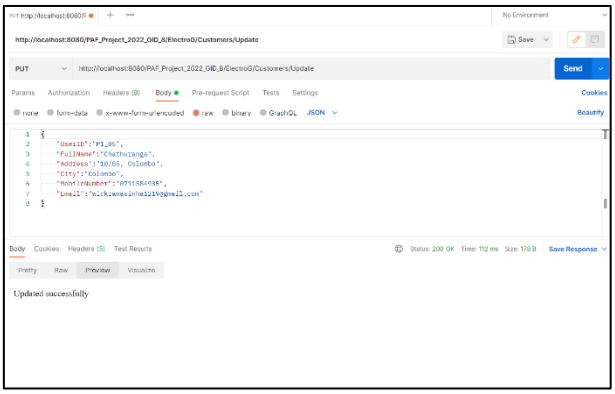
Payment



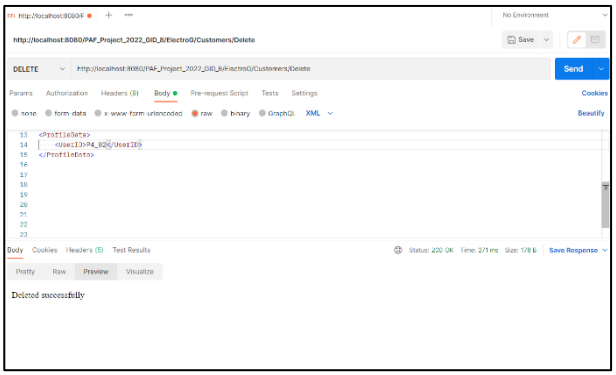
View Profile



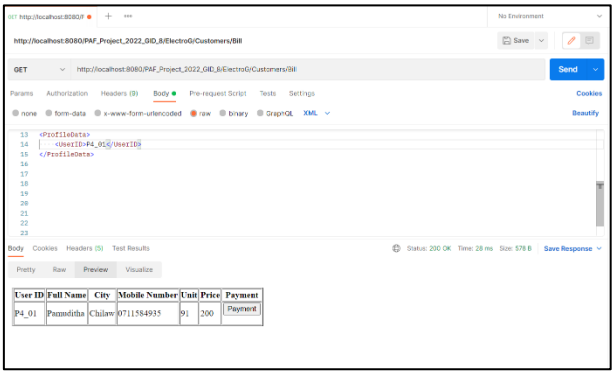
Update Profile



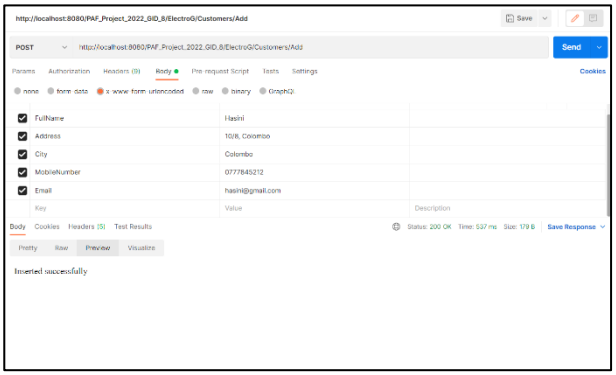
Delete Profile



View Bill



Register

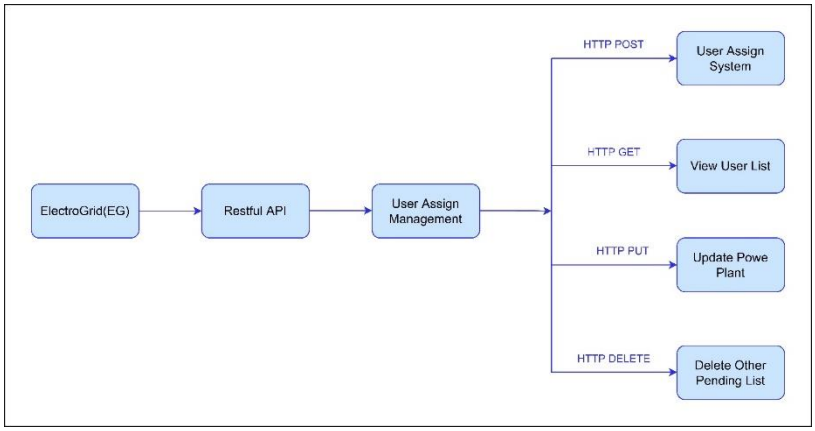


8.2. Individual Sections – IT20276232 – User Assign and Power Pant Management

8.2.1. Service design

This function selects only those who have registered correctly from the desired list, re-registers and inspects the relevant power plant for them. Also, the relevant power plant can be updated. If a customer is not properly registered or lacks accuracy, they can be removed from the waiting list.

8.2.1.1. API Of the Service



i. **Register Customer (POST)**
   - **Resource:** PowerPlant
   - **Request:** POST ElectroG/UserAssing/Add
   - **Media:** Form data - URL encoded
   - **Data:** UserID: “P4_01”, FullName: “Pamuditha”, Address: “No,10/8 Colombo”, City: “Colombo”, MobileNumber: “0711584935”, Email: “~~wickramasinha1219@gmail.com~~wickramasinha1219@gmail.com”, PowerPlant: “1”
   - **Response:** Inserted successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/UserAssing/Add

ii. **Update User Profile (PUT)**
   - **Resource:** PowerPlant
   - **Request:** PUT ElectroG/UserAssing /Update
   - **Media:** Form data – Application JSON

```
UserData {
    "UserID": "P1_05",
    "PowerPlant": "1"
}
```

   - **Response:** Update successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/UserAssing /Update

iii. **View User Profile (GET)**
   - **Resource:** Customers

- **Request:** GET ElectroG/UserAssing/Pending
- **Media:** Form Data
- **Response:** HTML table with all attributes in the User table
- **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/UserAssing/Pending

iv. Delete User Profile (DELETE)

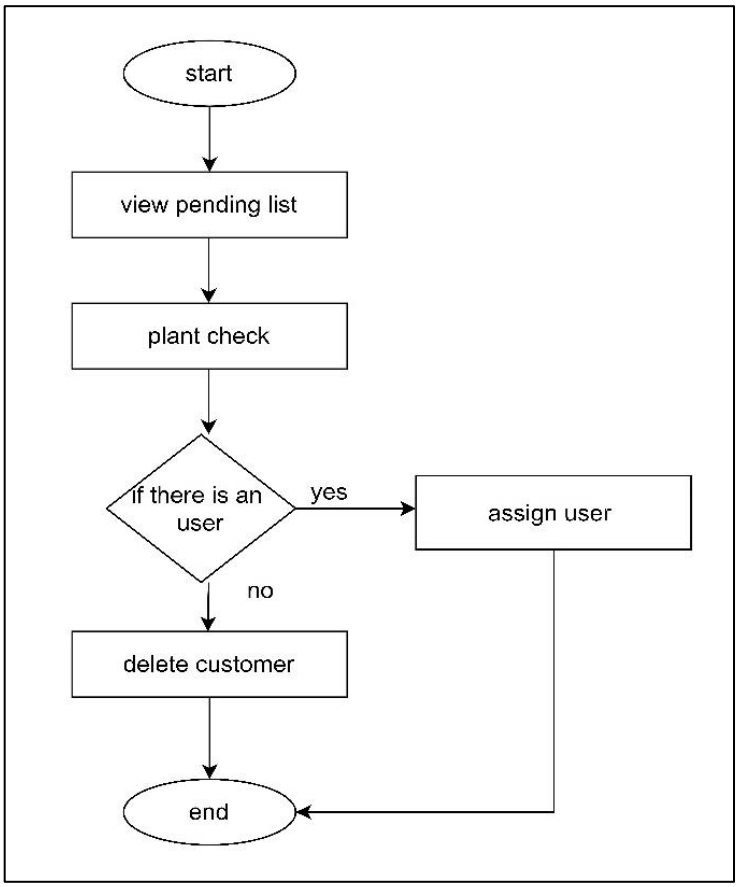
- **Resource:** Customers
- **Request:** DELETE
- **Media:** Application XML

```
<UserData>
    </UserID>P4_05</UserID>
</UserData>
```

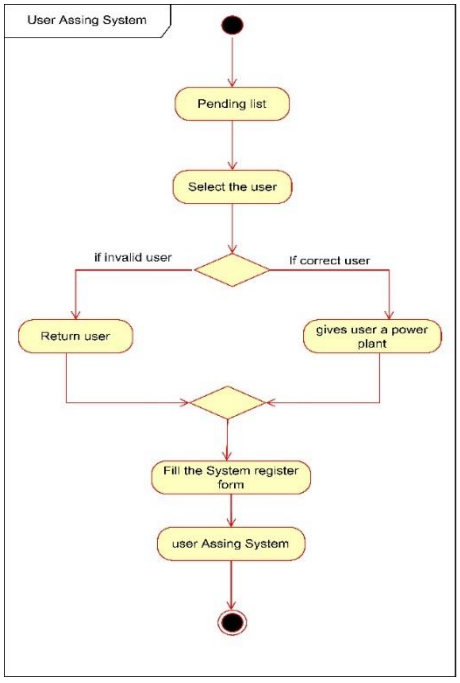
- **Response:** Deleted successfully
- **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/UserAssing/Delete

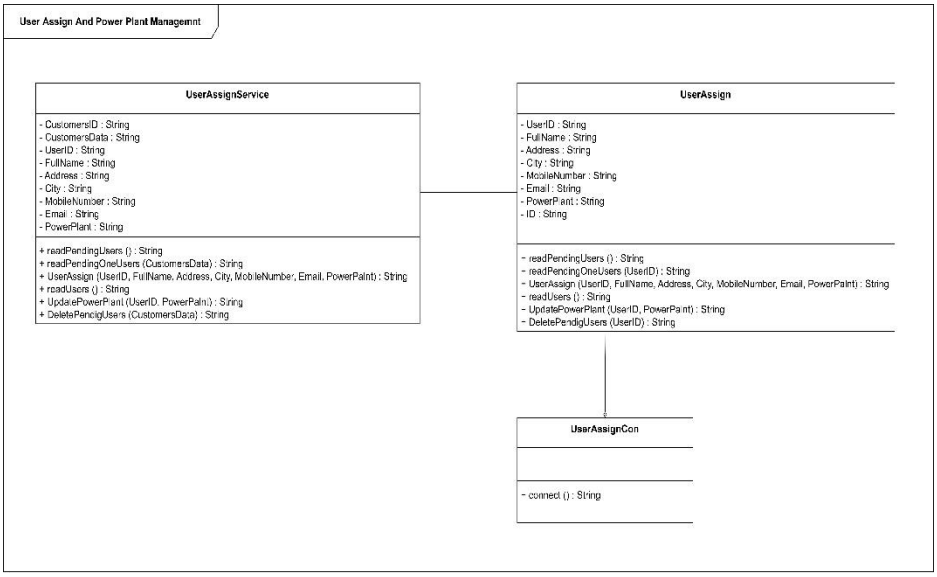
8.2.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)

- Flow Chart

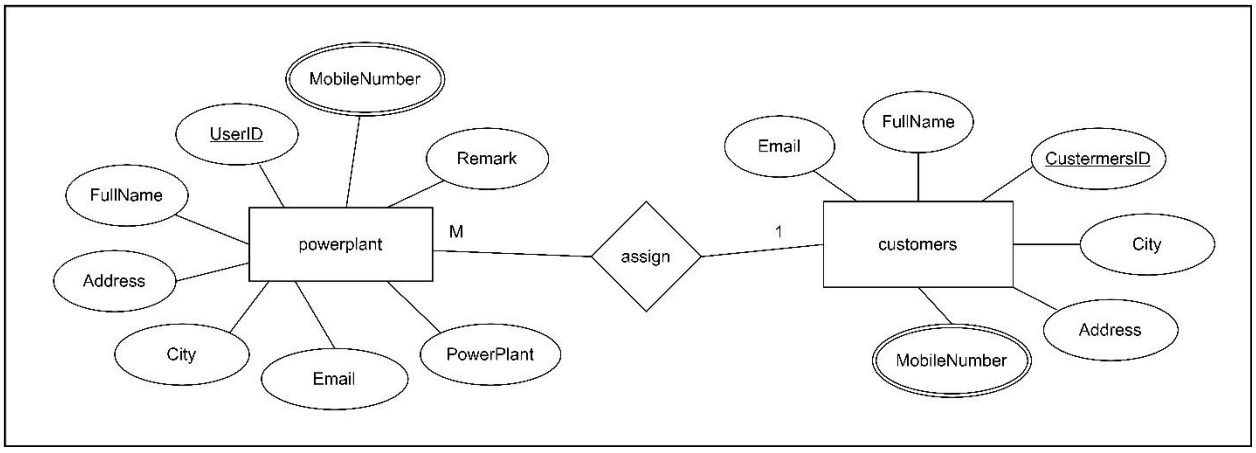


- Activity Diagram / Class Diagram





8.2.1.3. Database for the service (ER)



8.2.2. Service development and testing.

8.2.2.1. Tools used

- **Dependency Management Tool:** Maven
- **Testing Tool:** Postman
- **Version Control System:** Git
- **IDE:** eclipse
- **Programming Language:** Jersey framework (JAX-RS)
- **Programming Language:** Java
- **Database:** phpMyAdmin (MySQL)
- **Server:** Apache Tomcat Server

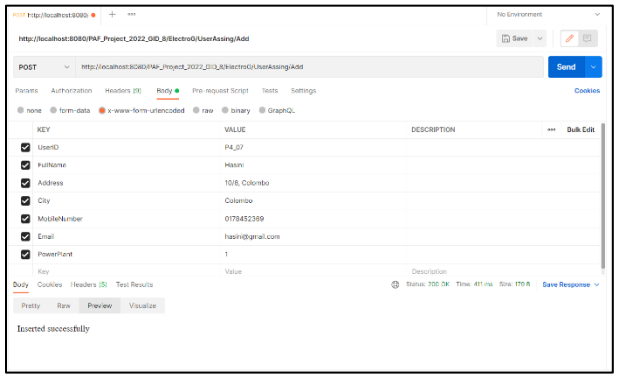
8.2.2.2. Testing Methodology and Results.

Test ID	Description	Input	Excepted Output	Actual Output	Result
1	Insert PowerPlant	UserID: “P1_05”, FullName: “Chathuranga”, Address: “10/85, Colombo”, City: “Colombo”, MobileNumber: “0711584935”, Email: “wickramasinha1219@gmail.com”, PowerPlant: “1”	Inserted Successfully	Inserted Successfully	Pass
2	View Pending Customers		Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass
3	Update User Profile	UserID: “P1_05”, PowerPlant: “2”,	Update Successfully	Update Successfully	Pass
4	Delete User Profile	UserID: “P1_05”,	Delete Successfully	Delete Successfully	Pass
5	Get All Users		Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass

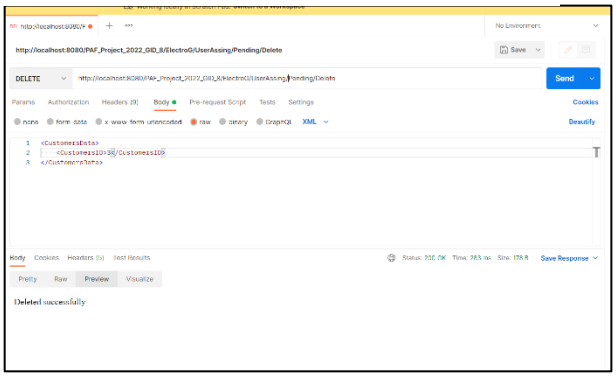
8.2.3. Assumptions and any other details.

Postman screenshots

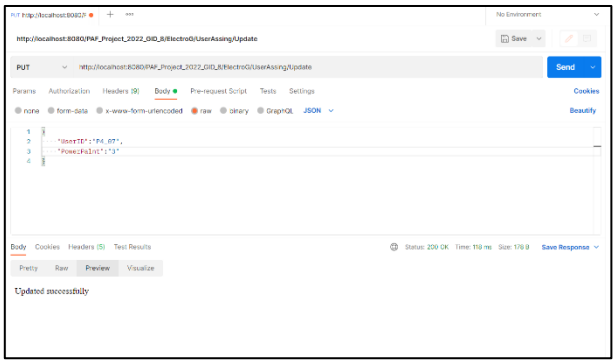
Add Users



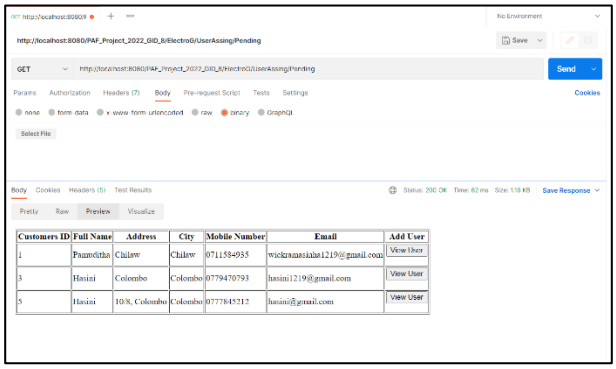
Delete Pending User



Update Power Plant



View Pending Users

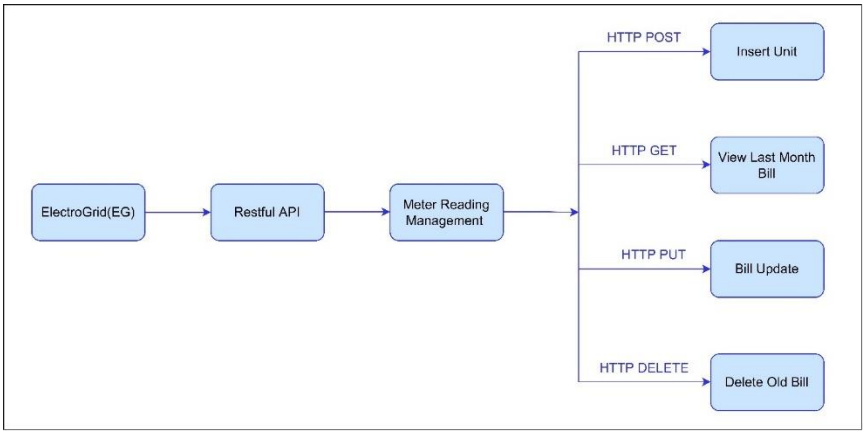


8.3. Individual Sections – IT20243876 – Meter Reader and Management

8.3.1. Service design

This function is used to Meter readings. You must first select a user. You should then check the old bill of that user. If the bill is not paid, leave a comment and you will receive the meter status. The details are put in the new bill and the old bill is deleted. The bill is created with the number of units used for the new bill

8.3.1.1. API Of the Service



i. **Register Customer (POST)**
   - **Resource:** MeterUnit
   - **Request:** POST ElectroG/MeterReader/Add
   - **Media:** Form data - URL encoded
   - **Data:** UserID: "P4_01", FullName: "Pamuditha", City: "Colombo", MobileNumber: "0711584935", Unit: "1", Remark: No
   - **Response:** Inserted successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/MeterReader/Add

ii. **Update User Profile (PUT)**
   - **Resource:** PowerPlant
   - **Request:** PUT ElectroG/MeterReader/Update
   - **Media:** Form data – Application JSON

```
UserData {
    "UserID": "P1_05",
    "Remark": "Not Pay Bill"
}
```

   - **Response:** Update successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/MeterReader/Update

iii. **View User Profile (GET)**
   - **Resource:** PowerPant
   - **Request:** GET ElectroG/MeterReader/
   - **Media:** Form Data
   - **Response:** HTML table with all attributes in the User table
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/MeterReader/One

iv. **Delete User Profile (DELETE)**
   - **Resource:** Bill

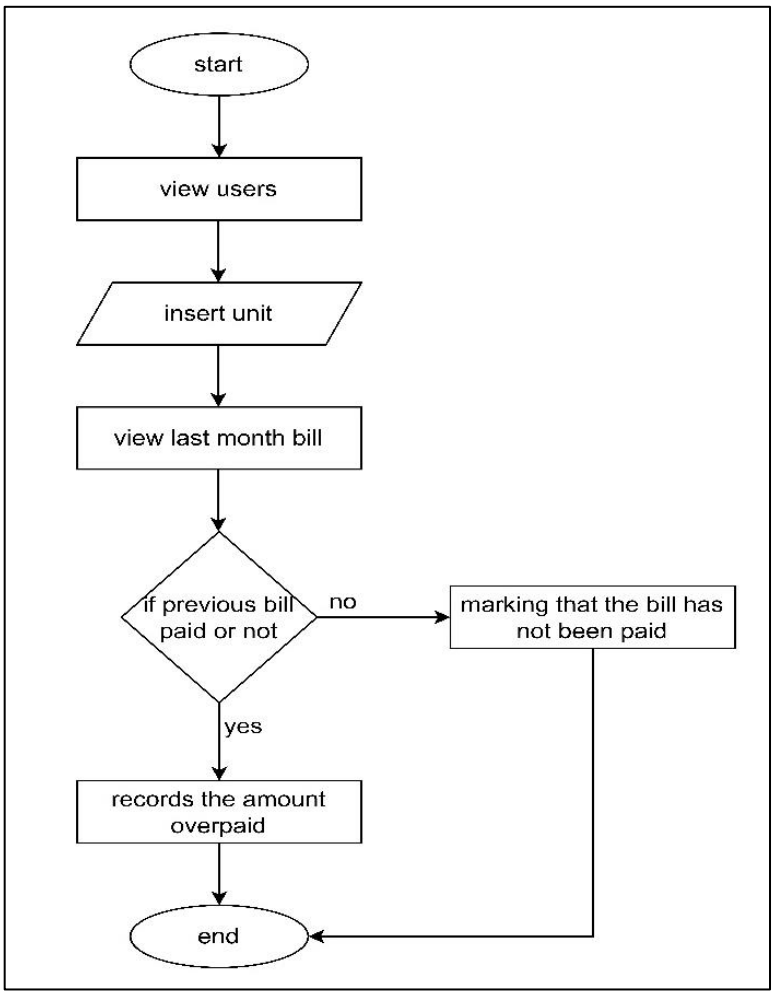
- **Request:** DELETE
- **Media:** Application XML

```
<UserData>
    </UserID>P4_05</UserID>
</UserData>
```

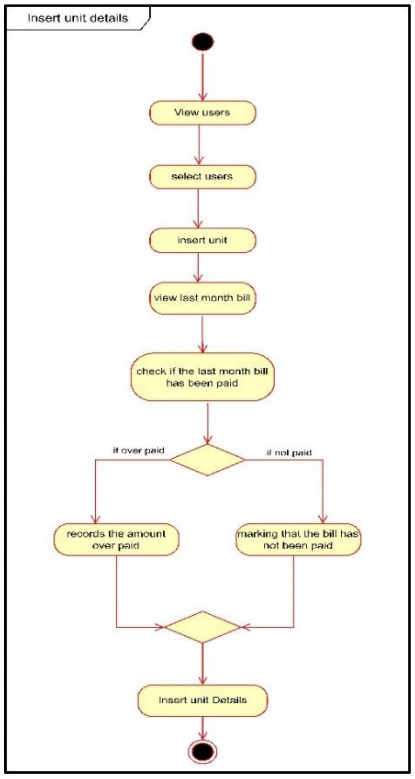
- **Response:** Deleted successfully
- **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/MeterReader/Delete

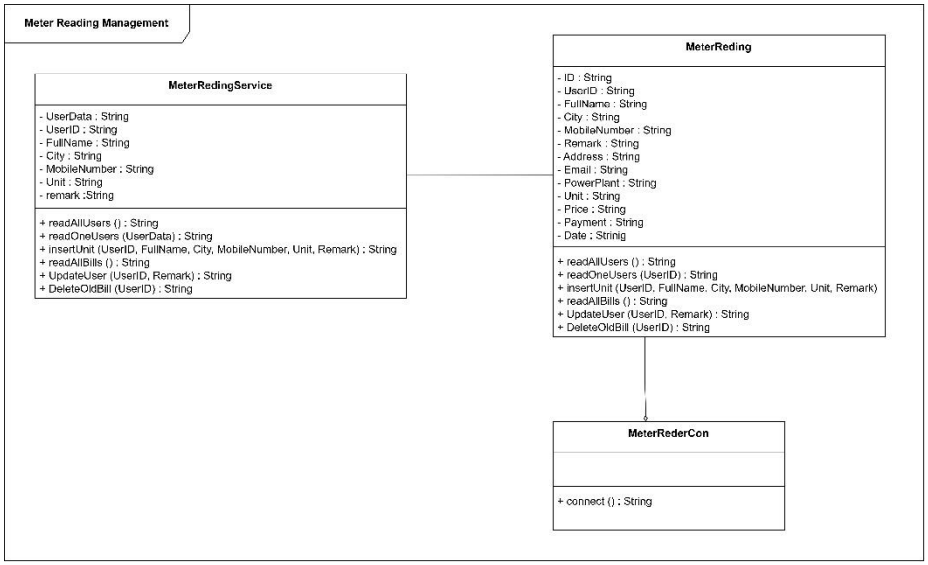
8.3.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)

- Flow Chart

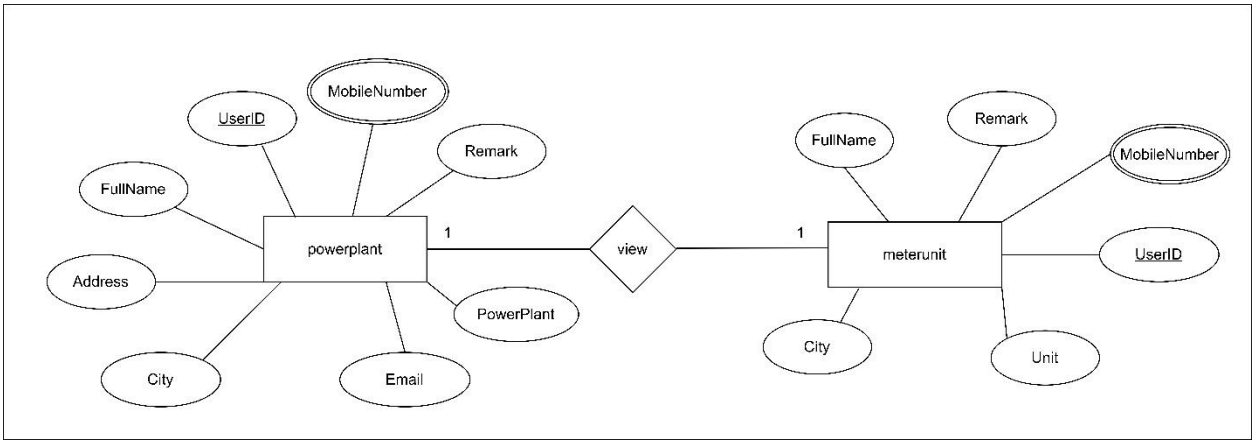


- Activity Diagram / Class Diagram





8.3.1.3. Database for the service (ER)



8.3.2. Service development and testing.

8.3.2.1. Tools used

- **Dependency Management Tool:** Maven
- **Testing Tool:** Postman
- **Version Control System:** Git
- **IDE:** eclipse
- **Programming Language:** Jersey framework (JAX-RS)
- **Programming Language:** Java
- **Database:** phpMyAdmin (MySQL)
- **Server:** Apache Tomcat Server

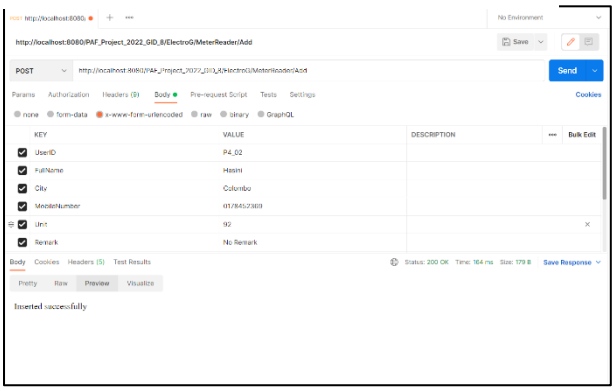
8.3.2.2. Testing Methodology and Results.

Test ID	Description	Input	Excepted Output	Actual Output	Result
1	Insert Meter Unit	UserID: “P1_05”, FullName: “Chathuranga”, City: “Colombo”, MobileNumber: “0711584935”, Unit: “25”, Remark: “No Remark”	Inserted Successfully	Inserted Successfully	Pass
2	View Users		Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass
3	Update User Profile Remark	UserID: “P1_05”, Remark: “Not Pay Bill”	Update Successfully	Update Successfully	Pass
4	Delete User Old Bill	UserID: “P1_05”	Delete Successfully	Delete Successfully	Pass
5	View Old Bill		Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass

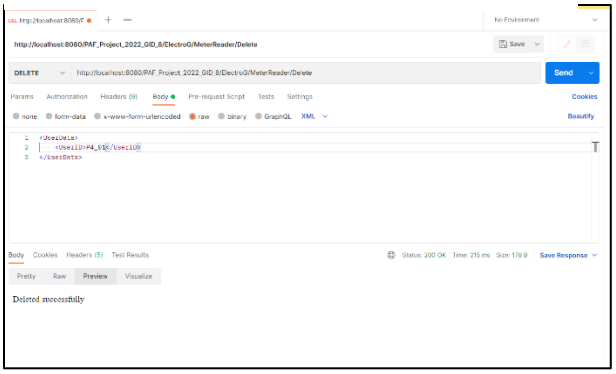
8.3.3. Assumptions and any other details

Postman screenshots

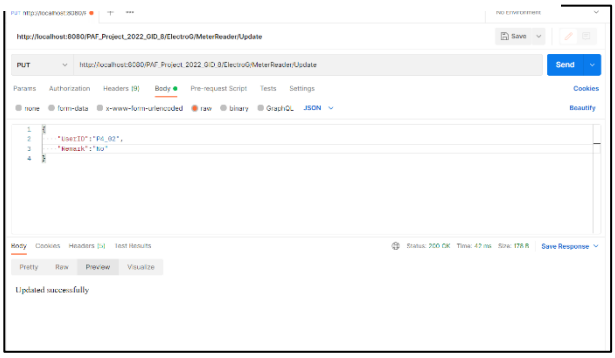
Add Meter Unit



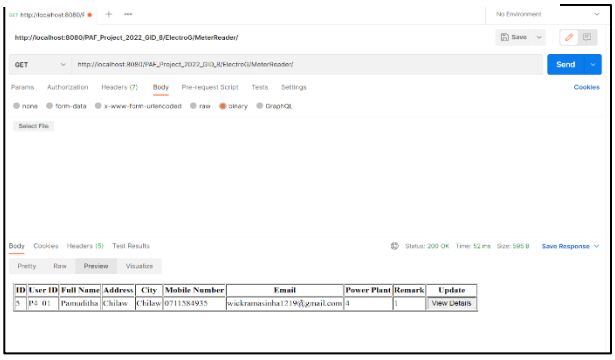
Delete Old Bills



Update User Remark



View Users

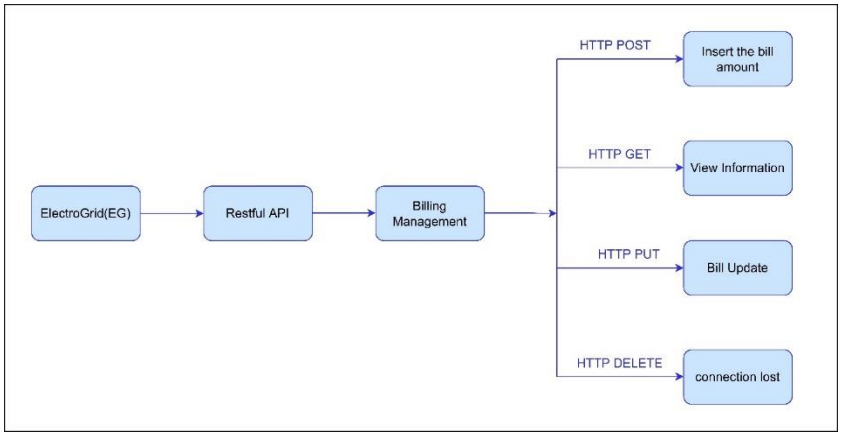


8.4. Individual Sections – IT20426958 – Billing Management

8.4.1. Service design

This function calculates the monthly bill based on the number of units consumed by the customer and automatically sends the bill to the customer. Also, if a customer does not make payments for two consecutive months, the customer's electricity will be cut off. It is also possible to update the bill as and when required

8.4.1.1. API Of the Service



i. **Register Customer (POST)**
   - **Resource:** Bill
   - **Request:** POST ElectroG/BillCreation/Add
   - **Media:** Form data - URL encoded
   - **Data:** UserID: “P4_01”, FullName: “Pamuditha”, City: “Colombo”, MobileNumber: “0711584935”, Unit: “1”, Remark: “No”, Price1_30: “5”, Price30_60: “10”, Price60_90: “20”, Price90: “40”
   - **Response:** Inserted successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/BillCreation/Add

ii. **Update User Profile (PUT)**
   - **Resource:** Bill
   - **Request:** PUT ElectroG/BillCreation/Update
   - **Media:** Form data – Application JSON

```
UserData {
    "UserID": "P1_05",
    "Price": "5320.0",
    "Remark": "Not Pay Bill"
```

   - **Response:** Update successfully
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/BillCreation/Update

iii. **View User Profile (GET)**
   - **Resource:** MeterUnit
   - **Request:** GET ElectroG/BillCreation/MeterReader
   - **Media:** Form Data
   - **Response:** HTML table with all attributes in the User table
   - **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/BillCreation/MeterReader/

iv. Delete User Profile (DELETE)

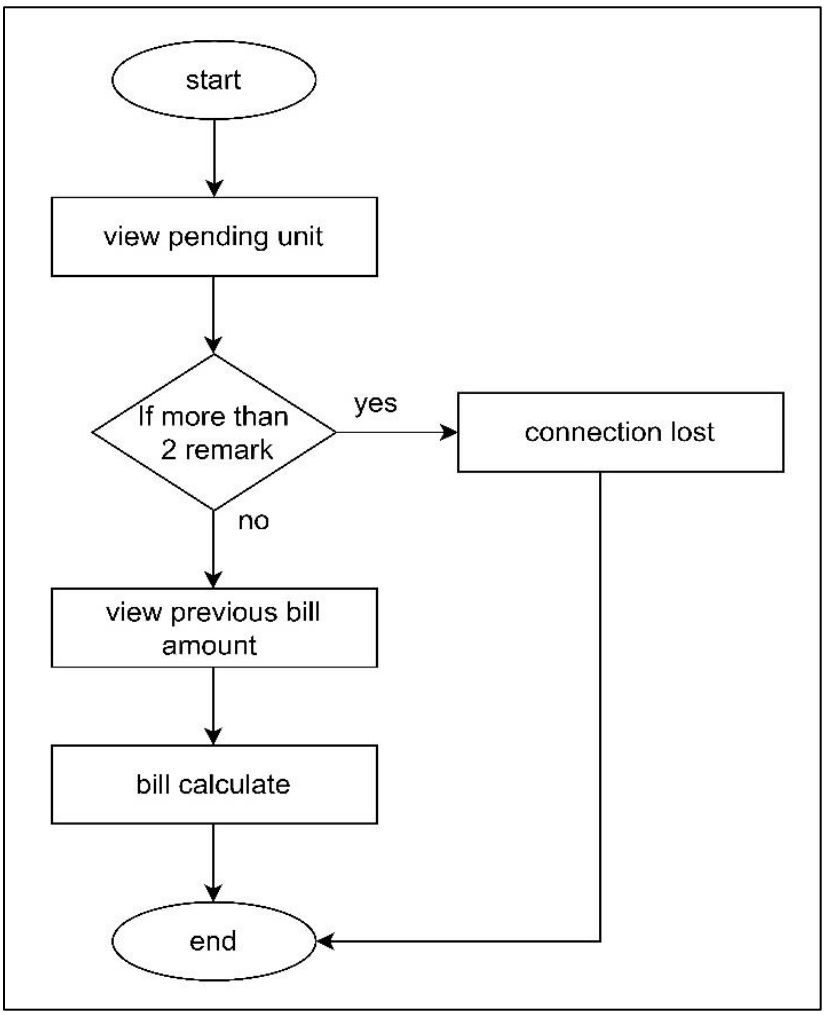
- **Resource:** PowerPlant
- **Request:** DELETE
- **Media:** Application XML

```
<UserData>
    </UserID>P4_05</UserID>
</UserData>
```

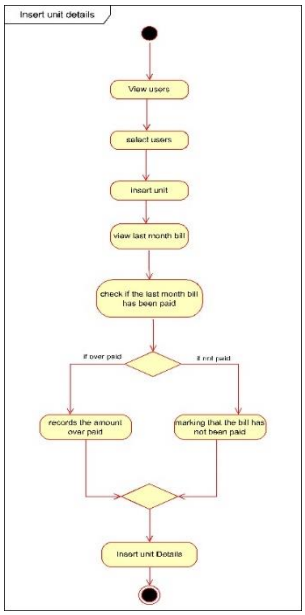
- **Response:** Deleted successfully
- **URL:** http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/BillCreation/Delete

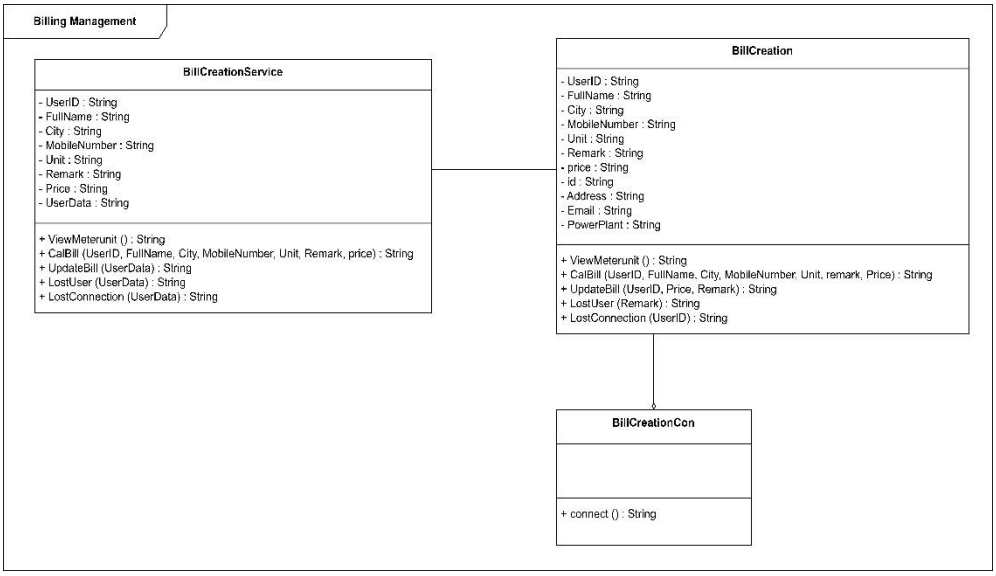
8.4.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)

- Flow Chart

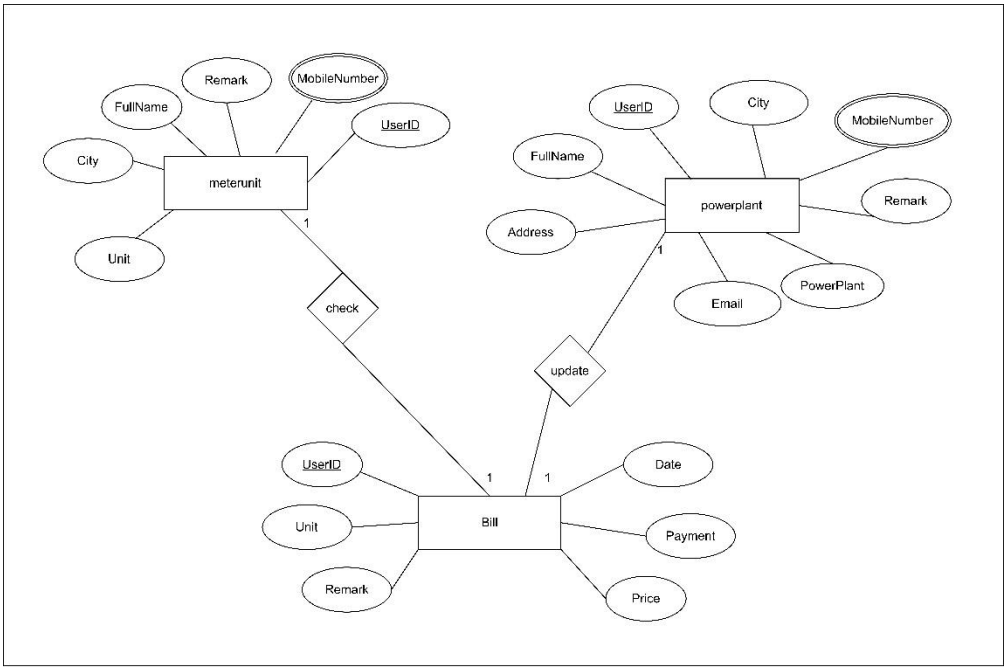


- Activity Diagram / Class Diagram





8.4.1.3. Database for the service (ER)



8.4.2. Service development and testing.

8.4.2.1. Tools used

- **Dependency Management Tool:** Maven
- **Testing Tool:** Postman
- **Version Control System:** Git
- **IDE:** eclipse
- **Programming Language:** Jersey framework (JAX-RS)
- **Programming Language:** Java
- **Database:** phpMyAdmin (MySQL)
- **Server:** Apache Tomcat Server

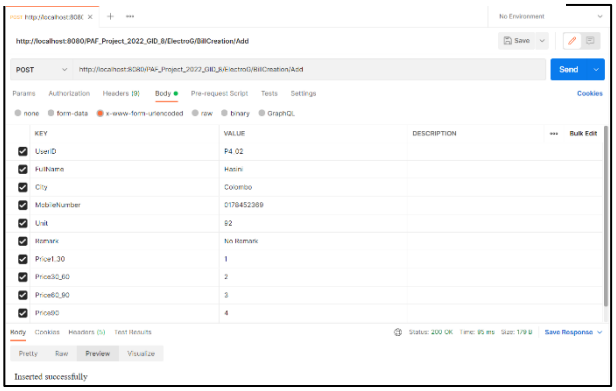
8.4.2.2. Testing Methodology and Results.

Test ID	Description	Input	Excepted Output	Actual Output	Result
1	Create Bill	UserID: “P1_05”, FullName: “Chathuranga”, City: “Colombo”, MobileNumber: “0711584935”, Unit: “25”, Remark: “No Remark”, Price1_30: “5”, Price30_60: “10”, Price60_90: “20”, Price90: “40”,	Inserted Successfully	Inserted Successfully	Pass
2	View User’s Unit		Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass
3	Update User Bill	UserID: “P1_05”, Price: “5462.0”, Remark: “Not Pay Bill”	Update Successfully	Update Successfully	Pass
4	Delete User Not Pay Users	UserID: “P1_05”	Delete Successfully	Delete Successfully	Pass
5	View Not Pay Users	Remark: “2 Month”	Display a HTML table with all the attributes in funds table	Display a HTML table with all the attributes in funds table	Pass

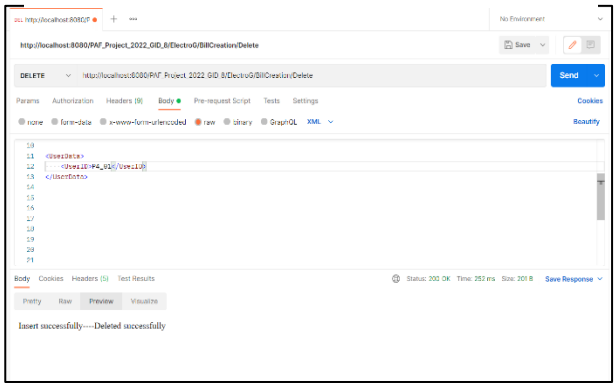
8.4.3. Assumptions and any other details

Postman screenshots

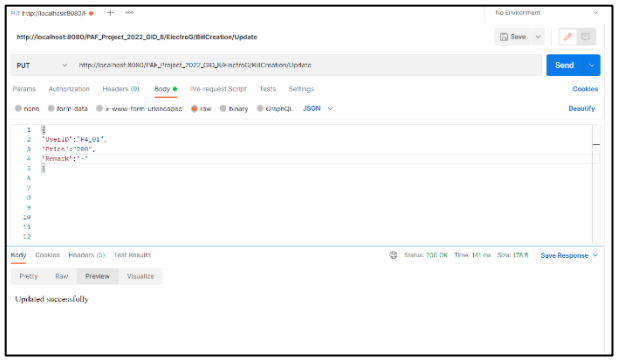
Create Bill



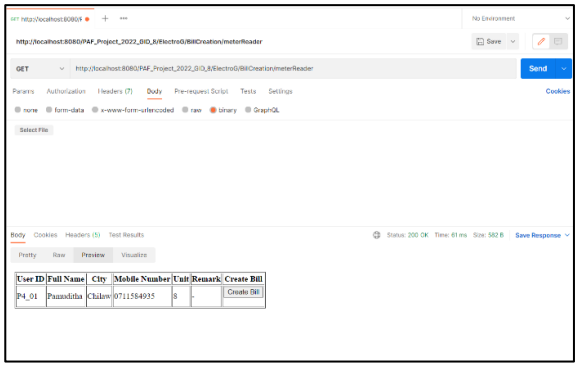
Delete Not Pay User



Update Bill

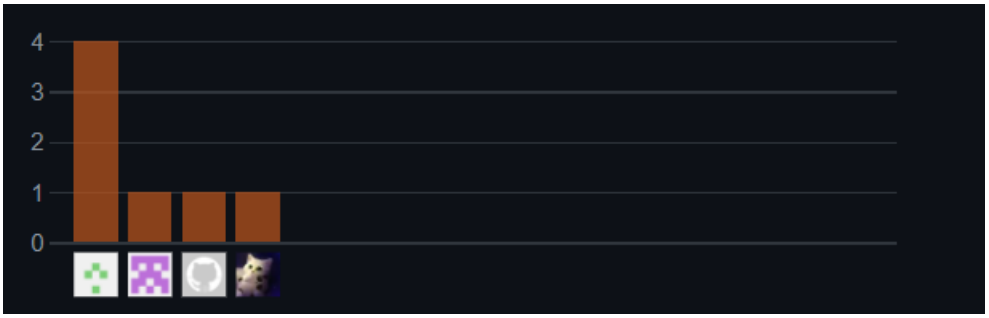


View Meter Reding Unit



9. Assumptions and Any Other Details

9.1. Git Commit Log



9.2. Reference

SE Methodologies : https://acodez.in/12-best-software-development-methodologies-pros-cons/