

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 4](#_Toc101893190)

[3. Git Repository 5](#_Toc101893191)

[4. SE Methodology/Methods (With Justification) 5](#_Toc101893192)

[5. Time Schedule 6](#_Toc101893193)

[6. Requirements 6](#_Toc101893194)

[6.1. Stakeholder analysis 6](#_Toc101893195)

[6.2. Requirements Analysis 7](#_Toc101893196)

[6.3. Requirements modelling (Use Case Diagram) 8](#_Toc101893197)

[7. System’s Overall Design 9](#_Toc101893198)

[7.1. Overall architecture 9](#_Toc101893199)

[7.2. Overall DB design (ER) 10](#_Toc101893200)

[7.3. Activity diagrams 11](#_Toc101893201)

[8.1. Individual Sections – IT20278380 – User Management 12](#_Toc101893202)

[8.1.1. Service design 12](#_Toc101893203)

[8.1.1.1. API Of the Service 12](#_Toc101893204)

[8.1.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram) 13](#_Toc101893205)

[8.1.1.3. Database for the service (ER) 14](#_Toc101893206)

[8.1.2. Service development and testing. 15](#_Toc101893207)

[8.1.2.1. Tools used 15](#_Toc101893208)

[8.1.2.2. Testing Methodology and Results. 15](#_Toc101893209)

[8.1.3. Assumptions and any other details 16](#_Toc101893210)

[8.2. Individual Sections – IT20276232 – User Assign and Power Pant Management 17](#_Toc101893211)

[8.2.1. Service design 17](#_Toc101893212)

[8.2.1.1. API Of the Service 17](#_Toc101893213)

[8.2.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram) 18](#_Toc101893214)

[8.2.1.3. Database for the service (ER) 19](#_Toc101893215)

[8.2.2. Service development and testing. 20](#_Toc101893216)

[8.2.2.1. Tools used 20](#_Toc101893217)

[8.2.2.2. Testing Methodology and Results. 20](#_Toc101893218)

[8.2.3. Assumptions and any other details. 21](#_Toc101893219)

[8.3.Individual Sections – IT20243876 – Meter Reader and Management 21](#_Toc101893220)

[8.3.1. Service design 21](#_Toc101893221)

[8.3.1.1. API Of the Service 22](#_Toc101893222)

[8.3.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram) 23](#_Toc101893223)

[8.3.1.3. Database for the service (ER) 24](#_Toc101893224)

[8.3.2. Service development and testing. 25](#_Toc101893225)

[8.3.2.1. Tools used 25](#_Toc101893226)

[8.3.2.2. Testing Methodology and Results. 25](#_Toc101893227)

[8.3.3. Assumptions and any other details 26](#_Toc101893228)

[8.4.Individual Sections – IT20426958 – Billing Management 26](#_Toc101893229)

[8.4.1. Service design 26](#_Toc101893230)

[8.4.1.1. API Of the Service 27](#_Toc101893231)

[8.4.1.2. Internal logic (Activity Diagram/ Flow Chart/ Class Diagram) 28](#_Toc101893232)

[8.4.1.3. Database for the service (ER) 29](#_Toc101893233)

[8.4.2. Service development and testing. 30](#_Toc101893234)

[8.4.2.1. Tools used 30](#_Toc101893235)

[8.4.2.2. Testing Methodology and Results. 30](#_Toc101893236)

[8.4.3. Assumptions and any other details 31](#_Toc101893237)

[9. Assumptions and Any Other Details 31](#_Toc101893238)

# **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. |

# **Git Repository**

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

# **SE Methodology/Methods (With Justification)**

What is Agile?

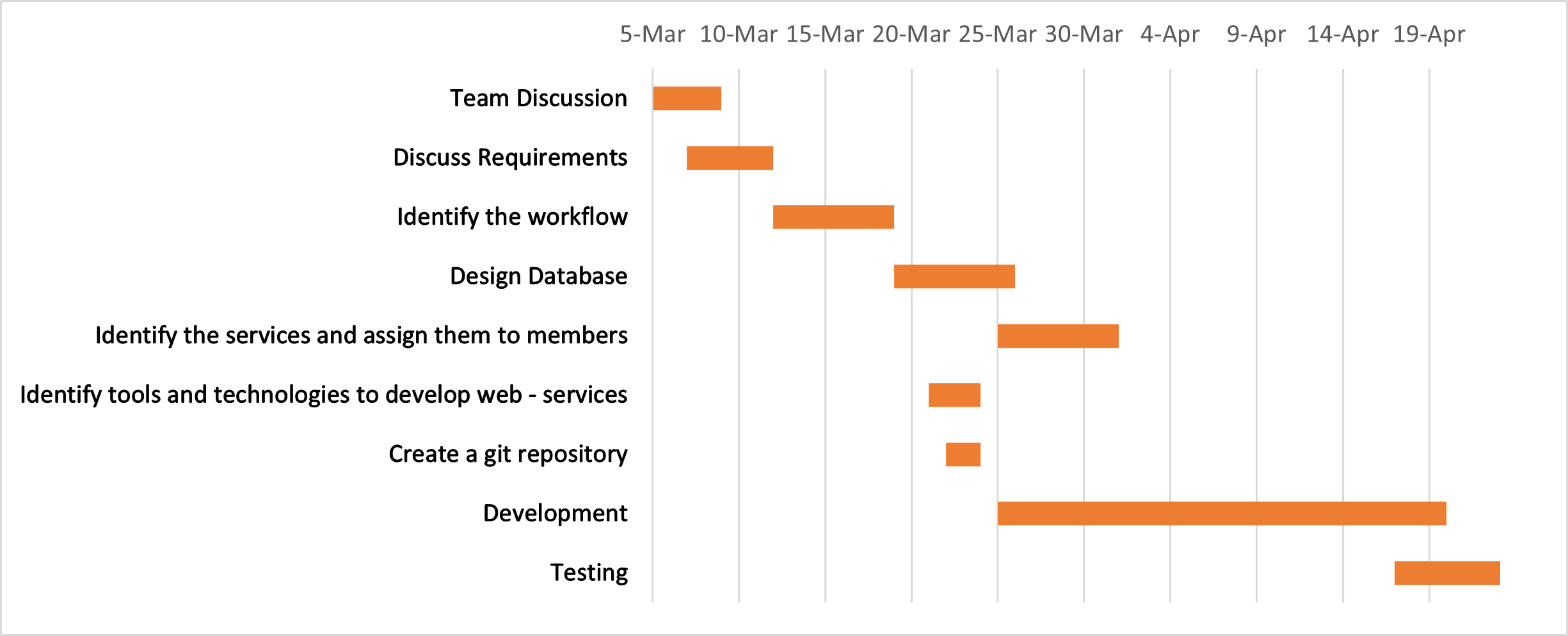
Agile Software Development is a methodology for creating a disciplined software management process that also allows for rapid changes to the development project. This methodology reduces risk by building software in short time periods known as iterations, which typically run one week to one month.

|  |  |
| --- | --- |
| 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.

# **Time Schedule**



**10-Mar**

**15-Mar**

**20-Mar**

**25-Mar**

**30-Mar**

**30-Mar**

**5-Apr**

**10-Apr**

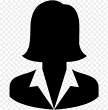
**15-Apr**

**20-Apr**

# **Requirements**

## **Stakeholder analysis**

Gadget Badget System



Tester

Developer

Analyst

User

System Admin

Customer

Researcher

Financial Beneficiaries

Funder



## **Requirements Analysis**

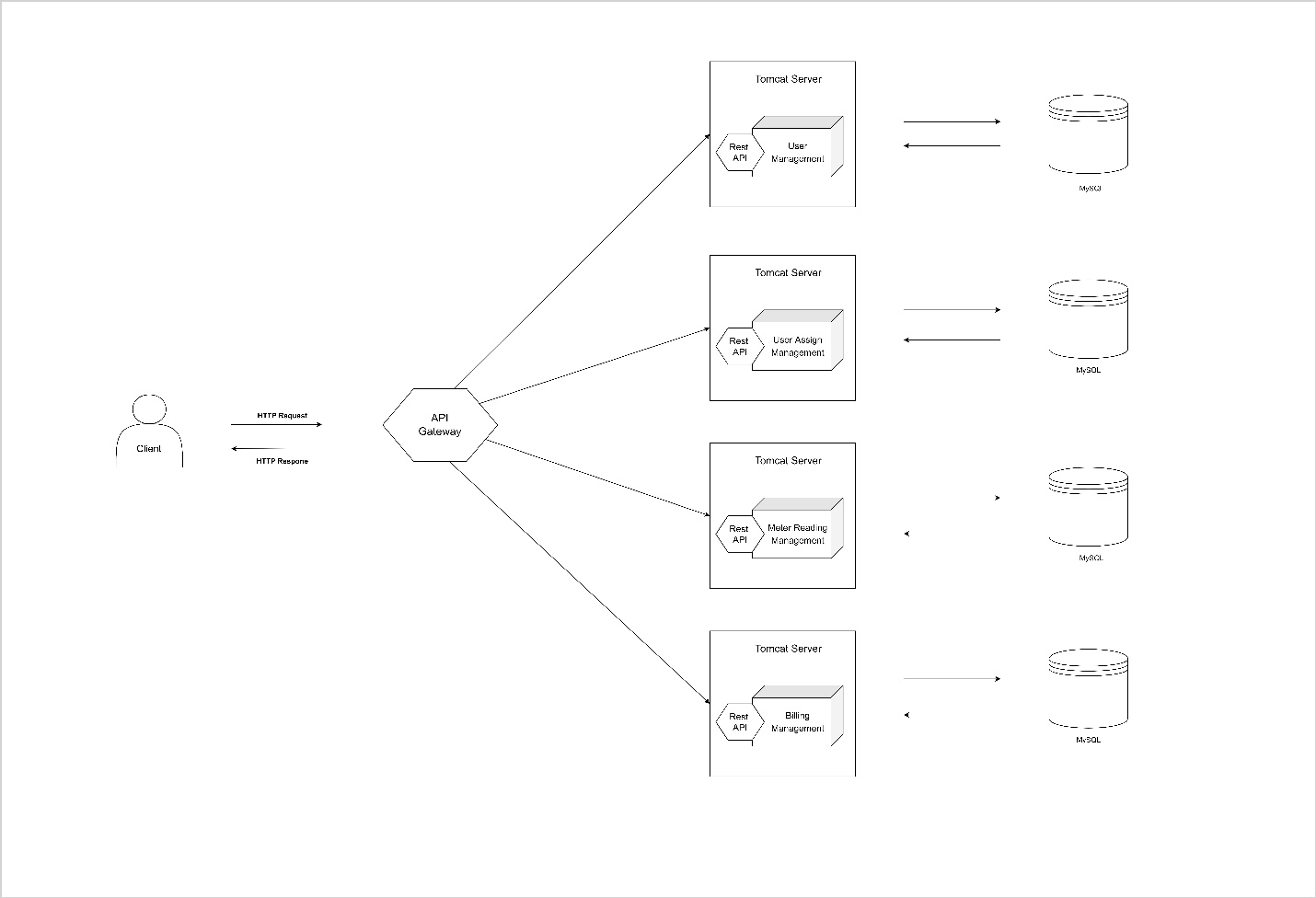
* Functional Requirement
* User Management – IT20278380
* Register User
* View User Profile
* Update User
* Delete User
* View Bill
* Bill Payment
* User Assign and Power Plant Management – IT20276232
* View Pending Users
* Assign User in Power Plant
* View All User
* Update User Power Plant
* Delete 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.

## **Diagram Description automatically generatedRequirements modelling (Use Case Diagram)**

# **System’s Overall Design**

## **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, Postman 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.

## Diagram Description automatically generated**Overall DB design (ER)**

## **Diagram, schematic Description automatically generatedActivity diagrams**

# **Individual Sections – IT20278380 – User Management**

## **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.

### Diagram Description automatically generated**API Of the Service**

* + 1. **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](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers%20/Add)
  + 1. **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](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers%20/Update)
  + 1. **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](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers%20/One)
  + 1. **Delete User Profile (DELETE)**
* **Resource:** PowerPlant
* **Request:** DELETE

<CustomerData>

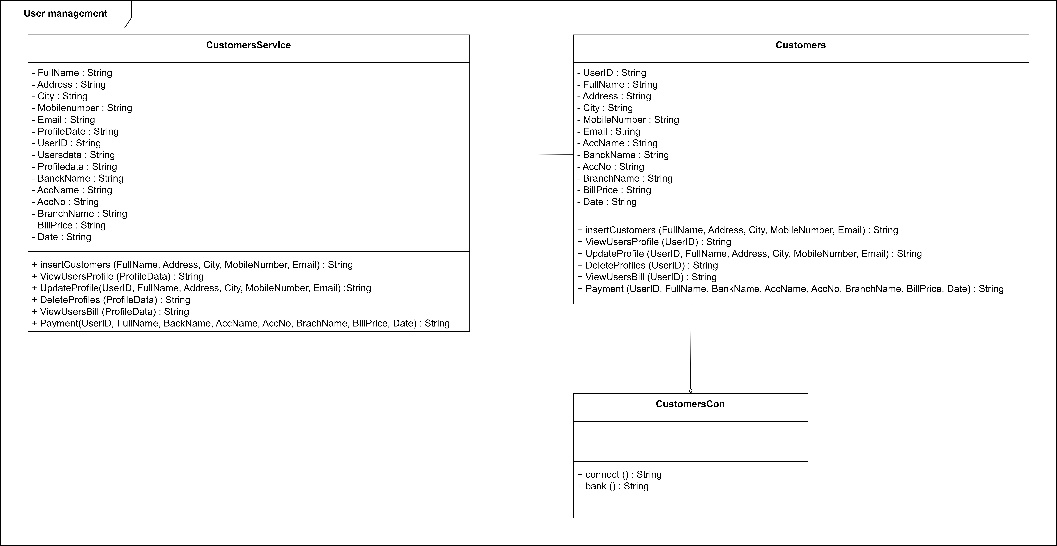
    </UserID>P4\_05</UserID>

</CustomerData>

* **Media:** Application XML
* **Response:** Deleted successfully
* **URL:** [http://localhost:8080/PAF\_Project\_2022\_GID\_8/ElectroG/Customers /Delete](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers%20/Delete)

### **Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)**

* Diagram

  Description automatically generatedFlow Chart
* Diagram

  Description automatically generatedActivity Diagram / Class Diagram

### **Database for the service (ER)**

Diagram

Description automatically generated

## **Service development and testing.**

### **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

### **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 |

## Graphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generated**Assumptions and any other details**Graphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generated

**Register**

**View Bill**

**Update Profile**

**Delete Profile**

**View Profile**

**Payment**

**Postman screenshots**

# **Individual Sections – IT20276232 – User Assign and Power Pant Management**

## **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.

### **API Of the Service**Graphical user interface, diagram Description automatically generated

* + 1. **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](mailto:wickramasinha1219@gmail.com)”, PowerPlant: “1”
* **Response:** Inserted successfully
* **URL:** [http://localhost:8080/PAF\_Project\_2022\_GID\_8/ElectroG/UserAssing/Add](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/Customers%20/Add)

1. **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](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/UserAssing%20/Update)

1. **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>

1. **Delete User Profile (DELETE)**

* **Resource:** Customers
* **Request:** DELETE

<UserData>

    </UserID>P4\_05</UserID>

</UserData>

* **Media:** Application XML
* **Response:** Deleted successfully
* **URL:** <http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/UserAssing/Delete>

### **Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)**

* Flow Chart

Diagram

Description automatically generated

* Activity Diagram / Class Diagram

A picture containing text

Description automatically generatedDiagram

Description automatically generated

### **Database for the service (ER)**

Diagram

Description automatically generated

## **Service development and testing.**

### **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

### **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 |

### **Graphical user interface, text, application Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedAssumptions and any other details**.

**View Pending Users**

**Update Power Plant**

**Delete Pending User**

**Add Users**

**Postman screenshots**

### **Individual Sections – IT20243876 – Meter Reader and Management**

## **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

### **API Of the Service**

Diagram

Description automatically generated

1. **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](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/MeterReader/Add%20)

1. **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>

1. **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>

1. **Delete User Profile (DELETE)**

* **Resource:** Bill
* **Request:** DELETE

<UserData>

    </UserID>P4\_05</UserID>

</UserData>

* **Media:** Application XML
* **Response:** Deleted successfully
* **URL:** <http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/MeterReader/Delete>

### **Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)**

* Flow Chart

Diagram

Description automatically generated

* Activity Diagram / Class Diagram

Diagram

Description automatically generatedDiagram

Description automatically generated

### **Database for the service (ER)**

Diagram

Description automatically generated

## **Service development and testing.**

### **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

### **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 |

## Graphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface Description automatically generated**Assumptions and any other details**

**Postman screenshots**

**View Users**

**Delete Old Bills**

**Update User Remark**

**Add Meter Unit**

## **Individual Sections – IT20426958 – Billing Management**

## **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

### Graphical user interface, diagram Description automatically generated**API Of the Service**

1. **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](http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/BillCreation/Add%20)

1. **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>

1. **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/>

1. **Delete User Profile (DELETE)**

* **Resource:** PowerPlant
* **Request:** DELETE

<UserData>

    </UserID>P4\_05</UserID>

</UserData>

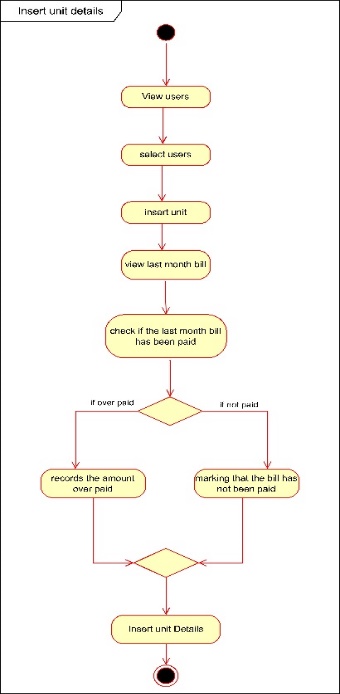
* **Media:** Application XML
* **Response:** Deleted successfully
* **URL:** <http://localhost:8080/PAF_Project_2022_GID_8/ElectroG/BillCreation/Delete>

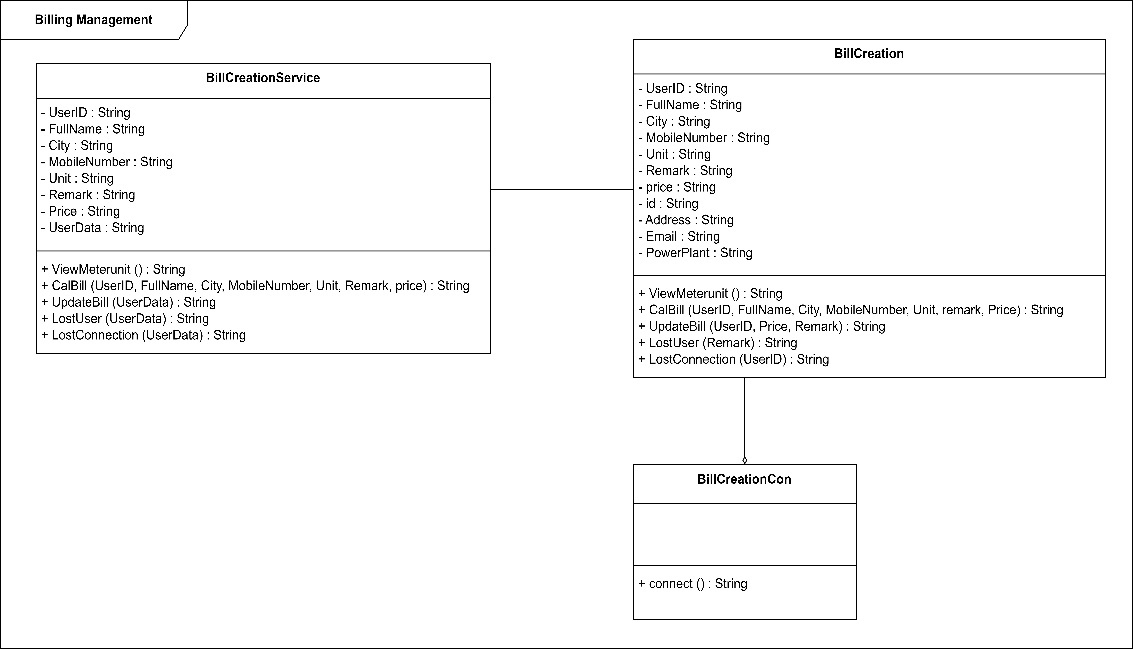
### **Internal logic (Activity Diagram/ Flow Chart/ Class Diagram)**

* Flow Chart

Diagram

Description automatically generated

* Activity Diagram / Class Diagram



### **Database for the service (ER)**

Diagram

Description automatically generated

## **Service development and testing.**

### **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

### **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 |

## Graphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generatedGraphical user interface, text, application, email Description automatically generated**Assumptions and any other details**

**Postman screenshots**

**Create Bill**

**Delete Not Pay User**

**Update Bill**

**View Meter Reding Unit**

## **Assumptions and Any Other Details**

* 1. **Git Commit Log**
  2. **Reference**

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