Programming Applications and Frameworks
(IT3030)
3rd Year, 1st Semester

Assignment

ElectroGrid

Submitted to

Sri Lanka Institute of Information Technology

Group – Y3.S1.WE.IT.02.01
Group ID – 220

IT20219598 – S.L.D.P Pramodya
IT20257040 – A.M.K.A.P Amarasingha
IT20235260 - D.R.N. Samarawila
IT20141356 - A.N Upathissa

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

2022.04.26

GitHub Link – https://github.com/PasinduPramodya/PAF_Electro.Grid.git

Contents

Software Engineering methodology

Introduction

The ElectroGrid company which supplies power to the country is in need of a more scalable system in order to maintain their users. The ElectroGrid system is a system that our team has developed a highly scalable online platform to monitor the power usages, generate bills and let their users make online payments.

The main functions that we have taken to implement are Customer management, power consumption management, Generate Bill which is a part of financial management and let the customers make online payments which is also function of financial management, Employment management where details about the existing staff is managed. Also, we have implemented a customer care where the users can lodge complaints regarding the breakdowns, inquiries regarding bill payment and other issues that may face, it is a part of the customer management.

To develop this system, we have chosen the Software Engineering methodology of Agile development because this company already has its system and need an extended version of the existing system in order to match the competition. Also, there is a fixed set of requirements that they need implemented.

The main stakeholders that we have identified are: -

Customers
Power consumption manager
Electrical Engineer
Customer service manager
Financial manager
Staff manager

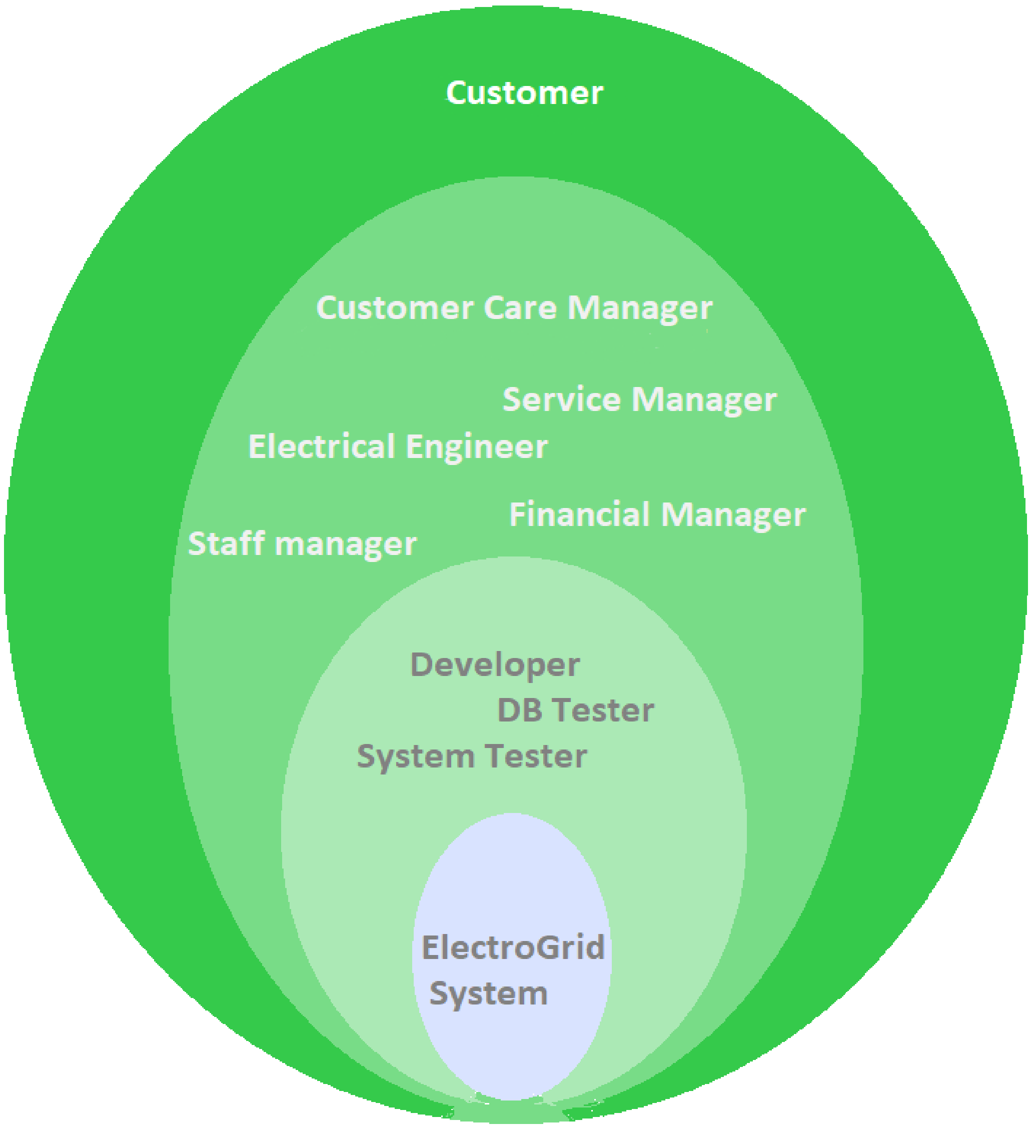
After identifying the main stakeholders, we have gathered all the necessary requirements needed for this system and analysed them to fix the most essential requirements.
Also we have categorised the requirements as functional, non-functional and technical to finalise the most important requirements and get a clear understanding about the logic.

IDE Used - Eclipse EE
Database – phpMyAdmin (MySQL)
Server - Tomcat

Gantt chart

09 April-24 April											🟥
07 April-08 April										🟦	
04 April-06 April									🟥		
01 April-03 April								🟦			
14 March-31 March							🟥				
12 March-13 March						🟦					
11 March-12 March					🟥						
08 March-10 March				🟦							
05 March-07 March			🟥								
03 March-04 March		🟦									
01 March-02 March	🟥										
	Stockholder Determination	Gather requirement	Analyzing requirements	Drawing UML	Drawing ER	Creating GitHub repository	Create individual functions	Create database	Connecting DB to individual part	Integrating the system	Report

Onion Diagram



Requirement analysis

1. Customer management
   - Customer Registration
   - Customer login
   - Customer profile update
   - Customer profile delete

2. Customer care management
   - Customer lodge a complaint regarding an issue they faced.

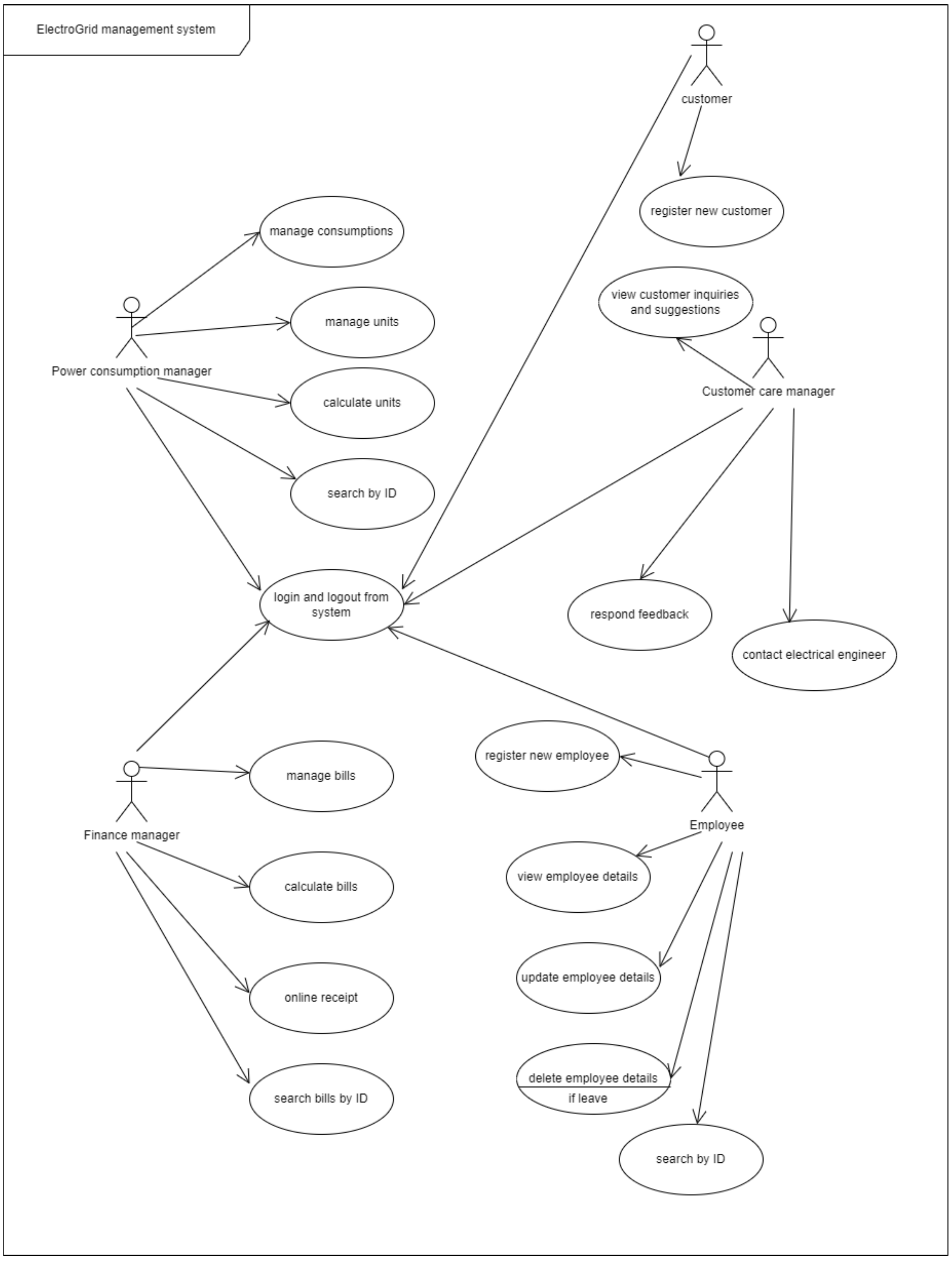
3. Power consumption management
   - Determine power usage of each customer
   - Send the power usage details to financial management
   - Have a table with units and their rates in the database

4. Finance management
   - Get the necessary details from power consumption
   - Get customer details
   - Generate a bill regarding the usage
   - Allow customer to make online payments.

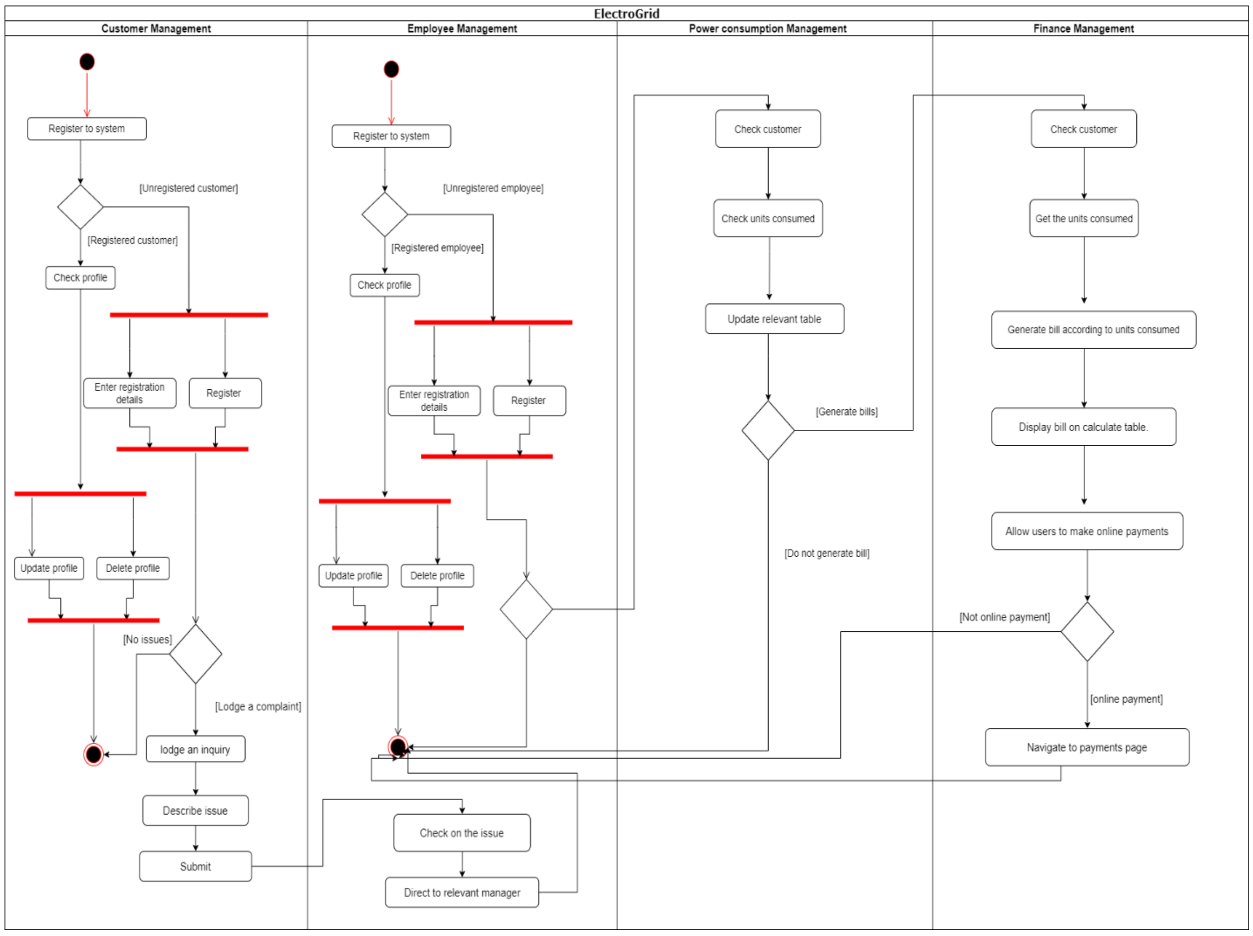
5. Employee management
   - Employee Registration
   - Employee login
   - Employee profile update
   - Employee profile delete
   - Engage in power management
   - Engage in billing management

Requirements modelling

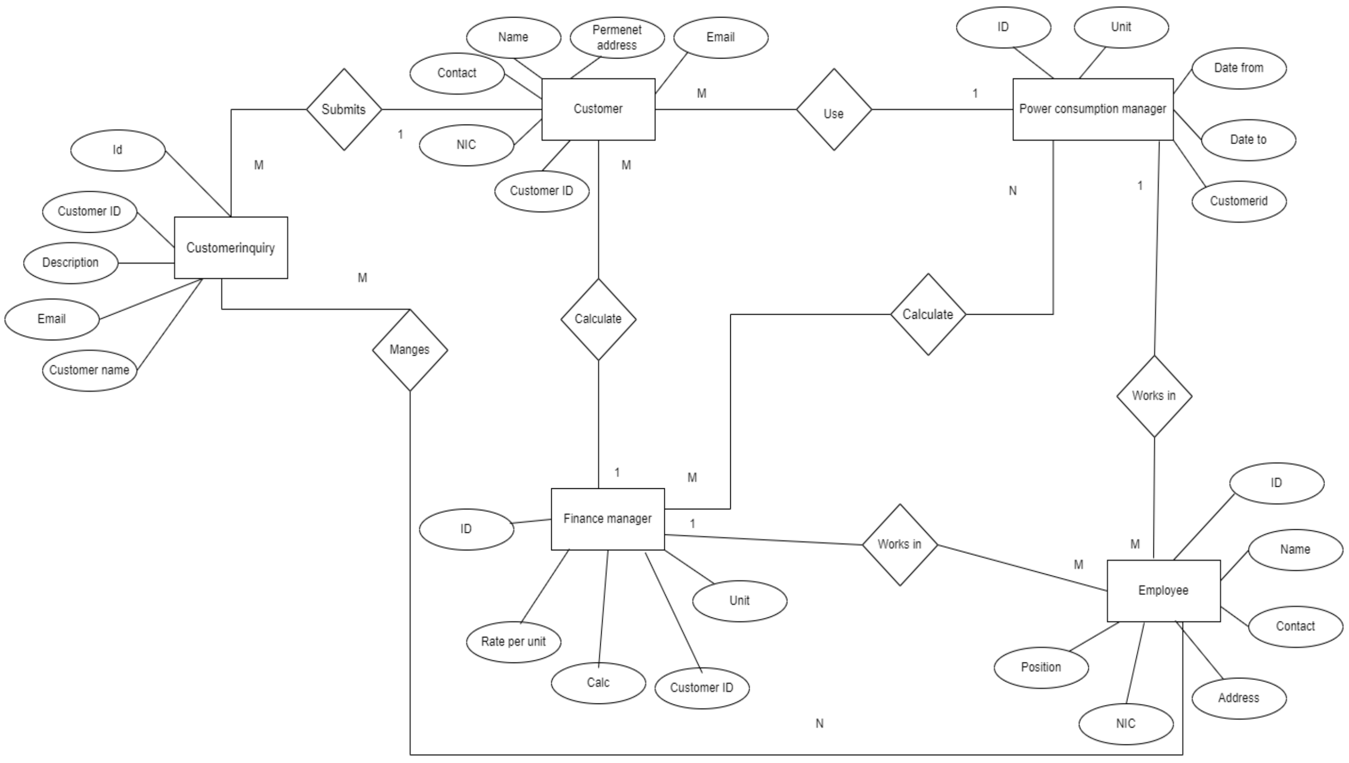
Usecase Diagram



Overall Activity Diagram



Overall Entity Relationship Diagram



Entity Relationship Diagram

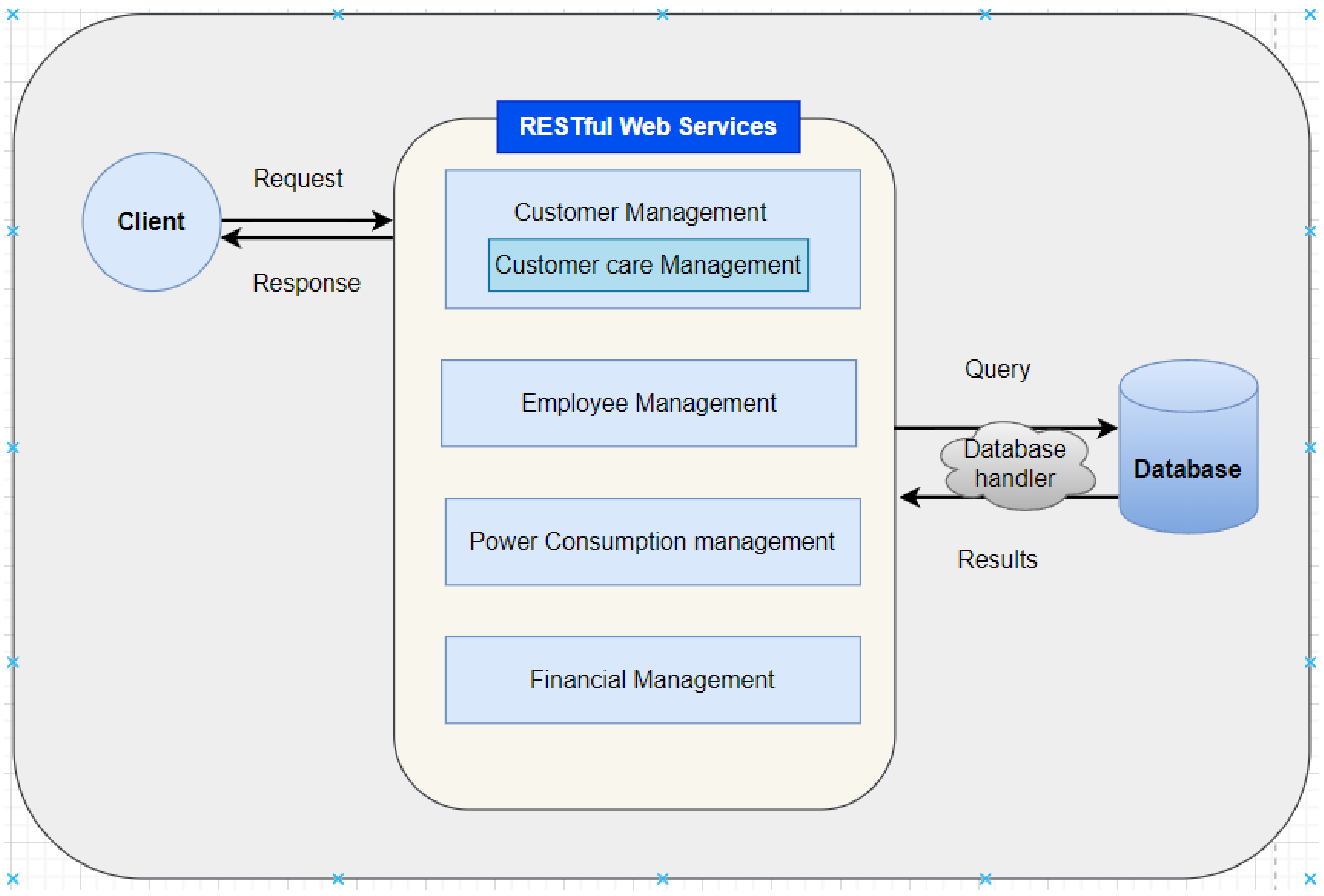
The main entities that we have identified are

Customer
Power consumption manager
Financial manager
Employee.

According to these entities we have identified the main tables in our Database after all normalisation processes and schema refinements as

Customer table (id, name, permanent address, NIC, Email)
Customer_contact (id, cusid, contact)
Power consumption (id, unit, date _to, date_from, customerid)
Finance (id, rate_per_unit, calc, units, customerid)
Employee (id, Name, address, nic, position)
Employee_contact (id, eid, contact)
Calculate (id, customerid, powerid, financeid, bill)
Customerinquiry (id. customerid, Description, email, customername)

Overall Architecture



This Explains the overall architecture of our RESTful web services. The client request using the PUT,GET,POST and DELETE methods, then the web services manage the request with the relevant web service and database handler.

The database handler executes the query and obtain the relevant results from the Database And the results are returned to the client as a response made to his/her request.

Main web services: -

Customer Management
Employee Management
Power consumption Management
Financial Management

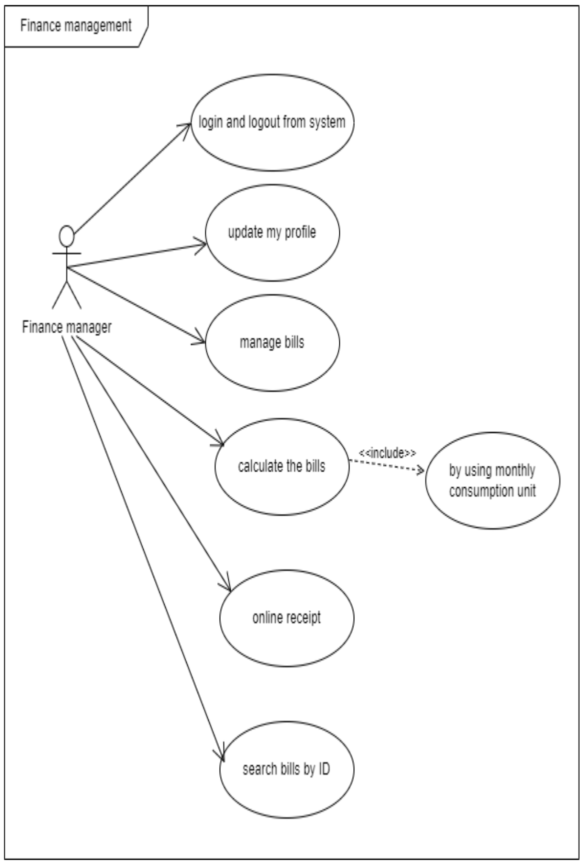
Requirements Categorization

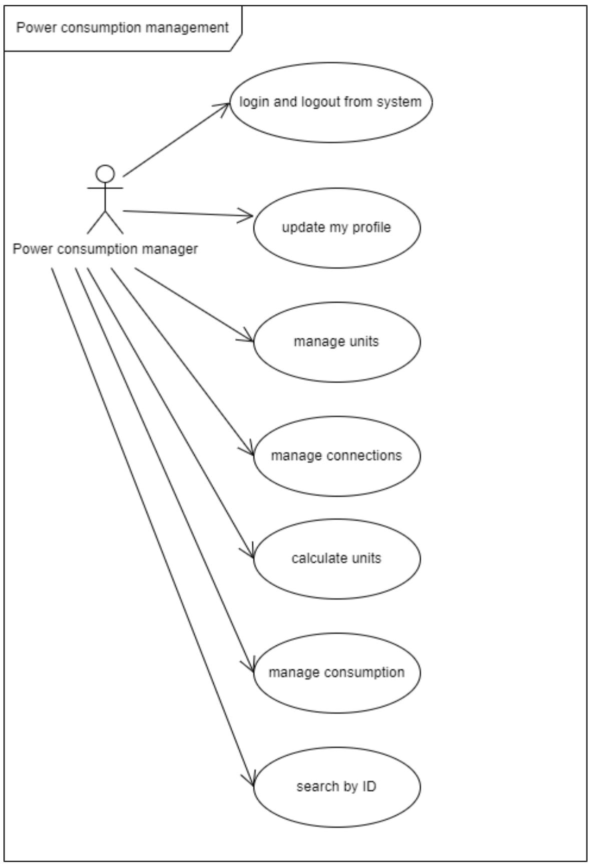
Functionality	**Function Requirements**	**Non-functional Requirements**	**Technical Requirements**
Customer Management	● Customer can sign in or log in to the system ● The system should allow a customer to check his or her monthly electricity bill. ● The customer should be able to use the system to file a complaint . .	• Reliability • Efficiency • Security • Usability • Privacy	● Customer can directly register to the system using the Customer management
Customer Care Management	● Should be able to view customer's inquiries and suggestions. ● Should be able to contact electrical engineer to resolve power consumption issues. ● Should be able to read and respond to feedback. ● Should be able to delete outdated feedback and irrelevant remarks.	• Reliability • Efficiency • Security • Usability • Privacy	● Customer's inquiry is directed to all management sections.

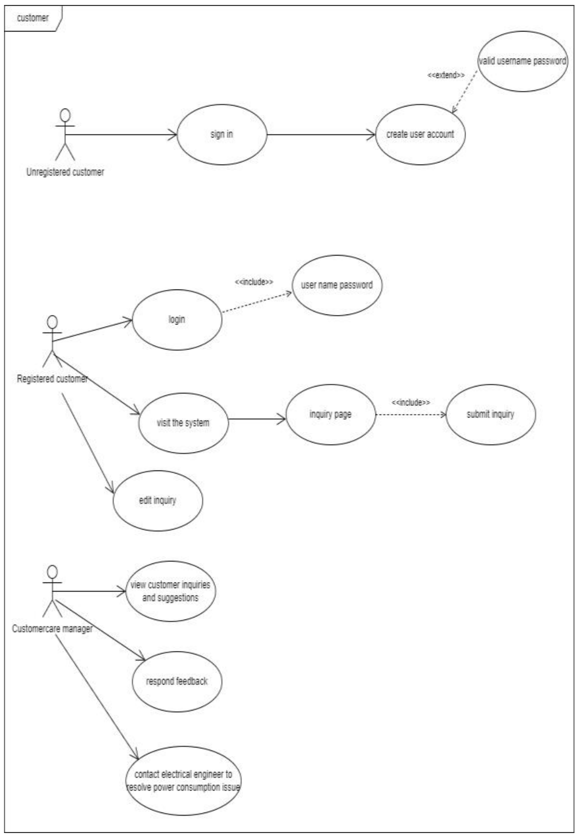
Power consumption management (Service manager and Electrical engineer)	● Should be to view complaints logged by the client ● Should be able to update the statuses of ongoing complaints ● Working on the power consumption issues ● Calculate the monthly consumption units	• Reliability • Efficiency • Security • Usability • Privacy	● Connecting with electrical engineers to solve issues.
Financial Management	● Should be able to manage bills. ● Should be able to Calculate the bills by using monthly consumption power units and update the bills ● Generate the online receipt	• Reliability • Efficiency • Security • Usability • Privacy	● Should be able to communicate with power consumption management and generate user's bills.

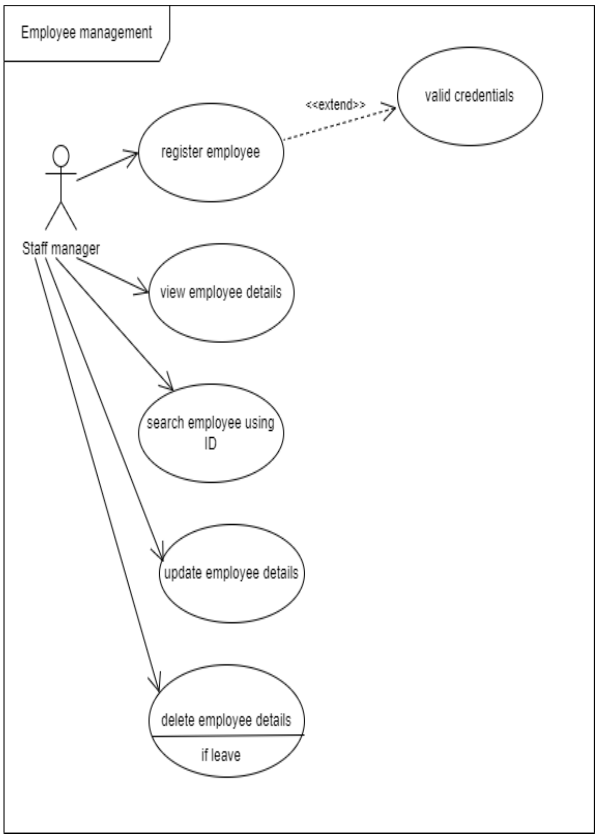
Employee Management	● All employee information should be visible. ● New employees should be able to be added to the system, and current employees should be able to be removed if they depart.	• Reliability • Efficiency • Security • Usability • Privacy	● Employees who have been verified have access to the personnel management feature.

Use Case Diagrams - Individual









Individual Contribution

S.L.D.P Pramodya - IT20219598

Web service - Power Consumption Management

Purpose - The main purpose of this web service is to get the details of power usage of each individual registered in the system and send those details to Billing section in order to generate a bill on monthly basis. Here the rates are entered for each unit that the customers consume and the bill is generated using those details.

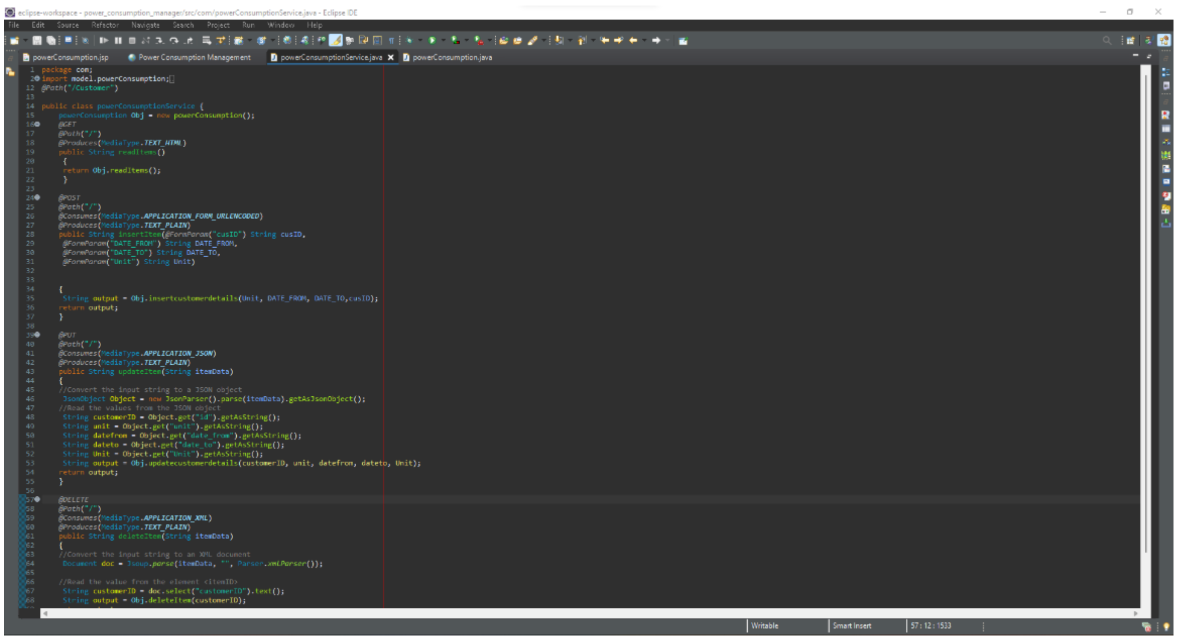
Main stakeholders - Electrical manager

Power consumption manager

Financial manager

Logic used - The power consumption manager gathers the relevant details of customer such as units consumed, date form, date to and enter them in the power consumption table along with customer's ID. Here data is passed from powerconsumption.jsp to powerconsumptionservice.java and finally to powerconsumption.java which sends the data database using the relevant inserting query and updating and deletion are performed accordingly.

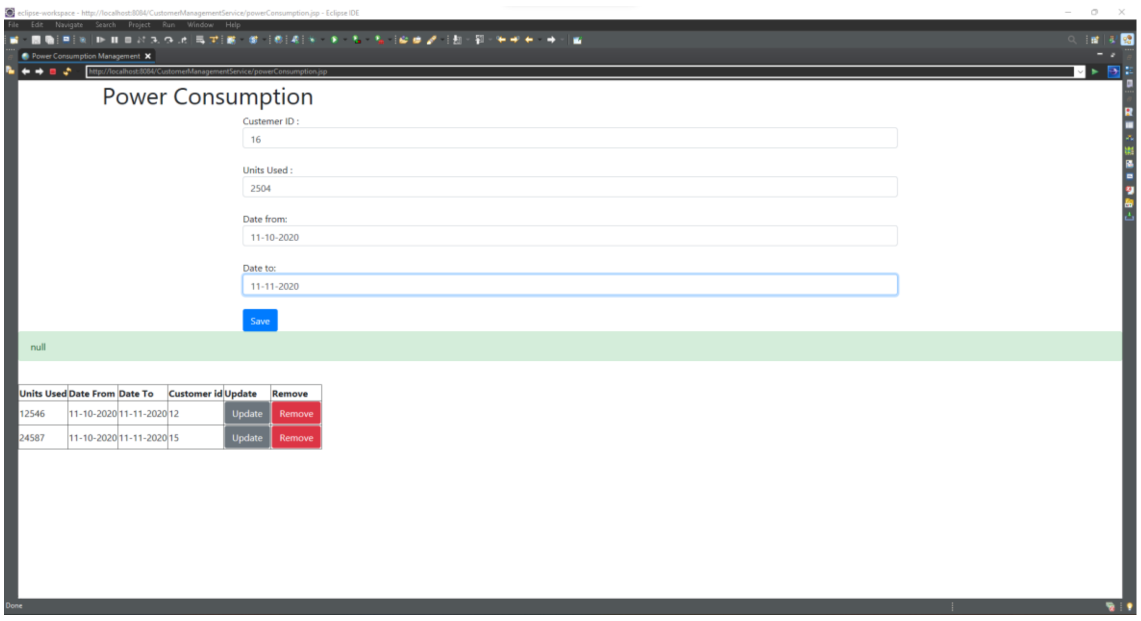
Code



eclipse-workspace - power_consumption_manager/src/com/powerConsumptionService.java - Eclipse IDE

```java
package com;
import model.powerConsumption;
@Path("/Customer")

public class powerConsumptionService {
    powerConsumption Obj = new powerConsumption();
    @GET
    @Path("/")
    @Produces(MediaType.TEXT_HTML)
    public String readItems()
     {
     return Obj.readItems();
     }

    @POST
    @Path("/")
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    @Produces(MediaType.TEXT_PLAIN)
    public String insertItem(@FormParam("cusID") String cusID,
     @FormParam("DATE_FROM") String DATE_FROM,
     @FormParam("DATE_TO") String DATE_TO,
     @FormParam("Unit") String Unit)


    {
     String output = Obj.insertcustomerdetails(Unit, DATE_FROM, DATE_TO,cusID);
    return output;
    }

    @PUT
    @Path("/")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.TEXT_PLAIN)
    public String updateItem(String itemData)
    {
    //Convert the input string to a JSON object
     JsonObject Object = new JsonParser().parse(itemData).getAsJsonObject();
    //Read the values from the JSON object
     String customerID = Object.get("id").getAsString();
     String unit = Object.get("unit").getAsString();
     String datefrom = Object.get("date_from").getAsString();
     String dateto = Object.get("date_to").getAsString();
     String Unit = Object.get("Unit").getAsString();
     String output = Obj.updatecustomerdetails(customerID, unit, datefrom, dateto, Unit);
    return output;
    }

    @DELETE
    @Path("/")
    @Consumes(MediaType.APPLICATION_XML)
    @Produces(MediaType.TEXT_PLAIN)
    public String deleteItem(String itemData)
    {
    //Convert the input string to an XML document
     Document doc = Jsoup.parse(itemData, "", Parser.xmlParser());

    //Read the value from the element <itemID>
     String customerID = doc.select("customerID").text();
     String output = Obj.deleteItem(customerID);
```

User Interface - Before inserting



eclipse-workspace - http://localhost:8084/CustomerManagementService/powerConsumption.jsp - Eclipse IDE

Power Consumption

Custemer ID : 16

Units Used : 2504

Date from: 11-10-2020

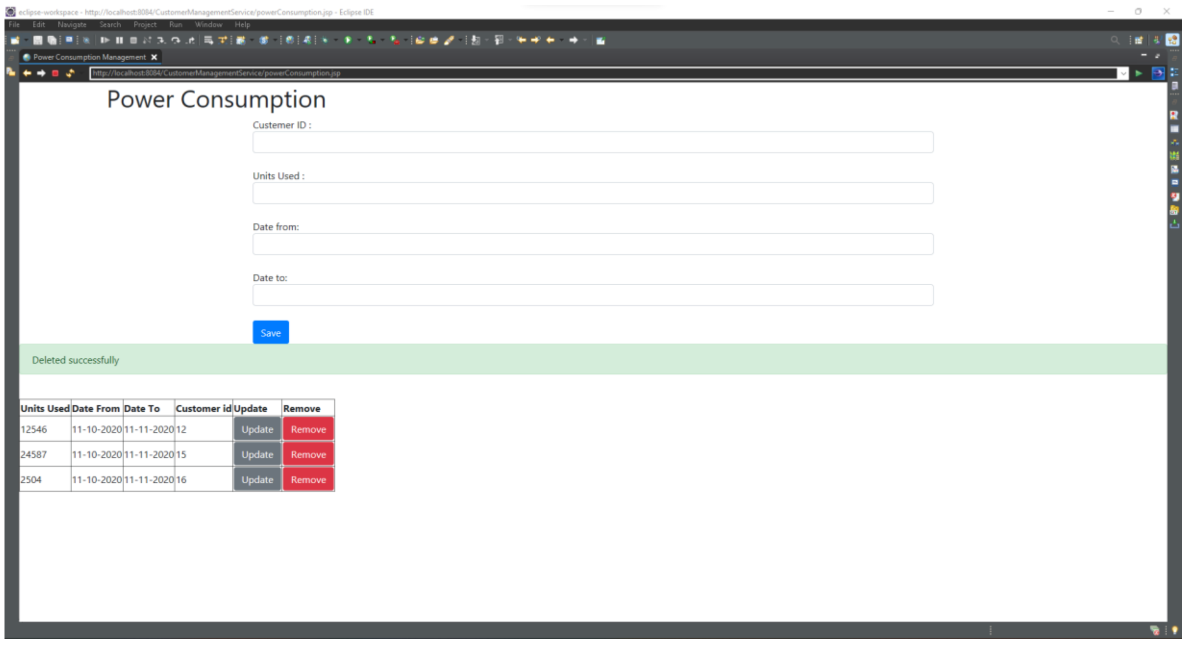
Date to: 11-11-2020

Save

null

Units Used	Date From	Date To	Customer id	Update	Remove
12546	11-10-2020	11-11-2020	12	Update	Remove
24587	11-10-2020	11-11-2020	15	Update	Remove

User Interface – After inserting



eclipse-workspace - http://localhost:8084/CustomerManagementService/powerConsumption.jsp - Eclipse IDE

http://localhost:8084/CustomerManagementService/powerConsumption.jsp

Power Consumption

Custemer ID :

Units Used :

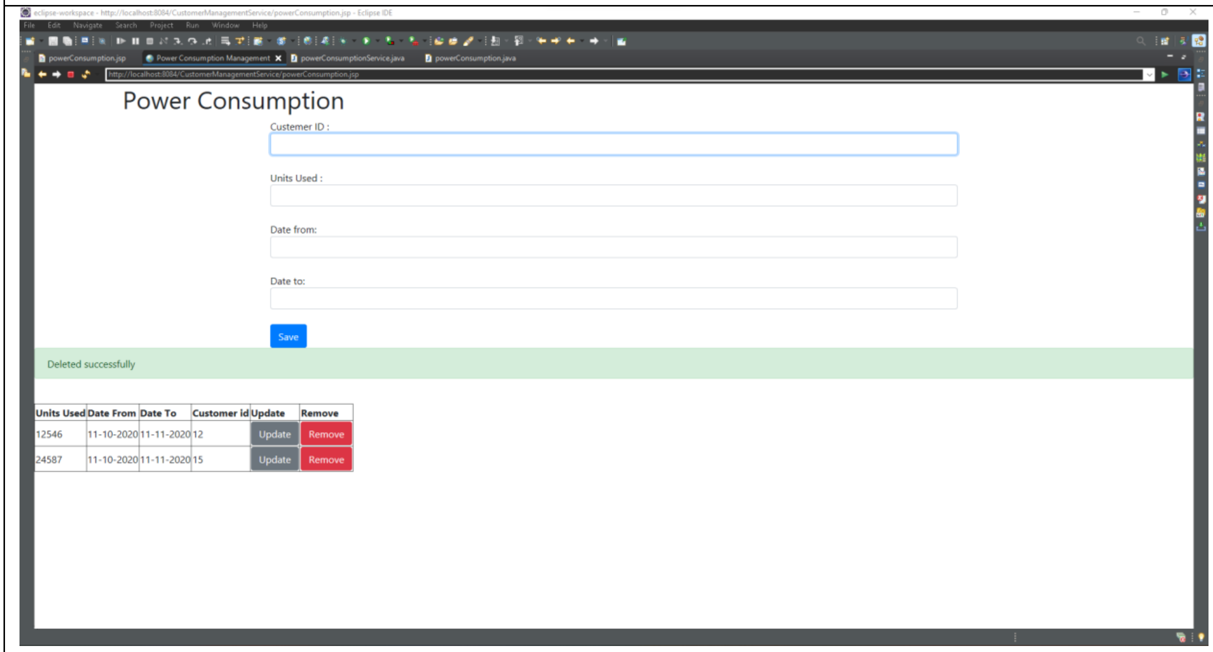
Date from:

Date to:

Deleted successfully

Units Used	Date From	Date To	Customer id	Update	Remove
12546	11-10-2020	11-11-2020	12	Update	Remove
24587	11-10-2020	11-11-2020	15	Update	Remove
2504	11-10-2020	11-11-2020	16	Update	Remove

User Interface – After deletion



eclipse-workspace - http://localhost:8084/CustomerManagementService/powerConsumption.jsp - Eclipse IDE

http://localhost:8084/CustomerManagementService/powerConsumption.jsp

Power Consumption

Custemer ID :

Units Used :

Date from:

Date to:

Deleted successfully

Units Used	Date From	Date To	Customer id	Update	Remove
12546	11-10-2020	11-11-2020	12	Update	Remove
24587	11-10-2020	11-11-2020	15	Update	Remove

A.M.K.A.P Amarasingha -IT20257040

Web service - Financial Management

Purpose - The main purpose of this web service is to get the details of power usage of each individual registered in the system generate a bill according their usage. on monthly basis. Here the rates which are entered in the power consumption management tables are used to calculate the monthly bills of the customers.

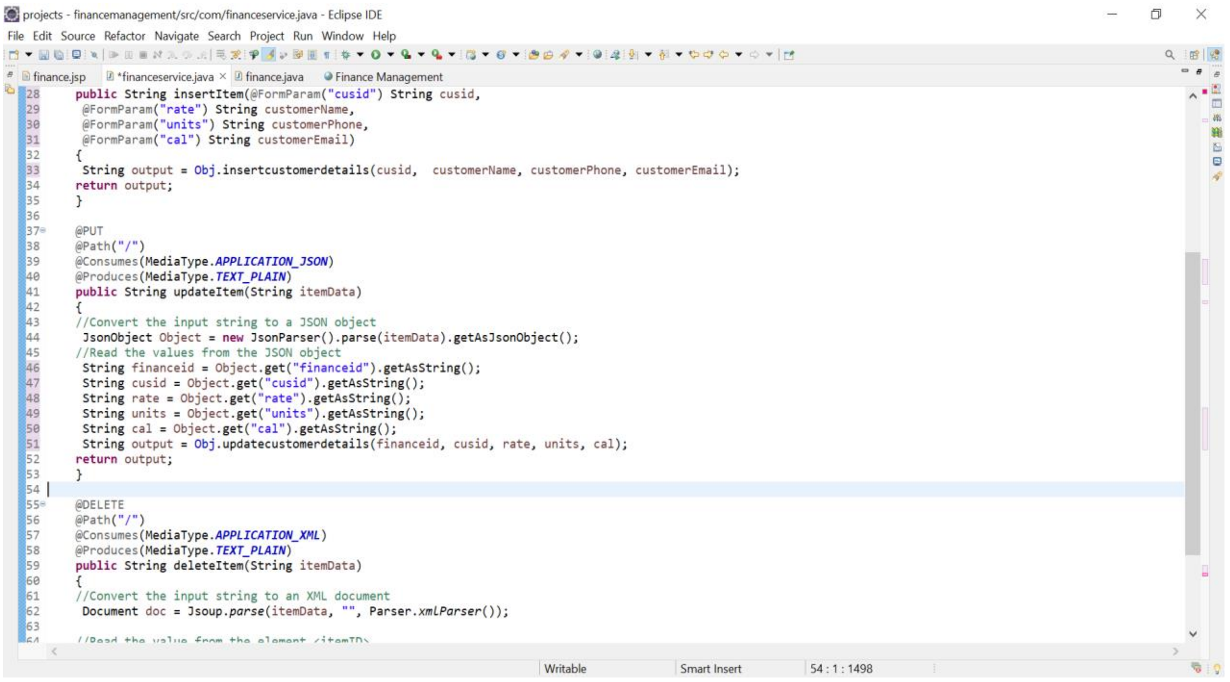
Main stakeholders - Electrical manager

Power consumption manager

Financial manager

Logic used - The finance manager gathers the relevant details such as id, rate per unit and units so that the calculation happens and enter them in the finance table along with ID. Here data is passed from finance.jsp to financeservice.java and finally to finance.java which sends the data database using the relevant inserting query and updating and deletion are performed accordingly.

Code

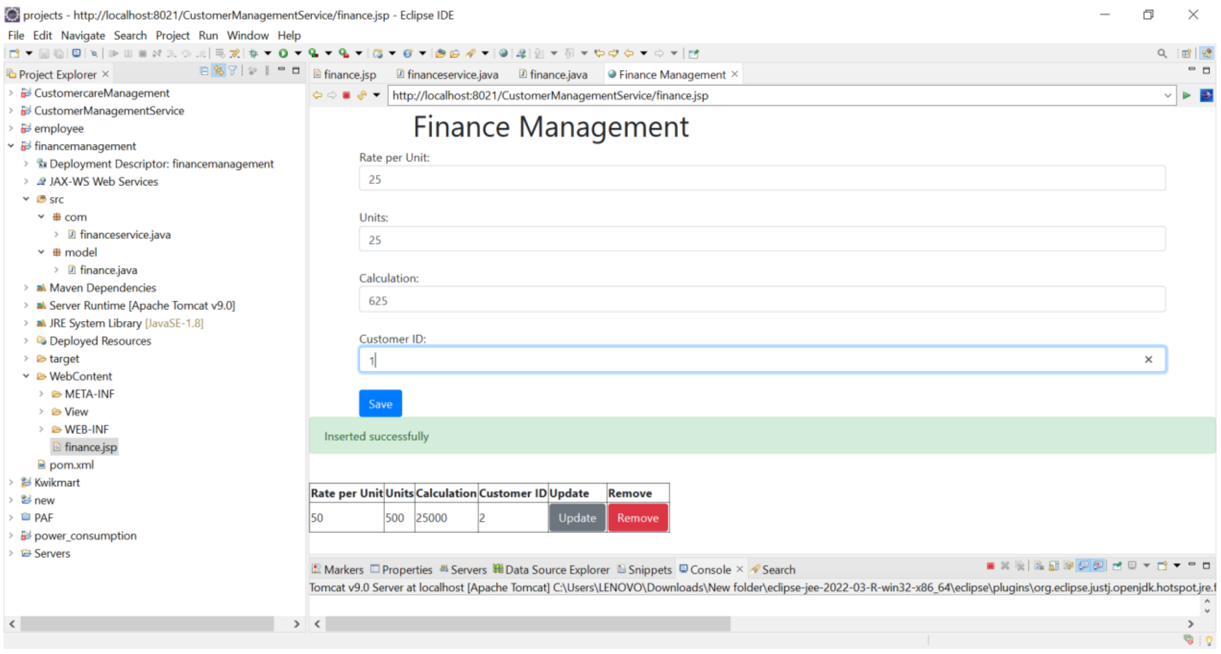


projects - financemanagement/src/com/financeservice.java - Eclipse IDE

```java
    public String insertItem(@FormParam("cusid") String cusid,
     @FormParam("rate") String customerName,
     @FormParam("units") String customerPhone,
     @FormParam("cal") String customerEmail)
    {
     String output = Obj.insertcustomerdetails(cusid,  customerName, customerPhone, customerEmail);
    return output;
    }

    @PUT
    @Path("/")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.TEXT_PLAIN)
    public String updateItem(String itemData)
    {
    //Convert the input string to a JSON object
     JsonObject Object = new JsonParser().parse(itemData).getAsJsonObject();
    //Read the values from the JSON object
     String financeid = Object.get("financeid").getAsString();
     String cusid = Object.get("cusid").getAsString();
     String rate = Object.get("rate").getAsString();
     String units = Object.get("units").getAsString();
     String cal = Object.get("cal").getAsString();
     String output = Obj.updatecustomerdetails(financeid, cusid, rate, units, cal);
    return output;
    }

    @DELETE
    @Path("/")
    @Consumes(MediaType.APPLICATION_XML)
    @Produces(MediaType.TEXT_PLAIN)
    public String deleteItem(String itemData)
    {
    //Convert the input string to an XML document
     Document doc = Jsoup.parse(itemData, "", Parser.xmlParser());

    //Read the value from the element <itemID>
```

Writable | Smart Insert | 54 : 1 : 1498

User Interface - Before inserting



projects - http://localhost:8021/CustomerManagementService/finance.jsp - Eclipse IDE

http://localhost:8021/CustomerManagementService/finance.jsp

Finance Management

Rate per Unit: 25

Units: 25

Calculation: 625

Customer ID: 1

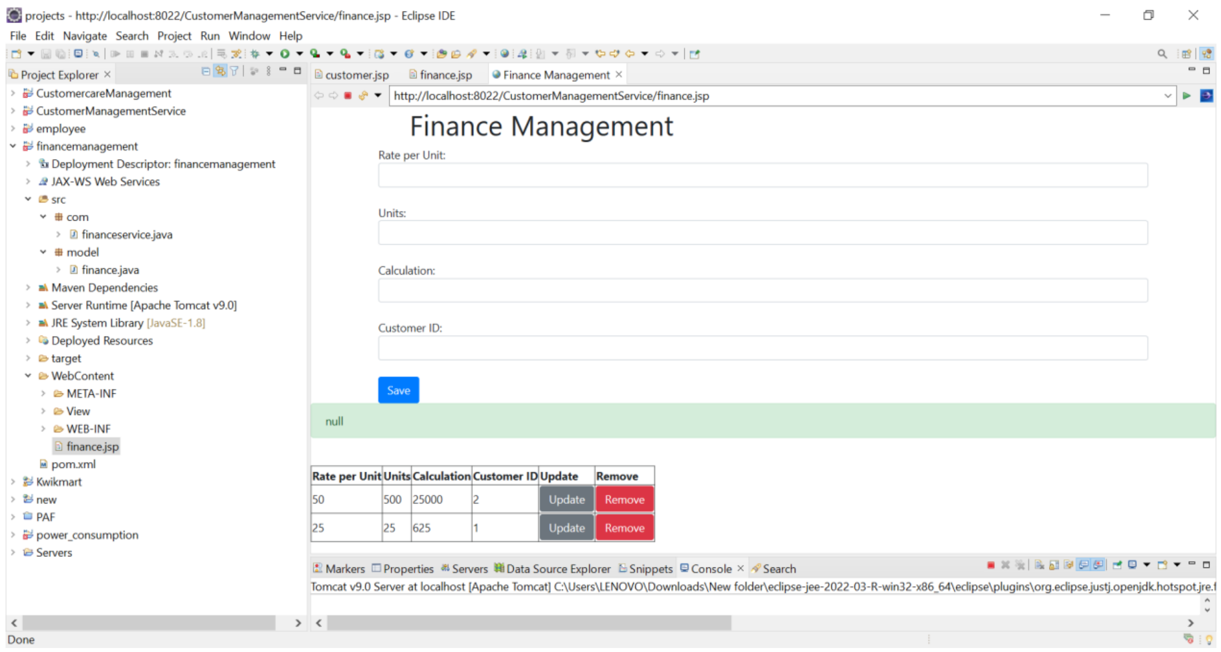
Save

Inserted successfully

Rate per Unit	Units	Calculation	Customer ID	Update	Remove
50	500	25000	2	Update	Remove

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f

User Interface – After inserting


projects - http://localhost:8022/CustomerManagementService/finance.jsp - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- CustomercareManagement
- CustomerManagementService
- employee
- financemanagement
  - Deployment Descriptor: financemanagement
  - JAX-WS Web Services
  - src
    - com
      - financeservice.java
    - model
      - finance.java
  - Maven Dependencies
  - Server Runtime [Apache Tomcat v9.0]
  - JRE System Library [JavaSE-1.8]
  - Deployed Resources
  - target
  - WebContent
    - META-INF
    - View
    - WEB-INF
    - finance.jsp
  - pom.xml
- Kwikmart
- new
- PAF
- power_consumption
- Servers

customer.jsp | finance.jsp | Finance Management

http://localhost:8022/CustomerManagementService/finance.jsp

Finance Management

Rate per Unit:

Units:

Calculation:

Customer ID:

Save

null

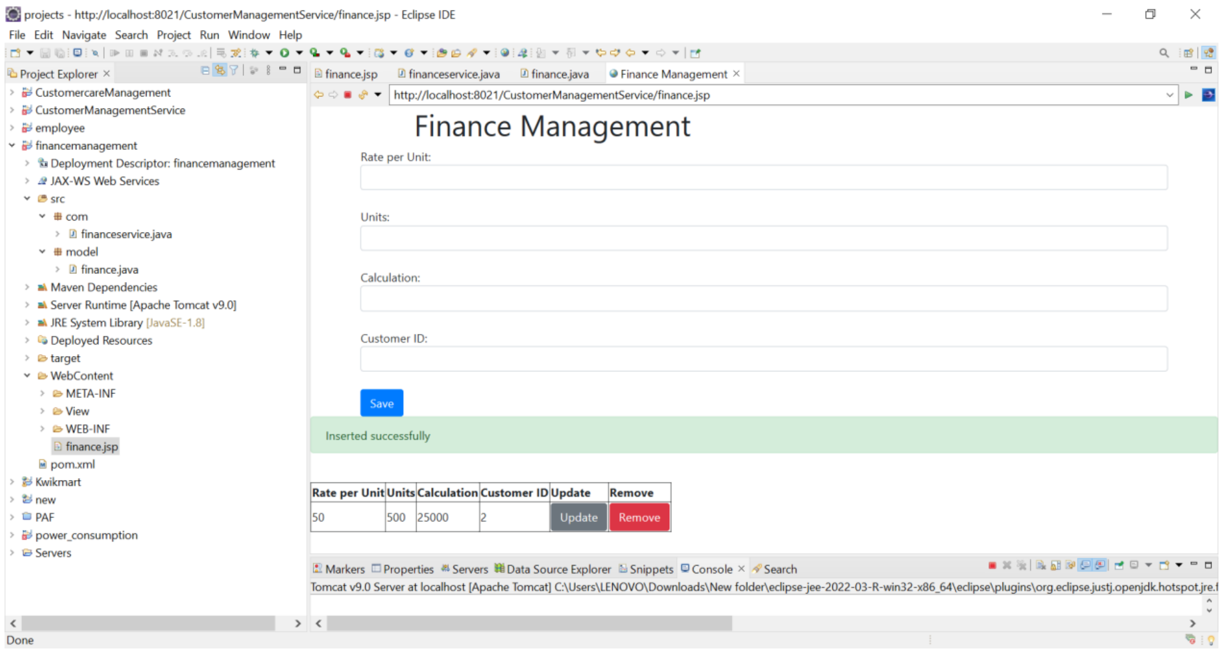
Rate per Unit	Units	Calculation	Customer ID	Update	Remove
50	500	25000	2	Update	Remove
25	25	625	1	Update	Remove

Markers Properties Servers Data Source Explorer Snippets Console Search

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
```

Done

User Interface – After deletion


projects - http://localhost:8021/CustomerManagementService/finance.jsp - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- CustomercareManagement
- CustomerManagementService
- employee
- financemanagement
  - Deployment Descriptor: financemanagement
  - JAX-WS Web Services
  - src
    - com
      - financeservice.java
    - model
      - finance.java
  - Maven Dependencies
  - Server Runtime [Apache Tomcat v9.0]
  - JRE System Library [JavaSE-1.8]
  - Deployed Resources
  - target
  - WebContent
    - META-INF
    - View
    - WEB-INF
    - finance.jsp
  - pom.xml
- Kwikmart
- new
- PAF
- power_consumption
- Servers

finance.jsp | financeservice.java | finance.java | Finance Management

http://localhost:8021/CustomerManagementService/finance.jsp

Finance Management

Rate per Unit:

Units:

Calculation:

Customer ID:

Save

Inserted successfully

Rate per Unit	Units	Calculation	Customer ID	Update	Remove
50	500	25000	2	Update	Remove

Markers Properties Servers Data Source Explorer Snippets Console Search

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
```

Done

D.R.N Samarawila – IT20235260

Web service - Customer Management and Customer care Management

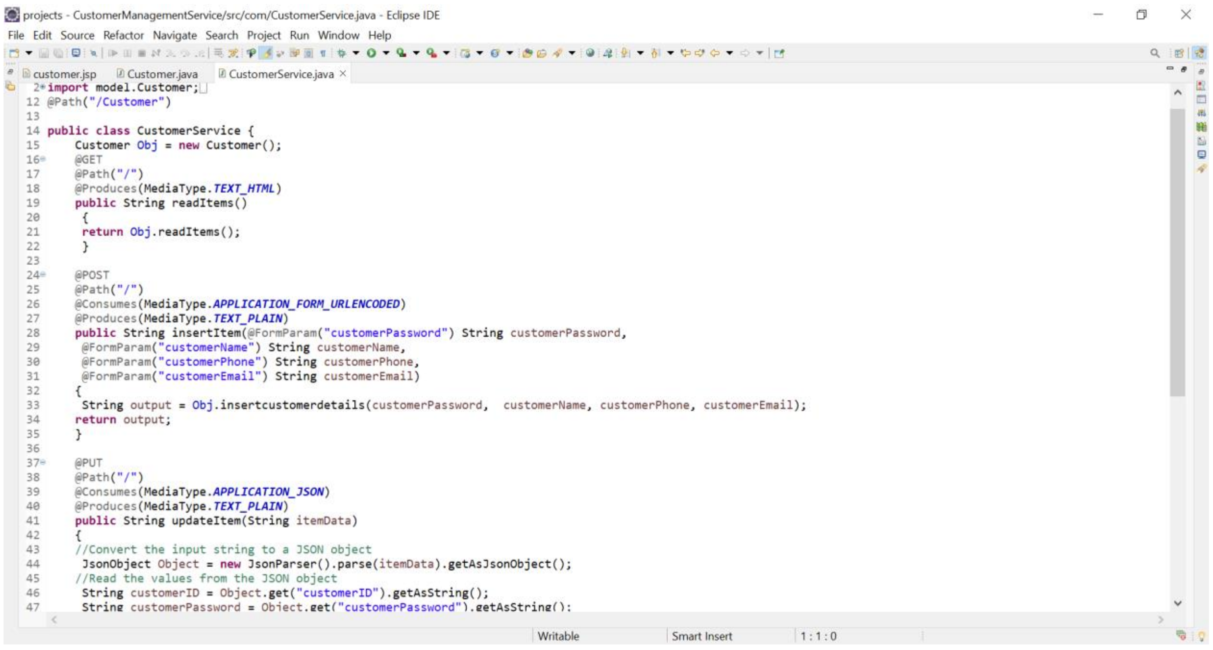
Purpose - The main purpose of this web service is to register new customers to the company and maintain a user profile of those customers. The customers can register themselves to the system and they can update and delete their profiles as they need. Also, these details are taken by the billing section and power consumption section to calculate the bills on a monthly basis. Also, there is a customer care management web service where it allows the customers to lodge he complaints regarding the service, breakdowns or any other interruptions that thy face.

Main stakeholders - Customer

Customer service manager

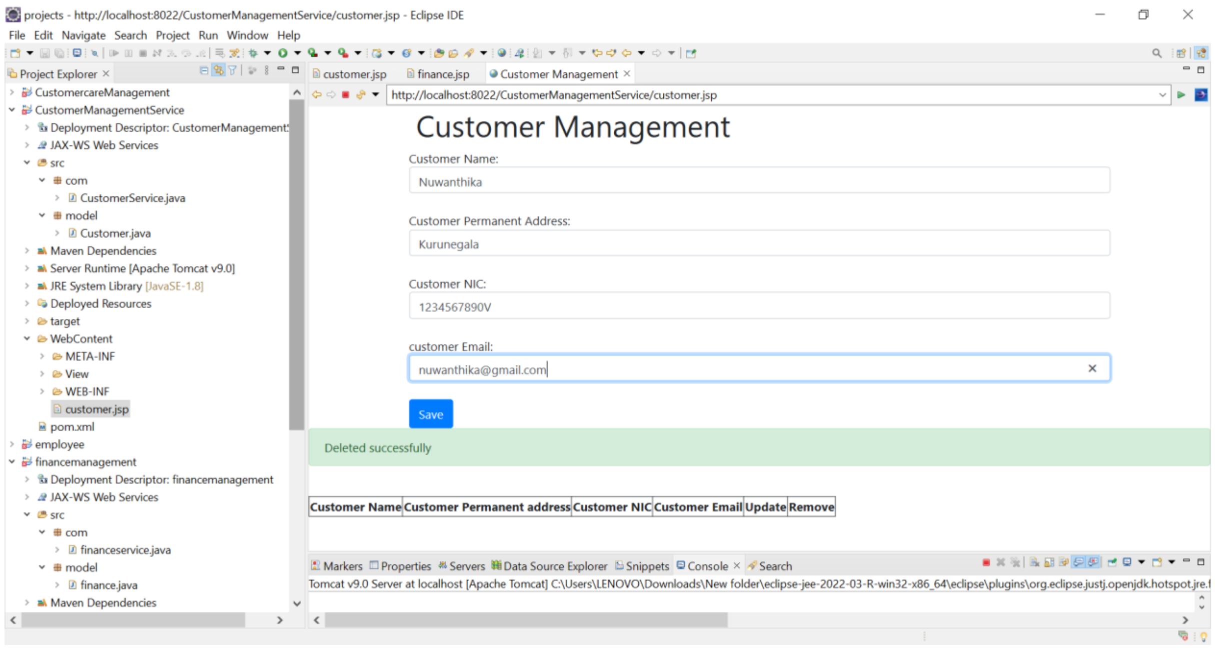
Logic used – The customer can register himself by entering name, address, nic, email and those details taken through customer.jsp are sent to customerservice.java, then those are taken to customer.java and inserted, viewed, updated and deleted according to customer’s request.

Code


projects - CustomerManagementService/src/com/CustomerService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
customer.jsp
Customer.java
CustomerService.java
Writable
Smart Insert
1 : 1 : 0

```java
import model.Customer;
@Path("/Customer")

public class CustomerService {
    Customer Obj = new Customer();
    @GET
    @Path("/")
    @Produces(MediaType.TEXT_HTML)
    public String readItems()
     {
     return Obj.readItems();
     }

    @POST
    @Path("/")
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    @Produces(MediaType.TEXT_PLAIN)
    public String insertItem(@FormParam("customerPassword") String customerPassword,
     @FormParam("customerName") String customerName,
     @FormParam("customerPhone") String customerPhone,
     @FormParam("customerEmail") String customerEmail)
    {
     String output = Obj.insertcustomerdetails(customerPassword,  customerName, customerPhone, customerEmail);
    return output;
    }

    @PUT
    @Path("/")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.TEXT_PLAIN)
    public String updateItem(String itemData)
    {
    //Convert the input string to a JSON object
     JsonObject Object = new JsonParser().parse(itemData).getAsJsonObject();
    //Read the values from the JSON object
     String customerID = Object.get("customerID").getAsString();
     String customerPassword = Object.get("customerPassword").getAsString();
```

User Interface - Before inserting



Customer Management

Customer Name:
Nuwanthika

Customer Permanent Address:
Kurunegala

Customer NIC:
1234567890V

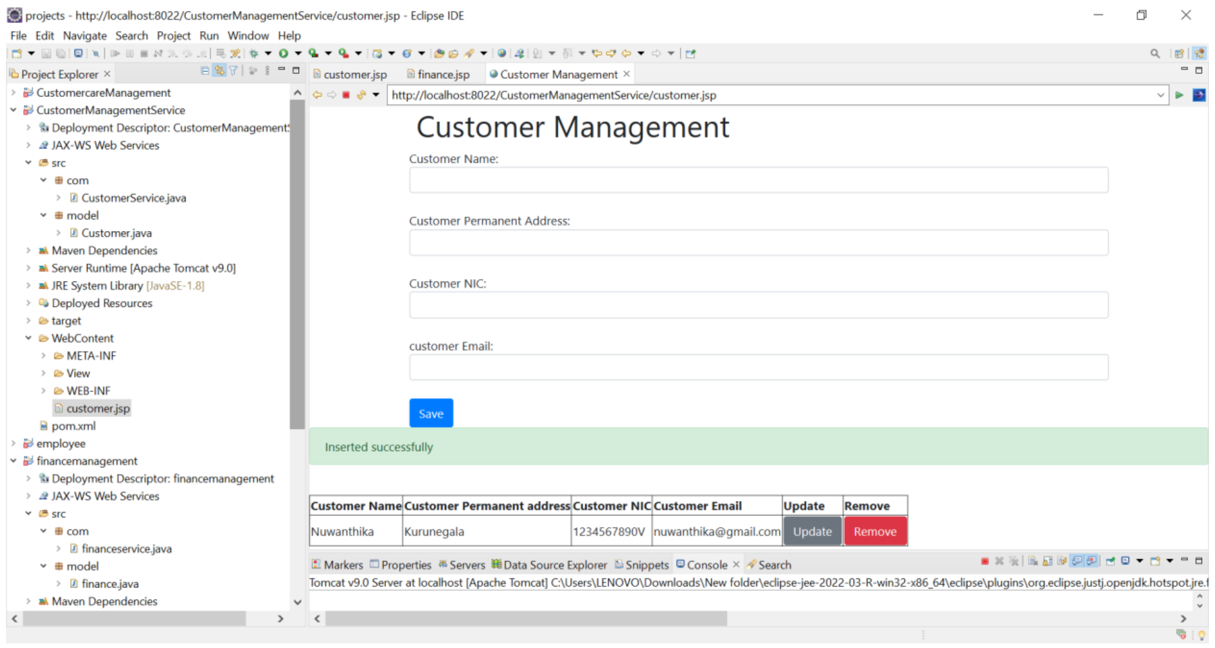
customer Email:
nuwanthika@gmail.com

Deleted successfully

Customer Name	Customer Permanent address	Customer NIC	Customer Email	Update	Remove

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
```

User Interface – After inserting


projects - http://localhost:8022/CustomerManagementService/customer.jsp - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
CustomercareManagement
CustomerManagementService
Deployment Descriptor: CustomerManagementService
JAX-WS Web Services
src
com
CustomerService.java
model
Customer.java
Maven Dependencies
Server Runtime [Apache Tomcat v9.0]
JRE System Library [JavaSE-1.8]
Deployed Resources
target
WebContent
META-INF
View
WEB-INF
customer.jsp
pom.xml
employee
financemanagement
Deployment Descriptor: financemanagement
JAX-WS Web Services
src
com
financeservice.java
model
finance.java
Maven Dependencies
customer.jsp
finance.jsp
Customer Management
http://localhost:8022/CustomerManagementService/customer.jsp
Markers Properties Servers Data Source Explorer Snippets Console Search

Customer Management

Customer Name:

Customer Permanent Address:

Customer NIC:

customer Email:

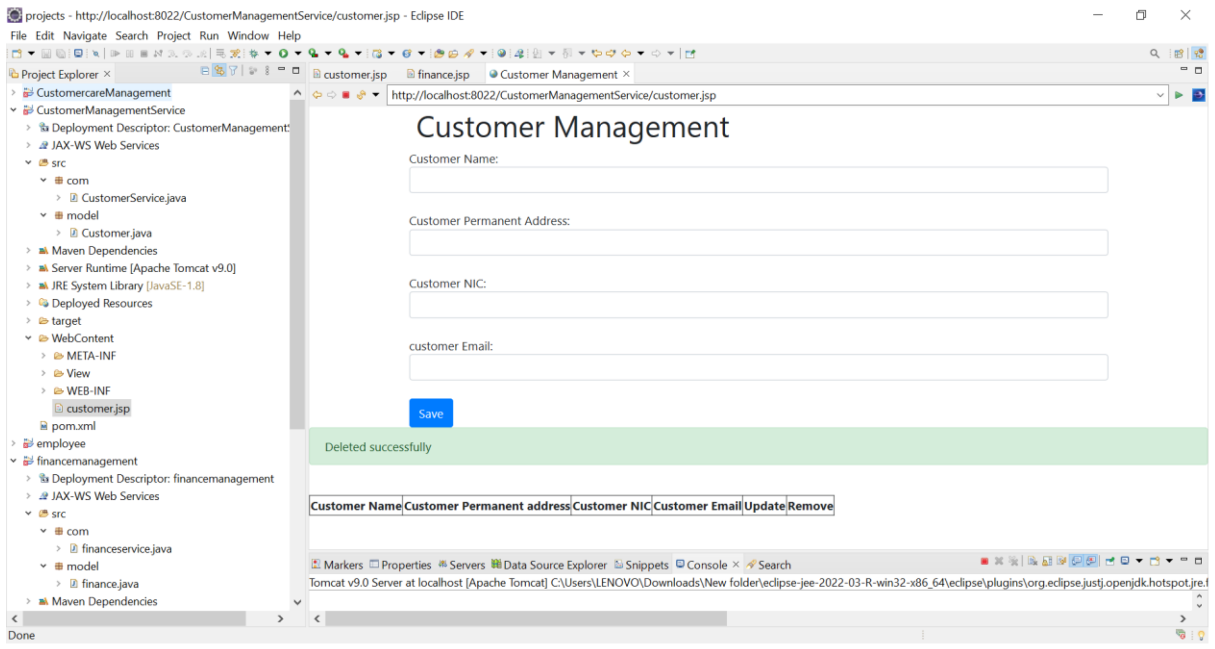
Save

Inserted successfully

Customer Name	Customer Permanent address	Customer NIC	Customer Email	Update	Remove
Nuwanthika	Kurunegala	1234567890V	nuwanthika@gmail.com	Update	Remove

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
```

User Interface – After deletion


projects - http://localhost:8022/CustomerManagementService/customer.jsp - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
CustomercareManagement
CustomerManagementService
Deployment Descriptor: CustomerManagementService
JAX-WS Web Services
src
com
CustomerService.java
model
Customer.java
Maven Dependencies
Server Runtime [Apache Tomcat v9.0]
JRE System Library [JavaSE-1.8]
Deployed Resources
target
WebContent
META-INF
View
WEB-INF
customer.jsp
pom.xml
employee
financemanagement
Deployment Descriptor: financemanagement
JAX-WS Web Services
src
com
financeservice.java
model
finance.java
Maven Dependencies
customer.jsp
finance.jsp
Customer Management
http://localhost:8022/CustomerManagementService/customer.jsp
Markers Properties Servers Data Source Explorer Snippets Console Search
Done

Customer Management

Customer Name:

Customer Permanent Address:

Customer NIC:

customer Email:

Save

Deleted successfully

Customer Name	Customer Permanent address	Customer NIC	Customer Email	Update	Remove

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
```

A.N Upathissa – IT20141356

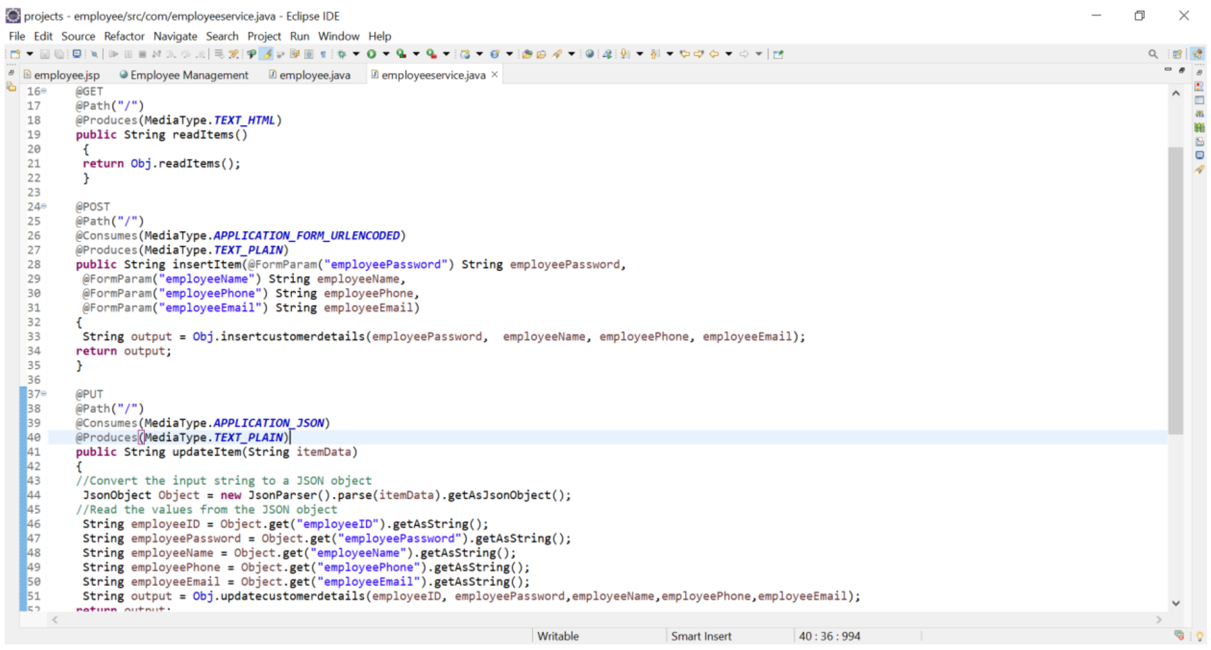
Web service - Employee Management

Purpose - The main purpose of this web service is to register new Employees to the company and maintain a user profile of those employees. The employees are registered by the staff manager. The main details id, name, address, nic, position and their contact number are taken in-order to register an employee to the system.

Main stakeholders - Staff manager

Logic used – The employee can register by entering id, name, contact, address, nic, position through the employee management and those details taken through employee.jsp are sent to employeeservice.java, then those are taken to employee.java and inserted, viewed, updated and deleted according to each employee.

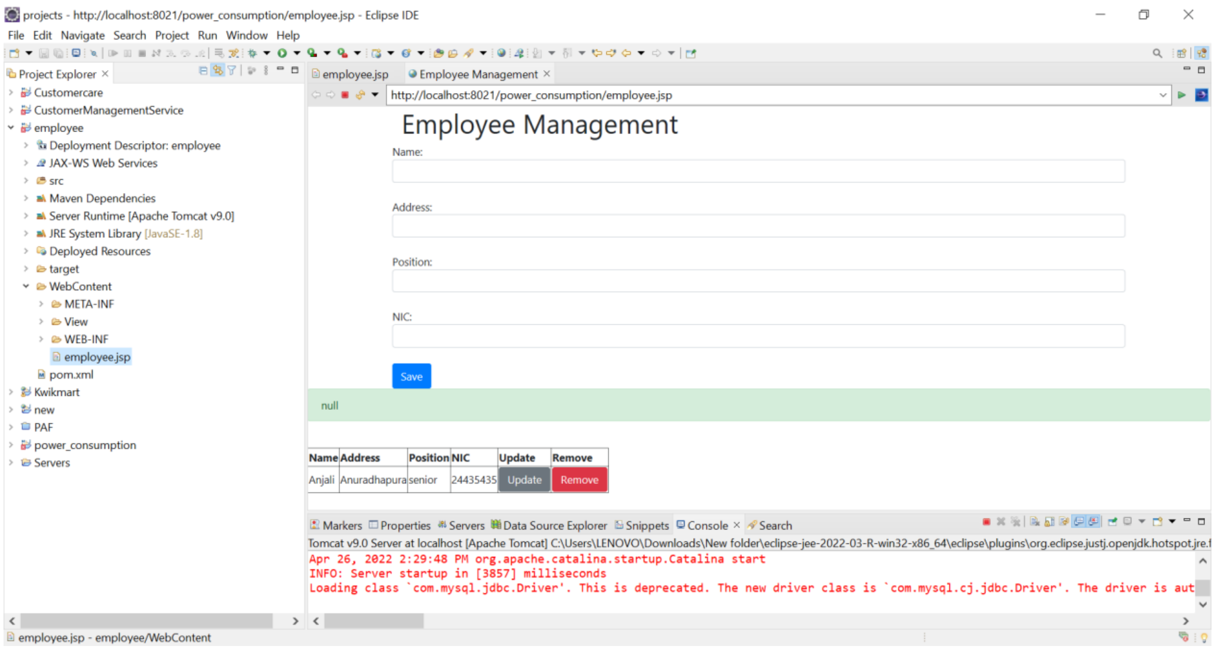
Code


projects - employee/src/com/employeeservice.java - Eclipse IDE

```java
    @GET
    @Path("/")
    @Produces(MediaType.TEXT_HTML)
    public String readItems()
     {
     return Obj.readItems();
     }

    @POST
    @Path("/")
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    @Produces(MediaType.TEXT_PLAIN)
    public String insertItem(@FormParam("employeePassword") String employeePassword,
     @FormParam("employeeName") String employeeName,
     @FormParam("employeePhone") String employeePhone,
     @FormParam("employeeEmail") String employeeEmail)
    {
     String output = Obj.insertcustomerdetails(employeePassword,  employeeName, employeePhone, employeeEmail);
    return output;
    }

    @PUT
    @Path("/")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.TEXT_PLAIN)
    public String updateItem(String itemData)
    {
    //Convert the input string to a JSON object
     JsonObject Object = new JsonParser().parse(itemData).getAsJsonObject();
    //Read the values from the JSON object
     String employeeID = Object.get("employeeID").getAsString();
     String employeePassword = Object.get("employeePassword").getAsString();
     String employeeName = Object.get("employeeName").getAsString();
     String employeePhone = Object.get("employeePhone").getAsString();
     String employeeEmail = Object.get("employeeEmail").getAsString();
     String output = Obj.updatecustomerdetails(employeeID, employeePassword,employeeName,employeePhone,employeeEmail);
    return output;
```

Writable | Smart Insert | 40 : 36 : 994

User Interface - Before inserting


projects - http://localhost:8021/power_consumption/employee.jsp - Eclipse IDE

http://localhost:8021/power_consumption/employee.jsp

Employee Management

Name:

Address:

Position:

NIC:

Save

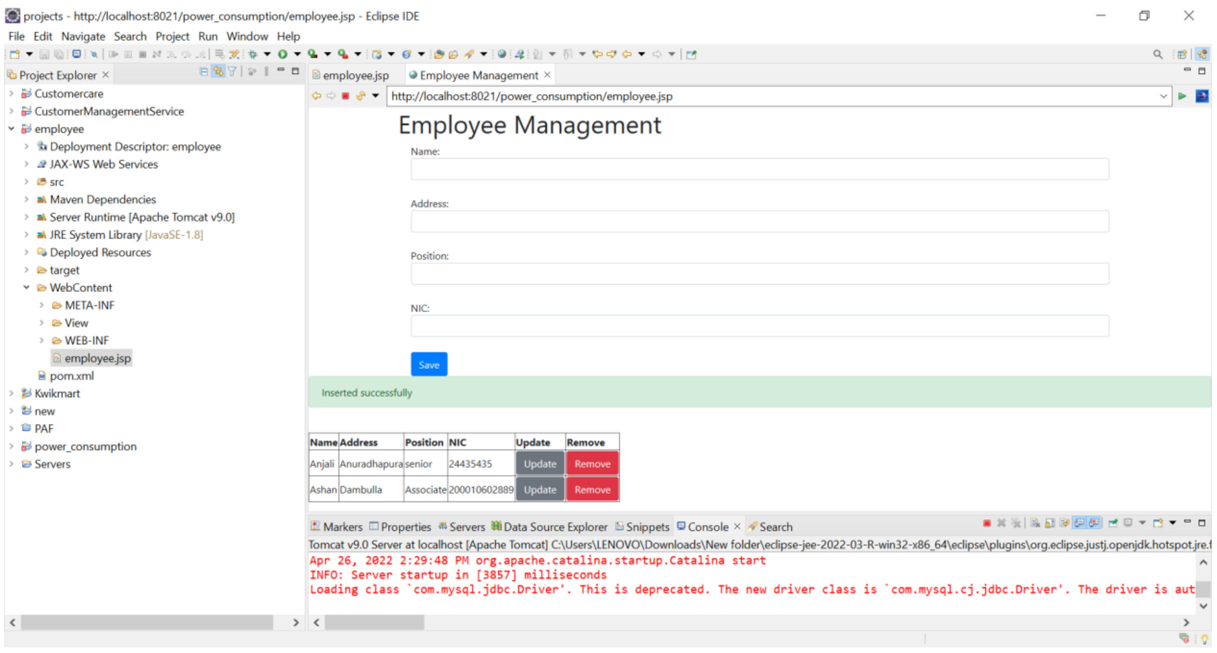
null

Name	Address	Position	NIC	Update	Remove
Anjali	Anuradhapura	senior	24435435	Update	Remove

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
Apr 26, 2022 2:29:48 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [3857] milliseconds
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is aut
```

Project Explorer
Customercare
CustomerManagementService
employee
Deployment Descriptor: employee
JAX-WS Web Services
src
Maven Dependencies
Server Runtime [Apache Tomcat v9.0]
JRE System Library [JavaSE-1.8]
Deployed Resources
target
WebContent
META-INF
View
WEB-INF
employee.jsp
pom.xml
Kwikmart
new
PAF
power_consumption
Servers
Markers Properties Servers Data Source Explorer Snippets Console Search
employee.jsp - employee/WebContent

User Interface – After inserting


projects - http://localhost:8021/power_consumption/employee.jsp - Eclipse IDE

http://localhost:8021/power_consumption/employee.jsp

Employee Management

Name:

Address:

Position:

NIC:

Save

Inserted successfully

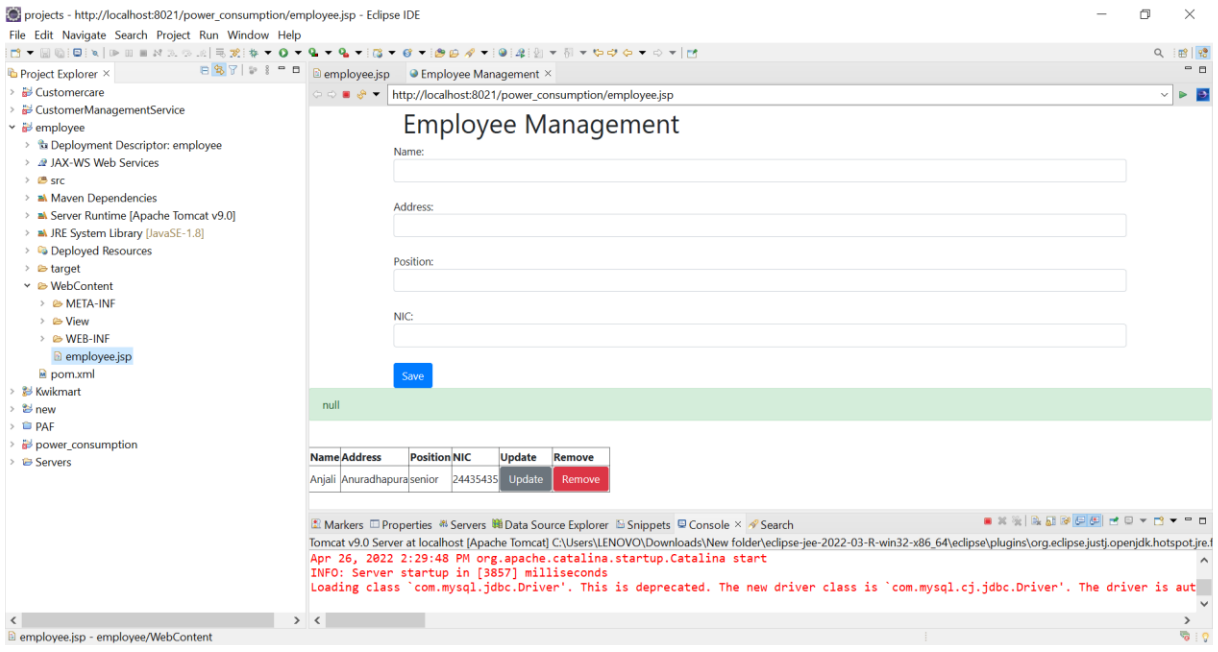
Name	Address	Position	NIC	Update	Remove
Anjali	Anuradhapura	senior	24435435	Update	Remove
Ashan	Dambulla	Associate	200010602889	Update	Remove

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f

```
Apr 26, 2022 2:29:48 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [3857] milliseconds
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is aut
```

File Edit Navigate Search Project Run Window Help
Project Explorer
Customercare
CustomerManagementService
employee
Deployment Descriptor: employee
JAX-WS Web Services
src
Maven Dependencies
Server Runtime [Apache Tomcat v9.0]
JRE System Library [JavaSE-1.8]
Deployed Resources
target
WebContent
META-INF
View
WEB-INF
employee.jsp
pom.xml
Kwikmart
new
PAF
power_consumption
Servers
employee.jsp
Employee Management
Markers Properties Servers Data Source Explorer Snippets Console Search

User Interface – After deletion


projects - http://localhost:8021/power_consumption/employee.jsp - Eclipse IDE

http://localhost:8021/power_consumption/employee.jsp

Employee Management

Name:

Address:

Position:

NIC:

Save

null

Name	Address	Position	NIC	Update	Remove
Anjali	Anuradhapura	senior	24435435	Update	Remove

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f

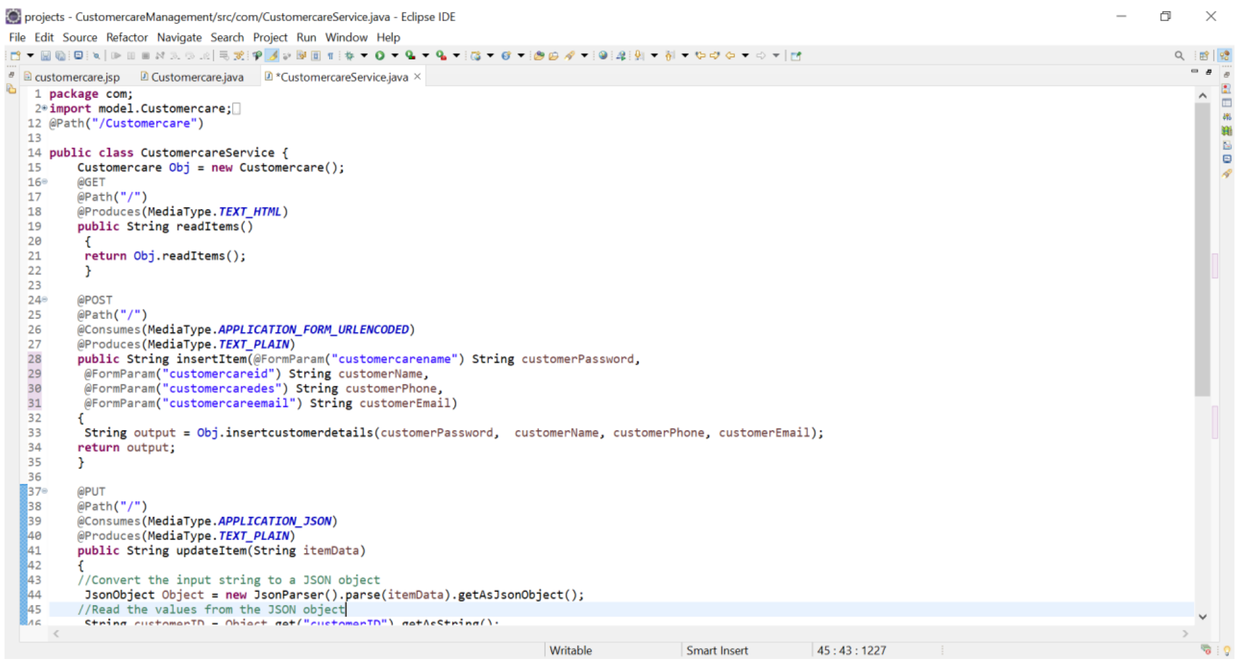
```
Apr 26, 2022 2:29:48 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [3857] milliseconds
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is aut
```

employee.jsp - employee/WebContent

File Edit Navigate Search Project Run Window Help
Project Explorer
Customercare
CustomerManagementService
employee
Deployment Descriptor: employee
JAX-WS Web Services
src
Maven Dependencies
Server Runtime [Apache Tomcat v9.0]
JRE System Library [JavaSE-1.8]
Deployed Resources
target
WebContent
META-INF
View
WEB-INF
employee.jsp
pom.xml
Kwikmart
new
PAF
power_consumption
Servers
employee.jsp
Employee Management
Markers Properties Servers Data Source Explorer Snippets Console Search

Customer Care Management

Code



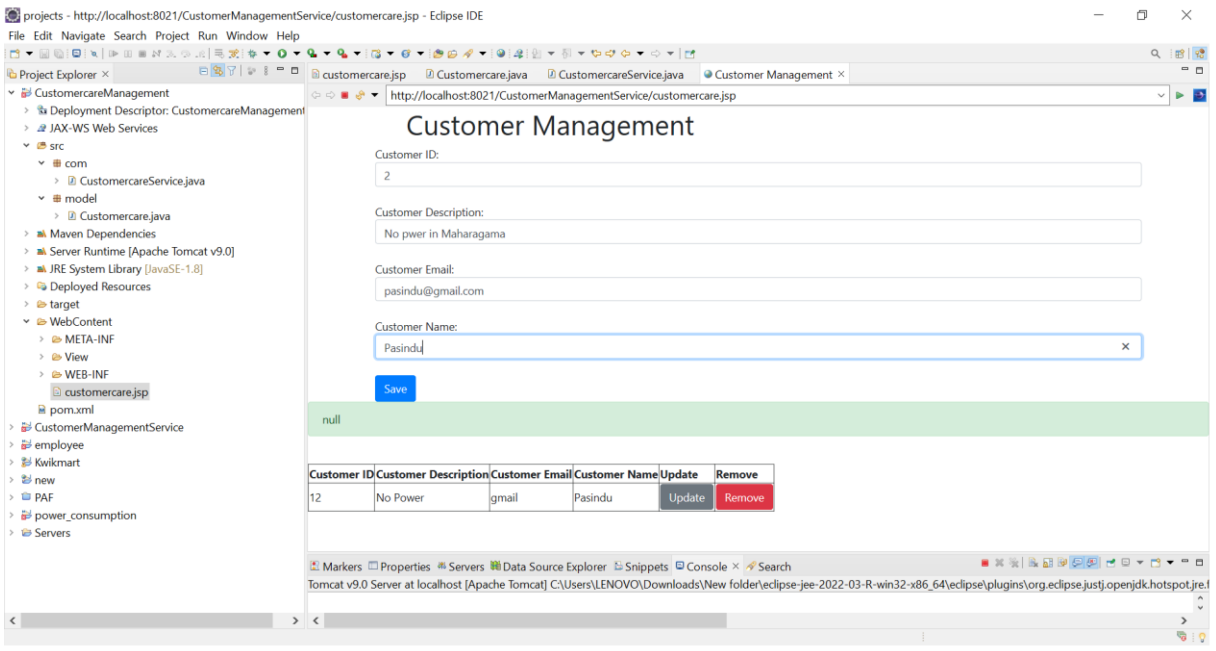
projects - CustomercareManagement/src/com/CustomercareService.java - Eclipse IDE

customercare.jsp | Customercare.java | *CustomercareService.java

```java
package com;
import model.Customercare;
@Path("/Customercare")

public class CustomercareService {
    Customercare Obj = new Customercare();
    @GET
    @Path("/")
    @Produces(MediaType.TEXT_HTML)
    public String readItems()
     {
     return Obj.readItems();
     }

    @POST
    @Path("/")
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    @Produces(MediaType.TEXT_PLAIN)
    public String insertItem(@FormParam("customercarename") String customerPassword,
     @FormParam("customercareid") String customerName,
     @FormParam("customercaredes") String customerPhone,
     @FormParam("customercareemail") String customerEmail)
    {
     String output = Obj.insertcustomerdetails(customerPassword,  customerName, customerPhone, customerEmail);
    return output;
    }

    @PUT
    @Path("/")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.TEXT_PLAIN)
    public String updateItem(String itemData)
    {
    //Convert the input string to a JSON object
     JsonObject Object = new JsonParser().parse(itemData).getAsJsonObject();
    //Read the values from the JSON object
     String customerID = Object.get("customerID").getAsString();
```

Writable | Smart Insert | 45 : 43 : 1227

User Interface - Before inserting



projects - http://localhost:8021/CustomerManagementService/customercare.jsp - Eclipse IDE

http://localhost:8021/CustomerManagementService/customercare.jsp

Customer Management

Customer ID: 2

Customer Description: No pwer in Maharagama

Customer Email: pasindu@gmail.com

Customer Name: Pasindu

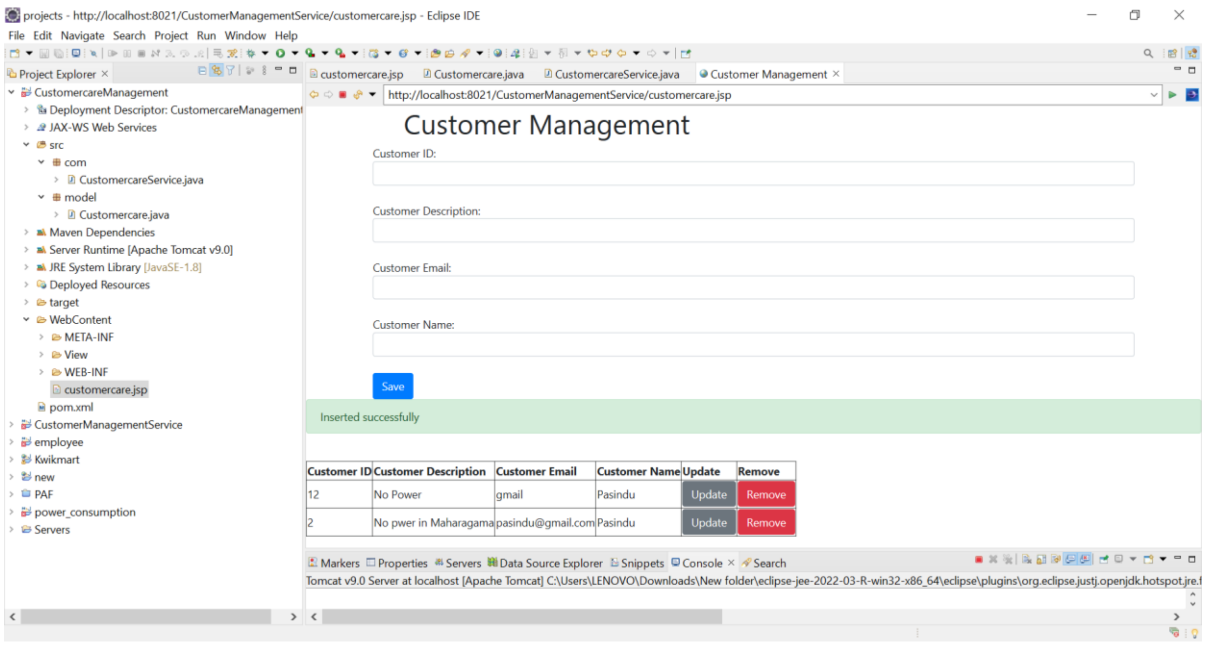
Save

null

Customer ID	Customer Description	Customer Email	Customer Name	Update	Remove
12	No Power	gmail	Pasindu	Update	Remove

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
```

User Interface – After inserting



projects - http://localhost:8021/CustomerManagementService/customercare.jsp - Eclipse IDE

Project Explorer: CustomercareManagement, Deployment Descriptor: CustomercareManagement, JAX-WS Web Services, src, com, CustomercareService.java, model, Customercare.java, Maven Dependencies, Server Runtime [Apache Tomcat v9.0], JRE System Library [JavaSE-1.8], Deployed Resources, target, WebContent, META-INF, View, WEB-INF, customercare.jsp, pom.xml, CustomerManagementService, employee, Kwikmart, new, PAF, power_consumption, Servers

http://localhost:8021/CustomerManagementService/customercare.jsp

Customer Management

Customer ID:

Customer Description:

Customer Email:

Customer Name:

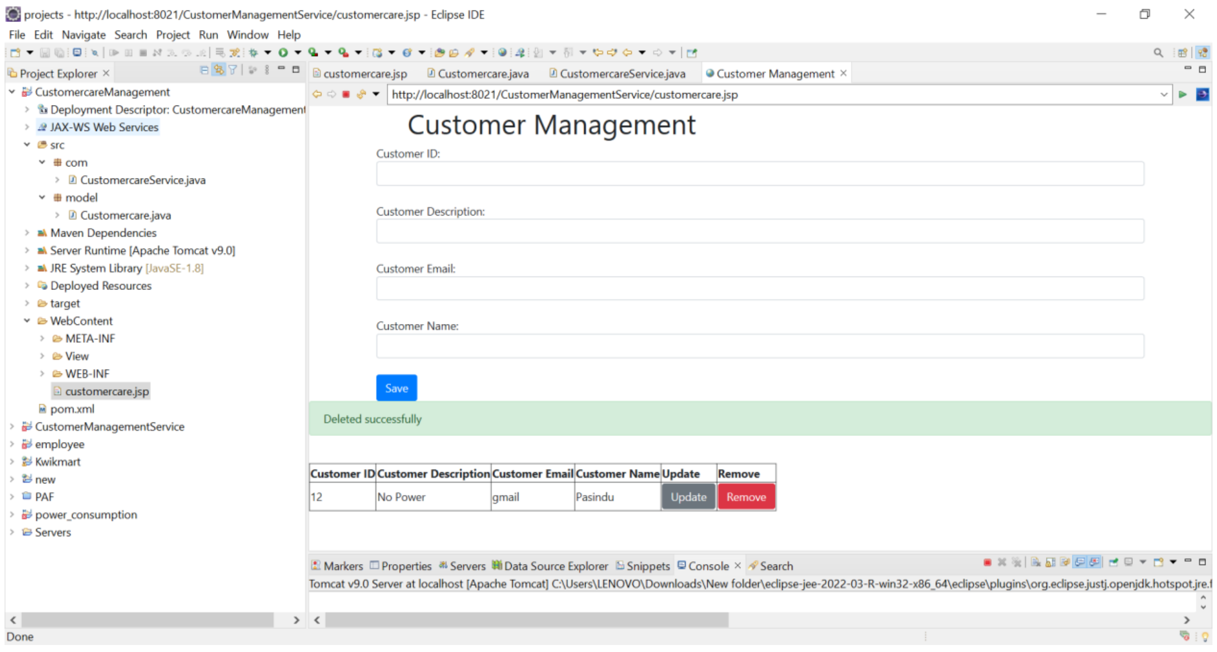
Save

Inserted successfully

Customer ID	Customer Description	Customer Email	Customer Name	Update	Remove
12	No Power	gmail	Pasindu	Update	Remove
2	No pwer in Maharagama	pasindu@gmail.com	Pasindu	Update	Remove

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f

User Interface – After deletion



projects - http://localhost:8021/CustomerManagementService/customercare.jsp - Eclipse IDE

http://localhost:8021/CustomerManagementService/customercare.jsp

Customer Management

Customer ID:

Customer Description:

Customer Email:

Customer Name:

Save

Deleted successfully

Customer ID	Customer Description	Customer Email	Customer Name	Update	Remove
12	No Power	gmail	Pasindu	Update	Remove

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Users\LENOVO\Downloads\New folder\eclipse-jee-2022-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f

Done

Thank You !