

IT Overview Final Project Report

2016

**Smart Feeder and Energy Manager (SFEM)**

**Submitted By**

**Prince Mathew**

**Rahul Thaivalappil**

Contents

[1 Introduction 3](#_Toc442755803)

[1.1 Purpose 3](#_Toc442755804)

[1.2 Overview 3](#_Toc442755805)

[1.2.1 Smart Feeder 3](#_Toc442755806)

[1.2.2 Energy Manager 3](#_Toc442755807)

[1.3 Background 4](#_Toc442755808)

[1.3.1 Identify problem 4](#_Toc442755809)

[1.3.2 Identify solution 5](#_Toc442755810)

[2 Detailed Overview 6](#_Toc442755811)

[2.1 User stories 6](#_Toc442755812)

[2.1.1 Energy Manager 6](#_Toc442755813)

[2.1.2 Smart Feeder 7](#_Toc442755814)

[2.2 Technical Details 8](#_Toc442755815)

[2.2.1 Software 8](#_Toc442755816)

[2.3 Hardware 22](#_Toc442755817)

[2.3.1 Energy Manager 22](#_Toc442755818)

[2.3.2 Smart Feeder 23](#_Toc442755819)

[3 How SFEM works? 24](#_Toc442755820)

[3.1 Smart Feeder 24](#_Toc442755821)

[3.2 Energy Manager 26](#_Toc442755822)

[4 Learning from Technology 26](#_Toc442755823)

[4.1 Tools 27](#_Toc442755824)

[4.2 Technology 27](#_Toc442755825)

[4.2.1 Server 27](#_Toc442755826)

[4.2.2 Web service 27](#_Toc442755827)

[4.2.3 Client 28](#_Toc442755828)

[4.2.4 Design Patterns 28](#_Toc442755829)

[5 Learning from Project 28](#_Toc442755830)

[6 Project Management 28](#_Toc442755831)

[7 Future Possibilities 29](#_Toc442755832)

[7.1 General 29](#_Toc442755833)

[7.2 Energy Manger 30](#_Toc442755834)

[7.3 Feeder Manager 30](#_Toc442755835)

[8 Bibliography 30](#_Toc442755836)

# Introduction

## Purpose

This document is the report for the project ‘Smart Feeder and Energy Manager’ which is abbreviated as SFEM.

## Overview

The application of computer interfaced controlling devices is increasingly rapidly in modern age. Smart Feeder and Energy Manager (SFEM) is a web based application which improves the operational efficiency and manages the energy consumption of an electrical power distribution system.

SFEM comes with two applications,

* Smart Feeder
* Energy Manager

### Smart Feeder

An advanced, Web-based monitoring and troubleshooting system that keeps 24-hour watch on an electric power distribution feeder system. This application targets the power utilities which generates, transmits and distributes electric power. The system allows configuring the layout of a particular feeder which needs to be monitored. The application notifies the engineer or lineman if there is a fault occurred in the FEEDER which can be caused by overload, broken electric poles or conductors and ground faults. This will enable the engineer or lineman to immediately act on the situation and rectify the faults.

The Smart Feeder consists of hardware and software units. The hardware units include fault detection sensors and the central unit which receives signal from each sensors located in the substation. The software unit includes a web based application which runs on a pre-configured server.

### Energy Manager

This Web-based dashboard application collects consumed energy information and presents it in a customized, easy-to-use format, helping the user to better understand their facility’s energy usage. This application targets the domestic consumers like houses and institutions who consumes the electric power distributed from a FEEDER.

Energy manger consists of hardware and software units. The hardware units include smart meters which are capable to record and send the energy consumption of equipment’s which are connected to it. The software unit includes a server and web based dashboard application.

## Background

This project is evolved from an idea to utilize the wide application of information technology to the energy sector.

### **Identify** problem

The two main challenges faced by electrical power sector are indiscriminative consumption of energy and power cuts.

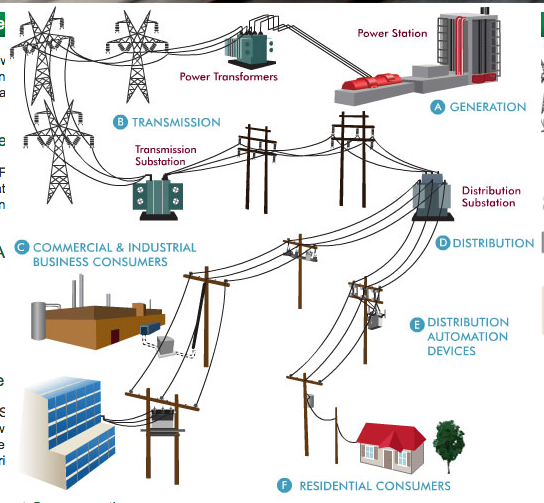
#### **Conserve energy**

Energy Conservation is unquestionably of great importance to all of us, since we rely on energy for everything we do every single day. Energy supplies are limited and, to maintain a good quality of life, we must find ways to use energy wisely. Reducing the amount of energy that we use is a good way to save

money, and there are also other benefits to decreasing energy consumption.

#### **Power cuts**

The phrase “Power Cuts” is quite well known and familiar to most people in India. Many of us have seen lot of power cuts in our lives, some planned and some unplanned. Natural disasters and unexpected events have turned into such important causes why power cuts happen frequently affecting mainly the distribution power lines.

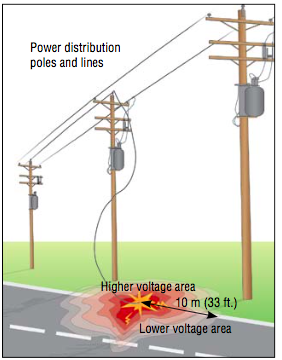


Below, are listed the main reasons for unexpected power cuts,

* Overload
* Broken electric poles or conductors due to natural disasters
* Ground faults (Short circuits) caused by momentary tree contact, bird or other animal contact, lightning strike and conductor clashing.



The distributed power lines are spread across kilometers and most of the time when the power cut occurs, consumer has to inform the engineer or lineman to rectify the issues. In the case of broken poles or electric lines, Power Company has to be informed immediately to de-energize the conductors. In the case of conductor clash due to tree contact or other objects, lineman has to manually check suspicious areas to find out the faulty locations. This can be a lengthy process and consumers need to wait patiently.



### Identify solution

#### **Energy Dashboard**

We realized that an energy manager dashboard for real time monitoring of energy use will help to effectively manage the energy consumption.

#### **Smart Feeder**

The identified solution to this problem is to develop an application by which the power company can keep track of the Feeder status which is mainly the distribution power lines.

The idea is to fix electric sensors and relays on the poles which can transmit the

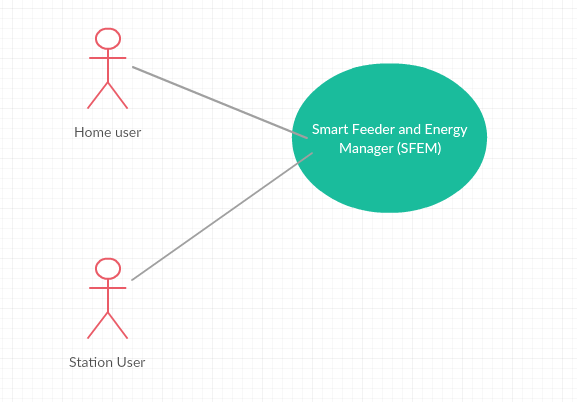
changes in the system to the substation and synchronize them with the application to display the status of the poles in the given distribution feeder.

The figure shows an overhead power system.

# Detailed Overview

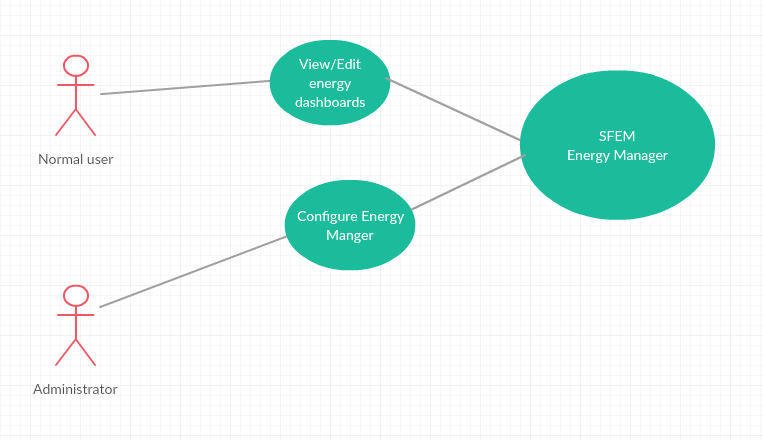
## User stories

The application smart manager is developed for power utility companies and energy manager is targeted for domestic consumers.



### Energy Manager

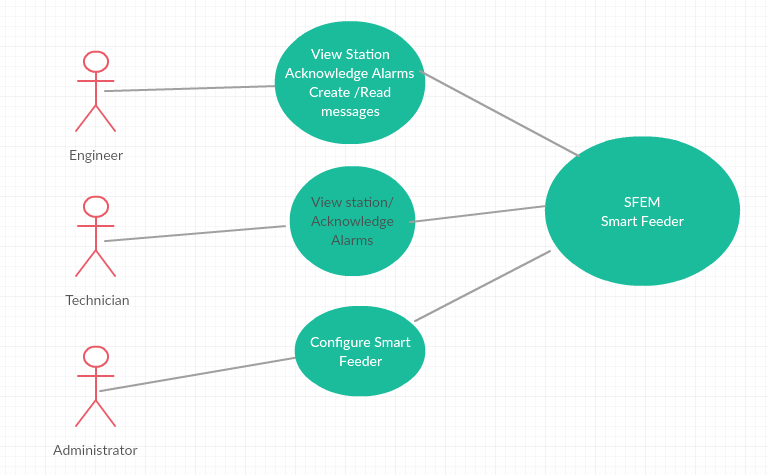
* There are two types of users in Energy manager,
* Normal user - User can create an account and add widgets, view and edit widgets.
* Admin User - Admin user is the one who configure the SFEM application during the installation.



### Smart Feeder

There are three types of users in Smart feeder,

* Engineer - Engineer can monitor the station, acknowledge alarms, assign task to the operator
* Operator - Operator can monitor the station, acknowledge alarms and act according the work assigned by engineer
* Admin User - Admin user is the one who configure the SFEM application during the installation



## Technical Details

### Software

SFEM is a dynamic web application which uses service oriented architecture (SOA). The application design is strictly following the object oriented design principles and the classes are open for extension and closed for modifications.

The architecture is designed in such a way that it is easily pluggable to different databases. The application is designed in such a way that both energy manager and smart feeder uses the same framework

**Server**

**Derby DB**

**Client**

**(Browser)**



#### **Server**

The server application is developed in Java and Derby database is used for storage.

The technologies or tools used are

* Java
* SQL
* Derby DB
* Jersey framework
* Eclipse

#### **Server-Client communication**

Service oriented architecture (SOA) based RESTful web services are used for the server client communications. Java Jersey frame work is used to achieve the same and the data is sending in XML format.

#### **Security**

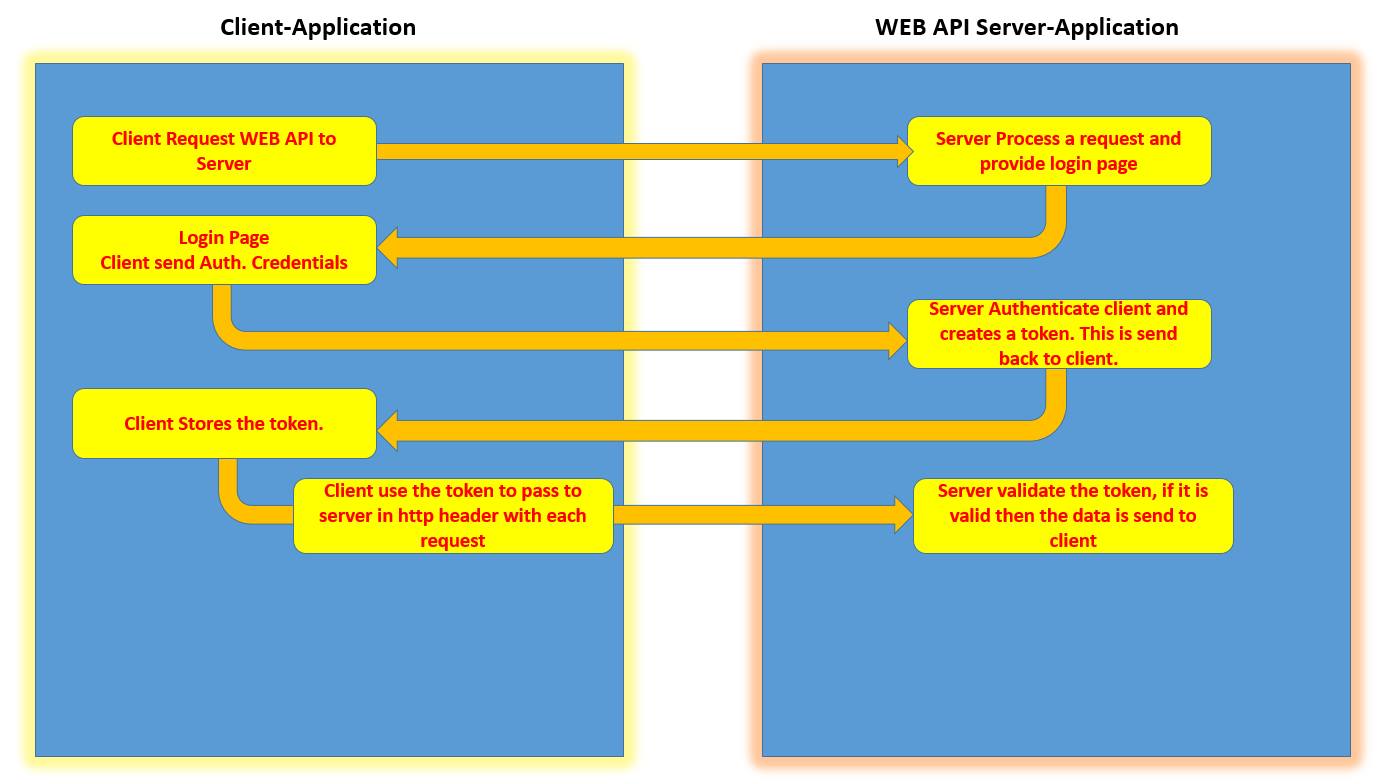
There are numerous network security and access control issues to be considered when implementing a client/server system. As the server is usually the central location for critical data, adequate physical security and operational security measures need to be taken to insure the safety of the data.

Following are the major possible threats for this application,

* A Man in the Middle attack in which an impostor claims to be the server for the clients and pretends to be the client for the server. The imposter then intercepts all messages and has access to all secure data.
* Manipulating a clientto pretend to be another client in order to gain access to sessions connected to the spoofed client.

**Token based Authentication**

SFEM is authenticated using token mechanism by which the security of the system has been improved.

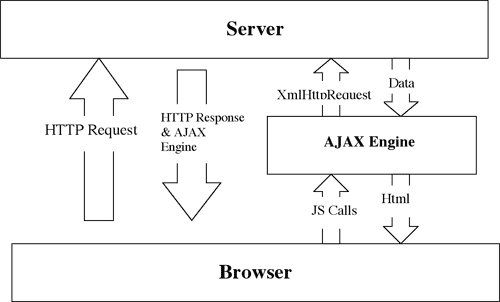


#### **Client**

The client side is designed on Ajax model and the application support cross browser compatibility.

The technologies used are

* HTML5
* CSS, SVG
* JQuery
* Bootstrap
* Java Primeface dashboards



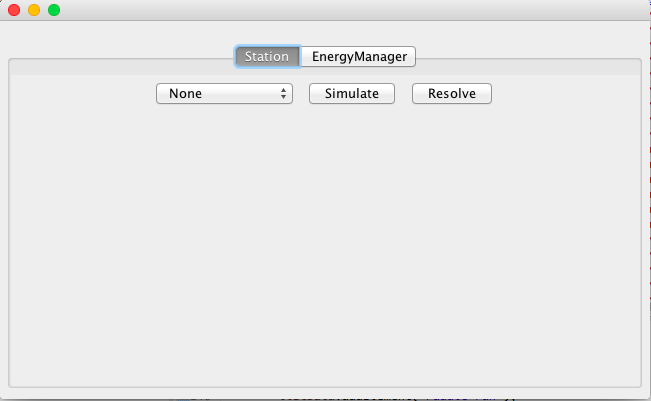
#### **Test Simulator**

Developing the hardware unit is out of this project and in order to simulate the data required for the application, a test simulator is developed. It is a dialog based application developed in java and communicates to the server using socket communication.

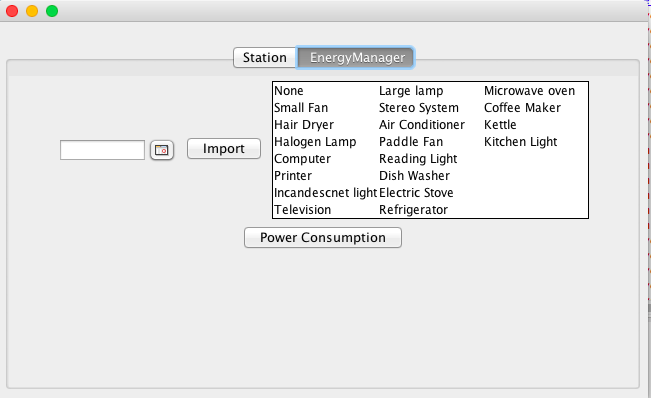
#### **Configuration files**

Two XML files are used to configure the feeder components and they are named like Feeder.xml and Binding.xml

* 1. **Smart Feeder Simulator**



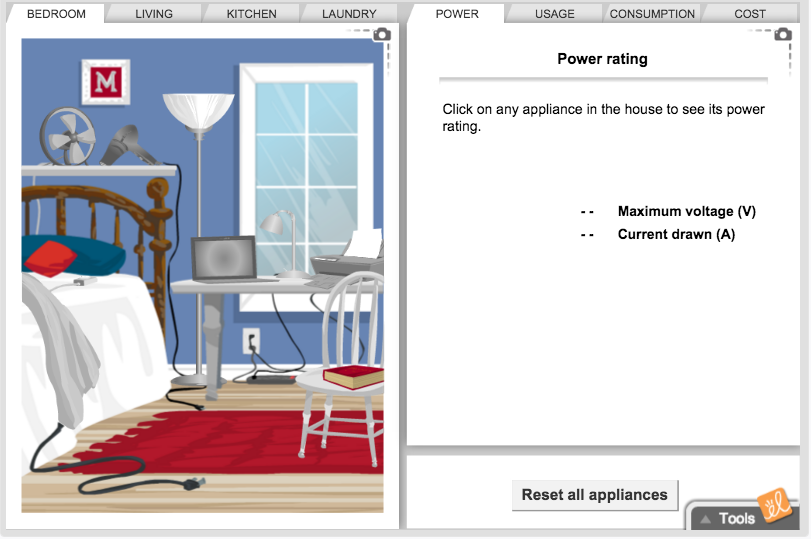
1. **Energy Manager Simulator**



1. **Energy consumption test data**

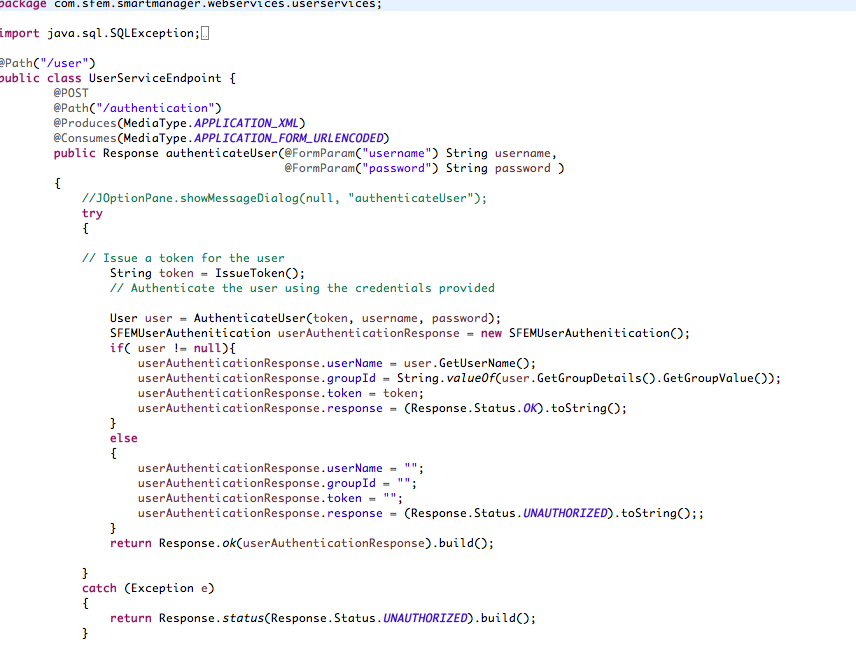
An online energy consumption simulator is used to generate the test data. It is available in

<https://www.explorelearning.com/index.cfm?method=cResource.dspDetail&ResourceID=444>

****

#### **Source code**

* 1. **UserServiceEndpoint.java**

****

****

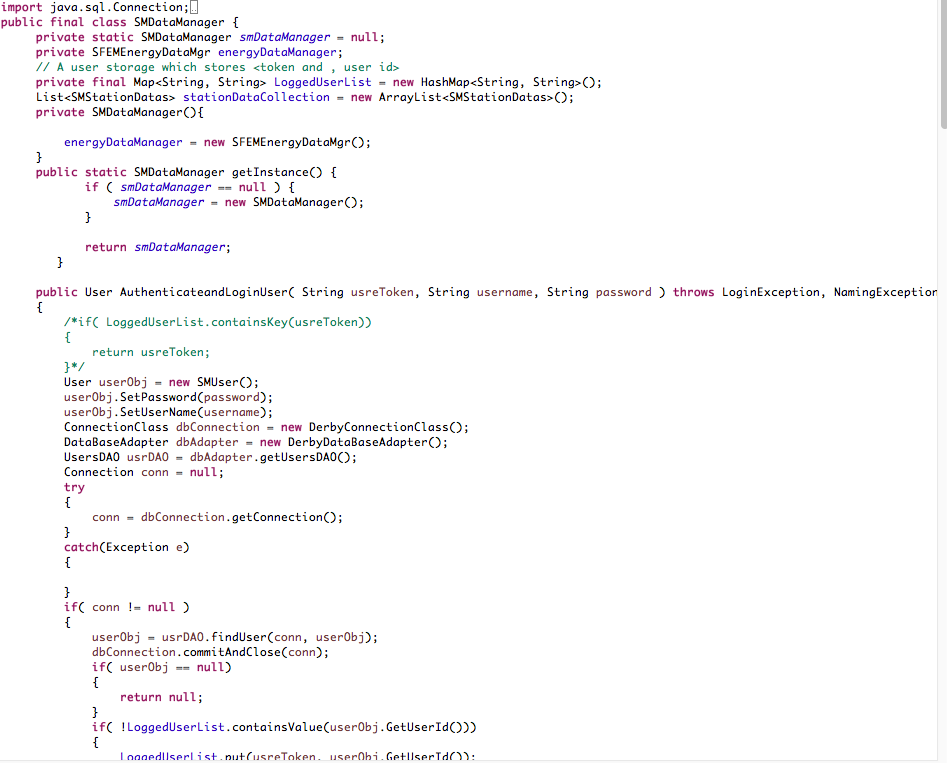
* 1. **SFEMSationService.java**

****

* 1. **EnergymanagerServices.java**

****

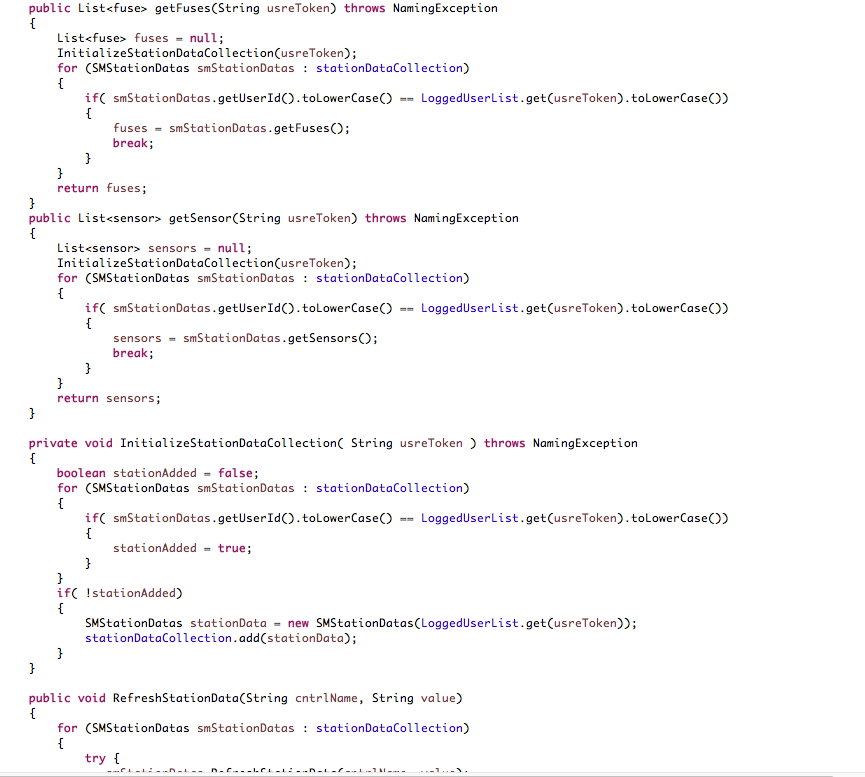
* 1. **SMDataManager.java**

****

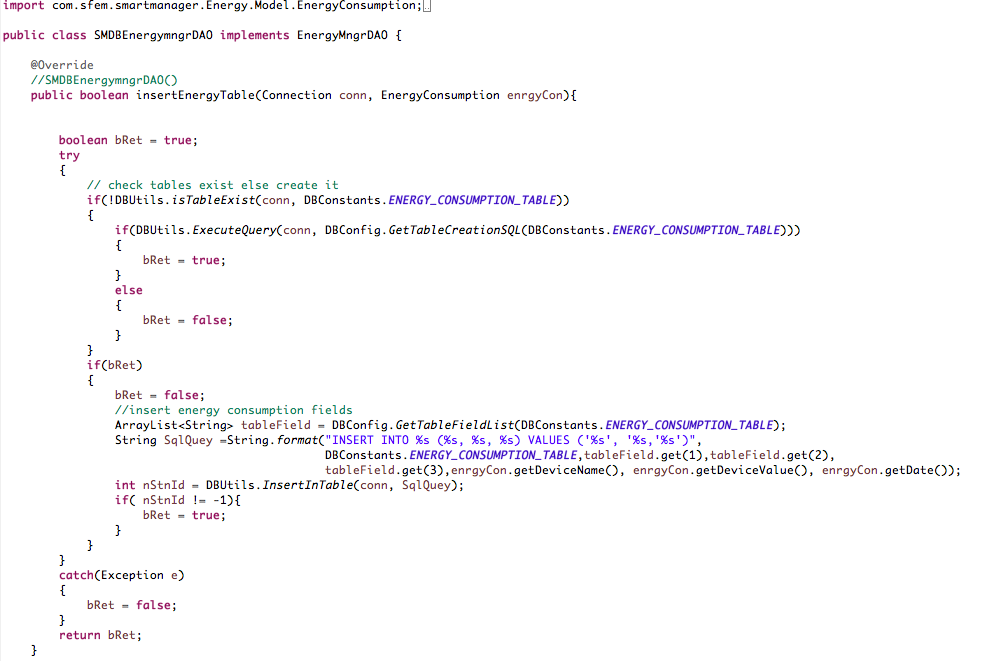
****

****

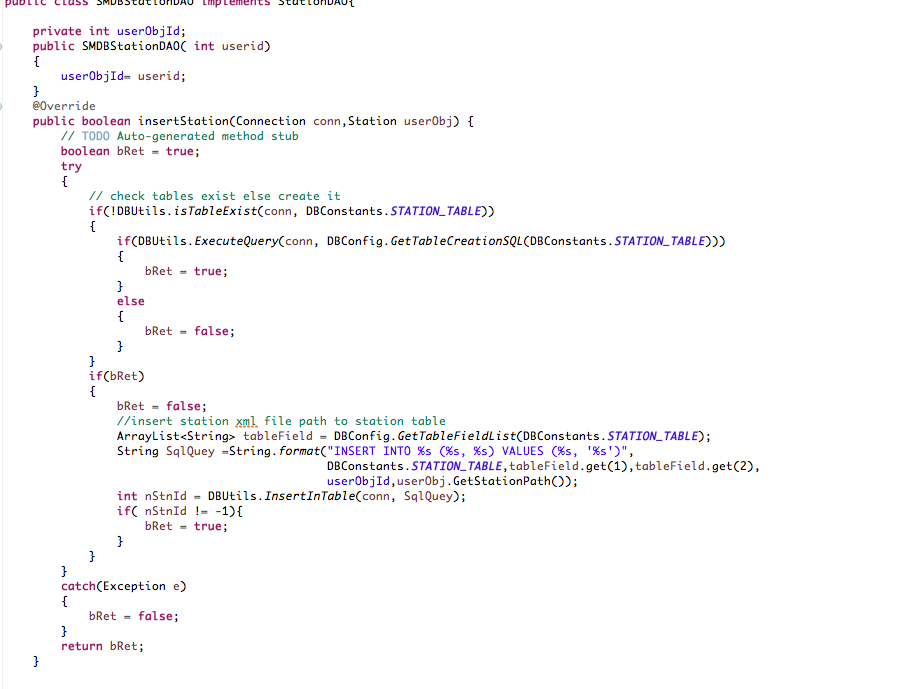
****

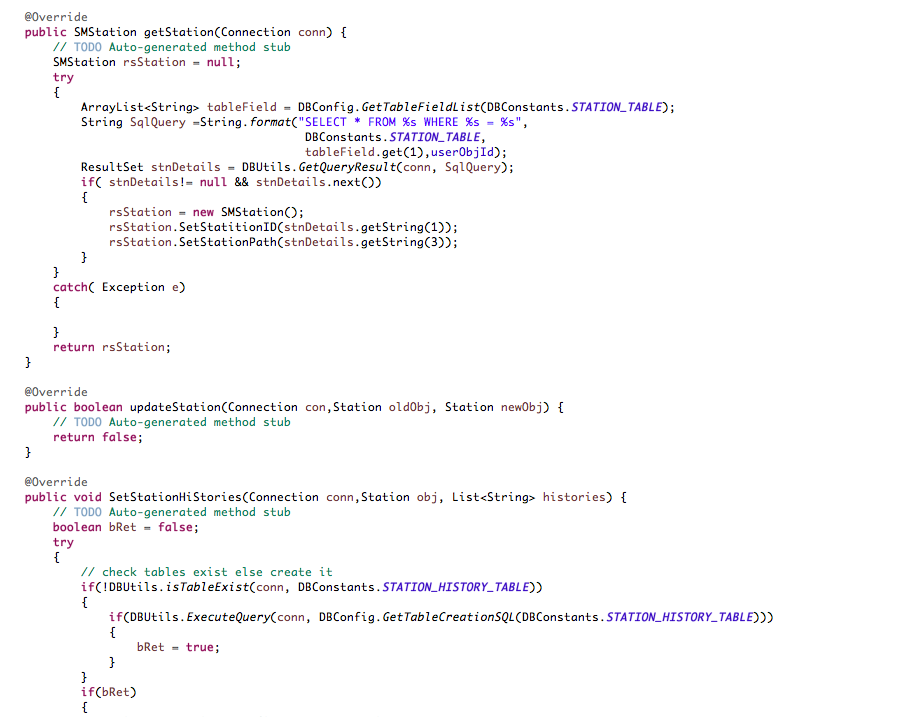
****

* 1. **SMDBEnergyManagerDAO.java**

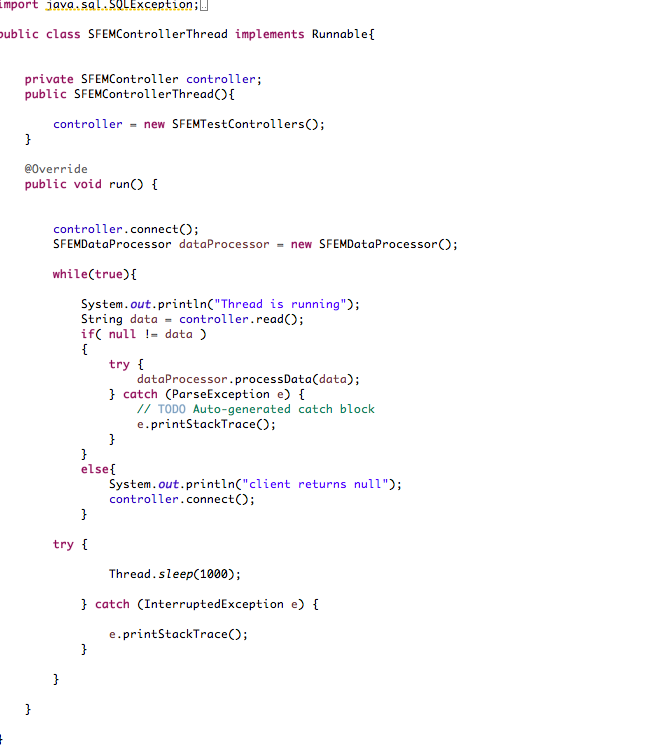
****

* 1. **SMDBStationDAO.java**

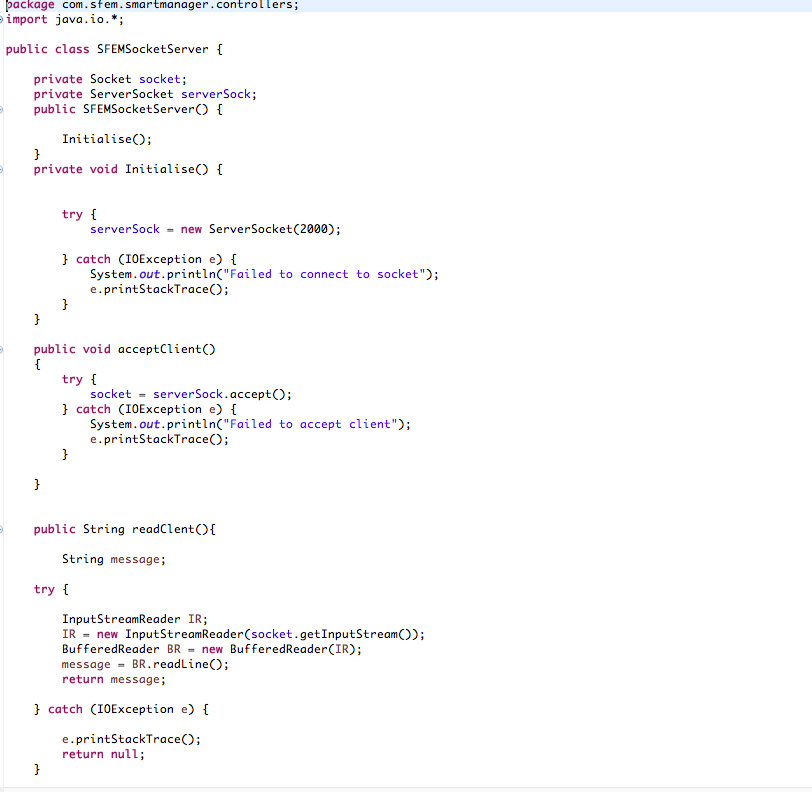
****

****

* 1. **SFEMControllerThread.java**

****

* 1. **SFEMSocketServer.java**

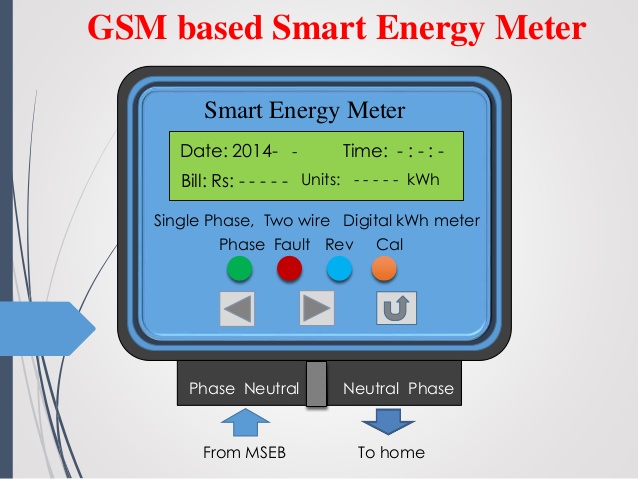


## Hardware

Developing the hardware unit is out of this project but the following components are identified.

### Energy Manager

Smart meters can be used for recording and transmitting the energy consumption details of the appliances to the server.



### Smart Feeder

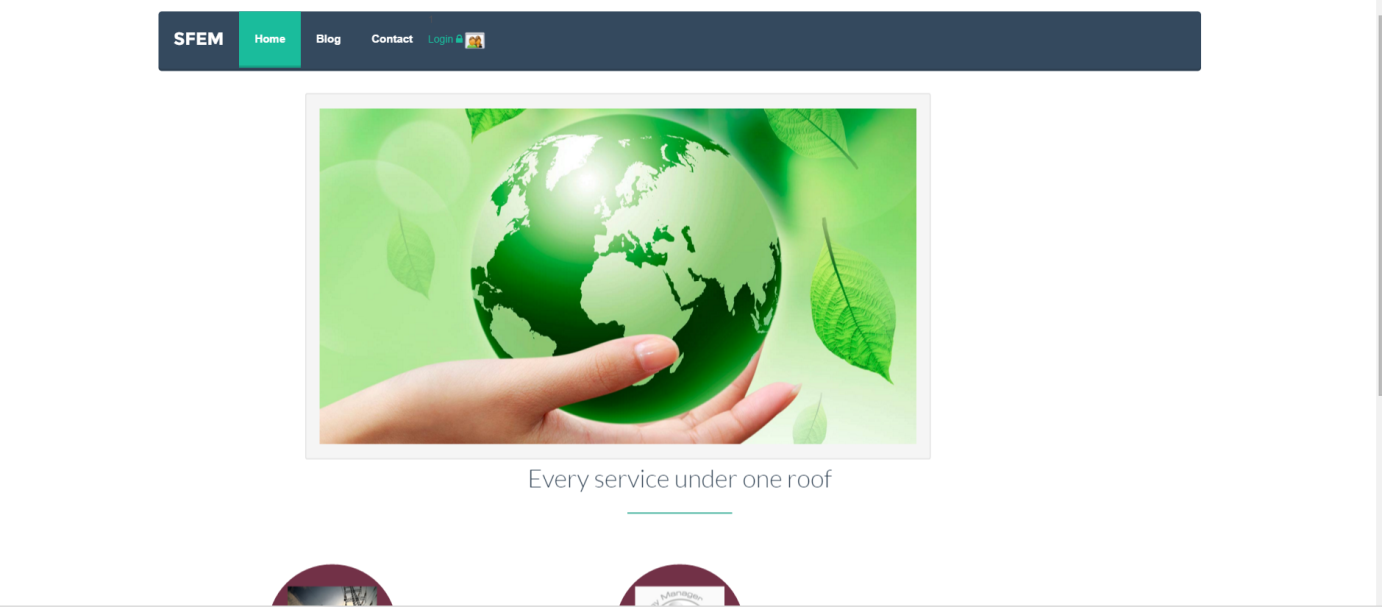
The main components required are electric sensors and relays which can detect the fault in the line and transmit the information to the central unit.

GSM or microwave communication can be used to transmit data from sensors to central unit which would be located in the substation.

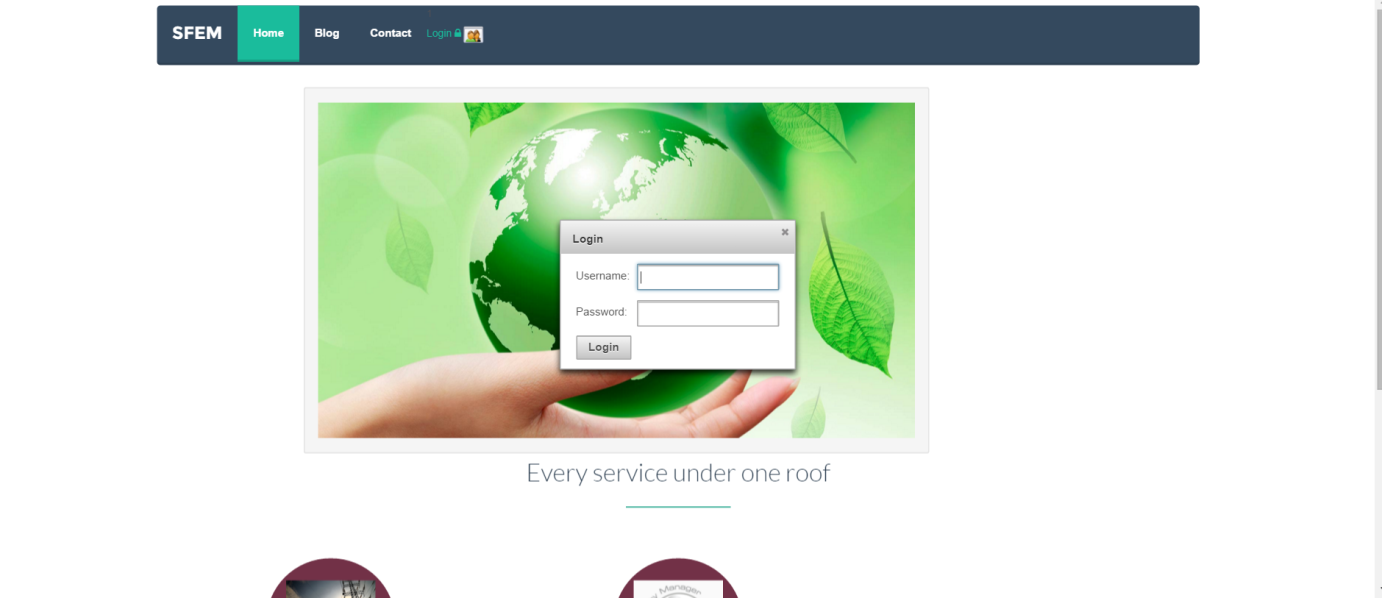


# How SFEM works?

SFEM has a home page which will be the gateway for Smart Feeder and Energy Manager.

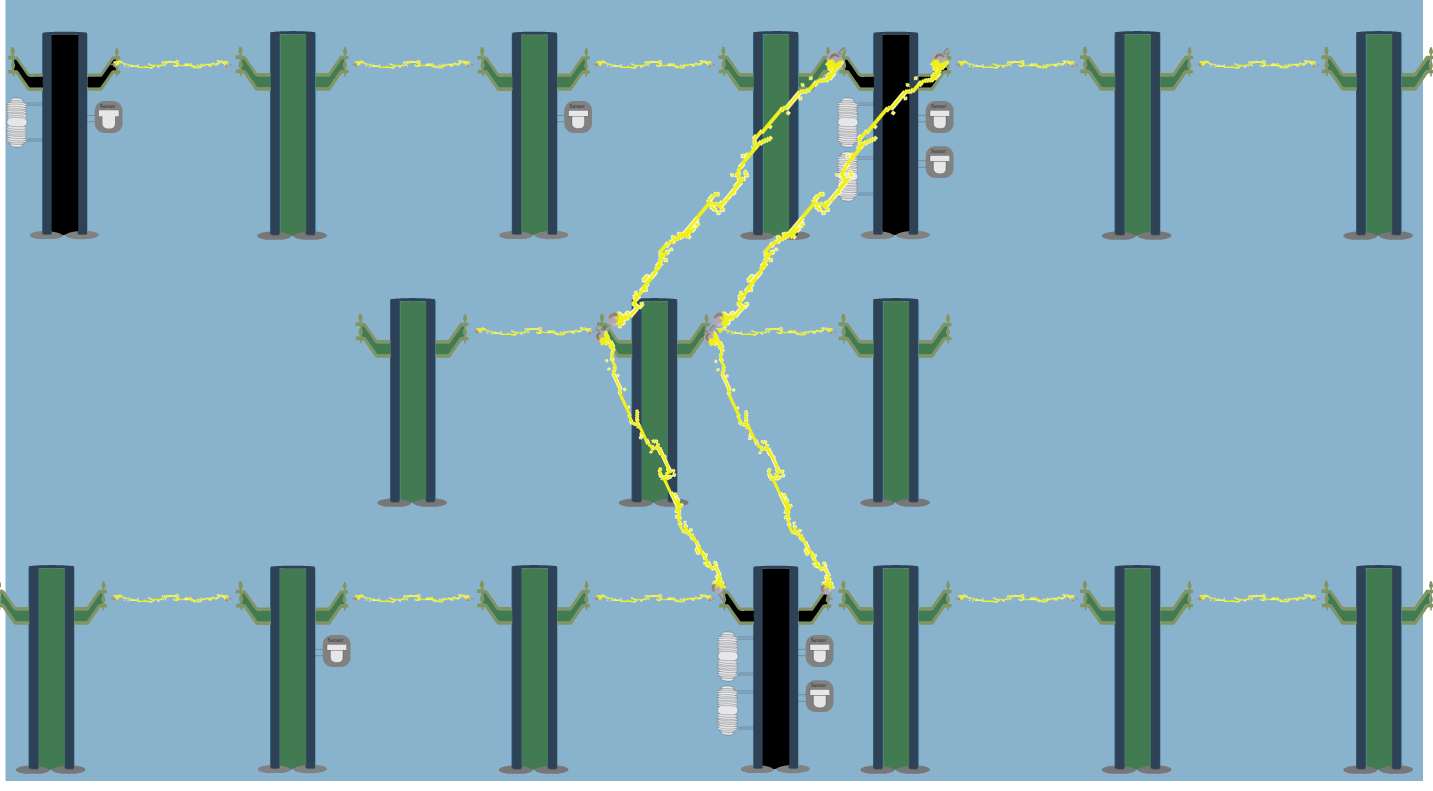


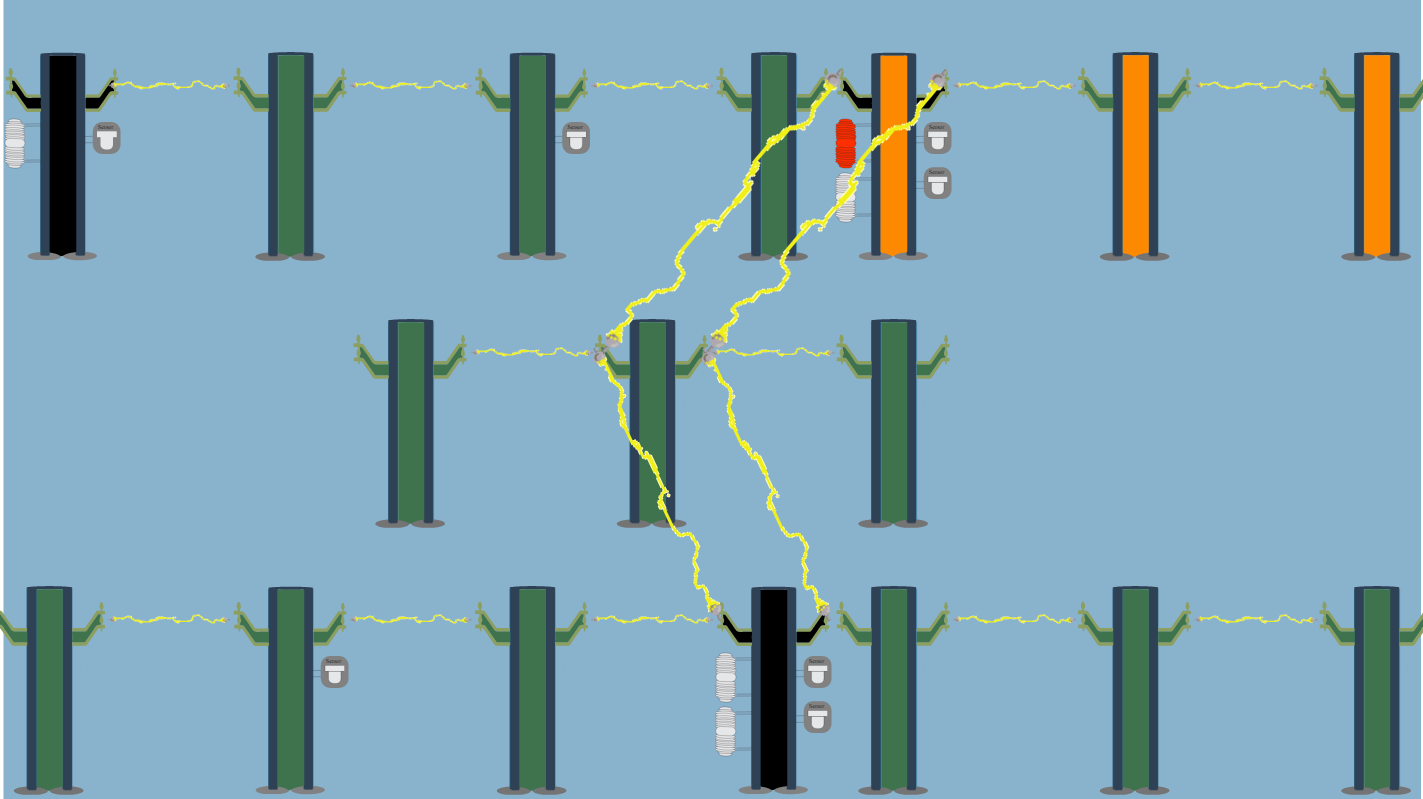
Login to the application using the credentials, and based on the credentials application will redirect either to Smart Feeder or to Energy Manager

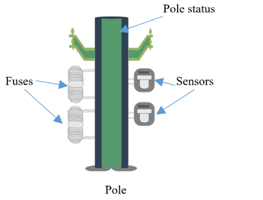


## Smart Feeder

When the feeder is working without any fault the station will indicate everything with Green color and when there is a fault occurs then corresponding fuses, sensors and poles will blink in red color which will help to identify the faulty location.







**Pole Status**

According to value of sensors or fuses with which it is associated, poles will show the status either green or red. Red indicates the conductor in that pole is not carrying the power supply.

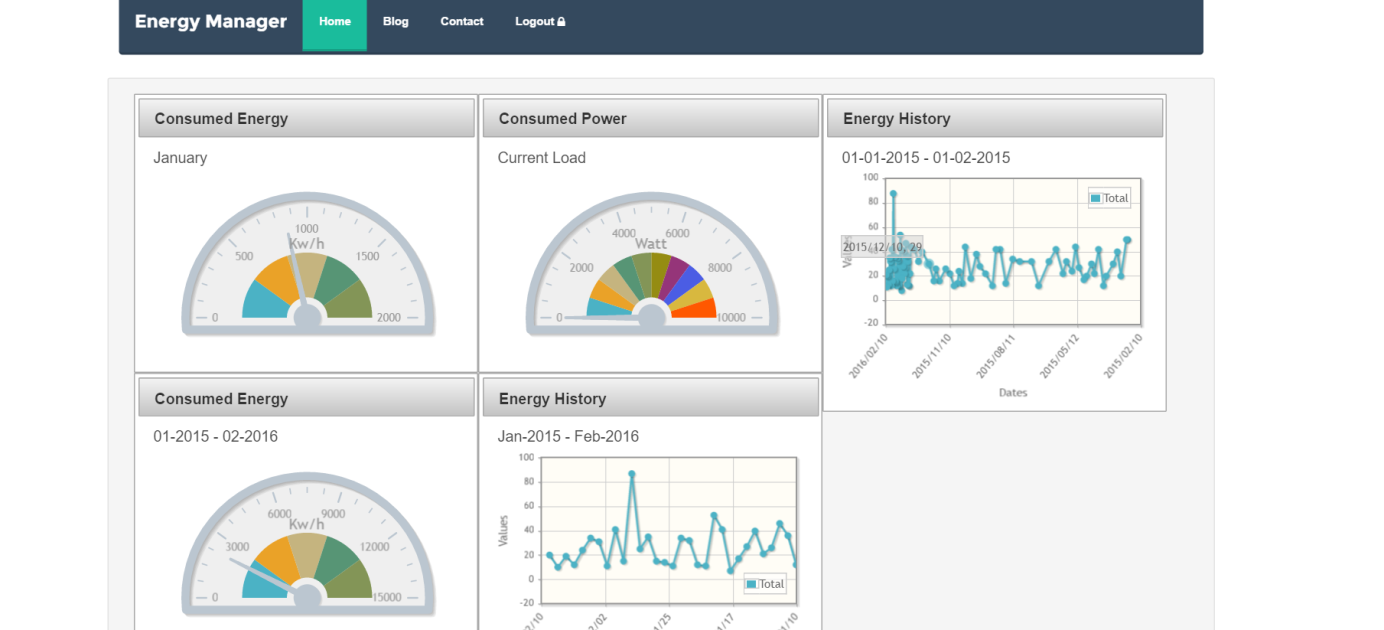
**Sensors**

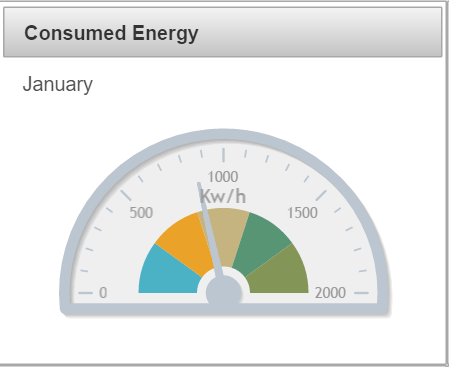
Depending on the real time value of sensors which are located in the selected poles in the feeder system, sensors in the application will show either green or red.

**Fuses**

Depending on the real time value of sensors which are located in the selected poles in the feeder system, sensors in the application will show either green or red.

## Energy Manager





# Learning from Technology

Different design concepts, technologies and tools are considered for the development of this application. By developing this application, each member of the team got opportunity to develop a dynamic web based application from the scratch.

## Tools

Following tools and plug in were used for the development of this application,

* Eclipse – Java development IDE
* Window builder – Test simulator application
* RazorSQL – SQL editor
* Chrome postman – REST Uri testing
* Creately.com – Online tool for drawing UML diagram.
* explorelearning.com – Energy consumption test data generation
* Primefaces dashboard framework

## Technology

Following technologies are used for this application,

* Java
* SQl
* Derby DB
* RESTful API(Jersy)
* XML
* Apache Tomcat
* JQuery
* HTML5
* CSS
* Bootstrap
* SVG

### Server

During the feasibility study we analyzed the possibility of using BIM server which is especially dedicated for building applications. We have checked the IFC data modeling but found that its applications are beyond the scope of our project. We decided to develop our own server and we used Apache Tomcat to host our server. Java is chosen as development language.

After considering different databases like SQLite, MySQL, BerkleyDB, Derby is selected for the database. It comes along with the java development package and it is an open source database.

### Web service

This was an opportunity for us to learn Service Oriented Architecture and we have considered SOAP and RESTful web services and decided to adapt RESTful Services which exactly matches our requirement. We learned to write resource URIs in server and consuming the same from the client. The application uses Java jersy framework which use JAX-RS.

### Client

The client side is designed on ajax model and the application support cross browser compatibility. This project was an opportunity to get hands on experience in web development using HTML5, SVG, and CSS. We also learned to bind the data from server to the browser using JQuery.

In order to develop the dashboard we have considered different open source dashboards like Slate, JQPolt and finally chosen Java Primeface dashboard. It uses AJAX model and JSF which is easy to learn and develop.

### Design Patterns

We have tried to strictly follow the design principles so that the application is open for future expansion. Following design patterns are used.

* Singleton
* Adapter Pattern
* strategy Pattern

# Learning from Project

This project is a learning curve and each team member got the opportunity to involve in all software life cycles. We learned to analyze the requirements, feasibility study, designing the architecture, choose right technology, develop the applications testing and deploying it.

We learned to work as a team regardless of gender and nationality. We learned to divide the task and manage the time. Developing this application in the limited time was a challenge and we were able to achieve it.

# Project Management

We have conducted a startup meeting and discussed the feasibility of the project. We identified team members and based on their technical experiences

We divided the tasks as follows,

**Client side (UI)**

* Favio Tejada

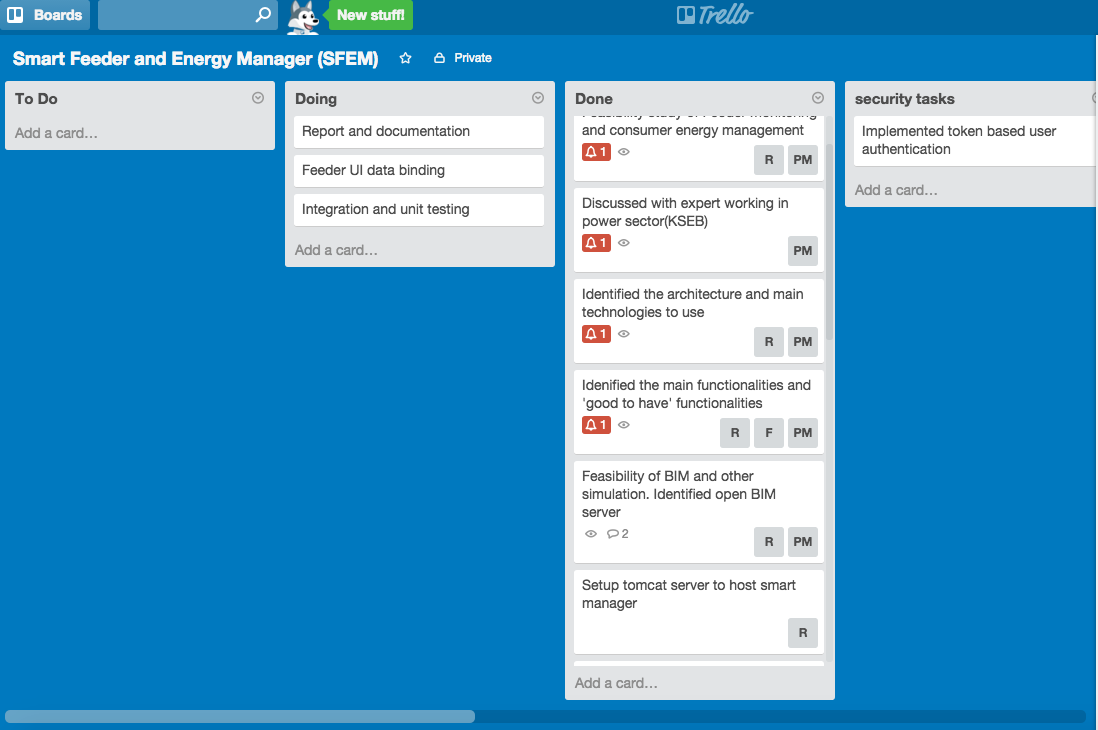
**Server and Test Simulator**

* Prince Mathew
* Rahul Kumar Thai Valappil

**Security**

* Neha Prabhu
* Rucha Prabhu

We have scheduled and tracked the tasks using Trelo and every week we conducted meetings to update and discuss the progress



# Future Possibilities

We have developed a prototype of the application and it has a wide scope for future expansion.

## General

1. The current product targets computer OS but a slight modification in the application will enable the product compatible to mobile and tablet platforms.
2. SFEM is currently developed as a thin client application but it can be modified as a thick client, which helps to implement more security mechanisms.

## Energy Manger

1. Currently dashboard provides a set of predefined widgets, but as part of future expansion user can create customized dashboard.
2. In the existing product there is no provision for the user to communicate with the power utility company. But in the future it can be implemented so that user can register complaints, provides feedbacks and receive alerts from the power utility company in the occasion of scheduled power cut.
3. This application can be enhanced in future in such a way that it can send alert to user based on predefined criteria for example if any appliances are running unnecessarily application give notifications to customer

## Feeder Manager

1. Back office tool-The display for the feeder application is currently static but it has to be changed according to the specific feeder layout. This can be achieved through developing a separate feeder layout builder application.
2. In the current application there is no provision to acknowledge the alarms, see the history of the alarms. This can be implemented in the future.

# Bibliography

# <https://www.youtube.com/watch?v=xkKcdK1u95s&list=PLqq-6Pq4lTTZh5U8RbdXq0WaYvZBz2rbn>

<http://www.codingpedia.org/ama/tutorial-rest-api-design-and-implementation-in-java-with-jersey-and-spring/>

<http://www.tutorialspoint.com/restful/restful_first_application.htm>

<http://www.w3schools.com/sql/sql_autoincrement.asp>

<http://simple.sourceforge.net/>

# <http://docs.oracle.com/javase/tutorial>

<http://stackoverflow.com/questions/26777083/best-practice-for-rest-token-based-authentication-with-jax-rs-and-jersey>