****

**Project 2.14**

**Electric Power Consume Management**

Student ID: GT60708

Full name: Nguyen Tien Khoa

**Introduction**

Vietnamese Electric Power Company has some problem to collect data from consume monitor device. So, they need a program to collect data information from those device quicker and cheaper. The program should support to get data about electric consume on monitor device though internet. Moreover, It can help customers to be aware about electric bill. Therefore, company can track consume electric of their customer then give good advices.

Because of using internet to collect data, the system must provide a sever. And all device must to have internet connection. But, device works 24/7 to monitor electric consuming, internet connection would be interrupted and device would be down sometimes by some reason. So, server should have a strategy to deal with offline device.

Surely, monitor devices are not perfect. They may be offline, conflict data, dirty data, etc. So, server should be save these problems to write report. Report will be read by administrators, the electric engineers. Then, they can track to solve problems faster and better. Therefore, the new system will satisfy customer and improve productivity and quality.

In short, the new system manage monitor device to collect data from customer then print it to report. It helps Vietnamese electric company manage and support electric consuming customers.

**Open point**

Customers regard to their money, product and service. Example, they need to know when electric bill become expensive, why electric lost, etc.

Customers can use the new system to interact and discuss about problems and needs.

**Analysis**

Base on requirement of this project and my experiences in Java, Java programming language is used to develop. Focus on available tools, low price and fast development, this project includes 3 modules to contain web service, device simulator and share project.

Web service, this is the main project.

Device simulator, this is project to simulate real device. Actually, I don't have any real device, this simulator would support to make system be visible.

Share project, this contain model classes, constant is used by both service and device.

**Requirement analysis**

Program must use java open source technique for keeping low price.

Program must delivery within 3 months with main functions.

Program must use an incremental development model for integrating other requirements.

Program must allow device to register into system automatically.

Program must be limited access, Administrator only on this version.

Program must allow admin to modify customer data.

Program must have a report reviewer, such as dashboard.

Program must have calculator to built electric bill on demand.

Program should validate device information to track problems on devices.

Program would allow admin to modify consume policy data.

Program would allow administrator edit some properties on device.

Program should allow to register new user account.

Program should update device on database when device reconnect.

Program should poll device every 10 minutes, in this version.

Device Simulator should able to simulate a device error.

Device Simulator should able to be polled by server.

Device Simulator should able to send data to server.

Device Simulator should able to store data in internal memory, such as files.

**Technology analysis**

**Srping Framework:** System must have server to provide service. With the main target is quicker and cheaper, using open source library to build the service. Spring framework seem to be the good choice with wide technique support and easy to use.

**Runable JAR:** In this article, monitor device is not developed and built so that a virtual device run on JVM is using to simulate real device. I considered to use Spring Shell, but It is a bit heavy with this requirement and I have no experience. So Spring Shell will be expensive.

**Java Message Service (JMS) and Active MQ:** Deep knowledge about Network is not require. For connection between device and server, JMS is good technique. ActiveMQ service is build to store, forward and broadcast to device and receive message from devices. Device and server just connect to ActiveMQ for communication through IP Address, port and protocol of ActiveMQ server broker.

**JavaScript Object Notation (JSON):** There is no important personal information is used. So, JSON is consider to used to convert Java Object to Text for transfer in communication. The reason are easy to use, easy to read, fast speed, light weight. Jackson JSON is used instead of Google GSON because of speed and feature powerful.

**Java Sevlet Facet (JSF) and Primeface:** It is famous of powerful and less javascript technology. For user interface, javascript is difficult to Java developer like me. Primeface framework is supported to integrate with Spring framework by Srping webflow library. So, with a bit javacript knowledge and time to research spring webflow and Primeface, JSF is considered to be best choice to build UI.

**MySQL:** System have Database server to store device information and customer information. MySQL is used to build a databse server. In future, with a huge database, the service should separate into 2 server, one other will use No-SQL data like mongoDB for lighter and faster query device information. Because the security and restrains on device information is not import.

**Hibernate:** It is a famous framework support Java Persistence Access(JPA). There is no reason to avoid Hibernate, this easy to use and powerful. But, I consider to remove it when separate data into SQL Database and No SQL database. In this case, Spring Database is good choice to support JPA because It supported both JPA and MongoDB

**Spring webflow and JSF:** As I mentioned in JSF, Spring webflow is the solution to integrate JSF with Spring web mvc. Spring web mvc and JSF are 2 difference framework use Model View Controller (MVC) Architecture, so that they need a midleware to work together. Moreover, Spring webflow also support to control life cycle and navigation such as define variable and view, follow step and check state. In this version, I just use Spring webflow to integrate Spring web mvc and jsf, especially Primeface.

**Spring Security:** Spring security support protect server from unexpected access. Encryption is support by Spring security by many algorithm such as MD5 on 32 characters and Bcrypt on 64 characters. Of course, It supports access with JPA database.

**Design**

**Implement**

**Maintenance**