**content**

**[1.0 Code/file version management](#_Toc2019035062_WPSOffice_Level1)** **[1](#_Toc2019035062_WPSOffice_Level1)**

**[2.0 Configuration Management](#_Toc437273400_WPSOffice_Level1)** **[1](#_Toc437273400_WPSOffice_Level1)**

[2.1 Nightly Build](#_Toc437273400_WPSOffice_Level2) [1](#_Toc437273400_WPSOffice_Level2)

[2.2 Mozilla Releases Firefox](#_Toc564993766_WPSOffice_Level2) [2](#_Toc564993766_WPSOffice_Level2)

[2.3 Advantages and Disadvantages of this system for the client](#_Toc1825021775_WPSOffice_Level2) [2](#_Toc1825021775_WPSOffice_Level2)

**[3.0 Request for Proposal (RFP)](#_Toc564993766_WPSOffice_Level1)** **[3](#_Toc564993766_WPSOffice_Level1)**

[3.1 Quick Fast](#_Toc632042324_WPSOffice_Level2) [3](#_Toc632042324_WPSOffice_Level2)

[3.2 Background on ABB](#_Toc1281221406_WPSOffice_Level2) [4](#_Toc1281221406_WPSOffice_Level2)

[3.3 Contact Information](#_Toc669642173_WPSOffice_Level2) [4](#_Toc669642173_WPSOffice_Level2)

[3.4 System description](#_Toc1861691331_WPSOffice_Level2) [4](#_Toc1861691331_WPSOffice_Level2)

[3.5 Basic summary of the project, including overall goals and timetable](#_Toc609463327_WPSOffice_Level2) [4](#_Toc609463327_WPSOffice_Level2)

## 1.0 Code/file version management

GitHub account name: UltraYang

## 2.0 Configuration Management

### 2.1 Nightly Build

Building a new version every night means that there are new problems every day. This will cause old problems to be unfinished and new problems to arise. The old problems still exist, so I don’t think it is necessary to build every day, and it can be extended by a week. Build once so that engineers have more time to polish and fix issues, and older issues don't get committed every day.

### 2.2 Mozilla Releases Firefox

Firefox is an open source browser, so you need to choose a suitable open source protocol, then set the version number to be released, and then go through the code, version approval, functional testing, writing update documentation, and then package and release the software to the Online, and provide feedback channel entrance

### 2.3 Advantages and Disadvantages of this system for the client

advantage

1) Free: Who can refuse something that doesn't cost money, besides, many free open source frameworks are good enough;

2) Transparency: All source codes of open source frameworks are public and can be seen by anyone;

3) Can be changed: Most open source projects are MIT or BSD open source copyrights with a high degree of freedom, and can be customized and developed on demand;

4) Collaborative: Github is the largest open source project platform, and developers around the world can participate in iterative open source projects;

5) Influence: Excellent open source projects can enhance the popularity and influence of authors or contributors in the industry.

shortcoming

1) Hidden security risks: Although many excellent open source projects are developed and maintained by enterprises or senior experts, but because they are not completely used by themselves, contributors are prone to negligence in security. There are countless examples of security vulnerabilities in well-known open source projects. For example, OpenSSL Heartbleed, Fastjson remote code vulnerability, Antd Christmas eggs, etc.;

2) Good and bad: Open source project developers, contributors and maintainers can be anyone, and their respective experiences and professional backgrounds are different, so it will inevitably lead to certain differences in the quality of code or open source projects; although the code specification (Coding Standard) can Avoid some problems, but excellent projects are few after all, look at NPM or Maven public repositories that host millions of projects;

3) Learning cost: The author admits that some excellent open source frameworks have mature and complete documentation systems, but most of them still lack effective documentation and tutorial support; even with detailed documentation, developers will spend a lot of time to read and learn ; Most paid products include professional technical support, which can effectively help developers save time;

## 3.0 Request for Proposal (RFP)

### 3.1 Quick Fast

Integrated system design for ABB.Add functions in and system, including management services, user management, statistical reports, sales management, quality supervision, production optimization(Esti Dwi Rinawiyanti, Sharif As Saber, C 2020).

### 3.2 Background on ABB

Aussie Business Buzz (ABB) is a Business that sells a wide range of technology products such as PCS, laptops, phones, routers, equipment repair and accessories for mobile devices.

### 3.3 Contact Information

Email Sida at:y.cui.21@student.scu.edu.au

### 3.4 System description

The backend adopts SpringCloud, and the streaming application starter is a spring integrated application based on Spring Boot, which provides integration with external systems. Spring Cloud Task, a short-lived microservices framework for rapidly building applications that perform limited data processing.

The granularity of service splitting is finer, which is conducive to the reuse of resources and the improvement of development efficiency. It is possible to formulate optimized service plans more accurately and improve the maintainability of the system. The microservice architecture adopts the decentralization idea, and Restful is used between services. Lightweight communication, lighter than ESB, suitable for the Internet era, and shorter product iteration cycle.

The front end uses Vue, which is a progressive framework for building user interfaces. Unlike other heavyweight frameworks, Vue adopts a bottom-up incremental development design. Vue's core library only focuses on the view layer, and is very easy to learn and integrate with other libraries or existing projects. On the other hand, Vue is fully capable of driving complex single-page applications developed with single-file components and libraries supported by the Vue ecosystem. Data-driven + componentized front-end development.

### 3.5 Basic summary of the project, including overall goals and timetable

The system is divided into functional modules, customer management, branch management, commodity management, inventory management, sales management, financial management, authority management and other modules. Customer management is mainly to manage the customer information of each branch, including the customer's name. , mobile phone number, address, email and other information; branch management is mainly to manage how many branches there are, the person in charge of each branch, telephone, address, staff, etc.; commodity management is mainly to manage the products sold, inventory management is the main management The inventory quantity of each commodity and each sku can be connected to the warehousing system; sales management mainly manages the sales, sales details, profits and other information of each store; financial management mainly manages the cost of commodities and the expenditure income of the store Information such as running water, profit status, employee salary status, etc.; authority management mainly allocates the functions of each role in the system, after all, the roles are unauthorized to operate.