**Stock Market Charting Design Document**

**Version <1.0>**

**Jack.Liu**

**2020-04-01**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2020/04/01 | <1.0> | Initial Version of Document | Jack.Liu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 4](#_Toc37338705)

[**1.1** **Purpose** 4](#_Toc37338706)

[2 Use Cases 4](#_Toc37338707)

[**2.1** **Admin Use Cases** 4](#_Toc37338708)

[**2.2** **User Use Cases** 5](#_Toc37338709)

[3 Micro-Services Architecture 6](#_Toc37338710)

[3.1 **Diagram** 6](#_Toc37338711)

[**3.2** **Micro-Services** 6](#_Toc37338712)

[4 Data Model 7](#_Toc37338713)

[**4.1** **User Microservice** 7](#_Toc37338714)

[**4.2** **Company Microservice** 8](#_Toc37338715)

[**4.3** **Stock Price Microservice** 9](#_Toc37338716)

[**4.4** **Exchange Microservice** 10](#_Toc37338717)

[**4.5** **Sector Microservice** 11](#_Toc37338718)

[**4.6** **E-R Model** 12](#_Toc37338719)

[5 Project Management 12](#_Toc37338720)

[**5.1** **Maven project** 12](#_Toc37338721)

[**5.2** **Modules** 13](#_Toc37338722)

[**5.3** **Project Repository (Mono Repo)** 13](#_Toc37338723)

[**5.4** **Angular application components** 14](#_Toc37338724)

# Introduction

This Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

## **Purpose**

This Software System lets Admin to upload Stock Price of a Company(which is listed in a Stock Exchange) at different points of time. It needs to support multiple Stock Exchanges. And the registered Users should be able to generate various charts to perform Stock Market performance of various Companies or Sectors over certain period.

# Use Cases

## **Admin Use Cases**

* Login:
  + - * + Login with username.
        + Validate the password.
* Logout:
  + - * + Logout from the system.
        + Clean the cache.
* Manage Stock Exchanges:
  + - * + List all the Stock Exchanges.
        + Edit current Stock Exchanges.
        + Add the new Stock Exchanges.
        + Delete the current Stock Exchanges.
* Manage Company
  + - * + List all the companies.
        + Edit current companies.
        + Add the new companies.
        + Delete the current companies.
* Import Data:
  + - * + Import excel format data.
        + Data should be stock price of a company at various points of time.
        + Display appropriate error message when import data failure.
        + While uploading Excel, specify the Stock Exchange to which the uploaded data belong to.
        + Validate values of company code, date ranges.
        + Data will be stored in a database.
        + Display a summary of data after successfully imported.

## **User Use Cases**

* Signup:
  + - * + Username, Password and E-mail address are required.
        + Sent an E-mail after user signup.
        + A confirmation link will be included in the E-mail.
        + Click confirmation link to active the account.
        + User data will be stored in user DB.
* Login:
  + - * + Login will E-mail address or Username.
        + Validate E-mail address is correct format.
        + Validate password.
* Logout:
  + - * + Logout from the system.
        + Clean the cache.
* Edit user account:
  + - * + Update he profile.
        + Reset the password.
        + Sent an confirm E-mail after password changed.
* Company information:
  + - * + List all the companies.
        + Fuzzy search a company by company code or name.
        + Display the company details information.
        + A chart will display by certain period.
        + Display the data by bar chart.
        + Display Average, Min, Max, Growth for that specific period.
* IPOs information:
  + - * + Display the IPOs information.
* Comparison charts:
  + - * + Single company over different periods of time.
        + Two different companies over a specific period.
        + A single sector over different periods of time.
        + Different sectors over a specific period.
        + Between a Sector and a company over a specific period.
* Export data:
  + - * + Export data and download in Excel

# Micro-Services Architecture

## **Diagram**

A screenshot of a cell phone

Description automatically generated

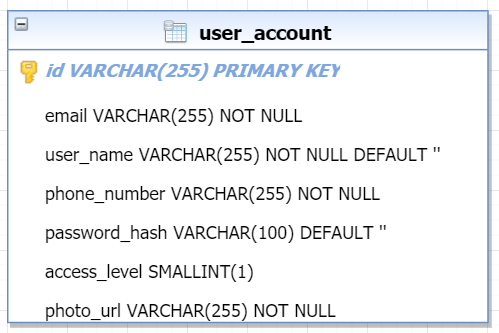
## **Micro-Services**

* The user or admin browses the stock web by link
* The API gateway receives the request from the user
* The API gateway forwards request to the backend for frontend (BFF), BFF processes the request and sends a response, rendering the stock Angular JS application (Stock SPA).
* The requests to browse the stock application is send to the web BFF, which makes an HTTP request to the backend microservices using private network.
* Stock price microservice expose the REST API interface to receive the excel file and process the validate data into database and get the stock price for company in certain period.
* Company microservice expose the REST API interfaces to get company list, company details, company stock price and company IPO details.
* Exchange microservice expose the REST API interfaces to get stock exchange list, add stock exchange
* Sector microservice expose the REST API interfaces to get company list in a sector and get sector price.
* User microservice expose the REST API interfaces to user login, admin login and user sign up.
* E-mail sender microservice expose the REST API interfaces to send confirmation mail when user sign up and reset password.
* Each Microservice is a Spring Boot Rest application by specifying required spring boot starter packages in pom.xml os by using Spring Initializr.
* REST Controllers, with the appropriate REST End points to perform corresponding CRUD operations. Along with End Points which are exposed to Angular Client.
* Each Microservice is a self-sufficient and standalone application, and owns data stored in specific database.
* Microservice interaction with corresponding DB it owns.
* Microservice need to interact with other Microservice by using FeignClient.
* Repository class which implements JPA.

# Data Model

## **User Microservice**

* User Account Data Model

****

* User Account APIs

|  |  |  |
| --- | --- | --- |
| **API Name** | **HTTP Method** | **Function** |
| createUserAccount | POST | Create a user account |
| getOrCreateUserAccount | POST | Get a user account or create user account if not exist |
| getUserAccountById | GET | Get a user account entity by user ID |
| getUserAccountByPhoneNum | GET | Get a user account entity by phone number |
| listAllUserAccount | GET | Get current user account list |
| updatePassword | PUT | Update user password for current account |
| verifyPassword | POST | Verify user password for current account |
| requestPasswordRest | POST | Sent a request for password reset |
| updateUserAccount | PUT | Update user account |

## **Company Microservice**

* Company Data Model

**A screenshot of a cell phone

Description automatically generated**

* IPO Details Data model

**A screenshot of a cell phone

Description automatically generated**

* Company APIs

|  |  |  |
| --- | --- | --- |
| **API Name** | **HTTP Method** | **Function** |
| getCompanyStockPrice | GET | Get company's stock price in period |
| listAllCompanies | GET | Get the list of all the current companies |
| getCompanyByCode | GET | Get a company entity by company code |
| getMatchingCompanies | GET | Get the list of companies by matching criteria |
| getCompaniesBySectorId | GET | Get the list of companies by sector ID |
| getCompaniesByExchangeId | GET | Get the list of companies by stock exchange ID |
| getCompanyIPODetails | GET | Get IPO Details of Company |
| updateCompany | PUT | Update company for current account |
| deleteCompany | POST | Delete current company |
| createCompany | POST | create a new company |

## **Stock Price Microservice**

* Stock Price data model

**A screenshot of a cell phone

Description automatically generated**

* Stock Price APIs

|  |  |  |
| --- | --- | --- |
| **API Name** | **HTTP Method** | **Function** |
| getStockPriceByTime | GET | Get stock price of company in certain time |
| listTwoStockPriceByTime | GET | Get the two different companies’ price in certain time |
| getStockPriceByCompany | GET | Get stock price in period by company |
| listTwoStockPriceByCompany | GET | Get stock price in two period by company |
| importExcelData | POST | import excel file of stock price data into DB |

## **Exchange Microservice**

* Stock Exchange Data Model

**A screenshot of a cell phone

Description automatically generated**

* Exchange APIs

|  |  |  |
| --- | --- | --- |
| **API Name** | **HTTP Method** | **Function** |
| getStockExchangesList | GET | Get the list of all of current stock exchanges |
| getStockExchangeByCompany | GET | Get the stock exchange by company name |
| getStockExchangeById | GET | Get the stock exchange by ID |
| createStockExchange | POST | Create a new stock exchange |
| updateStockExchange | PUT | update the current stock exchange |

## **Sector Microservice**

* Sector Data Model

**A screenshot of a cell phone

Description automatically generated**

* Sector APIs

|  |  |  |
| --- | --- | --- |
| **API Name** | **HTTP Method** | **Function** |
| getSectorList | GET | Get the list of all of current sectors |
| getSectorByCompany | GET | Get the sector by company name |
| getSectorById | GET | Get the stock exchange by ID |
| createSectorExchange | POST | Create a new sector |
| updateSectorExchange | PUT | update the current sector |

## **E-R Model**

**A close up of a map

Description automatically generated**

# Project Management

## **Maven project**

**A close up of a sign

Description automatically generated**

## **Modules**

**A screenshot of text

Description automatically generated**

## **Project Repository (Mono Repo)**

**A screenshot of a computer

Description automatically generated**

## **Angular application components**

**A screenshot of a cell phone

Description automatically generated**