|  |
| --- |
|  |

|  |
| --- |
| **Stock Market Charting (FSD) v3.0** |
| Case Study |
|  |
| This document covers Software Requirements of Stock Market Charting, along with list of Technologies to be used to develop this Software System, and also includes some details on the Architecture |
|  |
|  |
| **10/22/2019** |
|  |

Table of Contents

[1. Business Requirement (Stock Market Charting) 2](#_Toc22634436)

[1.1. Admin Use Cases: 2](#_Toc22634437)

[1.2. User Use Cases: 3](#_Toc22634438)

[1.3. Company related Data Fields: 4](#_Toc22634439)

[1.4. Stock Price details Excel: 4](#_Toc22634440)

[1.5. IPOs planned: 4](#_Toc22634441)

[1.6. Sectors data Fields: 4](#_Toc22634442)

[1.7. User DB Table: 4](#_Toc22634443)

[1.8. Stock Exchange Data Fields: 5](#_Toc22634444)

[2. Design Inputs 5](#_Toc22634445)

[3. UI/UX Wireframes 5](#_Toc22634446)

[4. Entity Classes – Mid Tier 9](#_Toc22634447)

[5. Model Classes 11](#_Toc22634448)

[6. Architecture Diagram 11](#_Toc22634449)

[7. Development of individual Microservices 13](#_Toc22634450)

[8. Database Tables 14](#_Toc22634451)

[9. Microservices Integration and Security 14](#_Toc22634452)

[10. Spring Microservices Tools to be used 15](#_Toc22634453)

[11. JWT Authentication 16](#_Toc22634454)

[12. Architecture/Design 16](#_Toc22634455)

[13. DevOps Activity 16](#_Toc22634456)

[14. Diagram 17](#_Toc22634457)

[15. Jenkins CI/CD 17](#_Toc22634458)

[16. Configure Jenkins and Docker for the Project 18](#_Toc22634459)

[17. Perform CI/CD 18](#_Toc22634460)

[18. Diagram 19](#_Toc22634461)

[19. Full Stack Technologies 19](#_Toc22634462)

[20. Technical Spec – Solution Development Environment 19](#_Toc22634463)

[20.1. Front End Layer 19](#_Toc22634464)

[20.2. Middle Tier Layer 20](#_Toc22634465)

[20.3. Database & Integration Layer 20](#_Toc22634466)

[20.4. Ancillary Layer 20](#_Toc22634467)

[20.5. Security 20](#_Toc22634468)

[20.6. Deployment & Infrastructure 20](#_Toc22634469)

[20.7. Editors 20](#_Toc22634470)

[21. Assessment Deliverables 21](#_Toc22634471)

[22. Important Instructions 21](#_Toc22634472)

# Business Requirement (Stock Market Charting)

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 need 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 of time. More details on the features which need to be supported are specified, below.

This Case study supports below two different Roles:

1. Admin
2. User

## Admin Use Cases:

Admin can perform below operations. It is mandatory to implement all the requirements, except the ones mentioned as optional.

1. **Login/Logout:** Login and Logout. To avoid Complexity, there can be a predefined username and password for Admin
2. **Manage Stock Exchanges:** This feature lists all the Stock Exchanges currently supported. BSE, NSE Stock Exchanges need to be available by default. It should be possible to add new Stock Exchanges. Deletion of Stock Exchange need not be supported.
3. **Manage Company:**
   1. Add a new company details with fields or edit an already existing Company details
   2. Deactivate an already existing company
   3. Update any IPO related data
4. **Import Data(Excel Format):**
   1. Data can be imported(in Excel format), it's basically to feed stock price of a company at various points of time.
   2. Uploaded Excel need to be in a specific format, if not error message need to be displayed. While uploading Excel, specify the Stock Exchange to which the uploaded data belong to. If such company exists.
   3. The company code, date ranges need to be appropriately checked, if any data will be over written.
   4. After successfully imported, data need to get stored in a database and Uploaded Summary need to be displayed like which company, Stock Exchange, how many records imported, from and to date range, etc…
5. **Missing Data:** Able to check the dates for which Stock Price of a Company is not available

## User Use Cases:

A User can perform below operations. It is mandatory to implement all the requirements, except the ones mentioned as optional.

1. **User Signup/Login/Logout:** An User can
   1. Signup for a new Account. When signed up, an email needs to be sent to the User with confirmation link.
   2. Login to an existing and Email confirmed account. User should be able to Login only after E-Mail Confirmation is done.
   3. Logout from an account.
2. User can update profile, password of an already existing Account
3. Able to search for a company to display Company profile & Turn over, CEO, board members, Industry, sector, brief write up, current/latest Stock market price
4. Whenever user requests charts/data for certain period, the period need to be divided into appropriate intervals(Week or Month or Quarter or Year), to display the chart
5. View IPOs planned in a Chronological order
6. When user types in 2 or more characters for a company name or company code, it should display matching company names(using ajax), so that user can select one of them, if required
7. Comparison Charts. It should be possible to perform below comparisons of
   1. a single company over different periods of time
   2. different companies over a specific period
   3. a single sector over different periods of time
   4. different sectors over a specific period
   5. between a Sector and a company over a specific period of time
8. Should be possible to select if comparisons need to be displayed in a single chart
9. Use different colors when multiple Companies/sectors are displayed in a single chart and display legend
10. Should be possible to select Chart type(line chart, bar chart, pie chart, etc…)
11. For a displayed chart, it should be possible to export data and download in Excel
12. Whenever a chart is displayed, display Average, Min, Max, Growth for that specific period
13. Possibility to perform multiple comparisons between Company’s or Sectors, over a period of time.
14. When data does not exist for certain period in between, that need to be appropriately indicated in the chart
15. View future Trend /Basic Prediction for a Company or Sector - Optional

## Company related Data Fields:

1. Company Name
2. Turnover
3. CEO
4. Board of Directors
5. Listed in Stock Exchanges
6. Sector
7. Brief writeup, about companies Services/Product, etc…
8. Stock code in each Stock Exchange

## Stock Price details Excel:

1. Company Code – to which Company this Stock Price Info belongs to
2. Stock Exchange – the Stock Price of the Company in this Stock Exchange
3. Current Price – Stock Price
4. Date – Date of the Stock Price
5. Time – Stock Price at this Specific time

## IPOs planned:

1. id
2. Company Name
3. Stock Exchange
4. Price per share
5. Total number of Shares
6. Open Date Time
7. Remarks

## Sectors data Fields:

1. Id
2. Sector Name
3. Brief

You may consider 3 or 4 sample sectors, as a sample data. For example Finance, Healthcare Services, Pharmaceuticals, Hotels, Internet Software & Services

## User DB Table:

1. Id
2. Username
3. Password
4. UserType (if Admin or normal User)
5. E-mail
6. Mobile number
7. Confirmed

## Stock Exchange Data Fields:

1. Id
2. Stock Exchange
3. Brief
4. Contact Address
5. Remarks

# Design Inputs

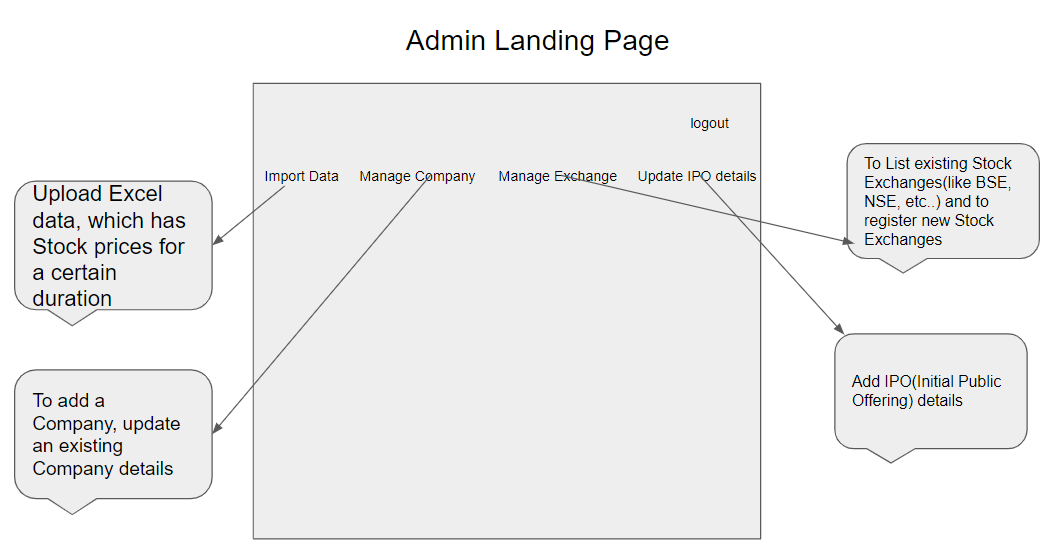
Next sections in this document provides inputs on designing the solution for above requirements.

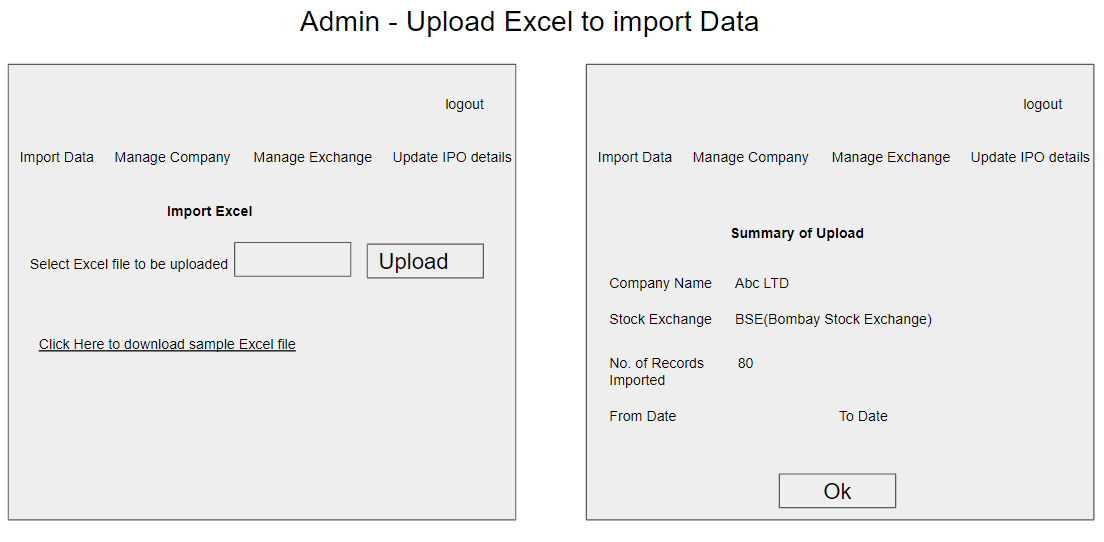
Design inputs provided in this document are just for your reference purpose, Associates can make changes or additions to the Design, based on their analysis.

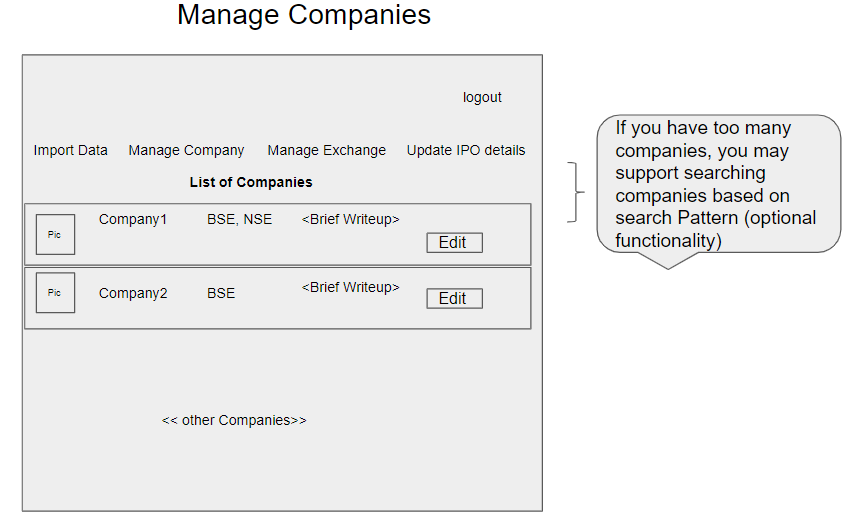
# UI/UX Wireframes

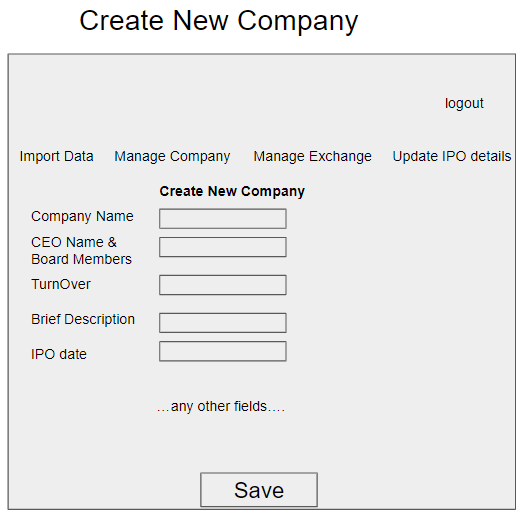
In this Phase you will develop Angular Client Application, with responsive UI using HTML5, HTML5 API, CSS3, Bootstrap/Material and Angular.

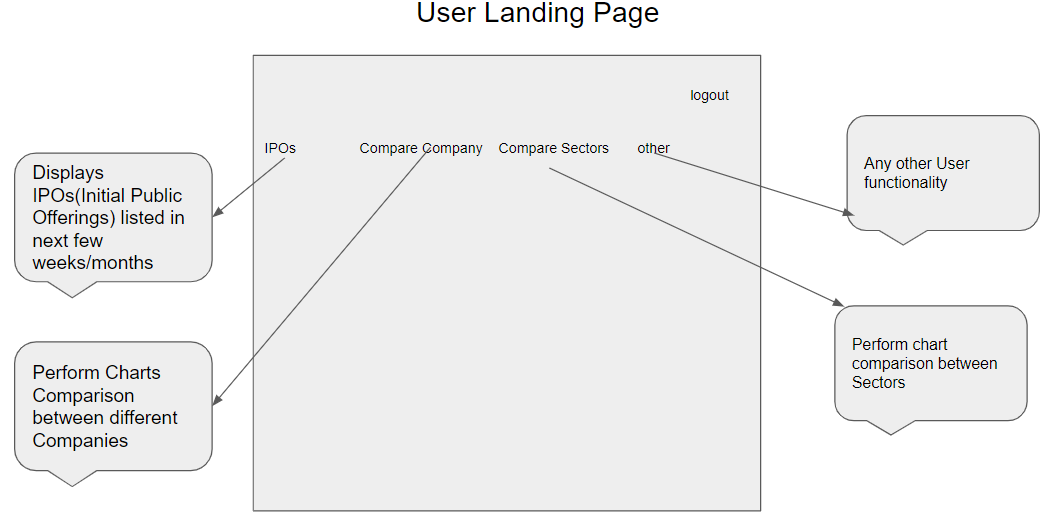
Below are some sample Wireframes

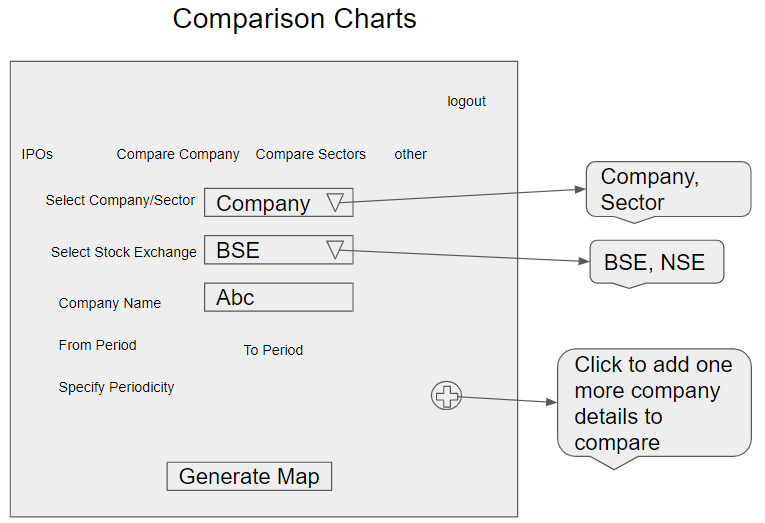












Below are some of the wireframes

1. Login/Signup for User/Admin
2. Display Comparison Charts
3. Admin – List existing Stock Exchanges
4. Admin – Add Stock Exchange
5. Admin – List/Add IPO Details
6. User – View IPO Details
7. User – Update Profile, Update Password

This Phase includes development of

1. Angular components with Templates
2. Angular Routing
3. Angular Services, to invoke REST End points.

Based on the requirements, Front End need to be divided into multiple components to accommodate above Wireframes. Angular Routing can be used to create navigation Links. For Authentication, store JWT token in Local or Session Storage. REST APIs are invoked from the corresponding Services,

Below are some of the Angular Components

1. ImportExcel
2. ManageCompanies
3. ManageIPO
4. ManageStockExchanges
5. CreateCompany
6. ComparisonCharts
7. UserLogin – Login for both user and Admin
8. UserSignup

Above Angular Components is just a sample list, more additional components need to be identified and developed.

Angular Routing feature is used to display Navigation links on Web Pages. When User clicks a Navigation link, the corresponding Component is displayed.

As known, Angular Services is used to interact with the REST end points, using Observable or Promises.

Fusion Charts library can be used to display Charts.

As known JWT token is generated on the Server side and received by Angular Client on successful authentication. Angular’s HttpInterceptor can be used to automatically send JWT Token thru Header of every HttpRequest.

# Entity Classes – Mid Tier

Below are the activities which need to be performed as part of this

1. Identify all Entity Classes. An Entity class is the one which is mapped to underlying DB Table
2. Identify relationship(such as One to One, One to Many, Many to One, Many to Many) between Entity classes, if required
3. Develop the source code of Entity classes

Below are sample Entity Classes

**CompanyEntity Class:** Below can be fields in Company Entity Class

1. Company Name
2. Turnover
3. CEO
4. Board of Directors
5. Listed in Stock Exchanges
6. Sector
7. Brief writeup
8. Company Stock code in each Stock Exchange

Snapshot of Entity class below

**StockPriceEntity Class:** StockPrice Entity class represents Stock Price of a company at a specific point of time. Below are the fields

1. Company Code
2. Stock Exchange
3. Current Price
4. Date
5. Time

**IPODetailEntity Class:** Indicates IPO details of a specific Company

1. id
2. Company Name
3. Stock Exchange
4. Price per share
5. Total number of Shares
6. Open Date Time
7. Remarks

**UserEntity class:** User login details

1. Id
2. Username
3. Password
4. UserType(if Admin or normal User)
5. E-mail
6. Mobile number
7. Confirmed

**StockExchangeEntity class:**

1. Id
2. Stock Exchange
3. Brief
4. Contact Address
5. Remarks

# Model Classes

Model Classes are the classes which are required to transfer the data between

1. REST APIs and Angular Client,
2. REST Controller and Service Layer
3. Service Layer and Repository Layer

As part of this Phase identify all Model classes, and develop source code for the same.

Model classes are just normal POJO classes with data members, constructors, setter/getter methods

# Architecture Diagram

Class diagram



Architecture of a Single Microservice with REST Controller, Service, Model & Entity Classes and Repository classes



# Development of individual Microservices

This specific Phase is to design/develop individual Microservices. Analyze the requirement and divide back end functionality into multiple Microservices. Based on the Mentor on Demand requirements, below can be possible Microservices

1. Microservice to upload excel. All required Error Handling including below cases need to be implemented

uploadExcel

1. Microservice to retrieve data of a company, for certain period, getCompanyStockPrice i/p Company ID, From Period, To Period, periodicity

getMatchingCompanies – used to retrieve list of Companies based on pattern matching of Company Name

getCompanyDetails – Basic information about company

getCompanyIPODetails – IPODetails of Company

1. Microservice to retrieve data of a sector, for certain period: Calculations need to be performed

getList of Companies in a Sector

getSectorPrice Sector ID, From Period, To period, periodicity

1. Microservice for User Login, Signup

User Login

Admin Login

User Signup

1. StockExchange

getStockExchangesList

addStockExchange

getCompaniesList – in a specific Stock Exchange

Each of above Microservice need to comprise below functionality, which need to be developed

1. Each Microservice is a Spring Boot Rest application by specifying required spring boot starter packages in pom.xml os by using Spring Initializr
2. REST Controllers, with the appropriate REST End points to perform corresponding CRUD operations. Along with End Points which are exposed to Angular Client, you may need additional End point(s) for interaction between Microservices
3. As known, each Microservice is a self-sufficient and standalone application, and owns data stored in specific DB tables or databases.
4. Services – Service Layer
5. Entity & Model classes, including appropriate relationship (like One-One, Many-One, etc…) between Entity Classes, if required. (Entity and Model classes have been developed in the Previous Phases)
6. In case specific Entity or Model classes are required across multiple Microservices, it is recommended to maintain separate copy of Entity or Model classes for each Microservice.
7. Microservice interaction with corresponding DB tables or Databases it owns.
8. It is possible that one Microservice need to interact with other Microservice(using RestTemplate or FeignClient)
9. Repository class which implements JPA or CrudRepository, if RDBMS is used
10. Usage of Custom Queries in JPA or CrudRepository using @Query where ever custom functionality required
11. Feign Client can be used to invoke one Microservice, from another Microservice
12. Use Postman to test the Microservices by directly passing requests to each REST end Point, of each Microservice
13. Unit Testing code can be developed using JUnit, Mockito, and perform Unit Testing

# Database Tables

Below are list of Database Tables, for actual fields refer corresponding Entity classes. Though, ideally each Microservice need to use separate database, it should be fine to place all below DB Tables in a single database

|  |  |
| --- | --- |
| Table Name | Purpose |
| Company | Stores list of Company details |
| StockPrice | Stores the Stock prices of all companies, which are imported thru Excel Sheet |
| IPODetails | Stores the IPO details of companies |
| Sectors | Stores the list of Sectors(like Health Care, banking, etc…) supported |
| Users | Stores profile & login details of Admin and normal Users |
| StockExchange | Stores the list of Stock Exchanges supported(like |

# 