|  |
| --- |
|  |

|  |
| --- |
| **Stock Market Charting (Mid-Tier Java Phase 2) v4.0** |
| 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 |
|  |
| **IIHT** |
| **1/10/2019** |
|  |

Table of Contents

[1. Business Requirement(Stock Market Charting) 2](#_Toc28698526)

[1.1. Admin Use Cases: 2](#_Toc28698527)

[1.2. User Use Cases: 3](#_Toc28698528)

[1.3. Company related Data Fields: 4](#_Toc28698529)

[1.4. Stock Price details Excel: 4](#_Toc28698530)

[1.5. IPOs planned: 4](#_Toc28698531)

[1.6. Sectors data Fields: 4](#_Toc28698532)

[1.7. User DB Table: 4](#_Toc28698533)

[1.8. Stock Exchange Data Fields: 5](#_Toc28698534)

[2. Design Inputs 5](#_Toc28698535)

[3. Entity Classes 5](#_Toc28698536)

[4. Model Classes 7](#_Toc28698537)

[5. Other Core Java Implementation 7](#_Toc28698538)

[6. Architecture Diagram(just for reference) 7](#_Toc28698539)

[7. Full Stack Technologies 8](#_Toc28698540)

[8. Technical Spec – Solution Development Environment 9](#_Toc28698541)

[8.1. Front End Layer 9](#_Toc28698542)

[8.2. Middle Tier Layer 9](#_Toc28698543)

[8.3. Database & Integration Layer 9](#_Toc28698544)

[8.4. Ancillary Layer 9](#_Toc28698545)

[Controllers can be tested using Postman Tool 10](#_Toc28698546)

[8.5. Security 10](#_Toc28698547)

[8.6. Deployment & Infrastructure 10](#_Toc28698548)

[8.7. Editors 10](#_Toc28698549)

[9. Assessment Deliverables 10](#_Toc28698550)

[10. Important Instructions 10](#_Toc28698551)

# 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(Optional):** If this feature is not implemented, you may use hard coded fixed records in database. 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(Optional):** If this feature is not implemented, you may use hard coded fixed records in database.
   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.

Below is sample Excel format, which can be used. (Associates can add more fields, if required)

<https://github.com/vskreddy652/Genc_BatchB/blob/master/StockExchange_CaseStudy/sample_stock_data.xlsx>

* 1. 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.
  2. The company code, date ranges need to be appropriately checked, if any data will be over written.
  3. 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 got imported, from and to date range, etc…

1. **Missing Data(Optional):** 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. Search can be based on Company Name.
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(Optional)
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. Two different companies over a specific period
   3. a single sector over different periods of time
   4. different sectors over a specific period(Optional)
   5. between a Sector and a company over a specific period of time(Optional)
8. Should be possible to select if comparisons need to be displayed in a single chart(Optional)
9. Use different colors when multiple Companies/sectors are displayed in a single chart and display legend
10. You may use Bar charts to display data.
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(Optional) for that specific period
13. Possibility to perform multiple comparisons between Company’s or Sectors, over a period of time.(Optional)
14. When data does not exist for certain period in between, that need to be appropriately indicated in the chart(Optional)
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.

# Entity Classes

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

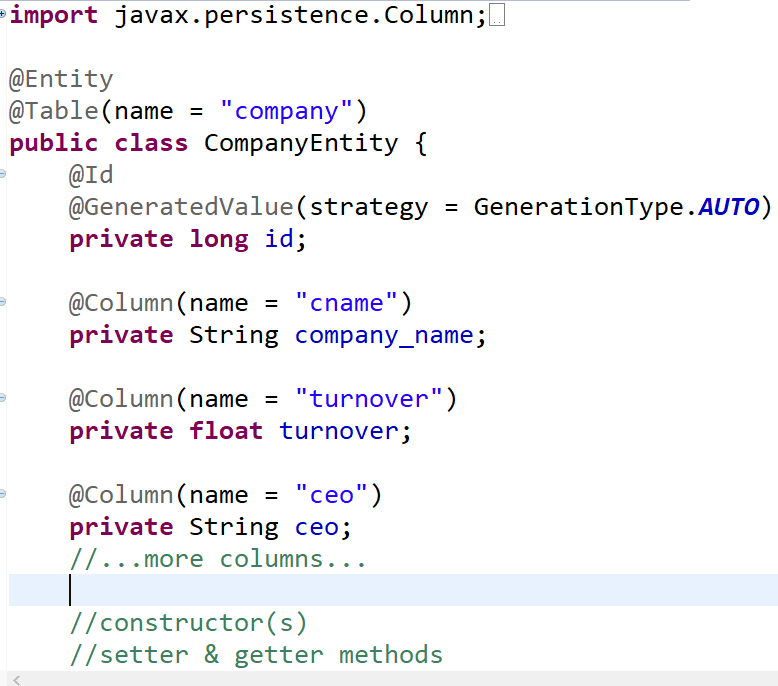
1. Identify all Entity Classes and its fields. An Entity class is the one which is mapped to underlying DB Table
2. Identify relationship(such as One to One, One to Many, Many to One, Many to Many) between Entity classes, if required
3. Develop the source code of Entity classes

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 its fields, and develop source code for the same.

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

# Other Core Java Implementation

This Phase also comprises development of other Core Java source code required for the Project.

# Architecture Diagram(just for reference)

Below Microservice Architecture would be required in next Phases



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



# Full Stack Technologies

The technologies included in Full Stack are not limited to following but may consist of:

* UI Layer (HTML5, CSS3, Bootstrap 4, JavaScript, Jquery, Angular 4/6)
* Middleware Restful API (Spring Boot Restful & MicroServices, JAX-RS, Spring MVC)
* Database Persistence ( Hibernate)
* Database layer (MySQL or MongoDB)
* Ancillary skills (GIT, Jenkins(CI/CD), Docker, Maven) etc.

To complete this case study, you should be comfortable with basic single page web application concepts including REST and CRUD. You may use angular-cli to create your template project. All web pages need to be responsive.

Ref1: https://cli.angular.io/

Ref2: <https://github.com/angular/angular-cli>

# Technical Spec – Solution Development Environment

## Front End Layer

|  |  |
| --- | --- |
| **Framework(s)/SDK/Libraries** | **Version** |
| Angular with TypeScript | 4/6 |
| Bootstrap | 3.0 or above |
| CSS | 3 |
| HTML | 5 |
| JavaScript | 1.8 or above |
| JQuery | 1.3 |

## Middle Tier Layer

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Java Stack | Spring Boot | 1.5 or above |
| Spring MVC | 4.0 or above |
| JDK | 1.7 or above |
| Maven | 3.x or above |

## Database & Integration Layer

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Java Stack | Hibernate | 4.0 or above |
| JAX-RS Jersey/ Spring Restful |  |
| MySQL | 5.7.19 |
| MongoDB | MongoDB | 3.4 |
| NoSQL |  |

## Ancillary Layer

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Source Code Management Tool | GIT | 2.14.2 |
| Build Tool/JAVA Stack | Maven | 3.x |
| Testing Tool/JAVA Stack | JUnit/Mockito | 4.x |
| Testing Tool/JAVA Stack | Spring Test | 4.x |

## Controllers can be tested using Postman Tool

## Security

|  |  |
| --- | --- |
| **Name** | **Version** |
| Spring Boot Security |  |
| JWT |  |

## Deployment & Infrastructure

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Docker | - |  |
| Apache Tomcat | - |  |
| Jenkins(CI/CD) | - |  |
| Node | - |  |

## Editors

|  |  |
| --- | --- |
| **Name** | **Version** |
| STS(Spring Tool Suite) |  |
| Visual Studio Code |  |

Agile/Scrum Software development Model can be used

# Assessment Deliverables

1. Source code of Compiled Entity Classes
2. Source code of Compiled Model Classes

# Important Instructions

1. Consider using below Java features
2. Lambda Expressions
3. Collection Streams
4. Generics
5. Sample Design provided is just for reference, Associates can make changes over it or follow their own Design.
6. Based on your current work, alternate Technologies can be used, for example ReactJS instead of Angular, etc…, however prior approval from the Mentor is required.
7. Please make sure that your code does not have any compilation errors while submitting your case study solution.
8. The final solution should be a zipped code having solution. Solution code will be used to perform Static code evaluation.
9. Implement the code using best design standards/family Design Patterns.
10. Use Internationalization for all the labels and messages in Rest API Development.
11. Do not use System out statements or console.log for logging in Rest API and FrontEnd respectively. Use appropriate logging methods for logging statements/variable/return values.
12. If you are using Spring Restful or Jersey JAX-RS to develop Rest API, then use Maven to build the project and create WAR file.
13. Write web service which takes input and return required details from database.
14. Use JSON format to transfer the results.

For any further queries you can contact fullstack@iiht.com