Stock Market Charting v4.0 UIUX (Phase1)

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

02/07/2019

Table of Contents

1. 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

1.1. 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.
   a. Add a new company details with fields or edit an already existing Company details
   b. Deactivate an already existing company
   c. Update any IPO related data
4. **Import Data(Excel Format):**
   a. 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
   b. 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.
   c. The company code, date ranges need to be appropriately checked, if any data will be over written.
   d. 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…
5. **Missing Data(Optional):** Able to check the dates for which Stock Price of a Company is not available

1.2. 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
    a. Signup for a new Account. When signed up, an email needs to be sent to the User with confirmation link.
    b. Login to an existing and Email confirmed account. User should be able to Login only after E-Mail Confirmation is done.
    c. 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
    a. a single company over different periods of time
    b. Two different companies over a specific period
    c. a single sector over different periods of time
    d. different sectors over a specific period(Optional)
    e. 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)

1.3. 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

1.4. 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

1.5. 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

1.6. 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

1.7. User DB Table:

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

1.8. Stock Exchange Data Fields:

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

2. Design Inputs

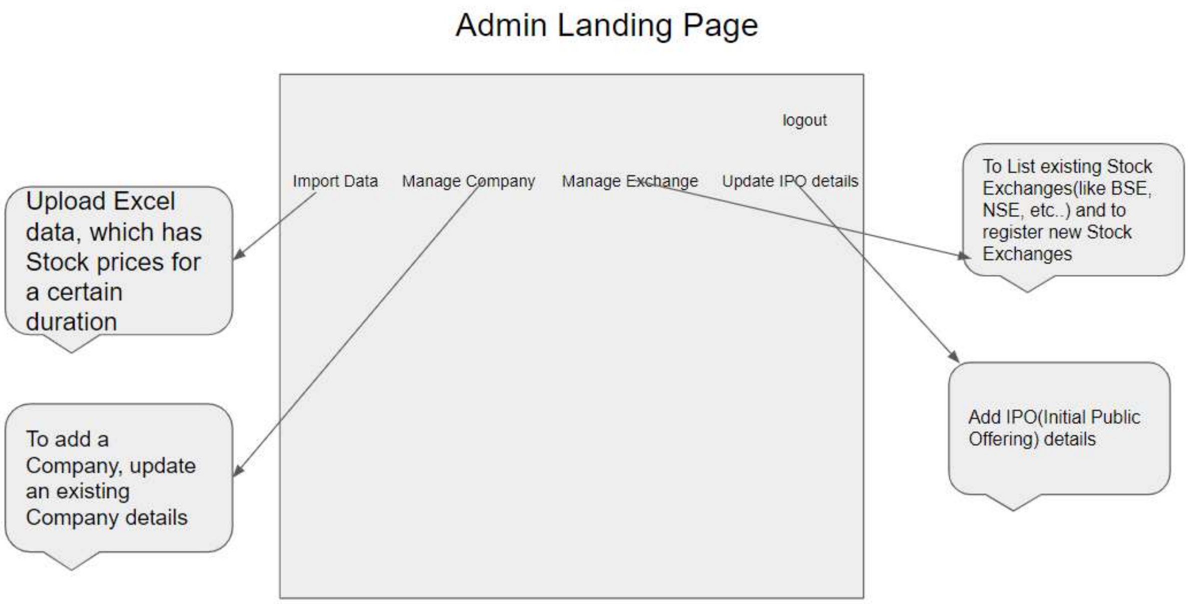
Next sections in this document provides inputs on designing the solution for above requirements.

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.

3. UI/UX Design

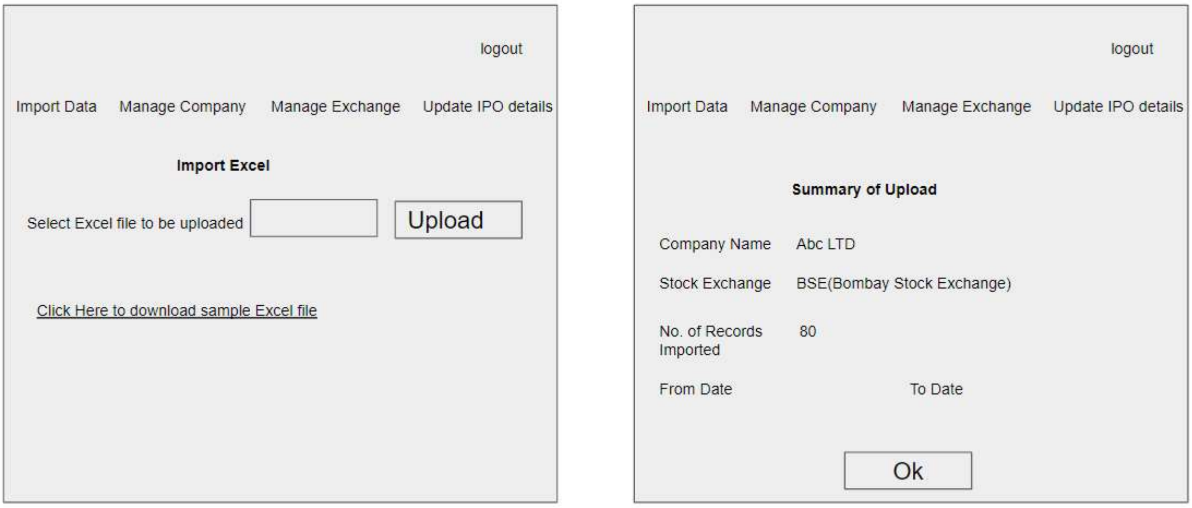
In this Phase you will develop Angular Client Application, with responsive UI using HTML5, HTML5 API, CSS3, Bootstrap/Material and Angular.

Below are some sample Wireframes



Admin Landing Page
logout
Import Data
Manage Company
Manage Exchange
Update IPO details
Upload Excel data, which has Stock prices for a certain duration
To add a Company, update an existing Company details
To List existing Stock Exchanges(like BSE, NSE, etc..) and to register new Stock Exchanges
Add IPO(Initial Public Offering) details

Admin - Upload Excel to import Data



Import Excel

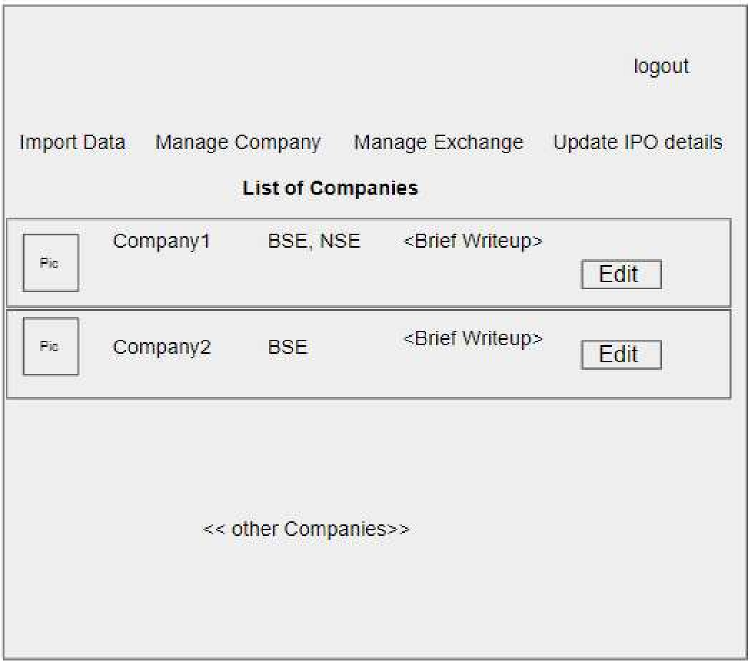
Select Excel file to be uploaded

Click Here to download sample Excel file

Summary of Upload

Company Name	Abc LTD
Stock Exchange	BSE(Bombay Stock Exchange)
No. of Records Imported	80
From Date	To Date

Manage Companies



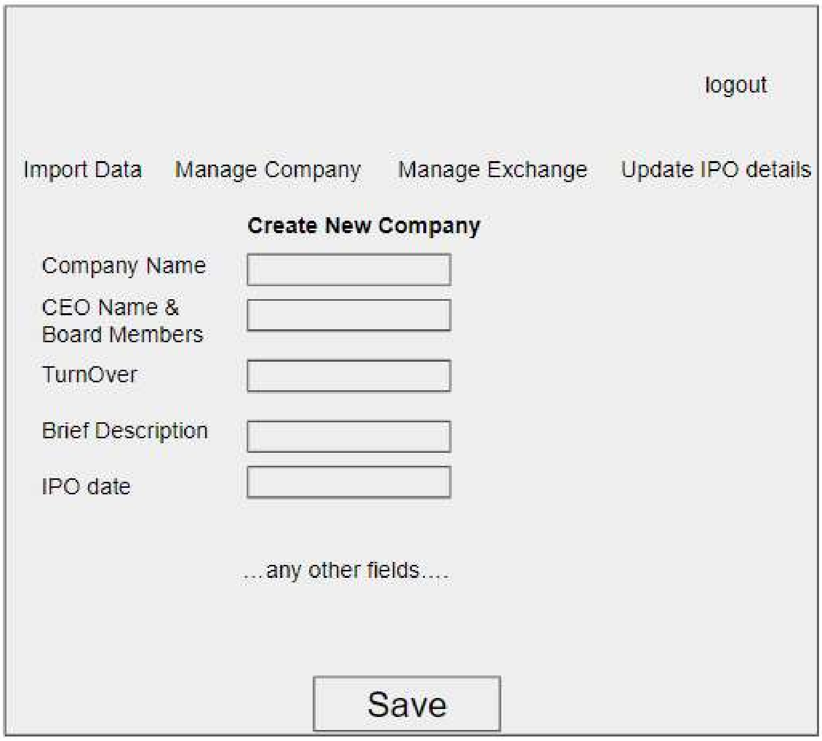
List of Companies

Company1	BSE, NSE	<Brief Writeup>	Edit
Company2	BSE	<Brief Writeup>	Edit

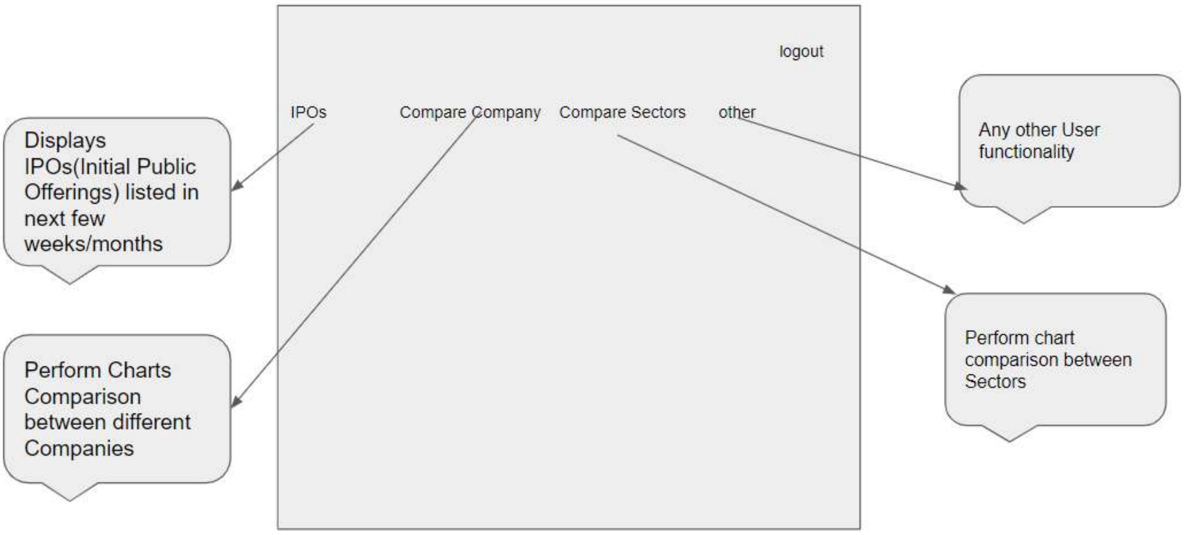
<< other Companies>>

If you have too many companies, you may support searching companies based on search Pattern (optional functionality)

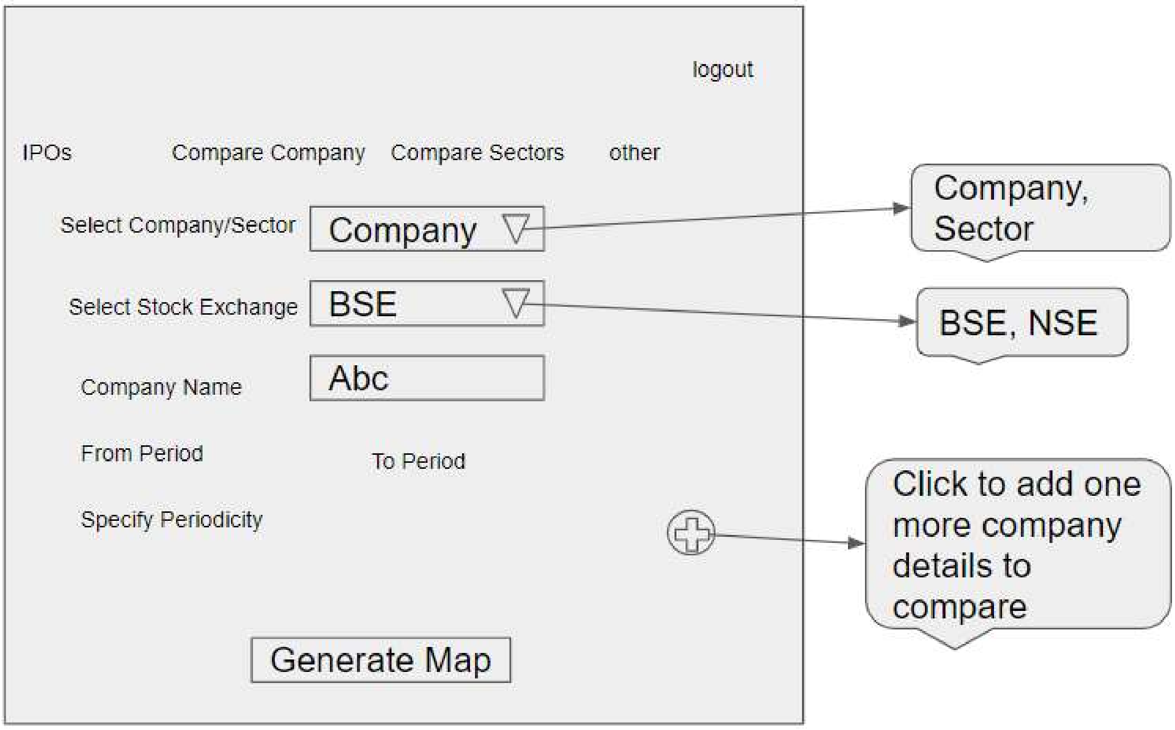
Create New Company



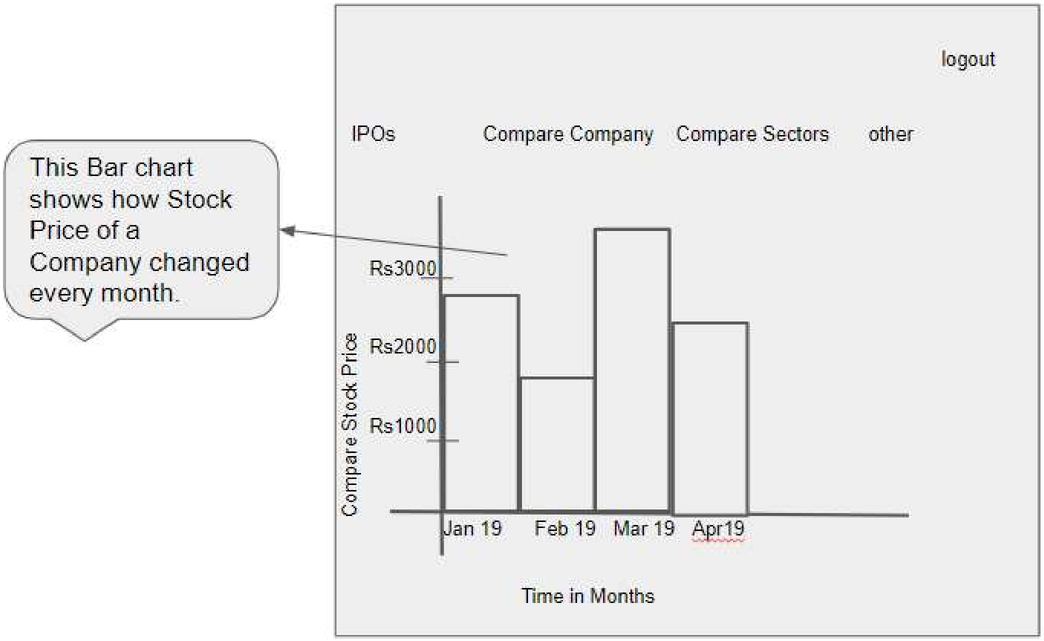
User Landing Page


logout
IPOs
Compare Company
Compare Sectors
other
Displays IPOs(Initial Public Offerings) listed in next few weeks/months
Perform Charts Comparison between different Companies
Any other User functionality
Perform chart comparison between Sectors

Comparison Charts



Company Stock Price(Bar Chart) for Reference only



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.

4. 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

5. Technical Spec – Solution Development Environment

5.1. 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

5.2. 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

5.3. 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	

5.4. 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		

5.5. Security

Name	Version
Spring Boot Security	
JWT	

5.6. Deployment & Infrastructure

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

5.7. Editors

Name	Version
STS(Spring Tool Suite)	
Visual Studio Code	

6. Deliverables of this Phase

1. Html templates along with Angular components
2. Unit Tested Angular Components using Jasmine

7. Important Instructions

1. Consider using below Java features
   a. Lambda Expressions
   b. Collection Streams
   c. Generics
2. Sample Design provided is just for reference, Associates can make changes over it or follow their own Design.
3. 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.
4. Please make sure that your code does not have any compilation errors while submitting your case study solution.
5. The final solution should be a zipped code having solution. Solution code will be used to perform Static code evaluation.
6. Implement the code using best design standards/family Design Patterns.
7. Use Internationalization for all the labels and messages in Rest API Development.

8. 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.
9. 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.
10. Write web service which takes input and return required details from database.
11. Use JSON format to transfer the results.

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