Vamsy Kiran Aripaka

Portfolio Management System

Contents

[Problem Statement: 2](#_Toc54163735)

[PMS rules of the game 2](#_Toc54163736)

[Identity Management 2](#_Toc54163737)

[BackOffice User Operations 2](#_Toc54163738)

[Client Operations 2](#_Toc54163739)

[Expected Development Platform 3](#_Toc54163740)

[Mile Stones 3](#_Toc54163741)

# Problem Statement:

The **PMS (Portfolio Management System)** is going to be a web application to offer various trading and investment services online. The below are the high level client expectations.

## PMS rules of the game

1. An investor is given a INR 2500 as welcome bonus on registration
2. Each investor is identified by his/her PAN and no double registrations allowed.
3. Each sell transaction is subjected to a 2% brokerage/commission.
4. No commission charged on buying’s.
5. Currency exchange rates shall be pulled from <https://exchangeratesapi.io/> or any other relevant api.

## Identity Management

1. The application is accessed by users of below roles:
   1. Investor
      1. A user can register himself as an investor by providing the login key and password.
   2. BackOffice Representatives
      1. A back office user account is created by an admin.
   3. Super User
      1. An admin / a super user’s credentials are pre-inserted in the database.

## BackOffice User Operations

1. Create/modify Listed Company IPO and Profile like Company Code, Company Title, Operations, ShareCount, Open Share Price in USD, Sector, Currency and Turnover
2. Add Stock Prices
   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 USD.
   4. Date – Date of the Stock Price
   5. Time – Stock Price at this Specific time
3. Create/modify Commodities like Gold, Silver..etc
4. Add Commodity Price each day.
5. Generate a commission report
   1. Annually
   2. Monthly
   3. Periodic from a given date to date.

## Client Operations

1. An investor once logins lands on a dashboard showcasing the below details:
   1. Current Portfolio value (a sum of current share values all shares purchased and commodity value)
   2. Recently viewed companies
   3. Amount invested as on date
   4. Amount earned as on date (Current Portfolio value – amount invested)
   5. A chart displaying the earning trend week wise for past 10 weeks.
2. An investor is expected to select his currency of trading.
3. An investor can search for a company and access its profile.
4. Choose companies from a sector and compare their current share value
5. Buy stocks (shares)
6. Sell stocks (shares) and the amount shall be posted into the portfolio wallet.
7. Deposited amount into the portfolio wallet
8. Withdraw from portfolio wallet
9. Generate a portfolio report
   1. Annually
   2. Monthly
   3. Periodic from a given date to date.

# Expected Development Platform

|  |  |
| --- | --- |
| Database | MySQL 5 or above |
| Backend | Java 8 or above |
| Persistence Layer | ORM – Spring Data JPA |
| Application Framework | Spring Web MVC on Spring Boot |
| Views | HTML 5, CSS 3, JSP with JSTL |
| Security | Spring Security - Role-based Authentication |
| SCM | GIT |
| Build Tool | Maven |
| CD/CI | Jenkins |
| Container | Docker |
| Testing | Spring Test |

Mile Stones

|  |  |  |
| --- | --- | --- |
| Mile Stone | Expectation | Deliverables |
| Phase-1, Week-1 | 1. Database script to create all required tables for this project. 2. Class diagram of entities / models. 3. Entity / Model class source code. | Create a GitHub repo common to the group and upload the following and share that URL.   1. Dbscript.sql 2. modelsClassDiagram.jpg 3. Java Project containing the entity / model class source code. |
| Phase-1,  Week-2 | 1. JSP pages for the entire project. | A Maven Web Project to be uploaded in to the Git Repo. This project is expected to contain the models and JSP pages. |
| Phase-2,  Week-1 | 1. CSS styling for JSP views 2. Entities and Model split up 3. ORM repositories for entities | A Spring boot Web MVC project to be uploaded in to the Git Repo. This project will have Entities, models, ORM based DAOs (repositories) and JSP pages |
| Phase-2,  Week-2 | 1. Controllers for first two modules 2. Services for first two modules 3. Exception Handling | A Spring boot Web MVC project to be uploaded in to the Git Repo. This project will have Entities, models, ORM based DAOs (repositories), Services, Controllers and JSP pages |
| Phase-3,  Week-1 | 1. Controllers for left over modules 2. Services for left over modules 3. Exception Handling | A Spring boot Web MVC project to be uploaded in to the Git Repo. This project will have Entities, models, ORM based DAOs (repositories), Services, Controllers and JSP pages |
| Final Review | 1. Spring Security Integration 2. Spring Test 3. Jenkins pipeline script 4. Docker script | A Spring boot Web MVC project to be uploaded in to the Git Repo. This project will have Entities, models, ORM based DAOs (repositories), Services, Controllers and JSP pages with spring security integration, spring test cases, Jenkinsfile and Dockerfile |