SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Stock Portfolio Tracking**

**Prepared by:-**

*Nithin Kumar R S*

*Muruga Perumal R*

*Kishore Kumar K*

# Introduction

## Purpose

The main objective of this document is to illustrate the requirements of the project Stock Portfolio Tracker. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The purpose of this project is to assist users in managing and monitoring their stock and know the performance of the stocks. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

## Document Conventions

* + - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

Stock Portfolio Tracker is basically creation, management and analysis of investments done on stocks. User can see Real-time, historical data tracking and performance evaluation through reports and visual representations. Personalized alerts and notifications will be sent.

The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules.

The language used for developing the project is Java as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 Common Stocks and Uncommon Profits by Philip Fisher

* + - Websites

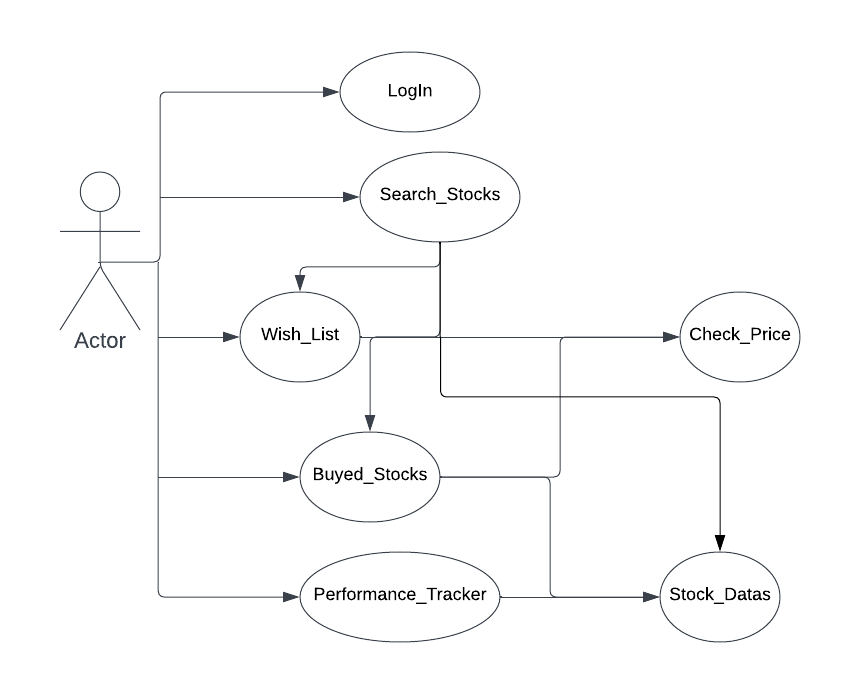
<https://polygon.io/>

<https://www.moneycontrol.com/stocksmarketsindia/>

# Overall Descriptions

## Product Perspective

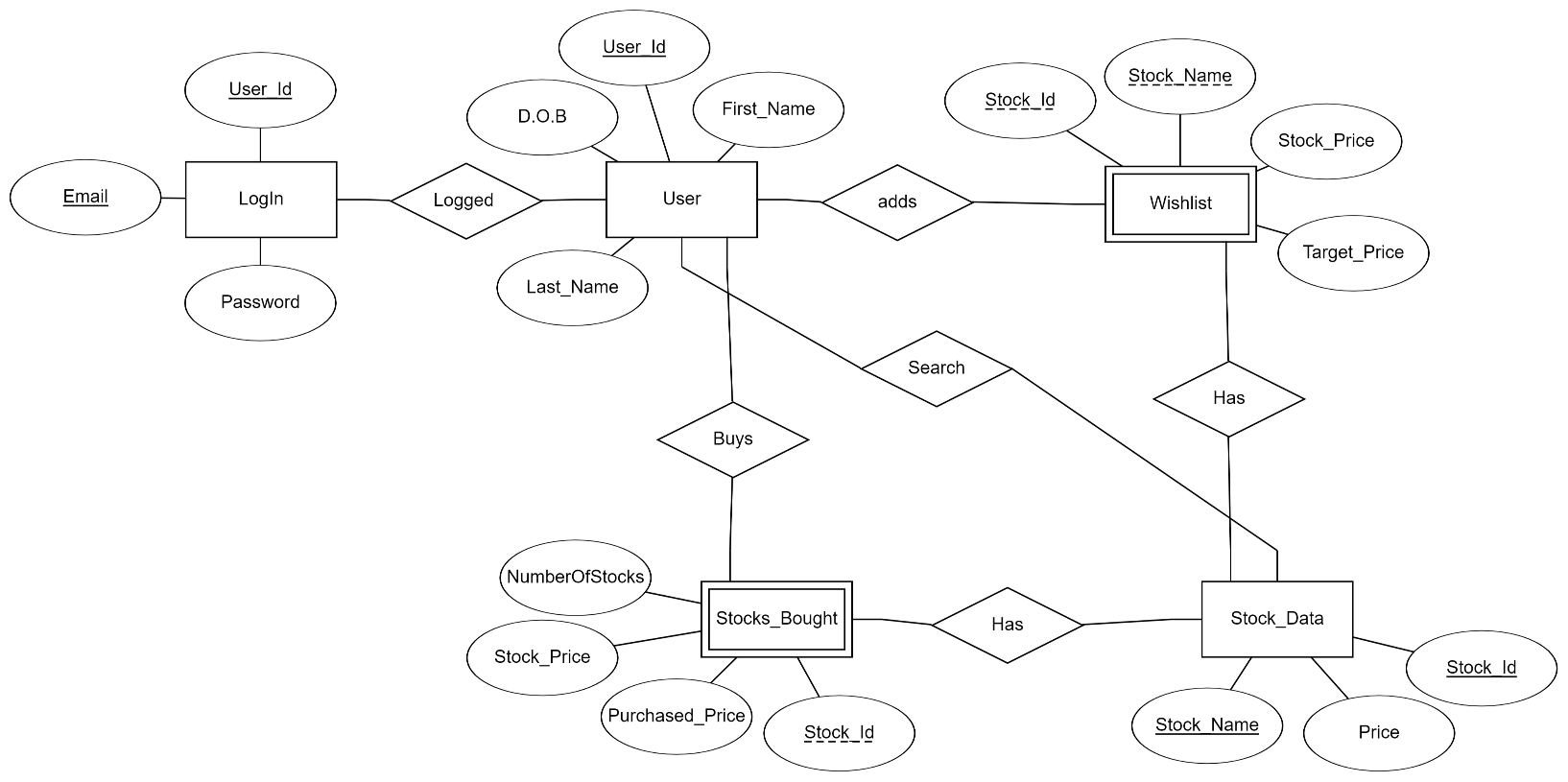
Use Case Diagram of Stock Portfolio Tracker



This is a broad level diagram of the project showing a basic overview of stock portfolio Java application. Users, portrayed as actors, engage with the system through multiple use cases. These encompass logging into the system, searching for stocks by symbol or name, viewing their purchased stocks and wishlist, checking stock prices, adding stocks to their portfolio or wishlist, and utilizing a performance tracker to monitor portfolio performance. This comprehensive diagram provides a holistic representation of the application's functionality and user-system interactions, facilitating the design and comprehension of the stock portfolio management system.

## Product Function

Entity Relationship Diagram of Library Management System



The presented Entity-Relationship (ER) diagram establishes a comprehensive framework for a Stock Portfolio Tracker. At its core, the "Login" entity captures user authentication details, linking to the "User" entity which stores personal information. The system incorporates Stocks Bought through the "Stocks Brought" entity, enabling users to track stock detials. This ER diagram delineates a sophisticated system architecture, seamlessly integrating user authentication, personal data, wishlist functionality, and detailed stock-related datas.

## User Classes and Characteristics

The system provides a platform where user can see all the stocks bought by user and the user can add stocks in wishlist which the user is interested in buying and the user can see a full analytical views.

The features that are available to the users are:-

* + - Can view the different stocks they have invested in.
    - Can view the Performance analysis and reports.
    - Can do real-time stock data and reporting.
    - Can add wishlist.

## Operating Environment

The product will be operating in windows environment. Stock Portfolio Tracker is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer,Google Chrome,and Mozilla Firefox.Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The system should be user-friendly so that it is easy to use for the users
    - The information of all users stocks must be stored in a database that is accessible by the website
    - The system should have more storage capacity and provide fast access to the database
    - The system should provide search facility and support quick transactions
    - The Stock Portfolio Tracker is running 24 hours a day
    - Users may access from any computer that has Internet browsing capabilities and an Internet connection
    - Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - The end users should have proper understanding of the product
    - The system should have the general report stored
    - Any update regarding the stock is to be recorded to the database and the data entered should be correct

## Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: MS SQL Server (back end)

Hardware Configuration:- Processor: Pentium(R)Dual-core CPU Hard Disk: 40GB

RAM: 256 MB or more

## Data Requirement

The inputs consist of the data to the database and the output consists of the analytics. The output also includes the user receiving the details of their accounts. In this project the inputs will be the data as fired by the users like create an account, selecting stocks and investments.

# External Interface Requirement

## GUI

The software provides good graphical interface for the user and the user can operate on the application, performing the required task such as seeing stock price updates, historical data view, performance metrics.

* + - It allows user to view stock price updates, historical data view, performance metrics.
    - It provides quick access to portfolios.
    - The user interface must be customizable if needed
    - All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined
    - The design should be simple and all the different interfaces should follow a standard

template

* + - The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

Login Interface:-

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

Search:-

The user can enter the name of the stock he is looking for and search the required details.

User Control Panel:-

This control panel will allow user to add/remove stocks; add, edit the number of stocks bought, or remove a stock.

# System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* User authentication and validation of members using their unique member ID
* Proper accountability which includes not allowing a member to see other member’s account.

# Other Non-functional Requirements

## Performance Requirement

* + - The performance of the system should be fast and accurate.
    - System response time (<2 seconds) for retrieving stock prices.
    - Support for concurrent user access without performance degradation.
    - The system should be able to handle large amount of data.

## Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## Security Requirement

* + - System will use secured database
    - Implementation of secure authentication mechanisms.
    - System will have different types of users and every user has access constraints.
    - Proper user authentication should be provided
    - No one should be able to hack users’ password
    - Encryption of sensitive user data and secure storage.

## Requirement attributes

* + - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

The proper user interface, user manual, online help and the guide to use and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The application provides certain facilities to the users in the form of:-

* + - Backup and Recovery
    - Forgot Password
    - Users should access the system across devices
    - Data migration i.e. whenever user registers for the first time then the data is stored in the server
    - Users should access real-time stock prices
    - Users should access historical stock data for analysis

# The Other Requirements

## Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - Administrator: A login id representing a user with user administration privileges to the software
    - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes. server must be maintained regularly and it has to be updated from time to time.

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘User’, ‘Stocks Bought’ and ‘’ are the most important classes which are related to other classes.

