SOFTWARE REQUIREMENTS SPECIFICATION

For

STOCK PORTFOLIO TRACKER

Prepared by:-

Janani M
Rithika S
Rithish S

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive and detailed blueprint for the development of the Stock Portfolio Tracker App. It serves as a communication bridge between stakeholders and the development team, outlining the functional and non-functional requirements essential for the successful implementation of the application. By clearly defining the app's features, user interactions, and system constraints, this document aims to ensure a shared understanding among all project participants. Additionally, the SRS serves as a reference guide throughout the development lifecycle, assisting in project management, quality assurance, and validation processes. Ultimately, the document facilitates a systematic and structured approach to app development, fostering clarity, accountability, and successful delivery of a robust and user-friendly Stock Portfolio Tracker.

1.3 Scope of Development Project

The scope of the Stock Portfolio Tracker App development project encompasses the creation of a comprehensive mobile application for both iOS and Android platforms. The primary focus is on empowering users to efficiently manage and analyze their stock portfolios in real-time. Key features include portfolio creation, addition, and modification of stock holdings, access to real-time stock prices, historical data analysis, personalized alerts, and in-depth performance analytics. The app will interface With reliable stock market APIs to ensure accurate and up-to-date information. The development scope emphasizes user-friendly interfaces, robust security measures, and seamless offline functionality,ultimately delivering a versatile tool that caters to both novice investors and experienced traders.

1.4 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

1.5 References

- Books

  Karl E. Wiegers, "Software Requirements" Second Edition (Microsoft).

  Roger S. Pressman, "Software Engineering: A Practitioner's Approach" Fifth Edition.

  Burton G. Malkiel, "A Random Walk Down Wall Street."

- Websites

  https://www.investopedia.com/markets/
  https://www.alphavantage.co/

2. Overall Descriptions

2.1 Product Perspective:

This is a high-level diagram of the stock portfolio tracker project, providing a fundamental overview. Users categorize into staff or investors. The system integrates a search feature for exploring stocks based on criteria like stock names or symbols. Financial staff can effortlessly add or remove stocks and monitor user portfolios. Investors can initiate actions such as adding or removing stocks, effectively managing their portfolio within the system.

2.2 Product Function

The Online Stock Portfolio Tracker offers real-time information on available stocks and user portfolios, aiming to streamline manual processes. This software effectively manages stock transactions, portfolio updates, calculates and manages gains or losses,and generates custom reports for record-keeping. Administered by the portfolio manager, the system ensures control over user accounts and stock management. The database maintains the status of stock transactions, and portfolio details can be accessed by the manager as needed.Valid users have the privilege to view their portfolio information and account details for enhanceduser engagement and transparency.

2.3 User Classes and Characteristics

End Users:
Description: The primary user class includes individual investors and traders who utilize the stock portfolio tracker to manage and monitor their investments in the stock market.

Characteristics:

Diverse demographic profiles ranging from novice investors to experienced traders.
Varying levels of familiarity with financial markets and investment strategies.
Typically interested in features such as portfolio tracking, real-time stock data, and performance analysis.

Add Stock:

- Users can add new stocks to their portfolio.
- Intuitive interface with auto-complete for stock symbols.
- Option to specify the quantity, purchase price, and purchase date.

Remove Stock:

- Users can remove stocks from their portfolio.
- Confirmation prompts to prevent accidental removal.
- Record of removed stocks in transaction history.

Performance Analysis and Reporting:

- Users can keep track of the stock price trend in different time periods to make a decision.

2.4 Operating Environment

The stock portfolio tracker operates on Windows, macOS, and Linux systems, using Java SE 8 or later, MySQL 5.7+, and Eclipse IDE for Java development. It requires internet connectivity for real-time stock updates, employs secure user authentication, and integrates with financial APIs. The development environment includes JavaFX for the graphical interface and Git for version control. User support is provided through comprehensive documentation and a helpdesk.
The hardware configuration includes Hard Disk: 40 GB, Monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, etc.

2.5 Assumptions and Dependencies

The assumptions are:-

- Coding for the stock portfolio tracker is error-free for system stability.
- The system assumes a user-friendly interface, facilitating easy stock portfolio management.
- Information on users, stocks, and transactions is stored in an accessible database.
- Adequate storage capacity and quick database access are assumed for efficient performance.
- The system assumes the provision of a search facility and support for swift transactions.
- Continuous 24/7 availability is assumed for users' access from any device with internet capabilities.
- Secure user authentication with correct credentials is required for system access.

The dependencies are:-

- Effective functioning of specific hardware and software configurations.
- Development and execution based on specified requirements for the stock portfolio tracker.
- The system's effectiveness depends on administrators' understanding of the stock portfolio tracker.
- Dependency on the system's ability to store and generate general reports for analysis.
- All user information must be stored in a database accessible by the stock portfolio tracker.

- ➢ Accurate recording of any stock updates in the database is crucial for system integrity.

2.6 Requirement

Software Configuration:-
This software package is developed using java as front end. Microsoft SQL Server as the back end to store the database.
Operating System: Windows Server 2016 and above or Linux (Ubuntu 20.04 LTS or CentOS 7).
Language: Java Runtime Environment (front end)
Database: MS SQL Server (back end)
Framework: JavaFX
Tools: SceneBuilder

Hardware Configuration:-
Processor: A dual-processor or higher
Hard Disk: 20GB
RAM: 4 GB or more

2.7 Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like creating an account, selecting books and putting them into account. Now the output will be visible when the user requests the server to get details of their account in the form of time, date and which books are currently in the account.

3. External Interface Requirement

3.1 GUI

External interface requirements for the Graphical User Interface (GUI) of a Stock Portfolio Tracker app describe how the application's user interface should interact with external elements. This includes user interaction, navigation, and visual representation.
Here are the external interface requirements for the GUI:

1. User Interface Design

Navigation Menu:
The application shall have a clear and intuitive navigation menu accessible from all pages.
Menu options should include Home, Portfolios, Stocks, Reports, and Settings.

Dashboard:
The Home page shall feature a dashboard providing an overview of the user's portfolios, performance metrics, and recent activities.

Portfolio Management:
Users shall be able to create, view, edit, and delete portfolios.
Each portfolio shall display key information, such as total value, gain/loss, and percentage change.

Stock Management:
Users shall be able to add and remove stocks from their portfolios.
Stock details should include real-time prices, historical data, and relevant company information.

2. Real-Time Data Display
Real-Time Stock Prices:
The application shall display real-time stock prices for stocks in the user's portfolios.
Prices should update dynamically without requiring manual refresh.

Price Alerts:
Users shall receive real-time notifications for price alerts triggered by stock price movements.

3. Reporting and Analysis
Portfolio Analysis:
The Reports section shall provide comprehensive analyses of portfolio performance, including ROI, volatility, and historical trends.

4. User Notifications
Alerts and Notifications:
The application shall provide visual and/or audible notifications for important events, such as price alerts and significant changes in portfolio value.

5. User Interaction
User Input Forms:
Input forms for creating and updating portfolios, adding stocks, and setting price alerts shall have clear labels and validation messages.

Interactive Charts and Graphs:
Reports and portfolio overviews shall include interactive charts and graphs to enhance data visualization and analysis.

6. Accessibility and Responsiveness
Cross-Browser Compatibility:
The GUI shall be compatible with the latest versions of major web browsers, including Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

Responsive Design:
The GUI shall be responsive, ensuring a consistent and user-friendly experience across various devices, including desktops, tablets, and smartphones.

7. Security Measures
Secure Authentication:
User authentication elements (login and registration pages) shall use secure protocols, such as HTTPS.
Passwords shall be hashed and stored securely.

8. Preferences and Settings
User Preferences:
Users shall be able to customize preferences, including currency and date format.

Profile Management:
Users shall have the ability to update their profiles, including username, email, and password.

9. Integration with External APIs
Financial Data API Integration:
The GUI shall seamlessly integrate with the chosen financial data API to retrieve real-time stock prices and historical data.

10. Error Handling
User-Friendly Error Messages:
In the event of errors, the GUI shall display clear and user-friendly error messages.

11. Compliance with Design Guidelines
UI Design Standards:
The GUI design shall adhere to established design standards and best practices for a modern and visually appealing user interface.

4. System Features

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

- ➢ Secure login: choose mobile number or various account options for seamless access.
- ➢ Track stocks with real-time data, analyze performance, & visualize trends like a pro.
- ➢ Manage holdings: add/remove, monitor dividends & gains, compare vs benchmarks.
- ➢ Stay informed: get news & alerts, engage with a supportive community for shared insights.
- ➢ Learn & refine: beginner tutorials, advanced analytics for informed investment decisions.
- ➢ Scalable & secure: built for long-term user growth & data protection, giving you peace of mind.

5. Other Non-functional Requirements

5.1 Performance Requirement

1. Real-time Performance:
   - The system must provide real-time updates on stock prices and portfolio values, ensuring swift calculations and responsive user interactions.

2. Error Handling and Security:
   - Robust error handling should address expected and unexpected errors, with secure

authentication measures in place to prevent unauthorized access.

3. Scalability and Data Management:
   - The system should scale seamlessly to accommodate a growing user base and large datasets, optimizing database queries for efficient data retrieval.

4. User-Friendly Interface:
   - The user interface should be intuitive, visually appealing, and responsive, enhancing the overall user experience for both novice and experienced investors.

5. Integration and Reliability:
   - Seamless integration with external financial data sources, efficient API calls, and high availability with minimal downtime are essential for the system's reliability and performance.

5.2 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.

5.3 Security Requirement

- The system will utilize a secure database.
- Regular users are restricted to read-only access, except for their personal information, and specific details, ensuring no unauthorized modifications.
- Various user types with specific access constraints will be implemented.
- Robust user authentication measures will be in place.
- Password security will be prioritized to prevent unauthorized access.

- Distinct admin and member accounts will be established, limiting database access to the admin for updates while ensuring members have restricted access.

5.4 Requirement attributes

- Multiple administrators involved in the project will possess the authority to introduce modifications to the system, while members or other users are restricted from making changes.
- The project is designed to be open source.
- Database quality is upheld to ensure user-friendliness for all database users.
- Users should experience ease in downloading and installing the system.

5.5 Business Rules

A business rule encompasses and puts into action business policies and practices, capable of enforcing business policies, making decisions, or deducing new data from existing information.

This extends to the regulations that users of the system must adhere to, encompassing aspects such as project costs and offered discounts. Users are required to steer clear of any illegal rules and protocols. Both administrators and members are obligated to comply with the established rules and regulations.

5.6 User Requirement

User Roles:

The system comprises members and administrators, where administrators serve as system maintainers. Members are expected to possess basic computer and internet browsing knowledge, while administrators should have a deeper understanding to handle potential issues like disk crashes and power failures.

User Resources:

To facilitate smooth system usage, comprehensive user resources are provided, encompassing a user-friendly interface, detailed user manual, online help, and an installation guide. These resources aim to equip users with the necessary knowledge to operate the system seamlessly.

Administrator Facilities:

Administrators extend crucial facilities to users, including:

- Backup and Recovery: Ensuring data protection and retrieval.
- Forgot Password: Providing a mechanism for password recovery.
- Data Migration: Storing user data on the server during initial registration.
- Data Replication: Safeguarding against data loss through redundancy.
- Auto Recovery: Frequent automatic saving of information.
- File Organization: Maintaining structured and organized files.
- Regular Server Maintenance and Updates: Ensuring server reliability and periodic updates for system optimization.

6. Other Requirements

6.1 Data and Category Requirement

In the Stock Portfolio Tracker app, distinct user categories such as Investors, Portfolio Managers, and Admins exist. Access rights are tailored to each category; administrators have privileges to modify, delete, and append data, while other users, excluding Portfolio Managers, possess rights solely for retrieving information from the database. Additionally, various stock categories will be available, and corresponding data for each category should be presented in a structured format.

6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions; I: Investors, Information; P: Portfolio, Performance Metrics; S: Stocks, Security; U: User, User Requirement;

6.3 Glossary

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

- Admin: A login ID with user administration privileges
- User: Investors, primary users of the app
- SQL: Structured Query Language for database information retrieval
- Layer: Represents a segment of the project
- User Interface Layer: Direct user interaction section
- Application Logic Layer: Web server, where computations occur
- 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

6.4 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 which are related to other classes required for their work. 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 'Investors', 'Portfolio' and 'Stocks' are the most important classes which are related to other classes.