Software Requirements Specification

for

Stock Market Prediction

Version 1.0 approved

Prepared by Prerna Dureja and Avishi Semwal

Manipal University Jaipur

24 January 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.   Introduction

1.1   Purpose

The Stock Market Prediction and Portfolio Management Web Application system is designed to help users make informed investment decisions by analyzing stock market trends using AI-driven predictions. It also allows users to manage their investment portfolios effectively. The document outlines the functional, non-functional, and external interface requirements of the system to provide clarity to developers, stakeholders, and project managers.

1.2   Document Conventions

This document follows standard IEEE formatting guidelines. Headings and subheadings are structured hierarchically using numbered sections. Important terms are emphasized using bold text, and technical terms are explained in the glossary section. Uniquely identified, all the requirements have, and references are made to documents or standards with which the specifications relate.

1.3   Intended Audience and Reading Suggestions

This document is intended for a variety of stakeholders, including developers who will implement the system, project managers who will oversee development, financial analysts who will provide domain expertise, and end-users who will interact with the application. Readers should start with the introduction for an overview before diving into specific sections relevant to their roles. Developers should pay particular attention to the system features and external interface requirements, while testers should focus on performance and security requirements.

1.4   Product Scope

The "Stock Market Prediction and Portfolio Management Web Application" is a web-based platform that utilizes AI and machine learning techniques to forecast stock prices and assist users in managing their investment portfolios. It provides real-time stock market insights, risk analysis, and personalized investment strategies. The application aims to cater to both beginner investors looking for automated recommendations and advanced traders seeking in-depth analytical tools.

1.5   References

This project relies on various external resources and standards, including financial data APIs such as Yahoo Finance and Alpha Vantage for real-time stock data. It also integrates with machine learning frameworks like

TensorFlow and Scikit-learn for predictive modeling. Additionally, web technologies such as React for frontend development and Node.js with PostgreSQL for backend functionality are used.

2. Overall Description

2.1 Product Perspective

The application is a standalone system developed specifically for stock market prediction and portfolio management. It leverages machine learning models to provide accurate forecasts and enhance investment decision-making. Unlike traditional stock market analysis tools, this application integrates AI-based insights with a user-friendly web interface.

2.2 Product Functions

The key functionalities of the system include the following:
• **Stock Market Prediction**: The system uses advanced AI and machine learning models to predict stock price trends, helping users make informed investment decisions. It analyzes historical data, market sentiment, and external financial indicators.
• **Risk Assessment and Investment Insights:** The system evaluates risk levels of stocks and entire portfolios based on volatility, market trends, and economic factors. It generates comprehensive risk reports to assist users in making safer investment choices.
• **Automated Trading Suggestions**: The application can suggest automated buy/sell actions based on market patterns and AI-driven analytics. It can integrate with brokerage APIs for seamless execution of trades based on predefined risk thresholds and investment goals.
• **User Authentication & Authorization**: The platform ensures secure user access using multi-factor authentication (MFA) and role-based access control (RBAC). Users can securely store investment data and set permission levels for different actions. These features aim to provide an all-in-one solution for investors, offering AI-powered insights,
portfolio management, and automated trading suggestions for better financial outcomes.

2.3 User Classes and Characteristics

The system will be used by a diverse group of users, each with specific needs and levels of expertise. The primary user classes include:
• **Beginner Investors:** These users have limited knowledge of the stock market and require intuitive tools for investment decisions. The system will offer simplified portfolio recommendations, educational resources, and automated insights tailored to their needs.
• **Retail Traders:** These users actively trade stocks and require real-time updates, advanced analytics, and AI-driven predictions. The system will provide interactive dashboards, market sentiment analysis, and technical indicators to support their trading strategies.

• **Institutional Investors**: These are professional investors, including hedge funds and financial analysts, who require deep market analysis, portfolio risk assessment, and investment optimization. The system will provide AI-powered trend forecasting and automated trading support tailored to institutional needs.
• **Financial Advisors:** These users manage investments for clients and need portfolio management tools, investment insights, and risk assessment features. The system will allow advisors to create and manage multiple client portfolios efficiently.
• **Developers & Analysts:** These users may require API access to integrate stock prediction models with other financial systems. The system will provide secure data access, API documentation, and analytical tools for customized applications. Each user class will have tailored features and permissions to enhance their experience and meet
their specific investment needs.

2.4   Operating Environment

The system will be a web-based application accessible through standard web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. It will be hosted on cloud infrastructure, ensuring scalability and availability. The backend will be built using Node.js with a PostgreSQL database, while the frontend will be developed using React.js. The application must be compatible with both Windows and macOS environments, as well as mobile devices for on-the-go access.

2.5   Design and Implementation Constraints

The system must comply with financial regulations and data privacy laws, ensuring user data is securely stored and encrypted. Since stock market predictions require real-time data processing, the system must optimize computational efficiency to deliver quick and accurate insights. Additionally, integration with third-party APIs (such as Yahoo Finance and Alpha Vantage) introduces dependencies that may affect data availability and consistency. Security measures such as OAuth-based authentication and role-based access control (RBAC) will be implemented to protect user
accounts.

2.6   User Documentation

The system will provide a comprehensive user manual, online help guides, and video tutorials to assist users in understanding and utilizing its features. Additionally, an FAQ section and chatbot support will be available to answer common user inquiries. Advanced users will have access to API documentation to integrate the system with their existing financial tools.

2.7   Assumptions and Dependencies

The system assumes that users have a stable internet connection to access real-time stock market data. It also depends on third-party financial data providers, which may impact system performance if their services experience downtime. Machine learning models used for stock predictions require periodic retraining with updated data to maintain accuracy. Additionally, browser compatibility and mobile responsiveness assume users are on modern web browsers and devices.

3.   External Interface Requirements

3.1   User Interfaces

- The user interface will be clean, intuitive, and accessible, ensuring easy interaction with stock market data and portfolio features.
- The homepage will display real-time stock prices, trending stocks, and an overview of the user's portfolio.
- A login screen will allow users to enter their credentials for authentication.
- Upon login, users will be directed to a portfolio management dashboard to view investments, add stocks, and track performance.
- Standard buttons such as Add Stock, Remove Stock, and View Predictions will be present on every screen with a consistent layout and design.
- A Help button will be available on every page to provide access to a guide or FAQ section.
- Error messages will be displayed in a clear and consistent format to help users correct mistakes.
- Keyboard shortcuts will be available for easy navigation.
- The interface will be responsive, ensuring accessibility across desktops, tablets, and smartphones.
- GUI design elements such as color schemes, font sizes, and button placements will follow industry-standard practices for ease of use.

3.2   Hardware Interfaces

- The system will run on various hardware, including personal computers, tablets, and smartphones.
- No specialized hardware is required; users will interact using common input/output devices such as keyboards, mice, and touchscreens.
- The application will support standard web browsers, including Chrome, Firefox, and Safari, ensuring accessibility across different devices.
- A minimum display resolution of 1366x768 is required for optimal visual presentation of stock charts and portfolio details.
- An active internet connection is necessary to fetch real-time stock market data from external APIs and update user portfolios.
- The software will be capable of utilizing multi-core processors to enhance performance when executing stock predictions and data analysis.

3.3   Software Interfaces

- The software will interact with multiple components, including a database, APIs, and external libraries.
- PostgreSQL will be used as the relational database to store user profiles, portfolio data, and historical stock prices.
- The system will fetch real-time stock data from external APIs such as Alpha Vantage or Yahoo Finance.
- Python-based libraries like TensorFlow or Scikit-learn will be used for stock prediction models.
- Data will flow between components via RESTful APIs, where the front-end (HTML/CSS/JavaScript) sends requests to the back-end (Node.js and Express.js) for stock-related data and portfolio actions.
- The back-end will communicate with the database to manage user portfolios and stock records.
- External APIs will be used to retrieve real-time stock prices and market trends.
- All communication will be encrypted using SSL/TLS protocols to ensure data security.

3.4   Communications Interfaces

- The application will use HTTP/HTTPS protocols for data transmission between the client (user browser) and the server.
- HTTPS will be enforced to encrypt data and ensure user privacy.
- Stock data will be fetched using REST API calls from external providers like Alpha Vantage, delivering real-time prices, historical data, and market trends.
- WebSocket connections will enable live data streaming for real-time stock updates.
- The application will support email communication for user registration confirmation, portfolio alerts, and system updates.
- An SMTP server will be configured to handle outgoing emails securely.
- SSL/TLS encryption will be used for all sensitive data exchanges, including user credentials.
- All APIs and communication channels will follow industry-standard encryption and security protocols to ensure data protection.

4.   System Features

4.1 User Management

4.1.1 Description and Priority

This feature allows users to register, log in, update their profile, and log out securely.

It is of high priority, as authentication and user profile management are essential for platform access.

4.1.2 Stimulus/Response Sequences

Stimulus: A user enters credentials and clicks the "Login" button.

Response: The system verifies the credentials and grants access if authentication is successful. If MFA is enabled, it requests a second factor (e.g., OTP).

4.1.3 Functional Requirements

REQ-1: The system must allow users to register with email verification.

REQ-2: The system must support login/logout functionality with multi-factor authentication (MFA).

REQ-3: Users should be able to update their profile details, including contact information.

REQ-4: The system should enforce role-based access control (RBAC) for admin and user functionalities.

4.2 Stock Market Prediction

4.2.1 Description and Priority

This feature allows users to view stock predictions based on AI-driven models analyzing historical data, financial indicators, and market sentiment.

It is of high priority, as it is a core function that helps users make informed investment decisions.

4.2.2 Stimulus/Response Sequences

Stimulus: The user searches for a stock and selects "View Stock Prediction."

Response: The system fetches historical data, runs ML-based predictions, and displays forecasted trends (growth/decline percentages for different timeframes).

4.2.3 Functional Requirements

REQ-1: The system must fetch real-time market data via external APIs (e.g., Alpha Vantage, Yahoo Finance).

REQ-2: The system should display predicted stock trends and notify users of potential price changes.

REQ-3: The system must provide predictions for multiple timeframes (daily, weekly, monthly).

REQ-4: The system should handle invalid stock symbols with error messages.

4.3 Stock Search

4.3.1 Description and Priority

This feature enables users to search for specific stocks and retrieve relevant data.

It is of medium priority, as it assists users in gathering stock market information but is not the core feature.

4.3.2 Stimulus/Response Sequences

Stimulus: The user enters a stock symbol and presses "Search."

Response: The system retrieves real-time stock data, including price, market trends, and historical performance.

4.3.3 Functional Requirements

REQ-1: The system must provide a search functionality for stocks.

REQ-2: The system should fetch stock details from external APIs and display them to the user.

REQ-3: The system should allow users to filter and sort stock data based on price, trends, or volume.

4.4 Admin Panel - User & Data Management

4.4.1 Description and Priority

The admin can manage users, update stock data, and manage prediction models.

It is of high priority, as maintaining accurate data and monitoring user activities are crucial for system functionality.

4.4.2 Stimulus/Response Sequences

Stimulus: The admin logs into the panel and selects "Manage Users" or "Update Stock Data."

Response: The system provides the admin with options to modify user details, manage prediction models, or update stock market data.

4.4.3 Functional Requirements

REQ-1: The system must allow admins to view and manage user accounts (activate/deactivate users).

REQ-2: The admin must be able to update stock market data manually if needed.

REQ-3: The system should allow admins to modify prediction models to improve accuracy.

REQ-4: The admin should have access to stock market API integrations for real-time updates.

4.5 Security & Communication

4.5.1 Description and Priority

This feature ensures secure communication, including SSL/TLS encryption, and email notifications for users.

It is of high priority, as data security is essential for protecting user investments.

4.5.2 Stimulus/Response Sequences

Stimulus: A user attempts to log in or a system update is triggered.

Response: The system enforces secure authentication and sends email notifications (for login attempts, stock alerts, or updates).

4.5.3 Functional Requirements

REQ-1: The system must enforce SSL/TLS encryption for all communications.

REQ-2: The system should integrate with an SMTP server for email notifications (e.g., registration confirmation, and portfolio alerts).

REQ-3: The system must implement auto-logout for inactive users to enhance security.

REQ-4: Users should be notified of failed login attempts or unauthorized access.

5.   Other Nonfunctional Requirements

5.1   Performance Requirements

The system must handle multiple users simultaneously, processing real-time data with minimal delay. Stock predictions, portfolio updates, and risk assessments should be completed in under 3 seconds under normal usage. The system should support up to 500 concurrent users without performance issues, with scalability to accommodate growth. Data synchronization must happen within 1 second, ensuring consistency across devices, and batch tasks like portfolio assessments should be completed in 5 minutes or less.

5.2   Safety Requirements

The application should ensure data protection through regular backups, with a recovery plan in case of system failures. Sensitive financial data must be encrypted during storage and transmission. The system should comply with relevant financial regulations like GDPR and PCI DSS to safeguard user privacy and data integrity.

5.3   Security Requirements

Security is crucial, and multi-factor authentication (MFA) should be implemented for user access.

Data encryption (AES-256) is mandatory for both storage and transmission.
Role-based access control (RBAC) will restrict user actions based on their role, and API security protocols like OAuth 2.0 will ensure safe data exchanges. The system must also log user actions to create an audit trail and comply with financial security standards.

5.4   Software Quality Attributes

Usability is key, and the system should be intuitive for both novice and experienced users. It should offer high reliability (99.9% uptime), scalability to support growing user numbers, and easy maintainability with a clear, well-documented codebase. Flexibility to integrate new features or services is essential, and the system should be robust enough to handle failures without data loss. Interoperability with third-party services (like financial APIs) is also a priority.

5.5   Business Rules

Access to features will be role-based; for example, financial advisors can access client portfolios, while retail traders have limited access. Automated trading will only occur when pre-defined conditions are met, ensuring that trades align with user-configured risk levels and strategies. Data access will be restricted to ensure users can only view their portfolios, while advisors can manage client portfolios with explicit permission. The system will rely on authorized data providers for accurate, timely financial information.

6.   Other Requirements

Database Requirements: The system will use a relational database (e.g., PostgreSQL) to store user and market data with high availability and scalability. Backup procedures will be implemented to prevent data loss.
Legal Requirements: The system will comply with GDPR, PCI DSS, and SEC guidelines to ensure data protection and financial reporting compliance.
Reuse Objectives: The system will be modular, allowing for component reuse, especially AI models, for future expansion and integration.

Appendix A: Glossary

API: A set of protocols for software communication.
MFA: A security method requiring multiple verification factors.
RBAC: A system that restricts access based on user roles.
AI: Machine intelligence simulating human processes.
Portfolio Management: Managing financial assets to meet investment goals.
GDPR: EU law on data protection and privacy.
PCI DSS: Security standards for payment card data.

Appendix B: Analysis Models

• Data Flow Diagrams
• Class Diagrams
• State-Transition Diagrams
• Entity-Relationship Diagrams

Appendix C: To Be Determined List

• TBD-1: Final database schema.
• TBD-2: API integration details.
• TBD-3: Localization specifics.