

Software Requirements Specification

For

Stock Market Analysis

04/05/2023

Prepared by

Specialization	SAP ID	Name
BFSI	500087553	Prasoon Ghosh
IoT	500082017	Kirti Chandani
BFSI	500081119	Diethoselhu Sakhrie
BFSI	500081899	Priya Yadav



Department of Informatics
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY
STUDIES, DEHRADUN- 248007.
Uttarakhand

Table of Contents

Topic		Page No
Table of Content		
Revision History		
1	Introduction	3-4
	1.1 Purpose of the Project	3
	1.2 Target Beneficiary	3
	1.3 Project Scope	3
	1.4 References	4
2	Project Description	5-9
	2.1 Characteristics of data	5
	2.2 SWOT Analysis	5
	2.3 Project Features	6
	2.4 Design diagrams	7
	2.5 Assumption and Dependencies	8
3	System Requirements	10
	3.1 User Interface	10
	3.2 Software Interface	10
4	Other Requirements	11-12
Appendix A: Glossary		13

1) INTRODUCTION

1.1) Purpose of the Project

The purpose of this project is to provide a resource for students to learn about the stock market and how to analyze financial data in Python. The project aims to help students gain an understanding of the stock market itself, including how it works, how it is regulated, and what factors affect stock prices.

Students will learn how to collect financial data from sources such as Quandl or Yahoo Finance using Python libraries such as pandas and pandas_datareader. They will also learn how to preprocess the data by filling in missing values or removing outliers, and how to visualize the data to understand trends and patterns using libraries such as seaborn and matplotlib.

The project will demonstrate how to use technical indicators to develop trading strategies and evaluate their performance using machine learning models such as RandomForestClassifier, LinearRegression, LassoCV, RidgeCV, DecisionTreeRegressor, RandomForestRegressor, GradientBoostingRegressor, KNeighborsClassifier, SVR, Ridge, and Lasso.

Overall, this project will provide students with a practical example of how to analyze financial data using Python and machine learning techniques, and how to apply this knowledge to the stock market. The system requirements for this project include a computer with Python and the required libraries installed, access to financial data sources such as Quandl or Yahoo Finance, and a basic understanding of Python programming and financial markets.

1.2) Target Beneficiary

The target beneficiaries of this project are students who wish to learn about the stock market and how to analyze financial data using Python. The project aims to provide a practical example of how to collect, preprocess, analyze, and visualize financial data, and how to develop and evaluate trading strategies using machine learning techniques. The project will benefit students who are interested in finance, economics, or data analysis, and who want to acquire practical skills that will help them in their future careers. The project will help students gain a deeper understanding of the stock market, its mechanics, and the factors that influence stock prices, and provide them with a valuable resource for learning and practicing financial analysis.

1.3) Project Scope

The project scope was to create a financial analysis tool using Python and Yahoo Finance to help students learn about the stock market. The tool was designed to provide users with a practical example of how to collect, preprocess, analyze, and visualize financial data using Python libraries.

The project focused on using Yahoo Finance as the data source, and Python libraries such as pandas, numpy, and pandas_datareader for data collection and preprocessing. The data was cleaned and prepared by filling in missing values, removing outliers, and transforming the data to ensure it was suitable for analysis.

Data visualization was an important aspect of the project, and libraries such as seaborn and matplotlib were used to create graphs and charts that helped users identify trends and patterns in the data.

Trading strategies were developed using technical indicators and evaluated using machine learning models such as RandomForestClassifier, LinearRegression, LassoCV, RidgeCV, DecisionTreeRegressor, RandomForestRegressor, GradientBoostingRegressor, KNeighborsClassifier, SVR, Ridge, and Lasso. The models were trained using historical data, and their performance was evaluated using various metrics such as accuracy, F1 score, mean squared error, and R-squared.

The scope of the project was limited to the use of Yahoo Finance as the data source and did not include other financial data sources. The project assumed a basic understanding of Python programming and financial markets and did not provide an in-depth explanation of these topics.

1.4) References

1. Pandas' documentation: <https://pandas.pydata.org/docs/>
2. NumPy documentation: <https://numpy.org/doc/>
3. Seaborn documentation: <https://seaborn.pydata.org/>
4. Yfinance documentation: <https://pypi.org/project/yfinance/>
5. Sys documentation: <https://docs.python.org/3/library/sys.html>
6. tkinter documentation: <https://docs.python.org/3/library/tkinter.html>
7. warnings documentation: <https://docs.python.org/3/library/warnings.html>
8. Matplotlib documentation: <https://matplotlib.org/stable/contents.html>
9. Scikit-learn documentation: <https://scikit-learn.org/stable/documentation.html>

2) PROJECT DESCRIPTION

2.1) Characteristics of Data

1. Practical: The project is designed to provide students with a practical example of how to analyze financial data using Python, making it a valuable learning resource for anyone interested in finance, economics, or data analysis.
2. Comprehensive: The project covers all the essential steps of financial analysis, including data collection, preprocessing, visualization, and model development, making it a comprehensive guide for beginners.
3. Interactive: The project includes data visualization and interactive tools that enable users to explore financial data and evaluate trading strategies using machine learning models.
4. Open-source: The project uses open-source libraries and tools, making it accessible to anyone with a basic understanding of Python programming and financial markets.
5. Educational: The project aims to educate students about the stock market, its mechanics, and the factors that influence stock prices, making it a valuable resource for learning and practicing financial analysis.
6. Customizable: The project can be customized and extended to include additional data sources, financial models, or analysis tools, making it a flexible and adaptable learning resource for students.

2.2) SWOT Analysis

STRENGTH-

- 1) Comprehensive guide to financial analysis in Python, covering all essential steps from data collection to trading strategy development.
- 2) Uses open-source libraries and tools, making it accessible and affordable for students and professionals alike.
- 3) Includes data visualization and interactive tools to explore financial data and evaluate trading strategies using machine learning models.
- 4) Provides a practical example of how to apply financial analysis techniques to real-world scenarios.
- 5) Can be customized and extended to include additional data sources, financial models, or analysis tools.

WEAKNESS-

- 1) Limited to using Yahoo Finance as the primary data source, which may have limitations in terms of the depth and breadth of financial data available.
- 2) Assumes basic knowledge of Python programming and financial markets, which may be a barrier for some users.
- 3) The project is focused on the US stock market, which may limit its relevance for users interested in other markets or regions.

OPPORTUNITY-

- 1) Provides a valuable resource for students and professionals seeking to learn or improve their financial analysis skills.
- 2) Can be extended or customized to include additional data sources, financial models, or analysis tools to meet the needs of different users.

- 3) Has the potential to be used in educational settings, as a teaching resource for courses in finance, economics, or data analysis.

THREAT-

- 1) The project may become outdated if Yahoo Finance or the libraries used in the project undergo significant changes or become obsolete.
- 2) Competition from other open-source financial analysis projects or commercial software tools may limit the adoption of this project.
- 3) The project may not be relevant or useful for users who have more advanced or specialized needs in terms of finance.

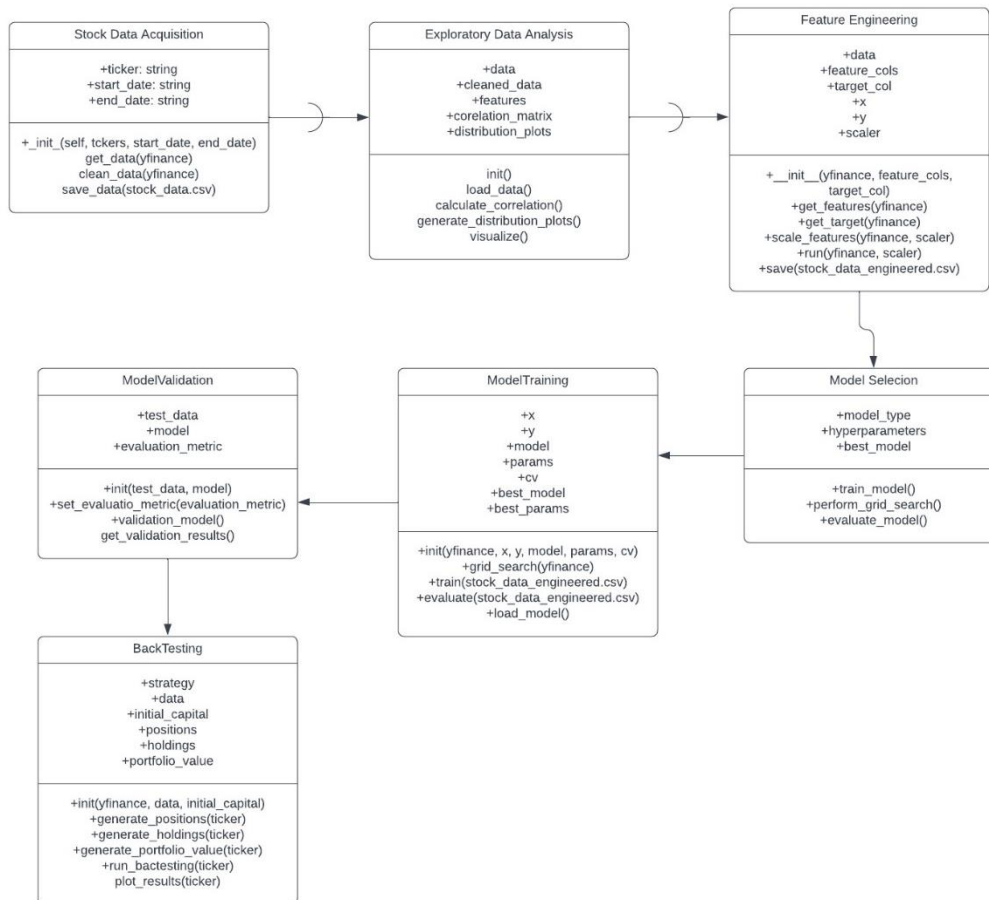
2.3) Project Feature

- 1) Data collection: The project provides Python code for retrieving financial data from Yahoo Finance. This includes stock prices, volumes, financial statements, and news articles related to a company or industry.
- 2) Data preprocessing: The project includes data cleaning and preprocessing techniques such as missing value imputation, outlier detection, and normalization.
- 3) Data visualization: The project provides a suite of data visualization tools to explore financial data, including stock price charts, financial statement visualizations, and correlation matrices.
- 4) Machine learning models: The project includes a set of machine learning models for financial analysis, including regression, classification, and clustering. These models can be used to develop trading strategies, predict stock prices, or identify patterns in financial data.
- 5) Trading strategies: The project provides a set of trading strategies based on technical analysis, such as moving averages, Bollinger Bands, and Relative Strength Index. These strategies can be back tested using historical data to evaluate their performance.
- 6) Back testing: The project includes a back testing module to evaluate the performance of trading strategies using historical data. This module includes metrics such as annualized returns, Sharpe ratio, and maximum drawdown.

Overall, this project provides a comprehensive toolkit for financial analysis in Python, covering all essential steps from data collection to trading strategy development and deployment.

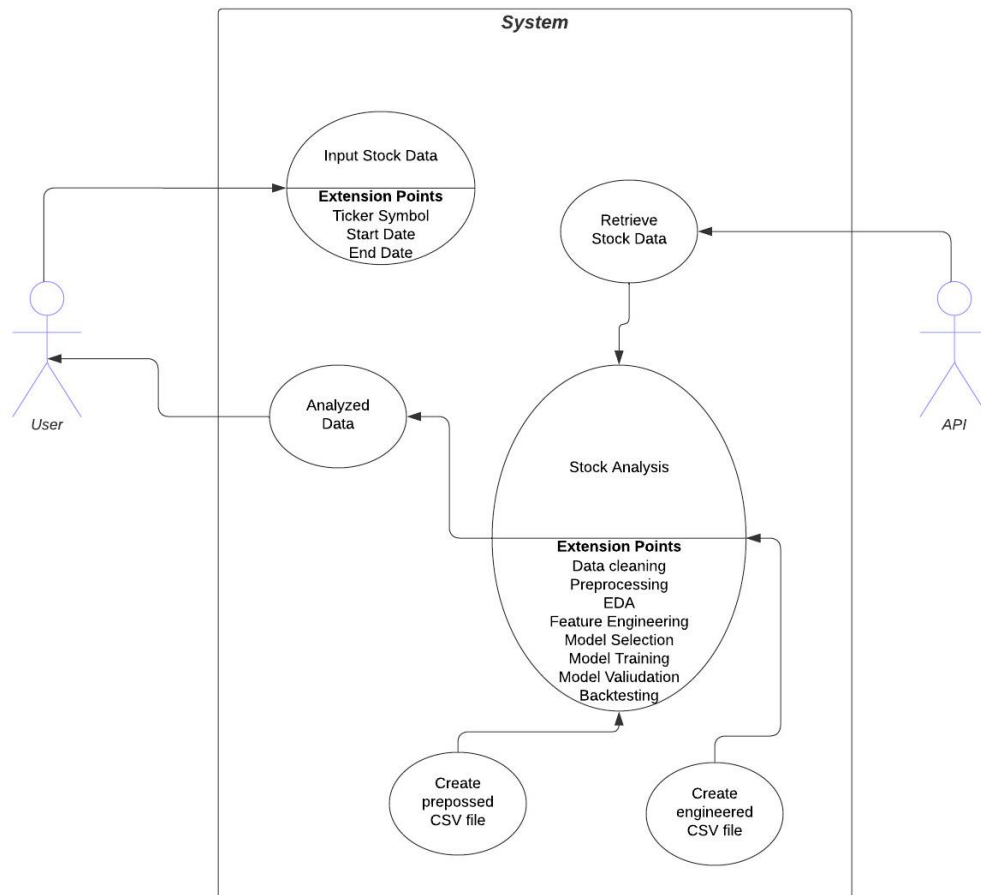
2.4) Design Diagram

1) Class Diagram



2) Use Case Diagram

Use case diagram(Stock Market Analysis)



2.5) Assumptions and Dependencies

ASSUMPTIONS:

- 1) The project assumes that the financial data retrieved from Yahoo Finance is accurate and up to date.
- 2) The project assumes that the selected trading strategies will perform similarly in both back testing and live trading environments.
- 3) The project assumes that the user has basic knowledge of stock market concepts and terminology.
- 4) The project assumes that the machine learning models, and trading strategies selected by the user are appropriate for the selected financial data and time period.

DEPENDENCIES:

- 1) The project depends on the availability and reliability of the Yahoo Finance API for retrieving financial data.
- 2) The project depends on the availability and compatibility of the Python libraries used for data retrieval, preprocessing, visualization, machine learning, and trading strategy development.
- 3) The project depends on the user interface for user input and output. Any issues with the user interface can affect the overall functionality of the project.
- 4) The project also depends on the live trading platform and brokerage firm used for deploying the trading strategies in a live trading environment. Any issues with the platform or the brokerage firm can affect the performance of the trading strategies in a live trading environment.

3) SYSTEM REQUIREMENTS

3.1) User Interface

- 1) Operating System: Windows, MacOS, or Linux
- 2) Screen Resolution: Minimum of 1024 x 768
- 3) Processor: Intel Core i3 or higher (or equivalent AMD or another processor)
- 4) RAM: Minimum of 4GB
- 5) Web Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge
- 6) Internet Connection: Required for accessing Yahoo Finance API

3.2) Software Interface

1. Python 3.x
2. pandas
3. numpy
4. pandas_datareader
5. seaborn
6. yfinance
7. tkinter
8. matplotlib
9. scikit-learn

The user interface will be designed using the tkinter library, which is built into Python. The user interface will allow the user to input their desired financial data, select their desired machine learning models and trading strategies, and view the results of the back testing and live trading. The software interface will include the libraries listed above for data retrieval, preprocessing, visualization, machine learning, and trading strategy development. The Yahoo Finance API will be used to retrieve financial data.

4) OTHER REQUIREMENTS

4.1) Functional Requirements

The functional requirements of the stock market analysis project can be divided into several categories. Firstly, the system must be capable of retrieving financial data from Yahoo Finance API for a user-specified company or stock symbol. This data must then be cleaned and preprocessed to eliminate any missing values, outliers, and inconsistencies. This ensures that the data is reliable and accurate for subsequent analysis.

Next, the system should be able to visualize the cleaned financial data in the form of line charts, scatterplots, and histograms. This will allow users to identify trends and patterns in the data and make informed decisions about their investments.

Machine learning algorithms such as linear regression, decision trees, random forests, and support vector regression can be used to predict stock prices and market trends. The system must be capable of performing these algorithms and provide users with accurate and reliable predictions based on the historical data.

Trading strategies can be developed based on the output of the machine learning algorithms and user-specified criteria. The system should allow users to develop and test these strategies using historical data to evaluate their performance. The system must be able to backtest the developed trading strategies using historical data to evaluate their performance.

Furthermore, the system must be capable of deploying the developed trading strategies on a live trading platform and execute trades in real-time. This will provide users with the opportunity to implement their trading strategies in a real-world setting and achieve financial success.

The user interface is an essential component of the system, and it should be user-friendly, allowing users to input their desired financial data, select their desired machine learning models and trading strategies, and view the results of the backtesting and live trading. The system must also be able to handle errors and exceptions that may occur during data retrieval, preprocessing, machine learning, trading strategy development, backtesting, and live trading.

Finally, the system must generate reports on the results of the backtesting and live trading, including performance metrics such as accuracy, F1-score, confusion matrix, mean squared error, mean absolute error, and r-squared value. These reports will help users to evaluate the effectiveness of their trading strategies and make informed decisions about their investments.

4.2) Environmental Requirements

- 1) Operating System: The system should be compatible with the Windows or Linux operating system.
- 2) Processor: The system should have a processor of at least 1GHz or higher.
- 3) Memory: The system should have at least 4GB of RAM for efficient performance.
- 4) Internet Connection: The system should have a stable and reliable internet connection to ensure seamless access to financial data.
- 5) Python Environment: The system should have Python 3.x installed with all the necessary libraries used in this project.
- 6) Screen Resolution: The system should have a screen resolution of at least 1024 x 768 pixels to ensure that the graphical user interface is displayed correctly.
- 7) Storage: The system should have sufficient storage space to store the financial data and any other relevant files.
- 8) Security: The system should have appropriate security measures in place to prevent unauthorized access to financial data. It is recommended to use secure passwords and enable firewall and antivirus software.
- 9) Third-Party Dependencies: The system should have all the necessary third-party libraries and dependencies installed to ensure proper functioning of the project. These libraries include pandas, numpy, seaborn, and yfinance among others.

APPENDIX A: GLOSSARY

- 1) **Stock Market:** A public market for trading company stocks and other securities. It allows companies to raise capital by selling ownership stakes to investors, who can then profit from any increase in the company's value. The stock market is an important tool for economic growth and investment, and is closely watched by economists, investors, and businesses around the world.
- 2) **Ticker Symbol:** A unique set of characters assigned to a security for trading purposes. It is used to identify and track the price of a stock, commodity, or other security on the stock market. Ticker symbols are typically composed of a series of letters, and sometimes numbers, and are assigned by stock exchanges or other market regulators.
- 3) **Financial Data:** Data relating to the financial performance of a company, including stock prices, financial statements, and other relevant metrics. Financial data is essential for investors and analysts to make informed decisions about whether to buy, sell, or hold a particular security. It is typically collected and analyzed using a range of tools and techniques, including data scraping, data cleaning, and statistical analysis.
- 4) **Data Preprocessing:** The process of cleaning, transforming, and preparing raw data for analysis. It involves a range of tasks, such as removing irrelevant or duplicate data, converting data into a standardized format, and handling missing data. Data preprocessing is a critical step in any data analysis project, as it helps to ensure that the data is accurate, reliable, and ready for further analysis.
- 5) **Data Visualization:** The representation of data in graphical or pictorial format to facilitate understanding and analysis. It is a key tool for exploring and interpreting complex data sets, and can help to reveal patterns, trends, and insights that might be difficult to discern from raw data alone. Data visualization can take many forms, including charts, graphs, maps, and interactive dashboards.
- 6) **Machine Learning:** A branch of artificial intelligence that enables computer systems to learn and improve from experience without being explicitly programmed. It involves the use of statistical models and algorithms to analyze and make predictions based on large data sets. Machine learning has many applications in finance, including stock market analysis, fraud detection, and risk management.
- 7) **Backtesting:** The process of testing a trading strategy on historical data to evaluate its effectiveness and potential profitability. It involves simulating trades based on historical data, and comparing the results to actual market performance. Backtesting is a key tool for investors and traders, as it can help to identify potential risks and improve overall trading performance.
- 8) **User Interface:** The visual interface that enables users to interact with the software, including inputting data and viewing results. A user interface is designed to be intuitive and user-friendly, and can take many forms, including web applications, desktop applications, and mobile apps.
- 9) **Software Interface:** The set of protocols and procedures that allow different software systems to communicate and exchange information with each other. It enables different programs to work together seamlessly, and is critical for data integration, analysis, and sharing. Software interfaces can be standardized using common

protocols such as REST or SOAP, or can be custom-designed for specific applications.