

System Architecture Specification for the Stock Market Simulator

Jakub Mierzwa

November 19, 2023

Contents

1	Introduction	2
1.1	Purpose of the Document	2
1.2	Scope of the Document	2
2	System Description	2
2.1	Data Import Module	2
2.1.1	Module Description	2
2.1.2	Classes and Dependencies	2
2.2	Data Analysis Module	3
2.2.1	Module Description	3
2.2.2	Classes and Dependencies	3
2.3	Charting Module	3
2.3.1	Module Description	3
2.3.2	Classes and Dependencies	4
2.4	Transaction Simulation Module	5
2.4.1	Module Description	5
2.4.2	Classes and Dependencies	5
2.5	Portfolio Management Module	5
2.5.1	Module Description	5
2.5.2	Classes and Dependencies	6
2.6	Strategy Testing Module	6
2.6.1	Module Description	6
2.6.2	Classes and Dependencies	7
2.7	User Interface	7
2.7.1	Module Description	7
2.7.2	Classes and Dependencies	8
3	Inter-Module Dependencies	9
3.1	Description of Dependencies	9
3.2	Conclusions	10

1 Introduction

1.1 Purpose of the Document

This System Architecture Specification aims to detail the design of the Stock Market Simulator. This document provides a comprehensive description of system modules, classes, dependencies between them, and the overall structure of the system. It is intended to serve as a knowledge source for development teams and project stakeholders, ensuring a unified understanding of the architecture and mechanisms of the application.

1.2 Scope of the Document

The document focuses on software architecture, describing key system components, their functions, interaction methods, and data structure. It also provides visual representations in the form of class and module diagrams, facilitating understanding of dependencies and data flow in the system.

2 System Description

The Stock Market Simulator is an advanced platform for simulating and analyzing financial markets. It allows users to simulate stock market trading, analyze market data, test investment strategies, and manage a virtual portfolio. The system consists of several modules, each responsible for different aspects of financial operations, from data import and processing to market analysis and transaction simulation.

2.1 Data Import Module

2.1.1 Module Description

The Data Import Module is responsible for collecting, validating, and normalizing market data from various sources, such as CSV, Excel files. It is a fundamental component of the system, providing essential input data for further analysis.

2.1.2 Classes and Dependencies

- **DataImporter:** Coordinates the data importing process, using different import methods.
 - *Dependencies:* Utilizes objects implementing the IDataImporter interface.

- *Helper Classes:* CSVImporter, ExcelImporter, DataValidator, DataNormalizer.
- **IDataImporter (Interface):** Defines a standard set of import operations.
 - *Implementations:* CSVImporter, ExcelImporter.

2.2 Data Analysis Module

2.2.1 Module Description

The Data Analysis Module is responsible for processing and analyzing imported market data. It performs functions such as statistical data analysis, market trend identification, and technical indicator generation.

2.2.2 Classes and Dependencies

- **DataAnalyzer:** Coordinates the analysis of market data.
 - *Dependencies:* Uses services of StatisticalProcessor, TechnicalIndicatorCalculator, TrendAnalyzer, and SignalGenerator.
- **StatisticalProcessor:**
 - *Tasks:* Conducts statistical analysis of data (averages, variances).
 - *Dependencies:* Directly used by DataAnalyzer.
- **TechnicalIndicatorCalculator:**
 - *Tasks:* Calculates market technical indicators.
 - *Dependencies:* Directly used by DataAnalyzer.
- **TrendAnalyzer:**
 - *Tasks:* Analyzes market trends and patterns.
 - *Dependencies:* Directly used by DataAnalyzer.
- **SignalGenerator:**
 - *Tasks:* Generates trading signals based on analyses.
 - *Dependencies:* Directly used by DataAnalyzer.

2.3 Charting Module

2.3.1 Module Description

The Charting Module in the stock market simulator is responsible for creating and presenting visualizations of market data. This includes generating various types of charts, such as line and candlestick charts, and tools for their customization and export. This module is key to the visual analysis of market trends and aids users in interpreting data.

2.3.2 Classes and Dependencies

Interfaces

- **IRenderer:**
 - *Tasks:* Defines methodologies for rendering charts.
 - *Dependencies:*
 - * *renderChart(chartData):* Requires data for rendering the chart.
- **IChartCustomizer:**
 - *Tasks:* Specifies methods for customizing the appearance of a chart.
 - *Dependencies:*
 - * *customizeAppearance(chart):* Requires access to the chart object for customization.
- **IChartExporter:**
 - *Tasks:* Defines methods for exporting charts.
 - *Dependencies:*
 - * *exportToImage(chart), exportToPDF(chart):* Requires a chart object for export.

Interface Implementing Classes

- **LineChartRenderer and CandlestickChartRenderer:**
 - *Tasks:* Realize the rendering of line and candlestick charts.
 - *Dependencies:* Implement *renderChart(chartData)* according to the chart type.
- **ChartCustomizer:**
 - *Tasks:* Customizes the appearance of a chart.
 - *Dependencies:* Implements *customizeAppearance(chart)*.
- **ChartExporter:**
 - *Tasks:* Exports charts to various formats.
 - *Dependencies:* Implements *exportToImage(chart)* and *exportToPDF(chart)*.

Main Class

- **ChartGenerator:**
 - *Tasks:* Coordinates the process of creating and managing charts.
 - *Dependencies:*
 - * Uses *IRenderer* for chart generation.
 - * Uses *IChartCustomizer* for chart customization.
 - * Uses *IChartExporter* for chart export.

2.4 Transaction Simulation Module

2.4.1 Module Description

The Transaction Simulation Module in the Stock Market Simulator plays a key role in simulating realistic market behavior. It is responsible for processing and executing trade orders, managing their impact on the market, and tracking transaction history. This module enables users to experiment with various trading strategies in a controlled, simulated environment.

2.4.2 Classes and Dependencies

Main Classes

- **TransactionSimulator:**
 - *Tasks:* Coordinates the transaction simulation process.
 - *Dependencies:*
 - * Uses Order and MarketState classes for processing orders and managing market state.
 - * Utilizes polymorphism in handling different types of orders.
- **MarketState:**
 - *Tasks:* Represents the current market state, including stock prices and transaction history.
 - *Dependencies:*
 - * Modified by TransactionSimulator during order processing.

Auxiliary Classes

- **Transaction:**
 - *Tasks:* Represents a completed transaction in the market.
 - *Dependencies:*
 - * Used by TransactionSimulator for recording and reporting transactions.

2.5 Portfolio Management Module

2.5.1 Module Description

The Portfolio Management Module in the Stock Market Simulator is a key component responsible for managing users' investment portfolios. It enables the creation, monitoring, and analysis of portfolios, encompassing both the buying and selling of assets, as well as assessing risk and investment performance. This module is essential for simulating realistic investment scenarios and strategies.

2.5.2 Classes and Dependencies

Main Classes

- **PortfolioManager:**
 - *Tasks:* Manages various user portfolios, allowing operations such as adding and removing assets.
 - *Dependencies:*
 - * Uses Portfolio objects to represent individual portfolios.
 - * Utilizes RiskAnalyzer for risk analysis.
- **Portfolio:**
 - *Tasks:* Represents an investment portfolio, storing a collection of assets.
 - *Dependencies:*
 - * Contains a collection of Asset, representing individual investments.
 - * Uses computational methods to assess value.

Auxiliary Classes

- **Asset:**
 - *Tasks:* Represents an individual investment asset, such as stocks or bonds.
 - *Dependencies:*
 - * Used by Portfolio for calculating the overall portfolio value.
- **RiskAnalyzer:**
 - *Tasks:* Provides tools for assessing the risk associated with a portfolio.
 - *Dependencies:*
 - * Analyzes portfolios stored in PortfolioManager.

2.6 Strategy Testing Module

2.6.1 Module Description

The Strategy Testing Module in the Stock Market Simulator allows users to test and evaluate different investment strategies in a simulated market environment. It is a key element of the system that enables the analysis of the effectiveness and efficiency of various investment approaches, using historical market data.

2.6.2 Classes and Dependencies

Main Classes

- **StrategyTester:**
 - *Tasks:* Coordinates the process of testing investment strategies, using available market data.
 - *Dependencies:*
 - * Uses the InvestmentStrategy class to conduct tests.
 - * Processes test results using the StrategyResult class.
- **InvestmentStrategy:**
 - *Tasks:* Represents an individual investment strategy that can be tested.
 - *Dependencies:*
 - * Contains details and parameters specific to the strategy.

Auxiliary Classes

- **StrategyResult:**
 - *Tasks:* Stores the results of testing a particular investment strategy.
 - *Dependencies:*
 - * Created and used by StrategyTester to evaluate test outcomes.
- **PerformanceEvaluator:**
 - *Tasks:* Assesses the results of strategy testing, using specified evaluation criteria.
 - *Dependencies:*
 - * Analyzes results provided by StrategyResult to assess strategy effectiveness.

2.7 User Interface

2.7.1 Module Description

The User Interface (UI) in the Stock Market Simulator is a key component that enables users to interact with various functions of the system. It offers an intuitive and easy-to-use interface that provides access to market analysis tools, portfolio management, strategy testing, and market monitoring. An effective and well-designed UI is key to ensuring a positive user experience and efficient use of the simulator's capabilities.

2.7.2 Classes and Dependencies

Main Classes

- **UserInterface:**
 - *Tasks:* Provides a central access point to various simulator functions.
 - *Dependencies:*
 - * Manages the Dashboard and various Views, such as MarketAnalysisView, PortfolioView, StrategyTestingView.
- **Dashboard:**
 - *Tasks:* The main user interface screen, presenting basic information and quick access to main functions.
 - *Dependencies:*
 - * Comprises various Widgets that provide specific information and functionalities.

Auxiliary Classes

- **View and its Subclasses:**
 - *Subclass MarketAnalysisView:*
 - * *Tasks:* Presents tools and data for market analysis.
 - * *Dependencies:* Uses widgets and tools for market data analysis.
 - *Subclass PortfolioView:*
 - * *Tasks:* Presents information and portfolio management tools.
 - * *Dependencies:* Enables interaction with PortfolioManager for asset management.
 - *Subclass StrategyTestingView:*
 - * *Tasks:* Provides access to strategy testing tools.
 - * *Dependencies:* Integrates with StrategyTester for conducting tests.
- **Widget:**
 - *Tasks:* UI elements presenting specific information or offering certain functions (e.g., charts, stock lists).
 - *Dependencies:*
 - * Integrated with Dashboard or respective Views.

3 Inter-Module Dependencies

3.1 Description of Dependencies

The Stock Market Simulator has been designed as an integrated system consisting of several modules that are closely interrelated. Below are the key dependencies between the individual system modules:

- **Data Import Module and Data Analysis Module:**
 - The Data Analysis Module depends on data provided by the Data Import Module. Processed data is key for further analysis and interpretation.
- **Data Analysis Module and Chart Module:**
 - The Chart Module uses data from the Data Analysis Module to generate visualizations, such as line and candlestick charts.
- **Data Analysis Module and Transaction Simulation Module:**
 - The Transaction Simulation Module uses the results of market analysis to simulate realistic transaction scenarios and investment decisions.
- **Transaction Simulation Module and Portfolio Management Module:**
 - Transaction decisions made in the Transaction Simulation Module have a direct impact on the portfolio state managed by the Portfolio Management Module.
- **Portfolio Management Module and Strategy Testing Module:**
 - The Strategy Testing Module uses data from the Portfolio Management Module to assess the effectiveness of different investment strategies.
- **Chart Module and User Interface:**
 - The User Interface utilizes the Chart Module to present market data visually, enabling users to conduct interactive analysis.
- **General Dependency of Modules on User Interface:**
 - The User Interface serves as a front-end for interacting with various modules, allowing users to access their functionalities.

3.2 Conclusions

These dependencies are crucial for ensuring smooth and efficient operation of the simulator. Each module is dependent on data and functionalities provided by other modules, ensuring coherence and integrity of the system as a whole. This provides users with a comprehensive environment for simulating and analyzing the stock market.

4 Summary and Future Directions

This document presents a detailed architectural specification for the Stock Market Simulator. Key system modules, their functions, dependencies, and data structure have been discussed, providing a comprehensive overview of the system. Each module is designed with maximum efficiency, scalability, and flexibility in mind for simulating and analyzing financial markets.

It should be emphasized that the Stock Market Simulator is a dynamic project that will evolve in response to changing market and technology requirements. Future expansions and improvements are planned as follows:

- **Integration with New Data Sources:** Expanding the data import module with new sources, allowing simulations based on an even wider range of market data.
- **Advanced Analytical Tools:** Developing the data analysis module with new algorithms and analytical tools, increasing the depth and accuracy of market analyses.
- **Performance Optimization:** Continuously improving system performance through code optimization and the application of new information technologies.
- **Interactive User Interface:** Enhancing the user interface to be even more intuitive and tailored to the needs of users at various levels of expertise.

The completion of the Stock Market Simulator project represents an important step towards creating advanced tools for market simulations. Further development and adaptation of the system will continue to meet the growing demands and expectations of the financial market.