Software Requirements Specification

Macedonian Stock Exchange Analysis Tool

David Dukoski

Jana Angjelkoska

Elena Panovska

06.11.2024

Table of Contents

1. Specific Requirements.. **2**

1.1 Functional Requirements.. 2

1.2 Nonfunctional Requirements... 2

2. Personas.. **3**

3. User Scenarios.. **3**

4. Descriptive Narrative.. **4**

1. Specific Requirements

1.1 Functional Requirements

1.1.1 The system shall automatically download a list of all valid issuers (companies and financial institutions) from the MSE (Macedonian Stock Exchange) website.

1.1.2 The system shall automatically get the stock exchange data of a listed company, if present.

1.1.3 The system shall ensure that only valid issuers (excluding bonds and codes containing numbers) are retrieved.

1.1.4 The system shall check the last available date of data for each issuer in the database. If no data exists for an issuer in the database, the system shall retrieve data for the last 10 years.

1.1.5 The system shall retrieve all missing data up to the current date for each issuer based on the last available date to avoid unneeded data fetching.

1.1.6 The system shall ensure that all newly retrieved data is properly added or merged with existing data in the database.

1.1.7 The system shall ensure that all dates are formatted consistently.

1.1.8 The system shall format the prices (comma for thousands, dot for decimals).

1.1.9 The system shall format all the date fields to a standard (MM/DD/YYYY)

1.1.10 The system shall store the complete and updated stock exchange data in the database (structured file).

1.1.11 The system shall implement the Pipe and Filter architecture for data processing, using independent functions (filters) for data transformation.

1.1.12 The system will utilize a web scraper to automatically retrieve data.

1.1.13 The system shall enable users to visualize and compare stock performance over custom time ranges or between multiple issuers.

1.1.14 The system will have basic error handling to ensure continuity in the dataset.

1.1.15 The system shall incorporate a basic machine learning (ML) model to predict future stock performance trends based on historical data.

1.2 Nonfunctional Requirements

1.2.1 The system will use a CSV (Comma-Separated Values) file to store stock exchange data for a company.

1.2.2 The system shall use multi-threading to retrieve data on a yearly level.

1.2.3 The system shall be made to be used in most web applications.

1.2.4 The system will ensure data accuracy and consistency across all functionalities.

1.2.5 The application should handle errors gracefully, ensuring that failures in data retrieval or processing do not cause complete application failure.

1.2.6 The codebase shall include comments on a function level.

1.2.7 The application should comply with relevant data privacy and protection regulations when handling financial data.

1.2.8 The application will provide an intuitive user interface (UI) for both basic users and professional analysts.

2. Personas

We shall define personas which would access the system and describe why they would do so, and in ***3. User Scenarios***, we would elaborate on how they would do it. The personas include the character’s name, age, background and goals they have set out.

Financial Analyst

Name: Marko

Age: 35

Background: Works for an investment firm analyzing stock market trends.

Goals: Needs accurate historical data from the Macedonian Stock Exchange to make informed investment decisions. Requires timely updates on data.

Software Developer

Name: Bojana

Age: 28

Background: Responsible for developing and maintaining the web application.

Goals: Wants to ensure the application efficiently retrieves and processes data. Needs clear documentation and user-friendly code for maintenance.

Stock Market Enthusiast

Name: Petar

Age: 22

Background: A university student studying finance with an interest in stock trading.

Goals: Seeks to understand market trends and access historical data to support his studies and trading strategies.

3. User Scenarios

Now, we are going to describe some common scenarios the characters in ***2. Personas*** would find themselves in.

Scenario 1: Data Retrieval by Financial Analyst

Marko logs into the web application and initiates the process to retrieve the latest data for all issuers listed on the Macedonian Stock Exchange. Steps: Marko accesses the web application. He selects the option to update historical data. The application uses the web scraper to automatically retrieve data for all issuers. The data is processed, formatted, and stored in the database. Marko receives a notification that the update is complete and can now analyze the latest trends.

Scenario 2: Data Quality Check by Software Developer

Bojana checks the application logs to verify that the data retrieval process has been successful and that the data quality is maintained. Steps: Bojana opens the application's backend and accesses the logs. She reviews the data fetched by the web scraper for completeness and correctness. Bojana checks that the dates and prices are formatted as specified (e.g., using commas for thousands and dots for decimals). She makes any necessary adjustments to the code or data handling processes based on her findings.

Scenario 3: Trend Analysis by Stock Market Enthusiast

Petar wants to analyze the historical performance of a specific issuer to make trading decisions. Steps: Petar logs into the application and selects a specific issuer. He views the historical data presented in a graph format, with options to filter by time frame (last month, last year, etc.). Petar analyzes the trends and decides on a potential investment based on the retrieved data.

4. Descriptive Narrative

Upon logging into the application, users can initiate data retrieval, prompting the system to check the last available data for each issuer. If the application identifies that data is missing or outdated, it will automatically fetch historical data for the last 10 years, transforming and formatting this data for easy analysis. Financial analysts like Marko benefit from timely updates, allowing them to make informed investment decisions. Software developers like Bojana can maintain the application with confidence, knowing that it adheres to data quality standards and operates smoothly. Lastly, stock market enthusiasts like Petar can engage with the application to analyze trends and support their studies, ultimately fostering a better understanding of market dynamics. This approach ensures that the application meets both the functional needs of its users and the non-functional requirements that guarantee performance, reliability, and usability.