

# Software Requirements Specification (SRS)

## Introduction

The project is designed to automate the process of retrieving historical stock market data for issuers listed on the Macedonian Stock Exchange (MSE). The system fetches data from the MSE website, processes it, and stores the data in CSV files for further analysis. The goal of this system is to allow for efficient and concurrent scraping of stock market data over multiple years, saving time and ensuring data accuracy. This document outlines both the functional and non-functional requirements for the system, ensuring that the project meets the necessary expectations for functionality, performance, and maintainability.

## Functional Requirements

### r1.1 Fetching Stock Codes (fetch\_codes)

The system shall retrieve a list of stock codes from the MSE website and filter out codes containing digits or starting with "E".

### r1.2 Fetching Last Update Date (fetch\_last\_update\_date)

The system shall check if a CSV file exists for a stock code and return the most recent trading date, or **None** if no data exists.

### r1.3 Fetching Data for a Stock Code (fetch\_data\_for\_code)

The system shall fetch historical stock data for a given stock code, including prices, percentage change, volume, and turnover, within a specified date range.

### r1.4 Updating Stock Data (update\_data\_for\_code)

The system shall fetch updated data by comparing the last update date to the current date, processing data in parallel using **ThreadPoolExecutor**.

### r1.5 Saving Data to CSV (save\_data\_to\_csv)

The system shall save the fetched stock data in CSV files, creating new files or appending data as necessary.

### r1.6 Main Execution (Main block)

The system shall initiate the scraping process for all stock codes concurrently, and print the total execution time when complete.

## Non-Functional Requirements

### r2.1 Performance

The system shall optimize performance by processing data in parallel using `ThreadPoolExecutor` to reduce execution time.

### r2.2 Scalability

The system shall scale by adjusting the `max_workers` parameter to handle more concurrent requests as needed.

### r2.3 Reliability

The system shall handle errors gracefully, logging them and continuing the scraping process without interruption.

### r2.4 Data Integrity

The system shall ensure data is correctly formatted and avoid duplication, updating only missing or new data.

### r2.5 Concurrency

The system shall process data concurrently to maximize throughput and reduce scraping time, ensuring smooth thread management.

### r2.6 Maintainability

The system's code shall be modular, enabling easy updates and extensions without complex changes to the overall structure.

### r2.7 Usability

The system shall be user-friendly, requiring minimal setup and automatically saving results in CSV files for easy analysis.

### r2.8 Security

The system shall securely handle web requests, using session management to avoid data leakage and network request failures.

## **r2.9 Error Handling and Logging**

The system shall log errors and continue processing other stock codes, with clear error messages for troubleshooting.

## **r2.10 Adaptability**

The system shall adapt to changes in website structure and data format, making it easy to update and expand in the future.

## **Conclusion**

This system is designed to automate the process of retrieving and storing stock market data from the Macedonian Stock Exchange. By leveraging concurrent processing and efficient error handling, the system ensures that large volumes of stock data are scraped, processed, and saved in a timely and reliable manner. The functional and non-functional requirements outlined above provide a clear understanding of how the system should perform and be maintained, ensuring that it meets the demands of scalability, reliability, and efficiency.