**Homework assignment 1**

Bojan Zavojcevski 221159

Viktor Karadjoski 221569

**Table of Contents**

1. **Introduction**
   * Project Overview
   * Purpose of the Web Application
   * Importance of Data Processing
2. **Scope**
   * Functional Requirements
   * Non-Functional Requirements
3. **Functional Requirements**
   * Automated Data Retrieval
   * Database Check for Last Available Data
   * Data Download for Missing Dates
   * Data Transformation and Formatting
   * Database Insertion
   * Process Completion Timer
4. **Non-Functional Requirements**
   * Performance
   * Reliability
   * Scalability
   * Maintainability
   * Data Consistency
5. **User Scenarios and Personas**
   * Persona: Dimitar Najak
   * Goals of the Persona
   * User Scenario and Narrative
6. **Conclusion**
   * Summary of Project Objectives
   * Future Enhancements and Use Cases

**Project Overview**

**The following paper is dedicated to the design of a web application of stock market analysis on MSE by using database-backed storage and several software architecture styles. The first phase will be devoted to implementation according to data processing by using a Pipe and Filter architecture that automates the retrieval, filtering, and transformation of historical daily stock data.**

**Therefore, the main purpose of this project is to collect and prepare all data about the issuers, companies, and other financial institutions listed in MSE at least over the last 10 years.**

**This initial stage in itself will provide a good basis for further analysis of such data by ensuring that it is full, clean, and correctly formatted. The implementation of the data processing pipeline includes filters that obtain the issuer codes, find the last date of available data, and fill in records missing up to the present time. Each "filter" in this pipeline performs certain transformations on the data: filtering out irrelevant data, standardizing the dates, or formatting prices with the aim of easing further handling and maximizing consistency.**

**Updates can then be more easily made, and high-quality data can be ensured along with effective processing and storage of stock information in the database. Meanwhile, these processes automate, meaning that users would have ready-to-analyze updated stock data with very minimal manual intervention.**

**Later on, this project will enable the financial analyst or user to perform analyses of stock trends and make informed decisions based on consistent historical data. Further successive enhancements to this base add more architectural styles, interactivity, and analytical features to the web application for robust and efficient stock market analysis.**

**Scope**

This project will go through the design of an interesting and feature-rich Web application for the analysis of the stock market for Macedonian Stock Exchange.

The application will automate the whole process of retrieving, transforming, and storing data about stocks for all MSE-listed issuers to make sure that at least 10 years of data is available.

While it does so, it has a core focus on the automation of stock data retrieval, checking of historical records for gaps, and updating of the same. Irrelevant data will be filtered out, and uniform formatting of dates and figures of finance will be carried out, with storage into a structured database.

The aim is to develop a reliable and easy-to-reach platform on which MSE stock data is analyzed without human interference, to support all financial analysts and investors in making decisions with a solid foundation, hence based on consistent high-quality historical information.

It will be scalable, reliable, and performance-oriented so that, at any time when new data is available, it gets processed efficiently.

**Functional Requirements**

1. **Automated Data Retrieval**

* Description: The system shall automatically fetch the list of all issuers listed in the Macedonian Stock Exchange excluding bonds or codes with numbers.
* Rationale: So that it captures the required issuer data at all times on its own without requiring human interference.
* Priority: High

1. **Database Check for Last Available Data**

* Description: The system shall check the database to identify the last date of recorded data for every issuer.
* Rationale: It prevents redundant data downloads. This system would save a lot of time and avoid duplication of data.
* Priority: High

1. **Data Download for Missing Dates**

* Description: In cases when some issuers have some missing or incomplete data, the system must be able to download historical stock data from the last recorded date to date.
* Rationale: Ensures that the database keeps updated information on all the issuers of the past 10 years.
* Priority: High

1. **Data Transformation and Formatting**

* Description: The application should transform raw data into a structured format. Examples include consistent date formatting such as YYYY-MM-DD and financial formatting such as comma for thousands, period for decimal: 21,600.00.
* Rationale: Properly formatted data will be sure to be accurate and readable for all subsequent analysis and queries that will have to be done.
* Priority: High

1. **Database Insertion**

* Description: The cleaned data, after transformation, should be inserted into the database, where the new data will merge correctly with any previous records.
* Justification: It will pull all the historical data to a single location, making the retrieving and analysis of the data efficient.
* Priority: High

1. **Process Completion Timer**

* Description: Include a timer that will help in tracking and showing the time taken to complete the population process.
* Justification: Enables insight into application performance as it helps optimize the speed of processing data.
* Priority: Medium

**Non-Functional Requirements**

1. **Performance**
   * Description: The application should retrieve, process and store data in an effective way that minimizes runtime.
   * Metric: Database population for an empty database should be done within reasonable time, optimized to either match or beat the benchmark set by the fastest solution.
   * Priority: High
2. **Reliability**
   * Description: The system shall process data updates correctly without duplicating any entries or loss of data.
   * Metric: Data for each issuer shall be correctly transformed and stored without any record missing in a 10-year period.
   * Priority: High
3. **Scalability**
   * Description: The architecture should allow for easy adjustments to accommodate more data sources if needed.
   * Metric: Ability to expand the number of issuers and years without major changes to the pipeline.
   * Priority: Medium
4. **Maintainability**
   * Description: The code should be modular and well-documented so that making updates or troubleshooting is easy.
   * Metric: Code complexity needs to be low, with clear comments and separation of concerns between modules.
   * Priority: Medium
5. **Data Consistency**
   * Requirement: All data should be in the same format and structure, so that analysis can be precisely done.
   * Metric: Consistency of date formats, numbers, and issuer code formats
   * Priority: High

**User Scenarios and Personas**

**Persona**  
*Name*: Dimitar Najak

*Role*: Financial Analyst  
*Background*:

Dimitar works for an investment firm based in Skopje. Having a basic understanding of stock data analysis, Dimitar is familiar with standard financial data formats. At the same time, he has very limited technical knowledge regarding the processing of data. The key objective for Dimitar would be to stay updated on stock trends at the MSE and to make use of latest data for analysis and forecasting with as little touching of raw data files as possible.

**Goals**

* Retrieval of daily stock data for all the issuers in the MSE has to be as fast as possible, with at least a decade of historical data.
* Ensure that the data being fed is cleaned up and is consistent and in the right format to suit various analytical applications.
* Minimize Manual intervention: The application has to fetch, transform and store data automatically.

**User Scenario and Narrative**

**Scenario**:  
Dimiter would like to analyze the trends in historical stock data to support his investment decisions. Each day, the incoming data from MSE must be preprocessed; it is at the same time tedious and prone to errors to download and clean the stock data manually. Dimitar uses this application, that automatically fetches and transforms the MSE stock data into an updated, cleaned, and prepared-for-analysis state. The application also verifies whether there is missing historical data and fills the gaps, if necessary.

**Narrative**:  
Dimitar opens the application and initiates a data update. The application first opens the MSE website, downloads a list of valid stock issuers, and automatically filters out the irrelevant codes for bonds.

Then, it checks Dimitar's database for the most recent date of recorded data on every issuer and fetches only new or missing data. It then standardizes dates and prices so that the analysis downstream will not be disrupted by formatting anomalies. After the refresh is complete, Dimitar gets an indication that the data is current, and he sees what the total runtime for the refresh was.

Confident that the data should be right and usable, Dimitar exports this to their analytics tools, knowing the data covers the last ten years and is in a uniform format across all issuers.

**Conclusion**

Finally, setting up a web application for stock market analysis in the Macedonian Stock Exchange is significant for improving access to and speed in decision-making for analysts in this field.

The system guarantees high-quality information that is up-to-date and instantly ready, thanks to its process of automatically collecting, transforming, and storing the data of stocks. Its robust architecture, with much attention to performance, reliability, and scalability, will give the system room for growth into the future.

This system will be able to empower users to make more informed decisions on investment backed by credible and accurate historical data, by ensuring strong data consistency and minimum manual intervention.