"Ss. Cyril and Methodius" University in Skopje
FACULTY OF COMPUTER
SCIENCE AND ENGINEERING

Software Design and Architecture

Homework 2

Team Members:

Bojana Andonova 221225
Kalina Jovanovska 221183
Atanas Vitanov 221128

Content

1. Conceptual Architecture

1.1. Categorization of key concepts

1. The web application shall enable the storage of processed data to facilitate easy access and analysis.
2. The web application shall provide an interface to display historical data through user-friendly graphical interfaces, with an option for users to select specific time periods.
3. The web application shall provide technical analysis of stock data for a selected issuer.
4. The web application shall provide fundamental analysis for a selected issuer, allowing users to analyze key financial metrics.
5. The web application shall generate predictions based on the results of technical and fundamental analyses, showcasing possible scenarios for future stock performance.
6. The web application shall offer a real-time table displaying the 10 most listed stocks on the Macedonian Stock Exchange.

Data	Function	Stakeholder	System	Abstract concept
processed data	enable (storage)	user	web application	storage
historical data	display			interface
time periods	technical analysis			user-friendly graphical interface
stock data	fundamental analysis			predictions
issuer	generate			scenarios
financial metrics	showcasing			stock performance
10-most listed stocks				real-time table

1.2. Conceptual Architecture Design

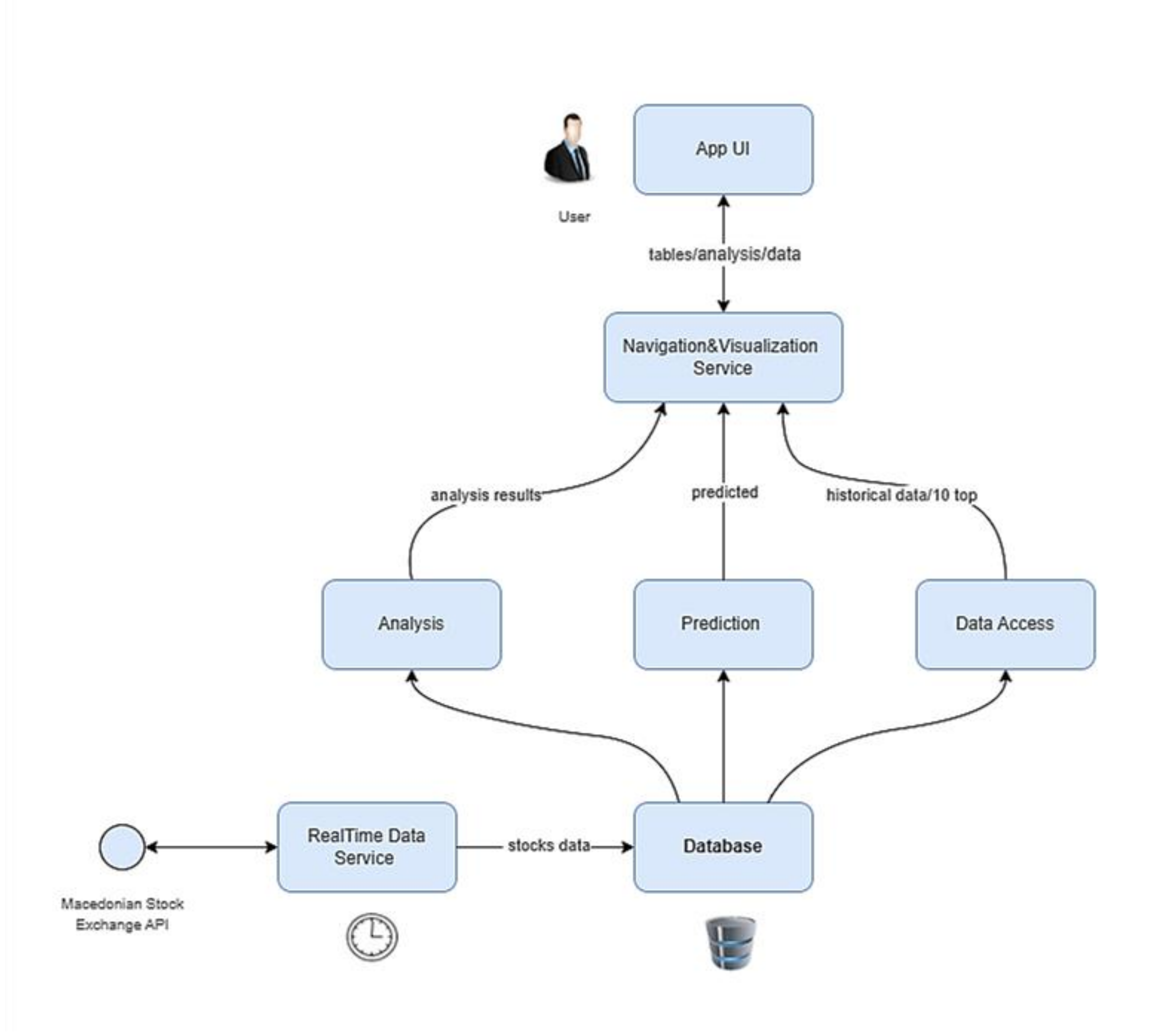

Image 1. Conceptual Architecture

1.3. Component Responsibilities

App UI

→ DisplayData – historical data, top 10 companies.
→ HandleUserInteraction
→ ShowPredictions
→ ShowAnalysisResults

Navigation & Visualization Service

- → RequestData
- → RequestAnalysis
- → RequestPredictions
- → PrepareVisualization

Analysis

- → PerformTechnicalAnalysis
- → PerformFundamentalAnalysis
- → ReturnAnalysisResults

Prediction

- → TrainLSTMModel
- → GeneratePredictions
- → ReturnPredictedData

Data Access

- → FetchHistoricalData
- → FetchTopCompanies
- → ReturnRequestedData

Real-Time Data Service

- → FetchRealTimeData
- → UpdateDatabase

Database

- → StoreData
- → ProvideData

Macedonian Stock Exchange API

- → retrieveMarketData

1.4. Behavior Model

Narrative:

The user – investor opens the app, navigates to the stock analysis section, and selects a specific stock to view its technical analysis. The system processes the request and displays clear and actionable results.

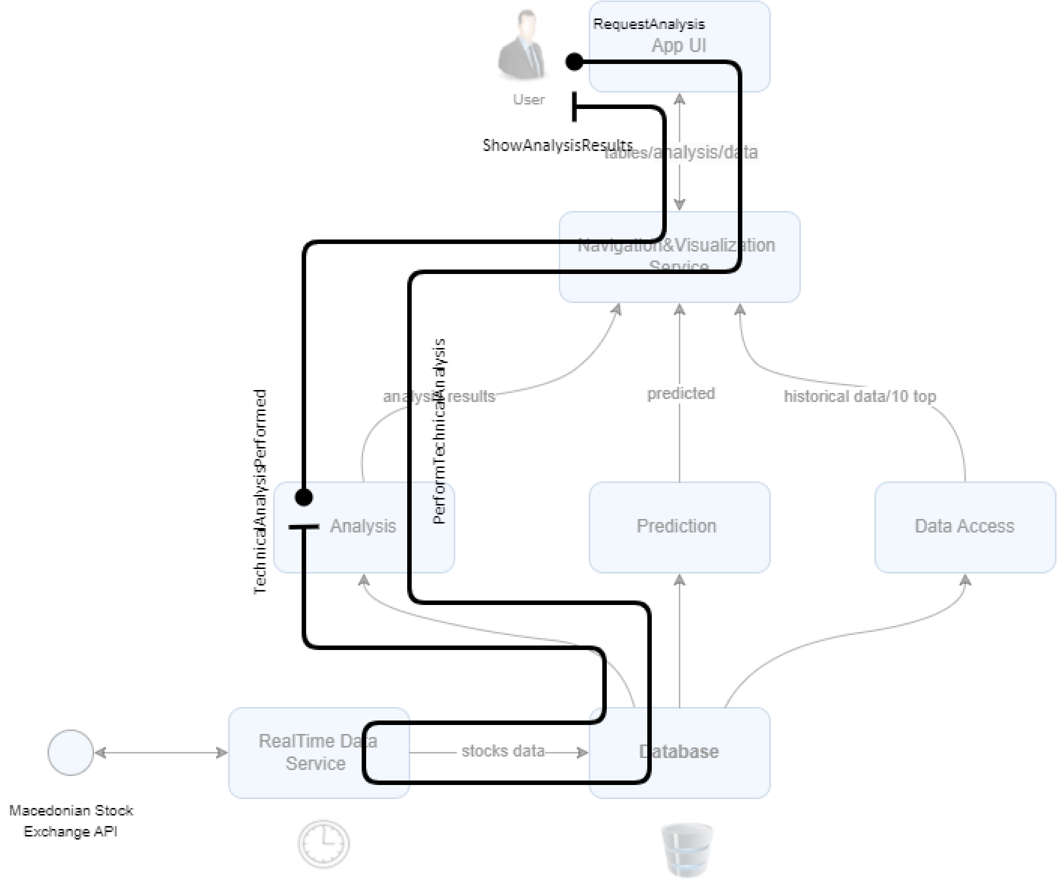

Image 2. Conceptual architecture behavior model – use case "technical analysis"

2. Execution Architecture

2.1. Execution Architecture Design

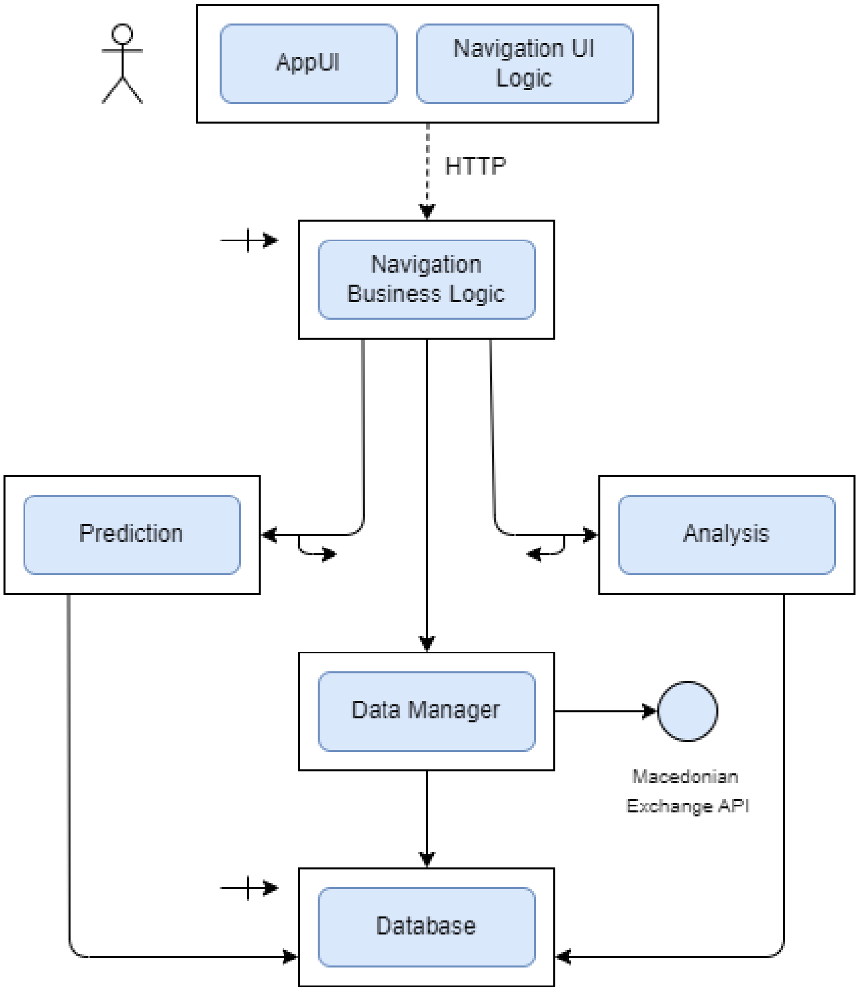

Image 3. Execution Architecture Design

2.2. Components of Execution Architecture

App UI

→ Handles client requests.
→ Provides visualization of analyses, predictions, and data tables.
→ Communicates with the Navigation Business Logic via HTTP.

Navigation UI Logic

→ Manages user interactions within the user interface.
→ Sends requests to the business logic and processes data for visualization.
→ Generates interactive elements for the user.

Navigation Business Logic

→ Acts as the central module for handling requests from the user interface.
→ Coordinates with the Prediction, Analysis, and Data Manager modules.
→ Sends processed data back to the user interface.

Prediction

→ Implements algorithms for forecasting future stock prices.
→ Generates structured prediction results for visualization.
→ Communicates with the Navigation Business Logic module.

Analysis

→ Performs technical and fundamental analysis of stock data.
→ Provides processed analysis data to the Navigation Business Logic module.

Data Manager

→ Manages data fetched from the Macedonian Stock Exchange.
→ Handles data storage and updates in the database.
→ Provides consistent access to data for other modules.
→ Communicates with the Macedonian Stock Exchange API for data retrieval.

Database

→ Stores historical stock data, analysis results, and predictions.
→ Provides fast and structured access to data for the application.
→ Connected to the Data Manager.

Macedonian Exchange API

→ Provides current and historical stock data.

2.3. Behavior Model

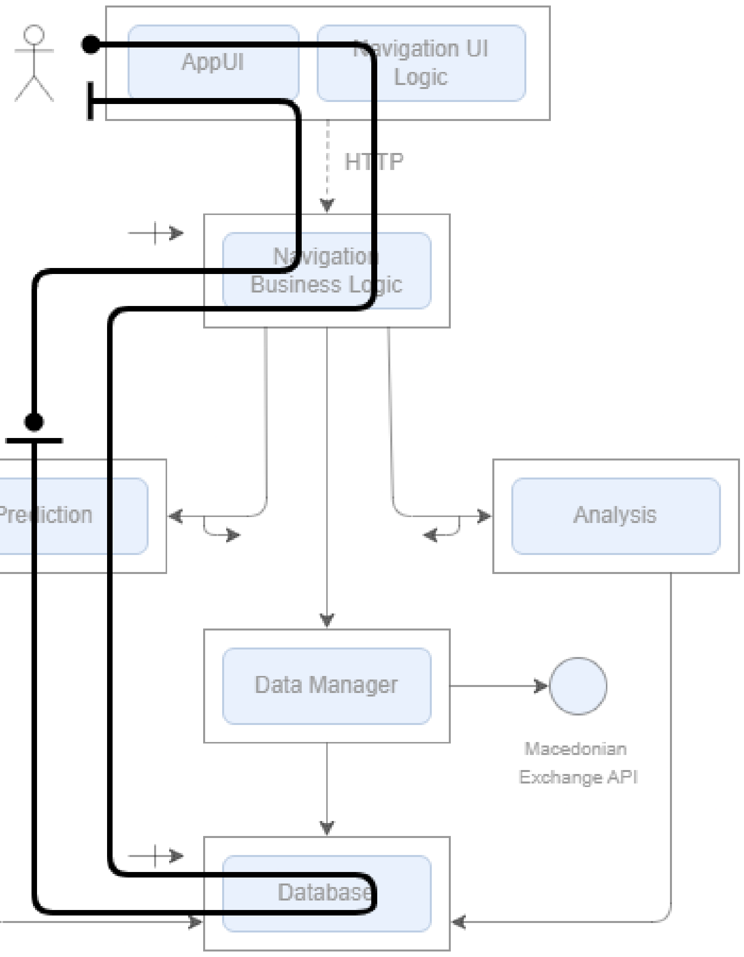

Image 4. Execution architecture behavior model – use case "predict stocks"

3. Implementation Architecture

3.1. Implementation Architecture Design

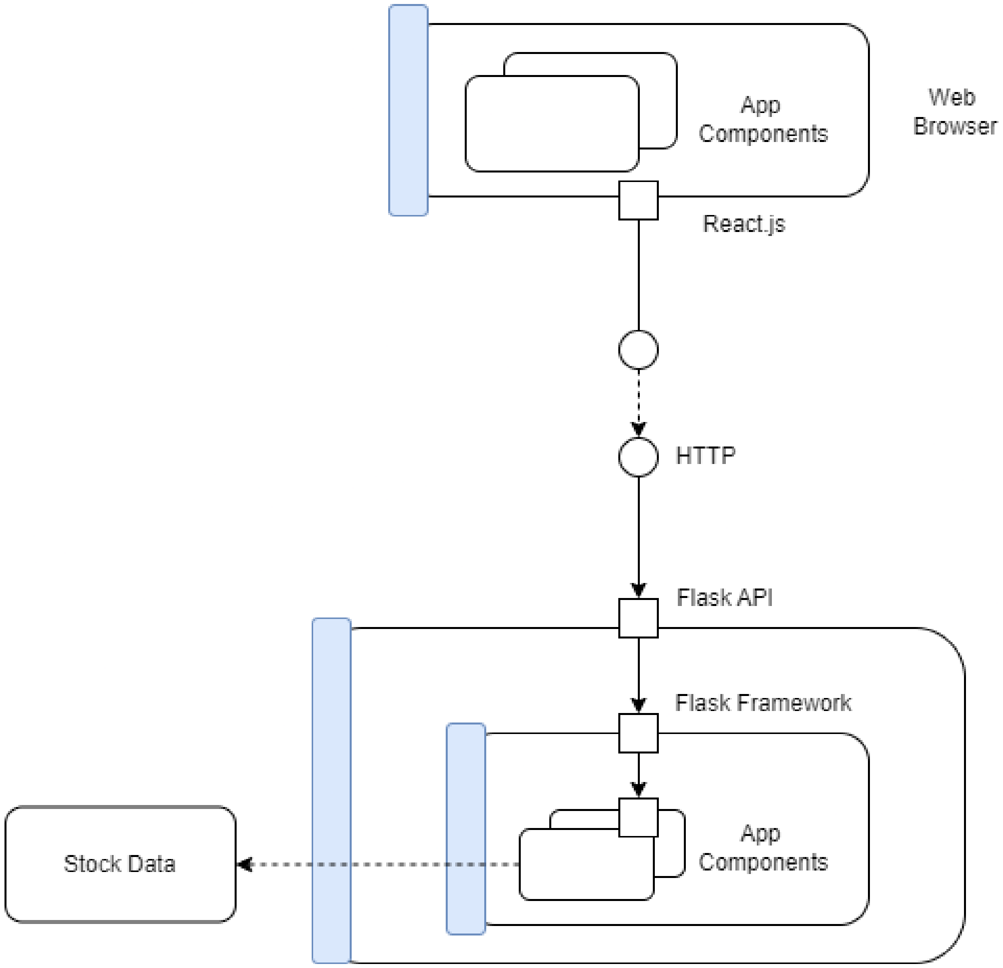

Image 5. Implementation Architecture Design

3.2. Components of Implementation Architecture

Web Browser

- → Acts as the user's gateway to interact with the application.
- → Executes JavaScript, enabling features like data visualization, user inputs, and HTTP requests to the back-end API.
- → Provides secure communication with the Flask API over HTTP.

React.js

- → Acts as the front-end of the application, running in a web browser.
- → Rendering the user interface dynamically.
- → Renders data visualizations, including analyses, predictions, and tables.
- → Provides forms for user inputs, such as selecting stock or time ranges.

Flask API

- → Handles HTTP requests.
- → Exposes endpoints for retrieving stock data, analysis results, and predictions.
- → Communicates with the Flask Framework to process business logic.

Flask Framework

- → Acts as the back-end business logic processor.
- → Coordinates data retrieval and storage operations with the database.
- → Handles requests for predictions and analyses by invoking respective components.

App Components – Flask

- → Provides the core functionality for data management, analysis, and prediction.
- → Includes modules for fetching data from the Macedonian Stock Exchange API.
- → Integrates with Flask Framework to serve results to the front-end.

Stock Data (Database)

- → Stores historical stock data fetched from the Macedonian Stock Exchange API.
- → Maintains data for analyses, predictions, and visualizations.
- → Provides a consistent data source for the back-end application components.

3.3. Behavior Design

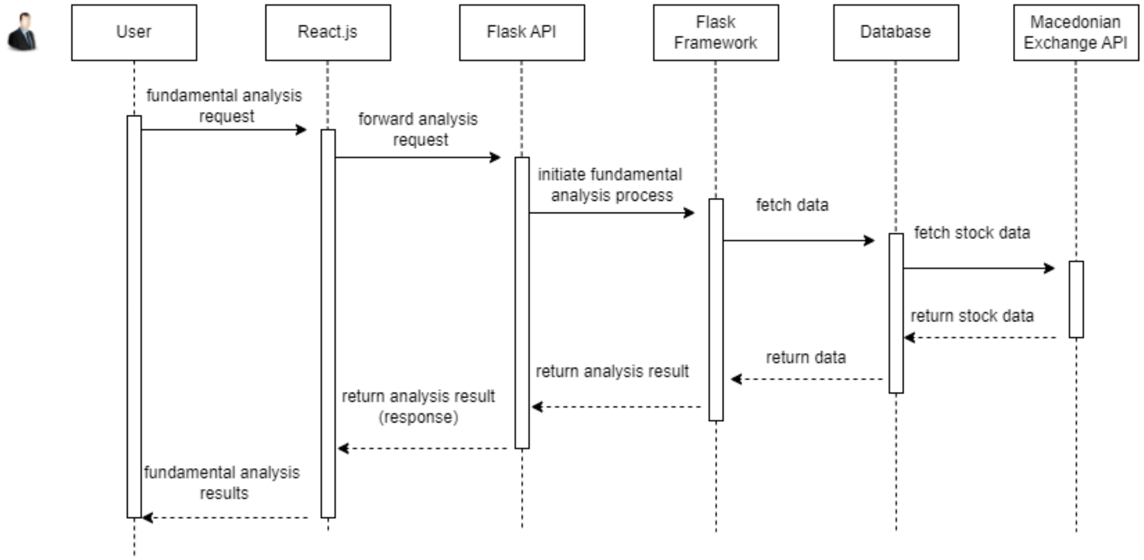

Image 6. Implementation architecture behavior model - sequence diagram (fundamental analysis)