Architectural design

This stock market analysis application is designed to automate the processing, storage, and analysis of stock data from the Macedonian Stock Exchange. In this document, we will show the conceptual, execution, and implementation architecture of our application.

1. Conceptual Architecture

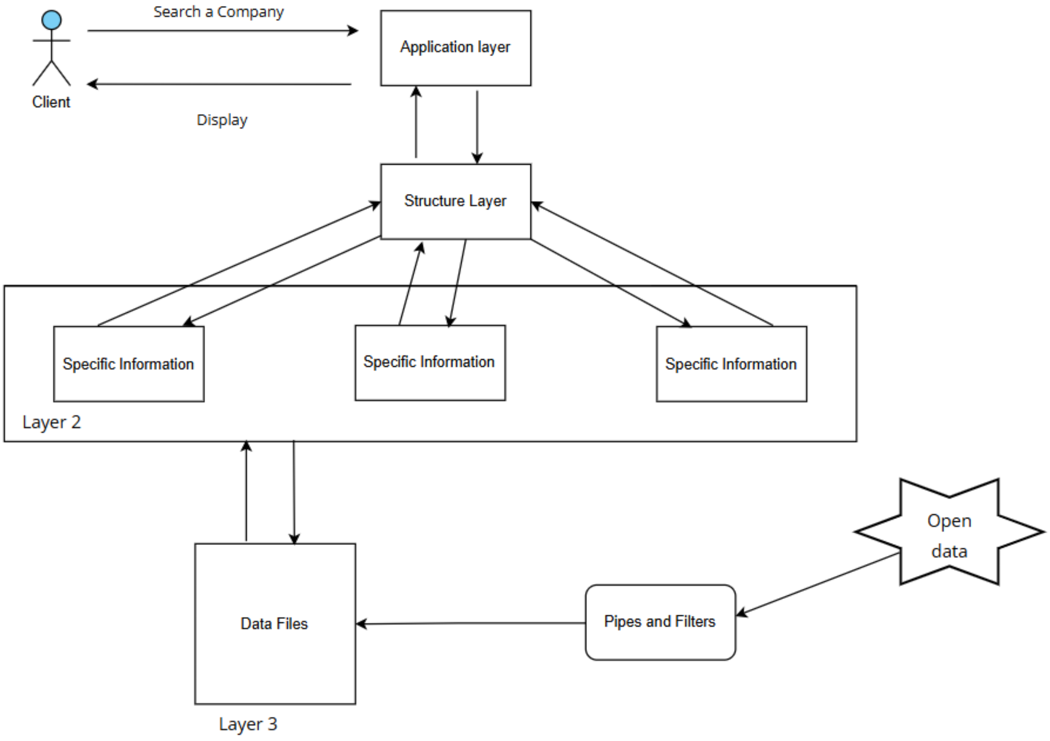

Figure 1. Diagram for Conceptual Architecture

The conceptual architecture illustrates the interaction between the key components of the system, focusing on the data flow and module responsibilities. This design ensures smooth communication across layers and efficient handling of client requests.

The client initiates a search request by selecting a company, which is processed by the Application Layer. The application layer ensures user input is handled efficiently and displays the corresponding results. Then the structure layer acts as an intermediary, organizing and delegating requests to the appropriate modules in Layer 2. It ensures modular handling of specific information based on the user's query. Layer 2 contains these specialized modules that retrieve detailed information about the selected company.

Each module focuses on a specific aspect of the data, ensuring clarity and separation of concerns. The Data Management or layer 3 consists of Data Fiels which act as the core repository for structured data, enabling efficient querying and storage of relevant information, then the Pipes and Filters who handle the processing and transforming of data from external sources, and the Open data which represents the external datasets or APIs.

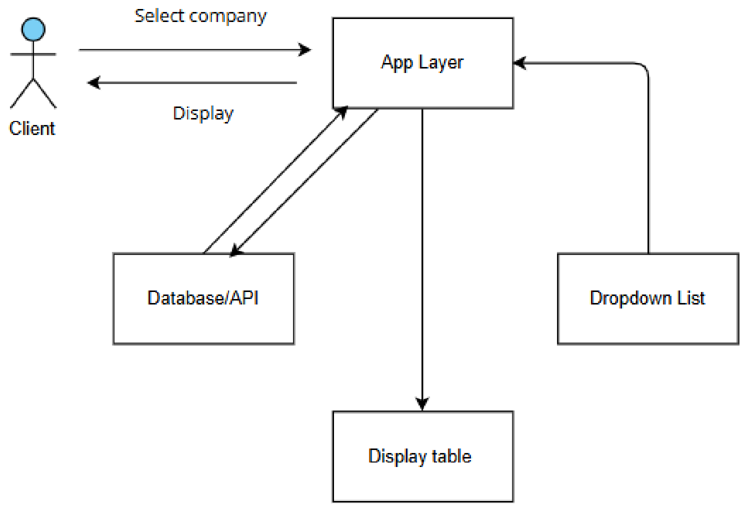

Figure 2. Conceptual Architecture Description for Dropdown and Table Display

The flow of data and interactions for selecting a company and displaying related information is outlined in a table format.

The client interacts with the App Layer through a Dropdown List to select a company for which they want the analysis. Once a selection is made, the App Layer processes the request and manages the data retrieval and display. The dropdown list serves as the interface for the user to select a company and then sends the selected company to the App Layer. This layer acts as the central processing unit of the system handling client requests, fetching data from the Database/API, and formatting it for display. The display table presents the retrieved company information to the user in a tabular format.

This architecture ensures a seamless flow of data from the user's input (Dropdown List) to the final output (Display Table) with efficient handling of backend data through the App Layer and Database/API.

2. Execution Architecture

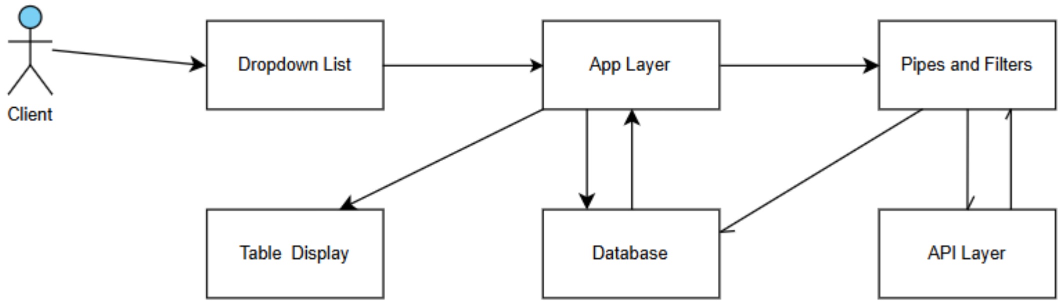

Figure 3. Diagram for Execution Architecture

This diagram represents the flow of interactions between the Client, system components, and external APIs for selecting a company and displaying its information.

The Client begins by interacting with the Dropdown List to choose a company. The selection triggers a communication with the App Layer. This acts as the interface for the user to input their selection. Then passes the selected company to the App Layer via a synchronous call. The App Layer sends requests to the Database for company information using a synchronous call. If external data is required, it forwards the request to the API Layer through the Pipes and Filters mechanism. Then it retrieves the company-specific details as requested by the App Layer and returns them synchronously.

3. Implementation Architecture

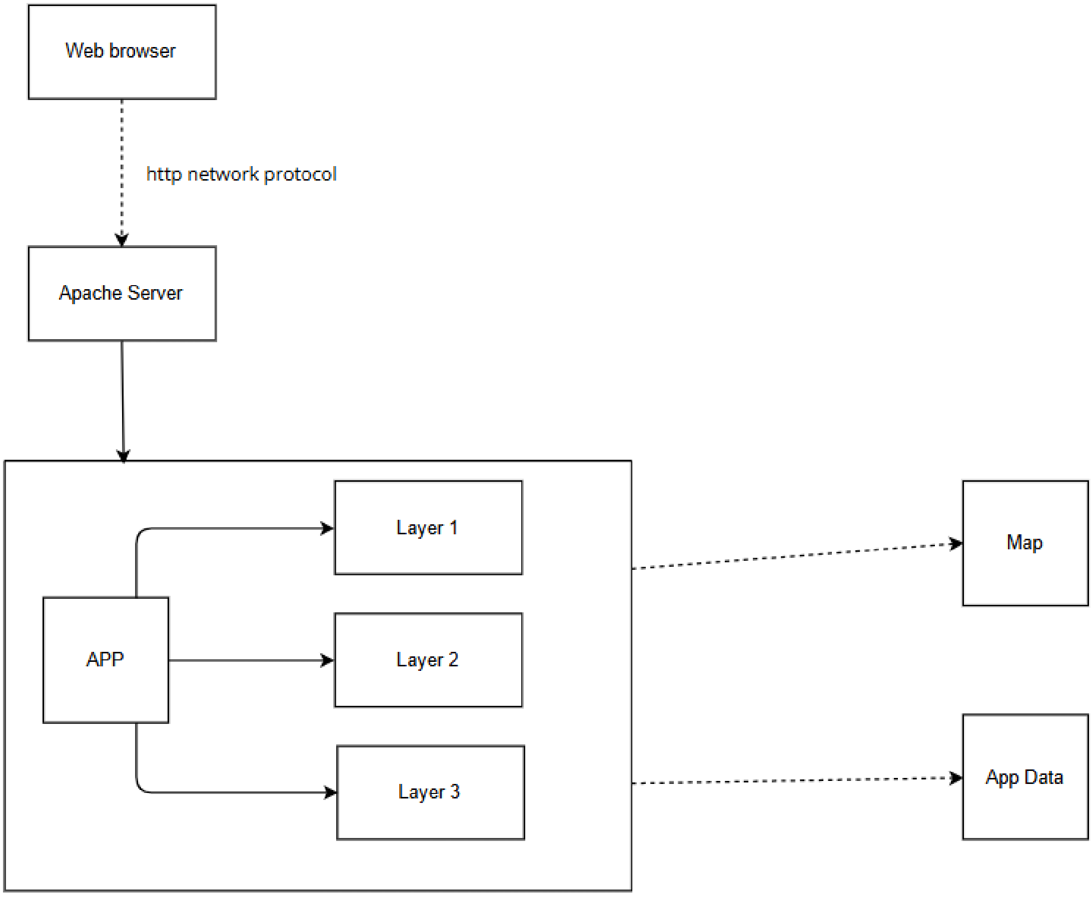

Figure 4. Diagram for Implementation Architecture

This implementation architecture details the mapping of components into actual software artifacts and systems that work together to realize the system's functionality.

The Web Browser acts as the user interface for interacting with the application. It communicates with the backend through the Apache Server using the HTTP network protocol. The Apache Server handles incoming requests from the Web Browser and forwards these requests to the Application for processing. The Application contains three internal layers and connects to both external data sources and the database. The layers handle the API routing and request management, business logic and data processes, and managing database access and external system communication. The map represents the external systems or APIs that communicate with the application to provide additional information. App Data or the Database stores application data, such as historical stock market records, and is accessible to the application for data retrieval and updates.